



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Uso de técnicas de conversión de voz a texto y
procesamiento de lenguaje natural, en el modelado
de dominio del problema y/o solución

presentada por

Ing. Ana Laura Millán Olivares

como requisito para la obtención del grado de
Maestra en Ciencias de la Computación

Director de tesis

Dr. Moisés González García

Cuernavaca, Morelos, México. Febrero de 2021.



“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

Cuernavaca, Mor., **08/diciembre/2020**

OFICIO No. DCC/120/2020
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

C. DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la **C. Ing. Ana Laura Millán Olivares**, con número de control M18CE065, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **“Uso de técnicas de conversión de voz-a-texto y proceso de lenguaje natural, en el modelado de dominio del problema y/o solución”** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

Dr. Moisés González García
Doctor en Ciencias en la Especialidad de
Ingeniería Eléctrica
7501724
Director de tesis

Dr. René Santaolaya Salgado
Doctor en Ciencias de la Computación
4454821
Revisor 1

Dr. Noé Alejandro Castro Sánchez
Doctor en Ciencias de la Computación
08701806
Revisor 2

C.c.p. Depto. Servicios Escolares
Expediente / Estudiante
JGGS/lmz



Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

Cuernavaca, Mor.,

18/enero/2021

No. de Oficio:

SAC/12/2021

Asunto:

Autorización de
impresión de tesis

**ANA LAURA MILLÁN OLIVARES
CANDIDATA AL GRADO DE MAESTRA EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E**

Por este conducto tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado *“Uso de técnicas de conversión de voz-a-texto y proceso de lenguaje natural, en el modelado de dominio del problema y/o solución”*, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E

*“Excelencia en Educación Tecnológica”
“Educación Tecnológica al Servicio de México”*

**DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO**



**CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA**

C.c.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares
Expediente
CMAZ/CHG

Dedicatoria

Es para mí una gran satisfacción y un gran orgullo poder dedicarles a las personas que quiero, este trabajo que con mucho esfuerzo, esmero y trabajo he logrado.

A mis padres Antonio Millán y Laura Olivares, que son el pilar fundamental y apoyo en mi formación académica, por haberme forjado como la persona que soy en la actualidad, con mis valores, mis principios, mi perseverancia, mi empeño y todo ello de una manera desinteresada y lleno de amor; muchos de mis logros se los debo a ustedes entre los que se incluye este. Gracias por motivarme constantemente para alcanzar mis objetivos y anhelos.

A mis hermanos por su afecto, su cariño, su alegría, por ser mi ejemplo y compañeros de mis desvelos cada noche para alcanzar mis metas.

Y finalmente, este proyecto también me lo dedico a mí, por el esfuerzo, dedicación y aprendizaje obtenido en este trabajo.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por proveer el apoyo necesario para realizar mis objetivos y el terminó de mis estudios.

Al Tecnológico Nacional de México (TEcNM) y al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), por facilitar las instalaciones necesarias y en óptimas condiciones para llevar acabo mis estudios, por tener un excelente servicio y nivel como plantel académico, por contar con doctores altamente preparados, y por tener en sus objetivos formar e impulsar el conocimiento científico.

A mi director de tesis, el Doctor Moisés González García, por su enseñanza, tiempo, esfuerzo, por su apoyo, orientación, paciencia, y valiosos conocimientos invertidos en este trabajo de tesis. Gracias, por hacer que saliera de mi zona de confort y por ser un excelente director de tesis, aprendí mucho de usted.

A mis revisores: el Doctor René Santaolaya Salgado y el Doctor Noé Alejandro Castro Sánchez, por brindarme cada uno de sus conocimientos, comentarios e intervenciones para mejorar este trabajo de investigación. Gracias, por ser unos buenos doctores y revisores. Por su tiempo, comprensión y apoyo invertido en cada una de las asesorías. Y así lograr que esto fuera una gran experiencia y se culminara con éxito.

A Daniel Ramírez, por ser mi apoyo incondicional, mi amigo, mi compañero en mi formación académica durante varios años, por ayudarme a crecer personal y profesionalmente. Por estar ahí en las buenas y en las malas, gracias, por cuidarme y ser mi compañía todos los días durante estos dos años.

Y en general, agradezco a cada una de las personas que conocí (Amigos, Doctores, Personal Administrativo, Compañeros) en este proyecto de vida, y a quienes formaron parte de esta gran aventura.

RESUMEN

El desarrollo de un software es un proceso que consta de varias etapas, una de ellas es la etapa de análisis. Esta etapa es donde se define el éxito o fracaso del sistema y se realiza por medio de la comunicación en lenguaje natural entre el analista y el interesado. Dada la complejidad del lenguaje en términos de interpretación del significado, en esta etapa pueden generarse problemas de comunicación. Es decir, si la información que se transmite es ambigua o inconsistente, incomprendida, mal interpretada, o conduce a que los requerimientos no se capturen asertiva y completamente, y los errores se transferirán a las siguientes etapas del desarrollo. Por consecuencia, se tendrán altos costos de corrección e incluso se dificultará la corrección de errores.

La etapa de análisis está dividida en dos fases: la etapa del dominio del problema (analizar y evaluar el objeto que se va a resolver) y la etapa de la solución (recolección de los requerimientos tanto funcionales como no funcionales y modelado UML). En esta investigación se trabajó la etapa del dominio del problema.

Con el objetivo de minimizar o eliminar las causas que originan los problemas de comunicación, en esta tesis se plantea recabar la información a través de un chatbot con reconocimiento de voz como apoyo al analista para recopilar el flujo de los datos y las entidades involucradas. Este chatbot se aplicó en el dominio administrativo de la jefatura del Departamento de Ciencias Computacionales del Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET, para analizar y comprender como funciona un proceso administrativo, así como el flujo de los datos, donde los usuarios involucrados en el dominio del problema fueron entrevistados por el chatbot. La información obtenida se procesó y se representó visualmente utilizando el diagrama de entidades a nivel usuario propuesto por Warnier Orr, el cual indica que se deben de identificar las entidades principales, las entidades secundarias, las relaciones y las transacciones entre entidades. De acuerdo con Orr, con estos elementos se obtiene un diagrama correcto y eficiente.

ABSTRACT

The development of a software is a process that consists of several stages, one of them is the analysis stage. This stage is where the success or failure of the system is defined and is carried out by means of communication in natural language between the analyst and the interested party. Given the complexity of language in terms of interpretation of meaning, this stage can generate communication problems. That is, if the information being transmitted is ambiguous or inconsistent, misunderstood, misinterpreted, or leads to requirements not being assertively and completely captured, and errors will be transferred to the next stages of development. As a consequence, there will be high correction costs and even the correction of errors will be difficult.

The analysis stage is divided into two stages: the problem domain stage (analyzing and evaluating the object to be solved) and the solution stage (collecting both functional and non-functional requirements and UML modeling). In this research, the problem domain stage was worked on.

With the aim of minimizing or eliminating the causes of communication problems, this thesis proposes to collect information through a chatbot with voice recognition to support the analyst to collect the flow of data and the entities involved. This chatbot was applied in the administrative domain of the Department of Computer Science of the National Center for Research and Technological Development CENIDET, to analyze and understand how an administrative process works, as well as the flow of data, where users involved in the domain of the problem were interviewed by the chatbot. The information obtained was processed and visually represented using the entity diagram at user level proposed by Warnier Orr, which indicates that the primary entities, secondary entities, relationships and transactions between entities must be identified. According to Orr, with these elements a correct and efficient diagram is obtained.

CONTENIDO

ÍNDICE DE ILUSTRACIONES	X
ÍNDICE DE TABLAS	XIII
CAPÍTULO 1 INTRODUCCIÓN	1
1.1 DESCRIPCIÓN DEL PROBLEMA	4
1.2 OBJETIVOS	5
1.2.1 OBJETIVO GENERAL	5
1.2.2 OBJETIVOS ESPECÍFICOS	5
1.2.3 METODOLOGÍA DE SOLUCIÓN PROPUESTA	5
1.3 ALCANCES	6
1.4 LIMITACIONES	7
1.5 ORGANIZACIÓN DEL DOCUMENTO	7
CAPÍTULO 2 MARCO TEÓRICO	9
2.1 INGENIARÍA DE SOFTWARE	10
2.1.1 Dominio	10
2.1.1.1 Dominio del problema	10
2.1.1.2 Dominio de la solución	11
2.1.2 Modelo	11
2.1.2.1 Modelo del dominio de problema	12
2.1.3 Diagrama de entidades	12
2.1.3.1 Diagrama de entidades a nivel usuario	13
2.1.3.2 Diagrama de entidades combinado	14
2.1.3.3 Diagrama de entidades a nivel aplicación	14
2.1.3.4 Representación de los diagramas	15
2.2 LENGUAJE NATURAL	18
2.2.1 Lenguaje	18
2.2.2 Lengua	18
2.2.3 Habla	18
2.2.4 Lenguaje Natural	19

2.2.5 Reconocimiento del habla	19
2.2.6 Fonética	19
2.2.7 Fonología	20
2.2.8 Procesamiento de Lenguaje Natural	20
2.2.9 Freeling	20
2.2.10 Análisis Morfológico	21
2.2.11 Análisis Sintáctico	22
2.2.12 Etiquetas Eagles	22
2.2.13 Intenciones DialogFlow	24
2.2.14 Entidades DialogFlow	26
2.2.15 Contextos DialogFlow	26
2.3 RESUMEN DEL CAPÍTULO	27
CAPÍTULO 3 ESTADO DEL ARTE	28
3.1 ANTECEDENTES	29
3.1.1 MAPEO SISTEMÁTICO DEL RECONOCIMIENTO DEL HABLA, PROCESO DEL LENGUAJE NATURAL Y USO DE ONTOLOGÍAS PARA IDENTIFICAR EL DOMINIO DEL PROBLEMA Y LOS REQUERIMIENTOS DE SOLUCIÓN	29
3.1.2 EDITOR PARA EL LENGUAJE DE DOMINIO ESPECIFICO DIAGRAMAS DE ENTIDAD	30
3.1.3 TABLA COMPARATIVA DE LOS ANTECEDENTES	30
3.2 TRABAJOS RELACIONADOS	32
3.2.1 A KNOWLEDGE-INTENSIVE METHOD FOR CONVERSATIONAL CBR	32
3.2.2 PROCESAMIENTO DE LENGUAJE NATURAL PARA ESTUDIAR COMPLETITUD DE REQUISITOS	33
3.2.3 PROCESAMIENTO SEMÁNTICO AUTOMÁTICO, ENFOCADO EN LA COHERENCIA TEXTUAL, PARA APOYAR LA PRODUCCIÓN ESCRITA DE NOTICIAS	34
3.2.4 HERRAMIENTAS LIBRES PARA PROCESAMIENTO DEL LENGUAJE NATURAL	34
3.2.5 ANÁLISIS DE LA DINÁMICA DEL CONTENIDO SEMÁNTICO DE TEXTOS	35
3.2.6 RAZONAMIENTO BASADO EN CASOS DE CONVERSACIÓN	36
3.2.7 UN CHATBOT DE APRENDIZAJE DE REFUERZO PROFUNDO	37
3.2.8 ENCUESTA SOBRE LAS TÉCNICAS DE DISEÑO DE CHATBOT EN LOS SISTEMAS DE CONVERSACIÓN DEL HABLA	37

3.2.9 TABLA COMPARATIVA DE LOS TRABAJOS RELACIONADO DE LENGUAJE NATURAL-----	38
3.2.10 STRUCTURED REQUERIMENTS DEFINITION -----	40
3.2.11 REPRESENTACIÓN DE LA REGLA FORMAL Y VERIFICACIÓN DE LOS REQUISITOS DEL LENGUAJE NATURAL UTILIZANDO UNA ONTOLOGÍA -----	41
3.2.12 MODELADO DE DOMINIOS CON METODOLOGÍA DE OBJETOS Y PROCESOS-----	42
3.2.13 LAS APLICACIONES DEL PROCESAMIENTO DEL LENGUAJE NATURAL (PNL) PARA LA INGENIERÍA DE REQUISITOS DE SOFTWARE - UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA -----	43
3.2.14 VOZ A MODELO: UN ENFOQUE PARA GENERAR MODELOS DE REQUISITOS A PARTIR DE MECANISMOS DE RECONOCIMIENTO DE VOZ-----	43
3.2.15 EXTRACTING SOFTWARE PRODUCT LINE FEATURE MODELS FROM NATURAL LANGUAGE SPECIFICATIONS -----	44
3.2.16 TABLA COMPARATIVA DE LOS TRABAJOS RELACIONADOS DE INGENIERÍA DE SOFTWARE -----	45
3.3 RESUMEN DEL CAPÍTULO-----	47
CAPÍTULO 4 METODOLOGÍA Y RESULTADOS -----	48
4.1 METODOLOGÍA DE SOLUCIÓN-----	49
4.1.1 FASE UNO - ARQUITECTURA DEL CHATBOT -----	50
4.1.2 IMPLEMENTACIÓN DE LA FASE UNO – CREACIÓN DEL CHATBOT -----	67
4.1.2.1 Página web Atenea (sección 1 página web)-----	69
4.1.2.2 Botón Chatbot y Botón Comenzar (Sección 2 página web) -----	70
4.1.2.3 Botón Autores (Sección 3 página web)-----	71
4.1.2.4 Botón Contacto (Sección 4 página web) -----	72
4.1.2.5 Entorno Google Assistant para presentar Atenea -----	72
4.1.3.1 Sub-Fase Uno Etiquetado -----	79
4.1.3.2 Resultados Sub-Fase Uno-Etiquetado-----	85
4.1.3.3 Sub-Fase Dos Archivo .csv-----	89
4.1.3.4 Resultados Sub-Fase Dos – Archivo .csv -----	93
4.1.5 RESULTADOS CREACIÓN DEL DIAGRAMA DE ENTIDADES-----	97
4.2 RESUMEN DEL CAPÍTULO-----	99
CAPÍTULO 5 PRUEBAS -----	100

5.1 PRUEBAS -----	101
5.1.1. Plan de Pruebas -----	102
5.1.2 Diseño de pruebas -----	102
5.1.3 Casos de pruebas -----	103
5.1.4 Procedimiento de Pruebas -----	105
5.1.5 Registro de pruebas-----	107
5.1.6 Reporte de Pruebas -----	107
5.2 ANÁLISIS DE RESULTADOS -----	108
5.3 DISCUSIÓN DE RESULTADOS-----	117
5.4 RESUMEN DEL CAPÍTULO-----	118
CAPÍTULO 6 CONCLUSIONES-----	119
6.1 CONCLUSIONES-----	120
6.2 APORTACIONES-----	121
6.3 TRABAJOS FUTUROS-----	122
REFERENCIAS BIBLIOGRÁFICAS-----	123
ANEXOS-----	126
ANEXO A -----	127
ANEXO B-----	133
ANEXO C -----	136
ANEXO D -----	137
ANEXO E-----	141

ÍNDICE DE ILUSTRACIONES

FIGURA 1- SOLUCIÓN PROPUESTA-----	6
FIGURA 2.- DIAGRAMA DE ENTIDADES A NIVEL USUARIO (1) (ZARAGOZA K., 2019) -----	15
FIGURA 3.- DIAGRAMA DE ENTIDADES A NIVEL USUARIO (3) (ZARAGOZA K., 2019) -----	16
FIGURA 4.- DIAGRAMA DE ENTIDADES COMBINADO (ZARAGOZA K., 2019) -----	16

FIGURA 5.- DIAGRAMA DE ENTIDADES COMBINADO CON FRONTERA (ZARAGOZA K., 2019)-----	17
FIGURA 6.- DIAGRAMA DE ENTIDADES A NIVEL APLICACIÓN (ZARAGOZA K., 2019) -----	17
FIGURA 7.- EJEMPLO PROCESAMIENTO DE FREELING (CATALUNYA, 2020) -----	21
FIGURA 8.- EJEMPLO DE ETIQUETADO " NOMBRE" -----	24
FIGURA 9.- DIAGRAMA EJEMPLO DE INTENCIÓN-----	25
FIGURA 10.- METODOLOGÍA DE SOLUCIÓN. -----	50
FIGURA 11.- SIMBOLOGÍA DE LA SECUENCIA DE PREGUNTAS-----	53
FIGURA 12.- EJEMPLO ENTIDAD DIALOGFLOW-----	54
FIGURA 13.- EJEMPLO ENTIDAD PROCESO WARNIER ORR -----	54
FIGURA 14.- DIAGRAMA DE LA SECUENCIA DE PREGUNTAS PARTE 1/2-----	55
FIGURA 15.- DIAGRAMA DE LA SECUENCIA DE PREGUNTAS PARTE 2/2-----	57
FIGURA 16.- RELACIÓN ENTRE COMPONENTES DENTRO DE LA PLATAFORMA DE GOOGLE -----	59
FIGURA 17.- ENTIDADES DEL CHATBOT ATENEA-----	60
FIGURA 18.- FRASES DE ENTRENAMIENTO INTENCIÓN DEFAULT WELCOME INTENT -----	65
FIGURA 19.- EJEMPLO DE VARIABLES DE LA ENTIDAD SIENTREVISTA -----	66
FIGURA 20.- EJEMPLO DE RESPUESTA PARA LA INTENCIÓN SIENTREVISTA -----	67
FIGURA 21.- INTERFAZ CHATBOT EN PÁGINA WEB-----	68
FIGURA 22. INTERFAZ DEL CHATBOT EN GOOGLE ASSISTANT -----	68
FIGURA 23.- BOTONES DE LA PÁGINA WEB ATENEA -----	69
FIGURA 24.- BOTÓN CHATBOT SECCIÓN DOS -----	71
FIGURA 25.- BOTÓN AUTORES SECCIÓN TRES -----	71
FIGURA 26.- BOTÓN CONTACTO SECCIÓN CUATRO -----	72
FIGURA 27.- ACTIVAR CHATBOT ATENEA DESDE EL ASISTENTE DE GOOGLE-----	73
FIGURA 28.- GOOGLE ACTIVA EL CHATBOT ATENEA -----	73
FIGURA 29.- ENTREVISTA AL USUARIO DESDE GOOGLE ASSISTANT-----	74
FIGURA 30.- HISTORIAL DE LA ENTREVISTA A LA SECRETARIA ARCHIVO .TXT-----	75
FIGURA 31.- HISTORIAL DE LA ENTREVISTA A LA SECRETARIA EN DIALOGFLOW-----	76
FIGURA 32.- HISTORIAL EN ARCHIVO .TXT DE LA ENTIDAD ALUMNO -----	77
FIGURA 33.- HISTORIAL DEL ALUMNO EN DIALOGFLOW -----	78
FIGURA 34.- SUB-FASES DE LA FASE ANÁLISIS DEL TEXTO -----	79
FIGURA 35.- NIVELES DEL PLN (LIMITED, 2020)-----	80

FIGURA 36.- EJEMPLO DE SINTAXIS DE DEPENDENCIA -----	82
FIGURA 37.- SIMBOLOGÍA DIAGRAMA ETIQUETADO CON FREELING-----	82
FIGURA 38.- DIAGRAMA CLASE ANALYZER. JAVA -----	83
FIGURA 39.- EJEMPLO DE CÓMO SE LEE EL HISTORIAL DENTRO DE LA CLASE ANALYXER.JAVA -----	84
FIGURA 40.- PRIMER COMANDO PARA EJECUTAR LA CLASE ANALYZER.JAVA-----	86
FIGURA 41.- SEGUNDO COMANDO PARA EJECUTAR LA CLASE ANALYZER.JAVA -----	86
FIGURA 42.- TOKENIZACIÓN DEL HISTORIAL DE CONVERSACIÓN -----	86
FIGURA 43.- SENTENCIA SPLITTER PARA PROCESAR EL HISTORIAL -----	87
FIGURA 44.- SENTENCIA PARA NOMBRAR LAS ENTIDADES DENTRO DEL TEXTO-----	87
FIGURA 45.- IMPRESIÓN EN LÍNEA DE COMANDOS DE LAS PALABRAS CON SU CATEGORIZACIÓN CORRESPONDIENTE-----	88
FIGURA 46.- HISTORIAL CON SU CATEGORÍA GRAMATICAL -----	89
FIGURA 47.- SIMBOLOGÍA DEL DIAGRAMA PARA LA CREACIÓN DEL ARCHIVO .CSV -----	90
FIGURA 48.- DIAGRAMA CLASE OR.RB -----	91
FIGURA 49.- LIBRERÍA .CSV EN RUBY -----	94
FIGURA 50.- ENCABEZADO DEL ARCHIVO .CSV -----	94
FIGURA 51.- MÉTODO INCLUDE EN RUBY-----	94
FIGURA 52.- ARCHIVO RESULTADO.CSV PARA LA ENTREVISTA A LA SECRETARIA -----	95
FIGURA 53.- ARCHIVO RESULTADO2.CSV PARA LA ENTREVISTA AL ALUMNO -----	96
FIGURA 54.- ARCHIVO RESULTADO3.CSV PARA LA ENTREVISTA A LA PRESIDENTA DE CLAUSTRO -----	96
FIGURA 55.- SIMBOLOGÍA PARA REPRESENTAR EL DIAGRAMA DE ENTIDADES A NIVEL USUARIO EN VISIO MICROSOFT -----	97
FIGURA 56.- DIAGRAMA DE ENTIDADES A NIVEL USUARIO - SECRETARIA (ENTIDAD PRINCIPAL) -----	98
FIGURA 57.- DIAGRAMA DE ENTIDADES A NIVEL USUARIO ALUMNO (ENTIDAD SECUNDARIA) -----	99
FIGURA 58.- DIAGRAMA DE ENTIDADES A NIVEL USUARIO PRESIDENTA DE CLAUSTRO (ENTIDAD SECUNDARIA) -----	99
FIGURA 59.- CREAR UN AGENTE (CHATBOT) EN DIALOGFLOW -----	136
FIGURA 60.- CREAR UNA ENTIDAD EN DIALOGFLOW -----	137
FIGURA 61.- CREAR UNA INTENCIÓN EN DIALOGFLOW -----	137

ÍNDICE DE TABLAS

TABLA 1.- REPRESENTACIÓN DE LAS ETIQUETAS EAGLES-----	23
TABLA 2.- NOMENCLATURA DE LA CATEGORÍA "NOMBRES" DE LAS ETIQUETAS EAGLES -----	23
TABLA 3.- COMPARATIVA DE ANTECEDENTES-----	31
TABLA 4.- TRABAJOS RELACIONADOS PLN -----	38
TABLA 5.- TRABAJOS RELACIONADOS INGENIERÍA DE SOFTWARE -----	45
TABLA 6.- RECURSOS CHATBOT -----	51
TABLA 7.- INTENCIONES DEL CHATBOT-----	61
TABLA 8.- CASOS DE PRUEBA-----	102
TABLA 9.- PRUEBAS A REALIZAR-----	102
TABLA 10.- MUESTRA DEL CASO DE PRUEBA TC-01-----	104
TABLA 11.- MUESTRA DEL PROCEDIMIENTO DE PRUEBAS TC-01 -----	106
TABLA 12.- MUESTRA DE LAS ACTIVIDADES Y EVENTOS DEL CASO DE PRUEBA TC-01 -----	107
TABLA 13.- RESULTADOS GENERALES DE LAS PRUEBAS -----	107
TABLA 14.- MATRIZ DE CONFUSIÓN DE PRECISIÓN PARA ENTIDADES DE LA ENTREVISTA UNO -----	110
TABLA 15.- MATRIZ DE CONFUSIÓN DE COBERTURA PARA ENTIDADES DE LA ENTREVISTA UNO-----	110
TABLA 16.- MATRIZ DE CONFUSIÓN DE PRECISIÓN PARA TRANSACCIONES DE LA ENTREVISTA UNO-----	111
TABLA 17.- MATRIZ DE CONFUSIÓN DE COBERTURA PARA TRANSACCIONES DE LA ENTREVISTA UNO ----	111
TABLA 18.- MATRIZ DE CONFUSIÓN DE PRECISIÓN PARA ENTIDADES DE LA ENTREVISTA DOS -----	112
TABLA 19.- MATRIZ DE CONFUSIÓN DE COBERTURA PARA ENTIDADES DE LA ENTREVISTA DOS -----	113
TABLA 20.- MATRIZ DE CONFUSIÓN DE PRECISIÓN PARA TRANSACCIONES DE LA ENTREVISTA DOS-----	113
TABLA 21.- MATRIZ DE CONFUSIÓN DE COBERTURA PARA TRANSACCIONES DE LA ENTREVISTA DOS ----	114
TABLA 22.- MATRIZ DE CONFUSIÓN DE PRECISIÓN PARA ENTIDADES DE LA ENTREVISTA TRES -----	115
TABLA 23.- MATRIZ DE CONFUSIÓN DE COBERTURA PARA ENTIDADES DE LA ENTREVISTA TRES -----	115
TABLA 24.- MATRIZ DE CONFUSIÓN DE PRECISIÓN PARA TRANSACCIONES DE LA ENTREVISTA TRES-----	116
TABLA 25.-MATRIZ DE CONFUSIÓN DE COBERTURA PARA TRANSACCIONES DE LA ENTREVISTA TRES ----	116
TABLA 26.- RESULTADOS DE LAS MEDIDAS DE EVALUACIÓN PARA LAS ENTIDADES-----	117
TABLA 27.- RESULTADOS DE LAS MEDIDAS DE EVALUACIÓN PARA LAS TRANSACCIONES-----	117

ACRÓNIMOS

ADOM: *Application-Based Domain Modeling* (Modelado de Dominio Basado en la Aplicación)

API: *Application Programming Interface* (Interfaz de Programación de Aplicaciones)

CCBR: *Conversational Case Based Reasoning* (Razonamiento Basado en Casos)

CENIDET: Centro Nacional de Investigación y Desarrollo Tecnológico

CHATBOT: Es una aplicación de inteligencia artificial que permite imitar una conversación real con un usuario con un lenguaje natural.

CQC: *Consistent Question Clustering* (Agrupamiento consistente de preguntas)

CQS: *Coherent Question Sequencing* (Secuencia Coherente de preguntas)

FI: *Feature Inferencing* (Inferencia de Características)

FM: *Feature Model* (Modelo de Características)

IA: Inteligencia Artificial

LEL: Léxico Extendido del Lenguaje

LSA: *Latent Semantic Analysis* (Análisis Semántico Latente)

MDA: *Model Driven Architecture* (Arquitectura Dirigida por Modelos)

MOF: Mecanismo de Meta-Objeto

NL: *Natural Language* (Lenguaje Natural).

NLP: *Natural Language Processing* (Procesamiento de Lenguaje Natural).

NLTK: *Natural Language Tool Kit* (Kit de Herramientas de Lenguaje Natural).

OPM: *Object and Process Methodology* (Metodología de Objetos y Procesos)

OWL: *Web Ontology Language* (Lenguaje Ontologías Web).

PLN: Procesamiento de Lenguaje Natural

POS: Posición

POST: *Parts Of Speech tagging* (Partes del habla).

QR: *Question Ranking* (Clasificación de preguntas)

TC- *Test Case* (Caso de Prueba)

TF-IDF: *Term frequency – Inverse document frequency* (Frecuencia de ocurrencia del término en la colección de documentos)

TPr- *Test Procedure* (Procedimiento de Prueba)

UML: *Unified Modeling Language* (Lenguaje Unificado de Modelado).

Capítulo 1

Introducción

CAPITULO 1. INTRODUCCIÓN

En el Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), se trabajó con el desarrollo de Atenea, un agente conversacional que interactúa con las personas a través de la voz y cuyo objetivo es obtener toda la información necesaria de un dominio de problema específico para elaborar un modelo de negocio de alto nivel de abstracción, como es propuesto por Warnier Orr (Orr K. T., 1981), en el que se señalan explícitamente las personas con las que el entrevistado interactúa (entidades del proceso) y los documentos que intercambian entre ellos (salidas). Esta información es recabada por medio de una entrevista al personal administrativo.

Actualmente, la voz es la forma básica, común y eficiente que la gente utiliza para interactuar entre sí. Por eso no es raro que las personas esperen tecnologías que abarquen este modo de interacción, debido a que las nuevas Tecnologías de la Información y la Comunicación (TIC), han sido el origen de importantes cambios que impactaron no sólo en la vida cotidiana de los individuos, sino también en organizaciones e instituciones sociales (Litwin, 2005). Estas permiten a las máquinas responder de manera correcta y confiable a la voz humana y proveer servicios útiles y valiosos. Dado que comunicarse de esta manera con algún aparato electrónico es más rápido que utilizar el teclado, ocasionando que cada día que pasa el teclado móvil sea menos usado por los usuarios.

El sector computacional diariamente ofrece sus servicios a una gran cantidad de clientes. A medida que un servicio es requerido por más clientes, las necesidades de ellos incrementan y las empresas se ven obligadas a invertir en mecanismos basados en las TICs para incrementar su productividad. Los chatbots facilitan el incremento de eficiencia (dado que realizan su tarea con la menor cantidad de recursos humanos) y de eficacia (logran el objetivo esperado) del proceso de atención al usuario en las empresas. Siempre y cuando contengan las características que apoyen en la comprensión del servicio y alcance los resultados esperados y estos sean satisfactorios (Paredes, 2019). Para lograr esta eficiencia y eficacia las empresas recurren a la ingeniería de software. Esta es una disciplina que ofrece métodos y técnicas para desarrollar y mantener software de calidad dentro de un dominio específico.

Una rama de la ingeniería de software la realizan analistas que se enfocan en el proceso de análisis de dominio, para identificar y documentar características de sistemas en un dominio específico. Este proceso implica distintas fases y actividades con el objetivo de evidenciar qué entidades y salidas derivados de estas actividades, deben ser rastreables y consistentes. En este contexto, realizar un proceso de análisis de dominio sin soporte de herramientas aumenta los riesgos de falla, aunque se requiere asegurar que la herramienta utilizada soporte el proceso completo y no solo una parte del mismo (Lisboa, 2010).

Durante el desarrollo de este trabajo, algunas de las tareas importantes fueron: seleccionar la plataforma, herramienta o IDE adecuada para crear el chatbot, conocer las características que empleaban, por ejemplo, la tecnología que aplica (Ej.: redes neuronales, procesamiento de lenguaje natural (PLN), aprendizaje automático o aprendizaje basado en reglas por mencionar algunas); la disponibilidad de la documentación en internet; el idioma; y el lenguaje acorde a esta investigación. Otro conjunto de tareas se realizó para encontrar las preguntas adecuadas, precisas y ordenadas que ayudarán a recolectar la información mínima necesaria para crear el diagrama de entidades a nivel usuario, y estas mismas fueran entendibles por el usuario sin ser redundantes ni se omitiera información.

Así mismo, se creó adecuadamente y satisfactoriamente el chatbot que ayudó a recolectar la información mediante una entrevista. Se presentó el chatbot en dos plataformas amigables para el usuario, y cada interfaz usa los dos medios definidos para esta investigación (Texto/voz). Por otro lado, las técnicas utilizadas del Procesamiento del Lenguaje Natural fueron las necesarias para procesar el texto y con ello lograr el objetivo deseado.

Por último, con la entrevista automática por medio del chatbot y con ayuda de las actividades del proceso realizado se generó un análisis para identificar exitosamente cada entidad involucrada en el dominio del problema, en este caso para el proceso de Maestría del Departamento de Ciencias Computacionales, dentro del Cenidet.

Por otro lado, durante este capítulo se expone el problema a resolver y la propuesta general de solución. Asimismo, se incluyen los objetivos, alcances y limitaciones, que se establecieron para esta investigación.

1.1 DESCRIPCIÓN DEL PROBLEMA

El conocimiento se organiza y estructura en dominios. Y un dominio se representa mediante un modelo del dominio que consta de vocabulario y los conceptos claves involucrados. Además, un dominio consta de dos partes importantes: 1) El dominio del problema: el entorno donde se trabajará, es decir, las ideas, el conocimiento y la información del problema que se intenta resolver; y 2) El dominio de solución: que define el entorno y los elementos incluidos en la solución. Así, en este entorno se provee una solución simple y estructurada basada en el conocimiento del problema específico. (Zaragoza K., 2019)

Un modelo de dominio del problema es el área de experiencia o aplicación que necesita conocerse para resolver alguna dificultad, e incluye las reglas, procesos y sistemas existentes que necesitan una solución que mejore la operación. Así, cuando un analista desea obtener este modelo necesita conocer el dominio y la terminología. Debido a que, en muchas ocasiones, los analistas desconocen esa terminología y no tienen la información necesaria para llevar a cabo este modelado, se necesita la intervención del experto del dominio (Ej. La persona que realiza una actividad de un proceso involucrado).

En consecuencia, se identificaron los siguientes problemas: El primero, es que puede presentarse una mala comunicación entre el analista y el experto en el dominio porque no hay un entendimiento respecto a los significados de las palabras usadas en las TICs y la terminología del dominio del problema. Esto conlleva a mantener largas entrevistas con el experto, y mayor número de interrupciones por parte de ambas partes, a fin de conocer el significado de las palabras utilizadas. El segundo problema identificado es que un analista espera que el usuario explique su proceso de una forma sistematizada y organizada, pero la realidad es que esto no sucede así, debido a que el interesado explicará su trabajo según se vaya acordando e incluso conforme lo va desempeñando. El tercer problema, es que el entrevistado puede omitir información relevante, en consecuencia, el analista no podrá identificar los elementos importantes y representar el conocimiento. El cuarto es que, si la manera de comunicarse es textual, el usuario requiere ser competente en redacción (requiere conocimientos gramaticales y ortográficos adecuados). Finalmente el quinto problema es que el entrevistado puede tener poca disponibilidad para ayudar al

analista porque la entrevista se desarrolla en horario laboral, y se pueden tener interrupciones ocasionadas por personas interactuando en los procesos involucrados.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Realizar por medio de la voz entrevistas estructuradas para identificar entidades y flujo de transacciones en un dominio administrativo.

Este objetivo disminuye las problemáticas mencionadas porque: 1º.- Al realizar entrevistas estructuradas se mejora la situación de la mala comunicación y las largas entrevistas por no conocer la terminología. Por otro lado se ayuda al experto a explicar el proceso de manera ordenada, así mismo evitamos que el experto omita información relevante ya que se ira preguntando de acuerdo al modelado, y por lo tanto, se facilita el tratamiento de la información recolectada con el propósito de identificar los elementos importantes del dominio utilizando técnicas de lenguaje natural; 2º.- El uso de la voz aminora la necesidad de habilidades para comunicarse usando textos; 3º.- La realización asíncrona de las entrevistas mejora la situación de la disponibilidad e interrupciones dentro de la entrevista, debido a que podrá realizarse en un periodo disponible dentro del horario de trabajo.

1.2.2 OBJETIVOS ESPECÍFICOS

- Obtener un framework o plataforma que procese la voz y el texto para la creación del chatbot que llevará acabo la entrevista con el interesado (usuario).
- Identificar las entidades principales, entidades secundarias y el flujo de las transacciones, por medio de técnicas de Procesamiento de Lenguaje Natural.
- Realizar el modelado del dominio administrativo mediante el diagrama de Warnier Orr.

1.2.3 METODOLOGÍA DE SOLUCIÓN PROPUESTA

Como se puede observar en la **Figura 1**, la propuesta de solución está dividida en tres fases. La primera fase es la construcción de un chatbot (programa informático con el que es posible mantener una conversación), el cual se comunicará con el experto del dominio a través de la

voz. Este contendrá una base de conocimientos con un vocabulario suficiente para el dominio administrativo, y **que** obtendrá el historial de la conversación en forma de texto.

En la segunda fase el texto obtenido del chatbot se procesará por medio de técnicas de Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) a fin de identificar los elementos importantes.

En la tercera fase con la información identificada, se realizará el modelado del dominio.

Las tres fases mencionadas anteriormente, se explicarán a mayor detalle en el capítulo 4.

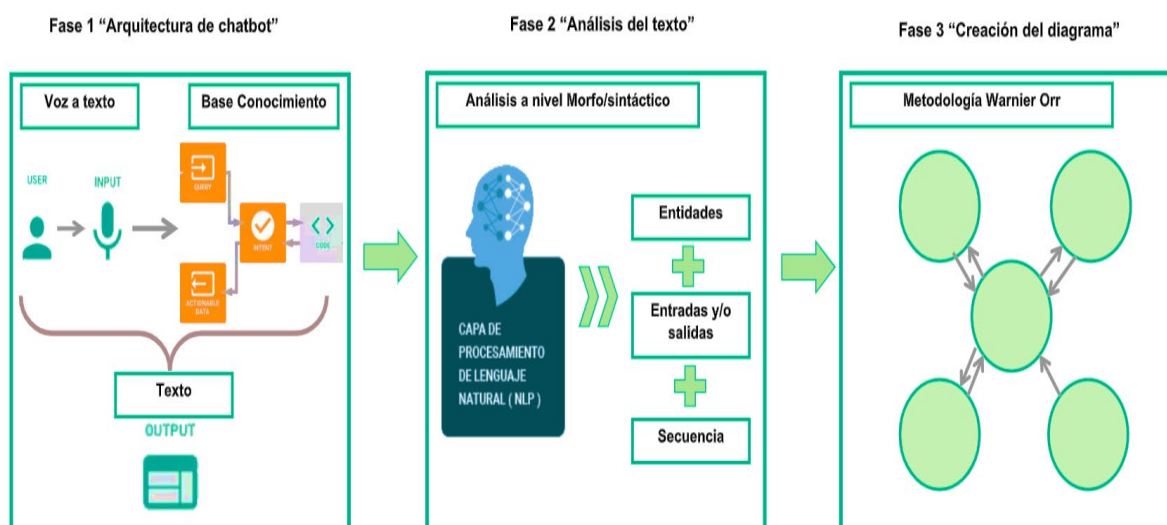


Figura 1- Solución Propuesta

1.3 ALCANCES

- Generar un marco de comparaciones de las plataformas de construcción de chatbots
- Obtener el texto de la conversación desde el reconocimiento de voz
- Construir el chatbot
- Etiquetar el historial de conversación del caso de estudio
- Obtener la información necesaria (entidades y transacciones)
- Crear el diagrama de entidades a nivel usuario

1.4 LIMITACIONES

- Acceder a las fuentes de datos del CENIDET por medio del CONRICYT: ACM digital library, ScienceDirect, IEEE Xplore, JSTOR y Springer Link.
- Consideraron artículos disponibles en texto completos.
- Utilizar una aplicación libre para generar el chatbot con reconocimiento de voz
- Aplicar solamente al idioma español
- Aplicar el procesamiento al texto resultante de la conversación de voz a texto a nivel análisis
- Aplicar solamente al caso de estudio administrativo del Departamento de Ciencias de la Computación
- Realizar manualmente el diagrama de entidades a nivel usuario

1.5 ORGANIZACIÓN DEL DOCUMENTO

Dentro de este documento encontraremos la organización de los capítulos de la siguiente manera:

En el capítulo 2 que corresponde al “Marco Teórico” se definen los conceptos utilizados a lo largo de este documento, para una mejor comprensión en la lectura.

En el capítulo 3 que corresponde al “Estado del Arte”, se presentan los antecedentes y trabajos relacionados para esta investigación. Se consideran como antecedente aquellos trabajos que se hayan realizado dentro del CENIDET, y los trabajos e investigaciones realizadas fuera de CENIDET serán los trabajos relacionados.

En el capítulo 4 que corresponde a la “Metodología y Resultados” se describen las fases que componen la metodología y los recursos utilizados en cada etapa para poder llevarla a cabo. Así como las tareas a realizar para completar cada una de las fases descritas y los resultados obtenidos dentro de cada etapa.

A lo largo del capítulo 5 correspondiente a las “Pruebas” se presenta un caso de prueba con base al estándar IEEE (2008) 829-2008- IEEE Standard for Software and System Test Documentation.

En el capítulo 6 correspondiente a las “Conclusiones” se analizan los resultados obtenidos, se presentan los hallazgos encontrados, y la experiencia obtenida de la investigación, así como los trabajos futuros y las aportaciones de esta investigación.

En el anexo A se encuentra el artículo presentado en la 3era Jornada de la Ciencia y la Tecnología.

En el anexo B, se presenta el resumen de la ponencia con la que se participó con en el Congreso Internacional de Lingüística Computacional y de Corpus. Tercero.

En el anexo C, se presenta el manual de cómo crear intenciones y entidades dentro de DialogFlow.

En el anexo D se presenta el código en Java de la clase utilizado.

En el anexo E se presenta el código en Ruby de la clase utilizado.

Capítulo 2

Marco Teórico

CAPITULO 2. MARCO TEÓRICO

En este capítulo se presentan los conceptos principales que van a establecer el ámbito contextual que se ocupan en esta investigación. Conceptos como: modelo de dominio, diagrama de entidades, Freeing, etiquetas Eagles, etc. Los cuales ayudan a tener un mejor entendimiento de todo el documento.

Este capítulo está dividido en dos secciones, en la primera sección se ubican términos de Ingeniería de Software, y en la segunda sección se incluyen los términos de Procesamiento de Lenguaje Natural.

SECCIÓN 1

2.1 INGENIARÍA DE SOFTWARE

2.1.1 Dominio

(Czarnecki, 2000) argumenta que un dominio es un área de conocimiento que incluye:

- Un conjunto de conceptos y terminología que comprenden los expertos del área.
- El conocimiento de cómo construir sistemas de software (o partes del sistema de software) en esa área. (citado en (Zaragoza K., 2019), p.18).

En otras palabras, por dominio se entiende aquello que se va a resolver, de lo que se trata el problema. Por ejemplo. El dominio será el negocio, la empresa, etc. (el problema que se va a resolver) esto podrá incluir las reglas, los procesos, el sistema, como opera. A veces, se tendrá que descomponer en subdominios, es decir, en el área contable, área de recursos humanos, etc.

2.1.1.1 Dominio del problema

En (Madeja, 2020) se describe que el objetivo principal de un sistema a desarrollar se centra en el alineamiento del sistema con las necesidades de negocio, con las posibles técnicas reales. Una fase muy importante, por ende, es la identificación de las necesidades de negocio.

En consecuencia, el objetivo principal del "estudio del dominio del problema" es obtener un conocimiento suficiente para poder comunicarse eficazmente con clientes y usuarios, comprender su negocio, entender sus necesidades y poder proponer una solución adecuada.

Por consiguiente, un dominio del problema es el conjunto de conceptos interrelacionados que es necesario conocer para entender el negocio del cliente, y, por lo tanto, para poder entender sus necesidades y proponer una solución adecuada.

2.1.1.2 Dominio de la solución

El siguiente texto citado en (Hernández G. S., 2019) describe el dominio de la solución. Es el dominio en el que los ingenieros utilizan su creatividad para resolver problemas. La característica principal que diferencia el dominio de la solución del dominio del problema es que, invariablemente, la ingeniería de requisitos se ubica en el dominio de la solución y describe la estructura y elementos adicionales a los del problema, que soportan una solución y consta de un conjunto dado de requisitos. En el dominio del problema, la ingeniería de requisitos comienza con un objetivo vago o una lista de deseos.

Dependiendo del grado en que los requisitos de entrada para el dominio de la solución están "bien formados" se desprende la calidad de la solución que utilizarán las personas, dentro de la organización del cliente que utilizará dicha solución. En un mundo ideal, todos los requisitos serían claramente articulados, requisitos que se pueden probar individualmente (Hull, 2017).

2.1.2 Modelo

(kimmel, 2008) refiere que un modelo es como una colección abstracta que puede representar diferentes aspectos de un sistema. De acuerdo a (Object Management Group, 2015), en el contexto de MDA, un modelo debe incluir el conjunto de información acerca de un sistema que está dentro de su dominio, las reglas de integridad aplicables a ese sistema, y el significado de los términos utilizados. (citado en (Zaragoza K., 2019), pp. 18).

Se entiende por modelo la forma en que se va a resolver (es como se va a plantear la solución)

- Seguramente será mediante una abstracción de la realidad.

- Es la especificación formal de una Función, Estructura, Comportamiento. Generalmente, esto se realiza mediante un contexto y un punto de vista en específico, frecuentemente representado por una combinación de dibujo y texto.

2.1.2.1 Modelo del dominio de problema

En (Taylor, 2008) los autores plantean un modelo de dominio como un conjunto de artefactos que capturan información sobre un dominio del problema. En un modelo de dominio se define un vocabulario para el dominio, se describen las entidades y datos en el dominio, se define cómo las entidades y los datos se combinarán para proporcionar características. Además, proporciona la base para la estandarización de las descripciones de los problemas a resolver en el dominio (citado en (Zaragoza K., 2019), p.19).

Con las definiciones anteriores, se llega a la conclusión de que con un modelo de dominio podemos lograr la organización y estructuración del problema, identificando los actores, componentes, relaciones y el alcance que se tendrá.

2.1.3 Diagrama de entidades

(Orr K. T., 1981), (citado en (Zaragoza K., 2019), p.20) Manifiesta que los diagramas de entidades ayudan a definir los requerimientos de una aplicación, de manera clara, completa y coherente, además de que esta metodología fue propuesta por este autor.

Orr propuso diferentes diagramas de entidades y son:

- Diagramas de entidades a nivel usuario.
- Diagramas de entidades combinado.
- Diagrama de entidades a nivel aplicación.

(González, 2006) Todos estos diagramas permiten mostrar una representación abstracta y estática, de la forma en que trabaja actualmente la organización, es decir, un modelo conceptual. Además, permiten la intervención de los expertos en el dominio de la aplicación cuando se inicia con elementos proporcionados para la representación estática del Diagrama de Entidades a Nivel Aplicación. Dentro los diagramas de entidades se manejan los siguientes conceptos generales:

- **Entidad.** - Son todos los usuarios involucrados en el dominio del problema. Estas entidades se dividen en: internas (pertenecientes a la organización) y externas (que no pertenecen a la organización).
- **Transacciones.** - Se representan por flechas para indicar el conjunto de todas las interacciones entre entidades de la organización (internas) y entidades del ambiente (externas).

El diagrama mencionado en la sección 2.1.3.1. Diagrama de entidades a nivel usuario, es el diagrama que se entrega como producto final de esta investigación, los demás diagramas presentados a continuación, solo se muestran como teoría para la mejor comprensión del documento.

2.1.3.1 Diagrama de entidades a nivel usuario

Este diagrama es un medio para recopilar información inicial a través de investigaciones y entrevistas. Estos diagramas proporcionan un mecanismo para capturar una vista general del dominio del problema, que se soportará mediante la aplicación a desarrollar.

Se deben identificar todas las entidades principales que interactúan en el contexto de la aplicación, dado que por cada usuario involucrado en el dominio del problema/aplicación se genera un diagrama, de acuerdo a las siguientes reglas (Orr K. T., 1981):

- a) Escribir el nombre de la organización, usuario y aplicación en la parte superior del diagrama.
- b) Colocar el nombre del usuario (participante) principal en una elipse, colocada en el centro del diagrama.
- c) Organizar las entidades que están relacionadas con el usuario principal.
- d) Dibujar las flechas (interacciones) entre las entidades.
- e) Revisar el diagrama y volver a dibujar si es necesario.

Cuando se terminan de dibujar todos los Diagramas de Entidad a Nivel Usuario, de todos los actores involucrados, se tienen las representaciones separadas de las tareas que abarca la aplicación. Si se requiere tener una idea completa, se integran todos los diagramas en el Diagrama de Entidades Combinado (González, 2006).

2.1.3.2 Diagrama de entidades combinado

Como su nombre lo indica este diagrama representa la vista general de todas las entidades involucradas en el sistema.

Para crear el diagrama combinado se tienen las siguientes reglas:

- a) Identificar las entidades y relaciones en cada diagrama de entidades a nivel usuario.
- b) Unir las entidades que sean iguales o similares, encontradas en los diagramas a nivel usuario, para evitar redundancias.
- c) Descartar aquellas entidades que no son necesarias.

En el diagrama de entidades combinado se tiene mucha información, incluye todas las interacciones entre las entidades relacionadas con el sistema, como se mencionó anteriormente. Para tener sólo las interacciones críticas se construye el Diagrama de Entidades a Nivel Aplicación, que se logra dibujando una frontera alrededor de las entidades internas. Se representa la aplicación cuando se consolidan todas las entidades internas en una sola elipse (González, 2006).

2.1.3.3 Diagrama de entidades a nivel aplicación

El diagrama de entidades a nivel aplicación muestra las principales entidades que se incluirán en la aplicación.

El proceso para desarrollar el diagrama es el siguiente (Orr K. T., 1981):

- a) Dibujar una frontera alrededor de las entidades internas para representar la aplicación en el diagrama de entidades combinado.

- b) Dibujar el diagrama a nivel aplicación, consolidando todas las entidades internas en una sola elipse que representa a la empresa (aplicación, dentro de la organización). Identificando las transacciones que interactúan con las entidades externas.

A partir este diagrama, es posible identificar los objetivos principales del sistema (aplicación). Para definir estos objetivos, se parte de la entidad principal del diagrama de entidad a nivel aplicación, donde cada transacción representa un objetivo. Si la flecha que simboliza la transacción sale de la entidad principal, entonces el objetivo se declara como “Enviar nombre de la transacción”. En caso que la flecha vaya en dirección contraria, se declara “recibir nombre de la transacción”.

Después de definir los objetivos y de elabora el diagrama de entidades a nivel aplicación, se construyen los diagramas de línea de producción. Estos diagramas están fuera del alcance de esta investigación.

2.1.3.4 Representación de los diagramas

En esta apartado se muestra una representación de los diagramas descritos anteriormente. El diagrama de entidades a nivel usuario es un medio para recopilar información inicial a través de investigaciones y entrevistas. En la **Figura 2** y la **Figura 3**, se muestran dos diagramas de entidades a nivel usuario, las cuales se emplean para generar el segundo diagrama, el diagrama de entidades combinado.

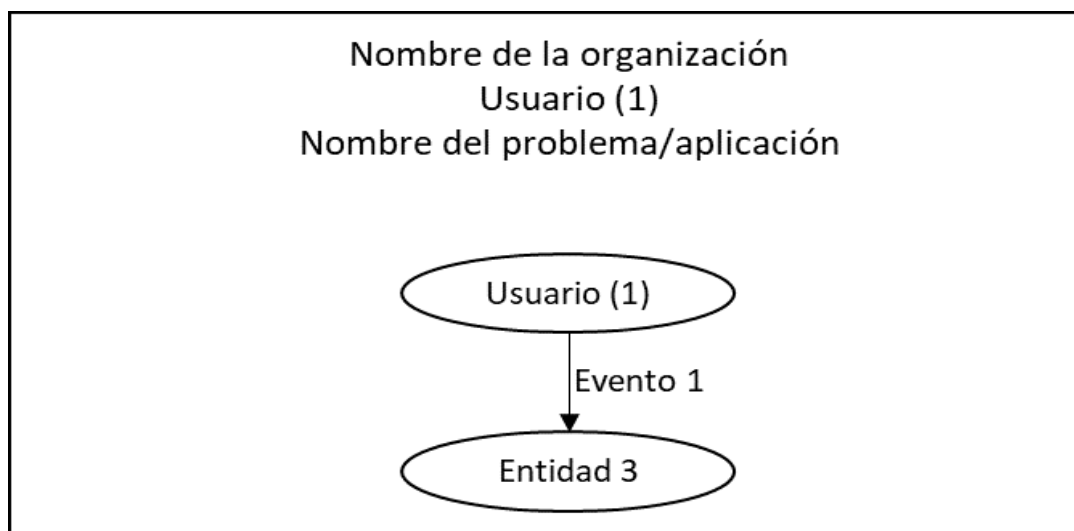


Figura 2.- Diagrama de entidades a nivel usuario (1) (Zaragoza K., 2019)

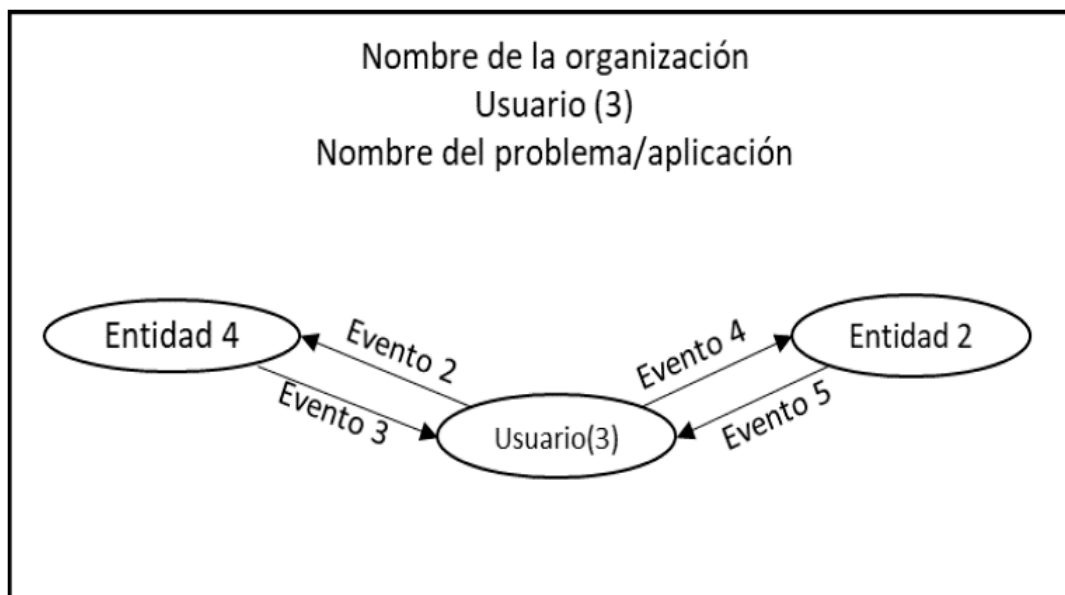


Figura 3.- Diagrama de entidades a nivel usuario (3) (Zaragoza K., 2019)

Al construir el diagrama de entidades combinado a nivel usuario se pueden descubrir inconsistencias en la información (eventos). Por ejemplo, una transacción solo puede mostrarse en un diagrama, y no verse reflejada en otros diagramas.

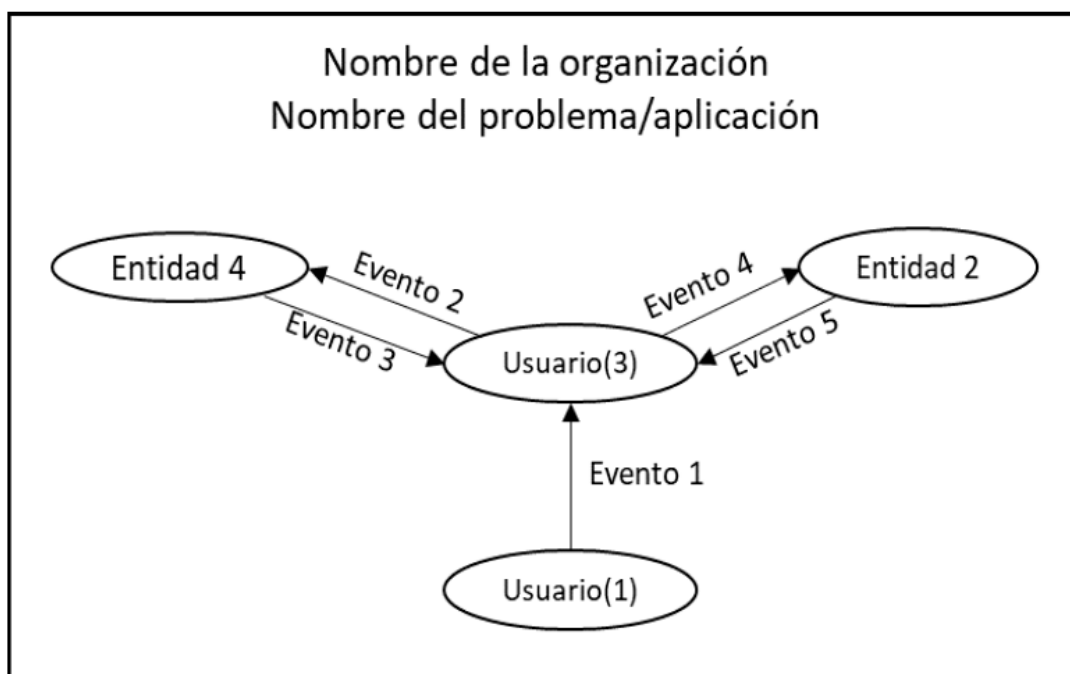


Figura 4.- Diagrama de entidades combinado (Zaragoza K., 2019)

El diagrama de entidades combinado contiene una gran cantidad de información, dado que se representan todas las interacciones dentro del sistema. Tomando como referencia el diagrama anterior (ver **Figura 4**), se obtiene el diagrama de entidades a nivel aplicación, donde se delimitará mediante una línea puntada las entidades que formarán parte del sistema, como se muestra en la **Figura 5**.

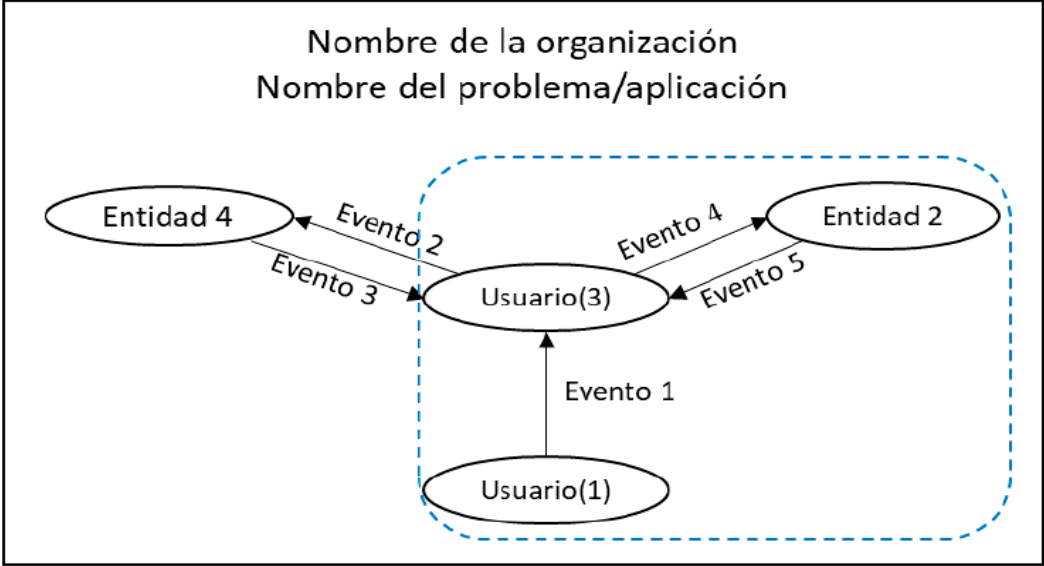


Figura 5.- Diagrama de entidades combinado con frontera (Zaragoza K., 2019)

Una vez seleccionadas las entidades internas, se define una nueva entidad, que representa la aplicación o problema. En este ejemplo, la nueva entidad es Aplicación (ver **Figura 6**).

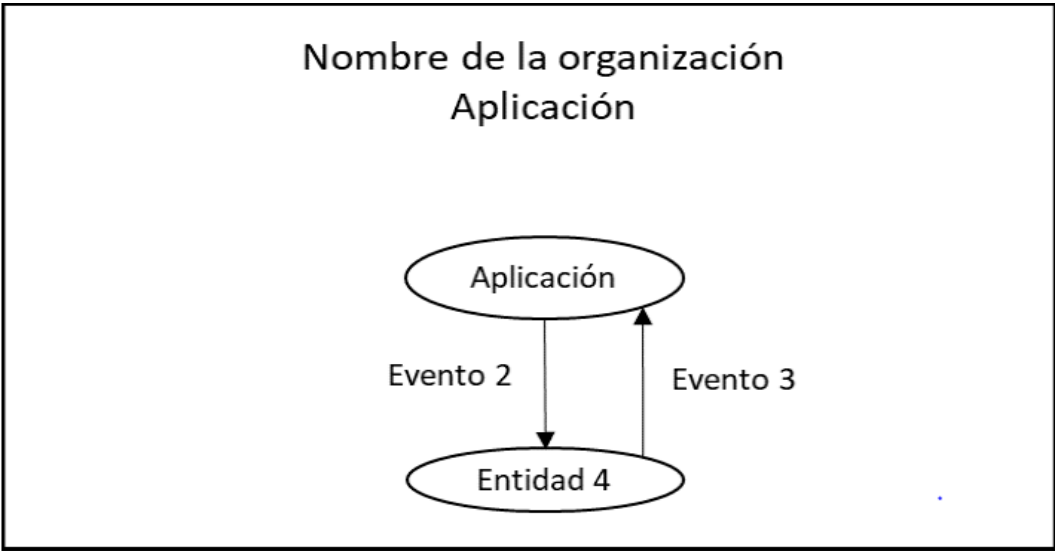


Figura 6.- Diagrama de entidades a nivel aplicación (Zaragoza K., 2019)

SECCIÓN 2

2.2 LENGUAJE NATURAL

2.2.1 Lenguaje

En (Raffino, 2020) se define al lenguaje como un sistema de signos orales, escritos o gestuales, que a través de su significado y la relación permiten que las personas puedan expresarse para lograr el entendimiento con otras personas.

(Cortez, 2009) refiere que un lenguaje se puede definir de diferentes formas: desde el punto de vista funcional lingüístico se define como una función que expresa pensamientos y comunicaciones entre la gente. Esta función puede realizarse mediante signos escritos (escritura) o mediante señales y vocales (voz). Desde un punto de vista formal se define como un conjunto de frases, que generalmente es infinito y se forma con combinaciones de elementos tomados de un conjunto (usualmente infinito) llamado alfabeto, respetando un conjunto de reglas de formación (sintácticas o gramaticales) y de sentido (semánticas).

2.2.2 Lengua

(Gelbukh, 2010) plantea que la lengua es un sistema (conjunto ordenado e interrelacionado) de signos (lingüísticos), elegidos de manera arbitraria, articulados conforme a ciertas reglas.

A diferencia del habla la lengua hace referencia a un sistema específico del que se valen las personas para comunicarse. Para que esto funcione, ese sistema debe ser retenerse en la memoria de los hablantes y debe conocerse por individuos con los que se quiere establecer la comunicación en cuestión.

2.2.3 Habla

(Gelbukh, 2010) menciona que el habla es la realización concreta de la lengua (hecha en cada momento por cada hablante), es decir, el habla es poner en uso aquel sistema de signos por parte de un grupo de individuos. Este término alude al acto individual y voluntario en la que se eligen los signos concretos que se necesitarán para entablar una comunicación.

2.2.4 Lenguaje Natural

(Cortez, 2009) menciona que al querer definir qué es lenguaje natural, nos hacemos la pregunta ¿Qué surgió primero las reglas gramaticales o el lenguaje?

El lenguaje natural(LN) es el medio que utilizamos de manera cotidiana para establecer nuestra comunicación con las demás personas. El LN ha venido perfeccionándose a partir de la experiencia a tal punto que puede utilizarse para analizar situaciones altamente complejas y razonar muy sutilmente. Los lenguajes naturales tienen un gran poder expresivo y su función y valor como una herramienta para razonamiento. Por otro lado, la sintaxis de un LN puede modelarse fácilmente por un lenguaje formal, similar a los utilizados en las matemáticas y la lógica.

2.2.5 Reconocimiento del habla

(Gaikwad, 2010) manifiesta que el habla es el principal modo de comunicación entre los seres humanos y también la forma más natural y eficiente de intercambiar información entre humanos. Por lo tanto, es lógico que el próximo desarrollo tecnológico sea el reconocimiento de voz en lenguaje natural para el área de interfaz humano computadora.

El reconocimiento de voz se puede definir como el proceso de conversión de la señal de voz a una secuencia de palabras mediante un algoritmo implementado como un programa de computadora (citado en (Hernández G. S., 2019). P 24). En otras palabras, el reconocimiento del habla es el conocimiento de los sonidos del lenguaje tanto fonético como fonológico.

2.2.6 Fonética

(Jurafsky, 2008) refiere que la fonética es la producción y percepción de los sonidos de una lengua con respecto a sus manifestaciones acústicas.

La fonética consiste, fundamentalmente, en una aproximación al aspecto físico del lenguaje verbal y a los distintos modos en que éste puede articularse, es decir, a la manera específica en que los hablantes de una misma lengua la ponen en práctica. (Raffino, 2020).

2.2.7 Fonología

(Jurafsky, 2008) plantea que la fonología es el funcionamiento del sonido a nivel abstracto o mental. Ésta también estudia cómo se deben usar cada una de las articulaciones de los órganos del habla para que los sonidos puedan ser dichos de una manera adecuada según su acento o entonación.

Según (Saussure, 1945) la fonología es del (francés *phonologie*). Y está fuera del tiempo, ya que el mecanismo de la articulación queda siempre semejante a sí mismo. Y lejos de confundirse estos dos estudios, ni siquiera se pueden oponer. El primero es una de las partes esenciales de la ciencia de la lengua; la fonología, en cambio —hay que repetirlo—, no es más que una disciplina auxiliar y no se refiere más que al habla.

2.2.8 Procesamiento de Lenguaje Natural

(Gelbukh., 2010) considera que por Procesamiento de Lenguaje Natural (PLN, denominado también NLP por sus siglas en inglés) se entiende la habilidad de la máquina para procesar la información comunicada, no simplemente las letras o los sonidos del lenguaje.

También el autor (Cortez, 2009) expresa que el Procesamiento de Lenguaje Natural (NLP) es una de las tareas fundamentales de la Inteligencia Artificial (IA) es la manipulación de lenguajes naturales usando herramientas de computación. En ésta, los lenguajes de programación juegan un papel importante, ya que forman el enlace necesario entre los lenguajes naturales y su manipulación por una máquina. El NLP consiste en la utilización de un lenguaje natural para comunicarnos con la computadora, debiendo ésta entender las oraciones que se le proporcionen. El uso de estos lenguajes naturales, facilita el desarrollo de programas que realicen tareas relacionadas con el lenguaje o bien, desarrollar modelos que ayuden a comprender los mecanismos humanos relacionados con el lenguaje.

2.2.9 Freeling

(Padró, 2011) indica que Freeling es una librería de código abierto para el procesamiento multilingüe automático, que proporciona una amplia gama de servicios de análisis lingüístico para diversos idiomas. Freeling ofrece a los desarrolladores de aplicaciones de Procesamiento del Lenguaje Natural funciones de análisis y anotación lingüística de textos, con la consiguiente reducción del costo de construcción de dichas aplicaciones. Freeling es

personalizable y ampliable, y está fuertemente orientado a aplicaciones del mundo real en términos de velocidad y robustez. Está concebido como una librería sobre la cual se puedan desarrollar potentes aplicaciones de PLN, y orientado a facilitar la integración con las aplicaciones de niveles superiores de los servicios lingüísticos que ofrece.

(Alemany, 2005) refiere que Freeling es de código abierto (LGPL), con diccionarios y gramáticas para español, catalán e inglés. Tiene un diccionario del español que cubre más del 90% de la lengua, el diccionario de más cobertura de uso totalmente libre. Como se puede ver en la **Figura 7**, el ejemplo del procesamiento de la frase “**Ramón está sentado en la silla**”.

▼ Sentences					
Sentence 1					
Ramón	está	sentado	en	la	silla
ramón	estar	sentar	en	el	silla
NP00000	VMIP3S0	VMP00SM	SP	DA0FS0	NP00G00

Figura 7.- Ejemplo procesamiento de Freeling (Catalunya, 2020)

2.2.10 Análisis Morfológico

(Cervantinos, 2020) Define que un análisis morfológico es básicamente un proceso mediante el cual se procede a examinar cada parte que integra un todo. Si esta noción la aplicamos a la lingüística; se estaría refiriendo al análisis formal de las partes que constituyen a las palabras y oraciones.

El estudio de las estructuras, y así como de la composición de las palabras, implica realizar un estudio etimológico buscando inclusive cuales son las raíces grecolatinas que poseen algunas de ellas. De esta manera; se hace más fácil dividir las estructuras en lexemas y morfemas para proceder a un análisis más exhaustivo. Estas son las unidades gramaticales más pequeñas; las cuales tienen un gran valor ya que aportan datos lingüísticos. Determina las clases o formas de las palabras dentro de una oración. Se busca ubicarlas en distintas categorías gramaticales.

2.2.11 Análisis Sintáctico

(Cervantinos, 2020) expone que el análisis sintáctico de oraciones consiste en analizar y buscar cuales son las funciones sintácticas que cumplen cada una de las palabras que integran dicha oración, así como también se estudia cuáles son sus relaciones de concordancia y de jerarquía entre ellas.

Es posible que en un principio sea confundido con el análisis morfológico, éste se diferencia en que este tipo de análisis busca cuales son las categorías gramaticales a las que pertenecen cada una de las palabras que integran dicha oración. El primer paso que se debe realizar al momento de iniciar con el análisis es aislar cada una de las palabras para poder determinar de qué clase son: sustantivos, verbos, determinantes o adjetivos por mencionar algunos. Se procede a identificar cual es el género al que pertenecen, así como su número, tiempo, modo y muy importante su conjugación.

2.2.12 Etiquetas Eagles

(Diaz, 2019) señala que para la ejecución del análisis morfológico de texto es necesario el uso de etiquetadores los cuales generan un sistema de etiquetado que ayuda a conocer el valor léxico y morfológico de cada palabra. En el caso del etiquetamiento de palabras en idioma español el grupo Eagles, crea el modelo de etiquetamiento el cumple con la identificación de adjetivos, adverbios, artículos, determinantes, nombres, verbos, pronombres, conjunciones, numerales, interjecciones, abreviaturas, preposiciones y signos de puntuación. El modelo de etiquetamiento crea etiquetas basadas en las características de tiene cada palabra.

Este conjunto de etiquetas es para la anotación morfosintáctica de lexicones y corpus para todas las lenguas europeas. Así pues, está previsto que recojan los accidentes gramaticales existentes en las lenguas europeas. Es por esto que dependiendo de la lengua hay atributos que pueden no especificarse. Si un atributo no se especifica significa que expresa un tipo de información o, bien, que no existe en la lengua o que la información no se considera relevante, la infraespecificación de un atributo se marca con el 0.

A continuación, presentamos las etiquetas que el analizador morfológico utiliza para el castellano en formato de tabla y algunos ejemplos de cada categoría, para cada categoría se

presentan los atributos, valores y códigos que puede tomar. Por ejemplo, en la **Tabla 1** se puede ver cómo están organizadas las columnas:

Tabla 1.- Representación de las etiquetas Eagles

Etiquetas Eagles			
Posición	Atributo	Valor	Código
Columna 1	Columna 2	Columna 3	Columna 4

En la *columna 1* encontramos un número que hace referencia al orden y posición en que aparecen los atributos. La *columna 2* hace referencia a los atributos, el número de los cuales varía dependiendo de la categoría. En la *columna 3* encontramos los valores que puede tomar cada atributo y, finalmente, la *columna 4* representa los códigos que se han establecido para su representación. Las etiquetas en sí sólo son los códigos (columna 4) y se sabe a qué atributo pertenecen por la posición (columna 1) en la que se encuentran.

En la **Tabla 2**, se puede observar la nomenclatura para la categoría “Nombres”, que será la categoría de las etiquetas Eagles que se estará ocupado en este trabajo de investigación.

Tabla 2.- Nomenclatura de la categoría "Nombres" de las etiquetas Eagles

Nombres			
Pos	Atributo	Valor	Código
1	Categoría	Nombre	N
2	Tipo	Común	C
		Propio	P
3	Género	Masculino	M
		Femenino	F
		Común	C
4	Número	Singular	S
		Plural	P
		Invariable	N

5	Caso	-	0
6	Género Semántico	-	0
7	Grado	Apreciativo	A

Los atributos que corresponden a *Caso* y *Género semántico* no se especifican, por lo tanto, el valor será 0.

Los nombres tienen como lema la forma singular, tanto si es de género femenino como masculino o neutro. Para los nombres invariables, es decir, aquellos que tanto para el singular como para el plural presentan la misma forma, el lema coincide con la forma. Como se puede observar en la **Figura 8**.

Forma	Lema	Etiqueta
Antonio	antonio	NP00000

Figura 8.- Ejemplo de etiquetado "Nombre"

2.2.13 Intenciones DialogFlow

(Google, 2020) refiere que una intención dentro de la plataforma clasifica la intención del usuario final para un turno de conversación. Para cada agente se definen muchos intents (intenciones); las intenciones combinadas pueden manejar una conversación completa. Cuando un usuario final escribe o dice algo, lo que se denomina *expresión de usuario final*, Dialogflow hace coincidir la expresión del usuario final con el mejor intent en nuestro agente. La coincidencia de un intent también se conoce como *clasificación de intent*. Es decir, una intención siempre asignará las entradas del usuario a las respuestas. En cada intención, se van a definir ejemplos de declaraciones de usuarios que pueden activar el intent, y nos dice qué extraer de la expresión y cómo responder. Cada intención está compuesta por frases de entrenamiento, acciones, parámetros (son las entidades a las que llamará) y la respuesta a esa intención.

- **Frases de entrenamiento:** Estas son frases de ejemplo de algo que podrían decir los usuarios finales. Cuando una expresión de usuario final se parece a una de estas frases, Dialogflow hace una coincidencia con el intent. No tienes que definir todos los ejemplos posibles, ya que el aprendizaje automático incorporado de Dialogflow expande tu lista con otras frases similares.
- **Acción:** Se puede definir una acción para cada intent. Cuando un intent coincide, Dialogflow proporciona la acción a tu sistema, y puedes usar la acción para activar ciertas acciones definidas en tu sistema.
- **Parámetros:** Cuando un intent coincide en el entorno de ejecución, Dialogflow proporciona los valores extraídos de la expresión del usuario final como parámetros. Cada parámetro tiene un tipo, llamado tipo de entidad, que dicta cómo se extraen los datos. A diferencia de la entrada sin procesar del usuario final, los parámetros son datos estructurados que se pueden usar con facilidad para realizar alguna lógica o generar respuestas.
- **Respuestas:** Se deben definir las respuestas de texto, de voz o visuales para mostrar al usuario final. Pueden proporcionar respuestas al usuario final, solicitar más información al usuario final o finalizar la conversación.

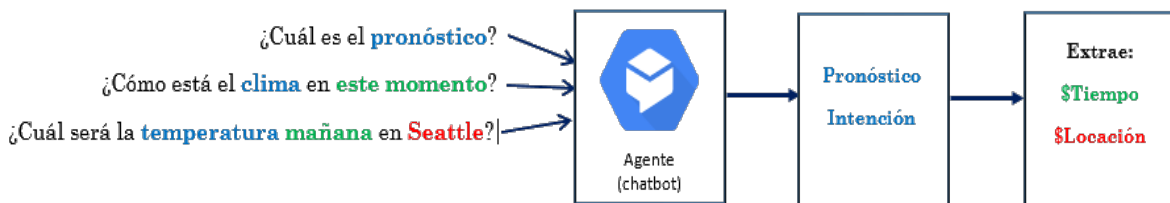


Figura 9.- Diagrama ejemplo de Intención

En la **Figura 9** se observa el comportamiento de una intención. Es decir, el chatbot tendrá una entrada por parte del usuario, estas entradas pueden ser preguntas (para esta investigación las entradas por parte del usuario serán respuestas) que contendrán palabras claves, las cuales serán reconocidas por el chatbot, en la figura anterior las palabras claves se pusieron en color azul. El segundo paso será encontrar la intención que coincida con dichas palabras, para este ejemplo la intención se llama “**pronóstico**”, esta tiene como frases de entrenamiento “**clima**

y **temperatura**”, esto es para que el chatbot reconozca las diferentes formas de decir o preguntar una misma cosa y sepa que intención se está llamando, y como último paso el chatbot extrae el tiempo (color verde) y la locación (color rojo), para poder responder adecuadamente lo que el usuario está preguntando.

2.2.14 Entidades DialogFlow

De acuerdo con (Google, 2020) se menciona que cada parámetro de intent tiene un tipo, denominado *tipo de entidad*, que determina de forma exacta cómo se extraen los datos de una expresión de usuario final. En otras palabras, las entidades son todas las palabras o conjunto de palabras o frases que se consideran equivalentes.

Cuando se analizan los detalles de la entidad, es importante entender términos más específicos:

- **Tipo de entidad:** Define el tipo de información que deseas extraer de la entrada del usuario. Por ejemplo, vegetal podría ser el nombre de un tipo de entidad.
- **Entrada de entidad:** Para cada tipo de entidad, hay muchas entradas de entidad. Cada entrada de entidad proporciona un conjunto de palabras o frases que se consideran equivalentes. Por ejemplo, si verdura es un tipo de entidad, puedes definir estas tres entradas de entidad:
 - zanahoria
 - cebollín, cebolla verde
 - pimentón, pimiento dulce

2.2.15 Contextos DialogFlow

(Google, 2020) plantea que los contextos son similares al contexto del lenguaje natural. Si una persona le dice "es de color naranja", necesita contexto para saber qué es de ese color. Del mismo modo, para que Dialogflow maneje una expresión de usuario final como esa, debe proporcionarse un contexto con el fin de que coincida de forma correcta con un intent. También nos ayudan a que nuestro chatbot tenga una memoria y sepa cuando se dispara o utiliza cada intent.

2.3 RESUMEN DEL CAPÍTULO

En este capítulo se presentaron los conceptos básicos para comprender el contexto del presente trabajo de tesis. Donde se describieron los conceptos utilizados durante todo el trabajo, tanto de Procesamiento del Lenguaje Natural como de Ingeniería de Software, entre estos términos se pueden resaltar, el dominio del problema y el dominio de la solución, el diagrama de entidades a nivel usuario, los conceptos que se utilizan en el Framework de DialogFlow, por ejemplo, los contextos, las frases de entrenamiento, las intenciones, etc. Durante los siguientes capítulos se estará haciendo mención de estos términos. Así mismo, para el estado del arte se realizó un mapeo sistemático para encontrar aquellos trabajos que aportaran información a este trabajo e incluso sirvieran como una guía, y en estos trabajos existen algunos que utilizan el reconocimiento del habla y el procesamiento del lenguaje natural.

En el siguiente capítulo se presentan los trabajos encontrados y los cuales tienen un aporte para este trabajo de investigación.

Capítulo 3

Estado del Arte

A continuación, se exhiben los resultados obtenidos del proceso de búsqueda de información con respecto a las investigaciones realizadas dentro y fuera del CENIDET, y que sirven de apoyo, punto de referencia y/o comparación para este trabajo de investigación.

3.1 ANTECEDENTES

Como se mencionó anteriormente en este apartado se describen de forma general las investigaciones que se han realizado dentro del CENIDET.

3.1.1 Mapeo sistemático del reconocimiento del habla, proceso del lenguaje natural y uso de ontologías para identificar el dominio del problema y los requerimientos de solución

En el trabajo de (Hernández G. S., 2019) , se presenta un Mapeo sistemático que reúne trabajos publicados acerca de las diferentes técnicas de reconocimiento del habla, el Procesamiento de Lenguaje Natural (NLP, por sus siglas en ingles) y el uso de ontologías en la Elicitación de requerimientos, la revisión bibliográfica de los trabajos se llevó a cabo del 2010 al 2019.

La aportación de este trabajo es evidenciar las técnicas más utilizadas dentro del Procesamiento de Lenguaje Natural (NLP) logrando la clasificación de cada técnica que incluye sus fases y descripciones correspondientes, esto se logró con la ayuda de un mapeo sistemático. Las técnicas que se observan en este trabajo son: pre-procesamiento de texto, análisis de texto, análisis de datos exploratorios, representación de texto, ingeniería de características y la extracción de patrones.

La aportación para esta investigación fue solo informativa, porque fue más sencillo identificar qué tipo de pre-procesamiento de texto se iba a requerir en esta investigación, e incluso ayudó a ya no realizar una revisión literaria para conocer las técnicas de lenguaje natural.

3.1.2 Editor para el Lenguaje de Dominio Especifico Diagramas de Entidad

El trabajo de investigación de (Zaragoza K., 2019), evalúa las diferentes herramientas de modelado que permiten la generación y aplicación de perfiles UML, que ayude a crear un editor con la finalidad de construir los diagramas de entidades a nivel usuario, diagramas combinados y de aplicación.

La aportación de este trabajo fue proporcionar un perfil UML para la construcción de diagramas de entidades, así como la identificación de una herramienta de modelado que permite obtener un editor mediante este perfil UML, el cuál ayuda a los desarrolladores a definir un modelo de dominio del problema sin requerir la intervención constante de los expertos del dominio. La herramienta que se investigó y utilizó fue la de *Enterprise Architect*.

Este trabajo aportó a esta investigación el conocimiento de la representación de los diagramas de entidades, cómo se realizan, los tipos de diagramas que se pueden crear y algunas herramientas para poder realizarlos de manera más sencilla.

En este trabajo de tesis se considera importante conocer el uso del reconocimiento del habla, procesamiento del lenguaje natural, el modelado del dominio del problema y dominio de la solución, técnicas y herramientas que ayuden a agilizar el proceso de atención al usuario que lleva el analista, sin embargo, se tiene muy poco conocimiento respecto a su estado, avance que hay, y herramientas que puedan ayudar al analista durante el proceso. Por lo tanto, se considera necesario realizar un primer inicio para conocer el dominio del problema y conocer los actores involucrados en el proceso.

3.1.3 Tabla Comparativa de los antecedentes

Los trabajos descritos en la sección anterior **3.1 ANTECEDENTES**, están resumidos en la **Tabla 3**, y muestran las investigaciones que se han estado realizando en el CENIDET, como se puede observar existen un trabajo anterior que se dedicó a la investigación del procesamiento de lenguaje natural, donde su principal objetivo fue recabar las referencias bibliográficas donde el lenguaje natural fuera empleado en la ingeniería de software. Al mismo tiempo, vemos un trabajo anterior que no utiliza el lenguaje natural, pero que estuvo

trabajando con el dominio del problema de la biblioteca del CENIDET, donde tuvo que representar a los actores involucrados en el proceso y a su vez, buscar una herramienta que ayudara a modelar este diagrama de entidades a nivel usuario.

Tabla 3.- Comparativa de antecedentes

Trabajo Antecedente	Uso del lenguaje natural	Uso de modelado del dominio (diagrama de entidades)	Creación de una herramienta (chatbot)	Conversión de voz a texto
Mapeo sistemático del reconocimiento del habla, proceso del lenguaje natural y uso de ontologías para identificar el dominio del problema y los requerimientos de solución. (Hernández G. S., 2019)	Se describe un panorama general acerca del reconocimiento del habla, procesamiento del lenguaje natural y uso de ontologías en apoyo al modelado del dominio del problema	No aplica	No aplica	No aplica
Editor para el Lenguaje de Dominio Especifico Diagramas de Entidad. (Zaragoza K., 2019)	No aplica	Se investigó una herramienta que ayudara a modelar el diagrama de entidades, el dominio del problema que se estuvo trabajando fue el de la biblioteca del Cenidet.	No aplica	No aplica

3.2 TRABAJOS RELACIONADOS

Los trabajos relacionados serán divididos en dos secciones. La primera sección está compuesta por las investigaciones más relevantes respecto a chatbots, procesamiento de lenguaje natural y sus técnicas. Y en la segunda sección, se describen las investigaciones sobresalientes de distintos autores en el tema de ingeniería de software y diagrama de entidades a nivel usuario. En cada investigación se describe de forma general el trabajo realizado por sus autores.

SECCIÓN 1 - LENGUAJE NATURAL

3.2.1 A Knowledge-Intensive Method for Conversational CBR

En este artículo (Aamodt, 2013), se habla del razonamiento basado en los casos conversacionales (CCBR, *Conversational Case Based Reasoning*), en el cual se comenta que el principal problema para crear un chatbot es el escoger las preguntas adecuadas. En el artículo se muestran cuatro tareas computacionales de un proceso computacional y ciertas características que se deben tomar en cuenta en la formulación de las preguntas. A continuación, se detallan las cuatro tareas mencionadas.

a) Inferencia de características (FI, *Feature Inferencing*)

En la tarea FI, se determina si existe una característica del problema que se puede deducir de la descripción del problema, esta característica se puede añadir a la descripción del problema automáticamente, en lugar de preguntarle al usuario.

b) Clasificación de preguntas (QR, *Question Ranking*)

En la tarea QR, se identifican preguntas discriminatorias y se clasifican antes de preguntarle al usuario, es decir, si la respuesta a la pregunta A se puede deducir de una de las posibles respuestas de la pregunta B, sería mejor saltarla la B y hacer la pregunta D.

c) Agrupamiento consistente de preguntas (CQC, *Consistent Question Clustering*)

En esta tarea, las preguntas con alguna relación semántica entre ellas deben agruparse y el orden de las preguntas en cada grupo debe establecerse intencionalmente.

d) Secuencia Coherente de preguntas (CQS, *Coherent Question Sequencing*)

En la tarea CQS, las preguntas formuladas en los ciclos secuenciales deben estar lo más relacionadas posibles, es decir, los contenidos semánticos de dos preguntas secuenciales deben evitar el cambio con demasiada frecuencia y de modo repentino.

Este artículo aportó a la investigación un apoyo para generar las preguntas correctas, teniendo una semántica adecuada pero que nos ayuden a recabar toda la información necesaria del dominio del problema a trabajar.

3.2.2 Procesamiento de lenguaje natural para estudiar completitud de requisitos

En el artículo (Litvak, 2016), el autor especifica que en la Ingeniería de Software se generan modelos que representan las necesidades de los clientes y/o usuarios (problema), así como las soluciones a dichos problemas. El desarrollo de modelos de requisitos en lenguaje natural facilita la interacción entre todos los involucrados, aunque contribuyen a generar ambigüedades. Por otro lado, se ha comprobado la existencia de omisiones en una cantidad muy superior a la imaginada intuitivamente. Por su propia naturaleza, las omisiones son notoriamente elusivas.

El autor menciona que diferentes estudios han atacado el tema de la completitud, obteniendo resultados limitados. Así, Litvak realizó estudios donde en la primera etapa se compara los modelos mediante la granularidad gruesa, debido a que solo se compararon los nombres de los símbolos, el modelo comparado fue el de *Léxico Extendido del Lenguaje* (LEL), el cuál ayudó a detectar omisiones significativas.

Este trabajo fue importante, debido a que Litvak ya había detectado algunas ambigüedades, y eso ayudó a que no se cometieran en este trabajo, y también sirvió para conocer nuevos modelos que ayudaran a detectar las omisiones de importancia.

3.2.3 Procesamiento semántico automático, enfocado en la coherencia textual, para apoyar la producción escrita de noticias

En el siguiente artículo (Cabrera, 2018), se presenta el diseño e implementación de un módulo de análisis semántico, enfocado en la predicción de la coherencia textual, programado en Python 3. Las etapas de implementación comprenden el trabajo realizado en el diseño de una herramienta para la recopilación automática del corpus (noticias sobre política), de otra destinada a preparar los textos reunidos para su procesamiento posterior, hasta llegar al diseño de la herramienta final que realiza el análisis de los textos.

El método empleado para esto es el Análisis Semántico Latente. El artículo concluye con la presentación de los resultados de las pruebas realizadas, con el fin de probar la herramienta mediante el procesamiento de textos, para observar su sensibilidad en la evaluación de la coherencia textual.

El aporte que dio este artículo para esta investigación fue informativo, debido a que realizaron un chatbot para detectar noticias sobre política, entonces su análisis lo hicieron en la etapa semántica, que es la que trabaja sobre el significado literal de las frases y enunciados, pero fue clave para saber hacia dónde dirigir la entrevista y enfocarlo al análisis morfo/sintáctico de esta investigación.

3.2.4 Herramientas Libres para Procesamiento del Lenguaje Natural

En este trabajo (Alemany, 2005), el autor describe que al intentar generar una herramienta semiautomática de comparación de oraciones, resulta imperioso recurrir a estrategias de Procesamiento del Lenguaje Natural. Utilizando herramientas que realizan el reconocimiento del idioma utilizado la segmentación de oraciones, palabras y secciones, y luego, dependiendo de la herramienta, un análisis morfológico, sintáctico y/o semántico.

Este artículo resulto importante para la investigación ya que nos explica en gran detalle los pasos que se deben seguir para hacer un pre procesamiento correcto del texto, el cual incluye: El reconocimiento del idioma, la segmentación de las palabras u oraciones, el análisis morfológico y análisis sintáctico. Que es al nivel que se estuvo trabajando en esta

investigación, con el texto obtenido de las entrevistas. Y termina por mencionar las herramientas que sirven para llevar a cabo este procesamiento.

3.2.5 Análisis de la dinámica del contenido semántico de textos

En este trabajo (Altszyler., 2015), el autor describe un proyecto que analiza la dinámica de distintos conceptos, desde un enfoque semántico, buscando identificar patrones temporales comunes en varios corpus de texto. En este trabajo se estudia la evolución semántica de conceptos a lo largo de las distintas novelas de la saga Harry Potter con el objetivo último de analizar variaciones del contenido semántico en textos utilizando herramientas de *Procesamiento de Lenguaje Natural* (NLP, por sus siglas en inglés). En este contexto, se muestra la primera aproximación a la comprensión del alcance y de las limitaciones de las herramientas clásicas de PLN para cuantificar la evolución del concepto “oscuridad” con el avance temporal de los libros. La saga de novelas de Harry Potter resulta ser un corpus de texto ideal para testear esas herramientas debido a que es de público conocimiento que dicha saga presenta un aumento gradual de la “oscuridad” a medida que se suceden los libros.

Dentro del campo del Procesamiento del Lenguaje Natural, la extracción de significados de textos escritos es, al día de hoy, un tema de gran interés. Dentro de esta sub-área, los métodos de cuantificación de distancia semántica entre palabras y /o documentos cumplen un rol fundamental. Existe una gran diversidad de algoritmos capaces de cuantificar la similitud semántica entre palabras, varios de los cuales parten del supuesto de que las palabras semánticamente relacionadas tenderán a utilizarse en contextos similares, y consecuentemente, a coexistir en documentos. Un algoritmo muy utilizado que parte de estas mismas hipótesis es el Análisis Semántico Latente (LSA por sus siglas en inglés).

Para el entrenamiento del LSA al corpus de texto crudo se le aplicó un filtrado de palabras frecuentes (se filtraron los “stopwords” del paquete NLTK para el idioma inglés) y se llevaron las palabras a su raíz con el algoritmo de Porter. Luego se entrenó un LSA de 300 dimensiones sobre la matriz de frecuencias términos-documentos normalizada según TF-IDF. Finalmente, se les aplicó el filtro de palabras frecuentes, se llevaron las palabras a su raíz y se eliminaron las palabras “Potter”, “Ron” y “Dark Lord” para evitar aportes espurios en los cálculos de distancia debido a la incapacidad de detectar polisemia del algoritmo.

Es por ello que, este trabajo se toma en cuenta en esta investigación para conocer cómo y cuándo implementar la similitud semántica junto con la técnica *Análisis Semántico Latente* (LSA, por sus siglas en inglés), el cual es capaz de identificar la cercanía de una palabra a un texto incluso en el caso en que el documento no contenga esa palabra, por otro lado, podría medirse con qué frecuencia se repiten algunos formatos durante el proceso, e incluso con que departamentos se tiene más comunicación. Esto para implementarlo en trabajos futuros.

3.2.6 Razonamiento basado en casos de conversación

En el artículo (Breslow, 2014), se muestra como una universidad crea una herramienta conversacional por medio del razonamiento basado en casos, El razonamiento basado en casos conversacionales (**CCBR**, *Conversational Case Based Reasoning*) fue la primera forma ampliamente difundida comercialmente exitosa de razonamiento basado en casos. Las herramientas comerciales de CCBR llevaron a cabo diálogos restringidos entre usuarios humanos y tareas específicas de soporte al cliente. Detallan su progreso en abordar tres de estos problemas: 1) simplificación de la creación de casos, 2) inferencia de diálogos y 3) planificación interactiva. En resumen, en este artículo se destacan los problemas importantes de CCBR, se evalúan los enfoques para resolverlos y se sugirieron alternativas a considerar para futuras investigaciones.

Este trabajo se tomó como referencia debido a que, en el primer artículo citado, mencionan este razonamiento basado en casos CCBR, pero a comparación del anterior artículo en este presentan las problemáticas que enfrenta este razonamiento, donde que no todo es tan funcional, y esas variaciones se tomaron en cuenta al momento de elegir las preguntas del chatbot. Por ejemplo:

- 1) Reutilizar las preguntas cuando sea posible
- 2) Ordenar el contexto antes de detallar las preguntas
- 3) Eliminar las preguntas que no distinguen los casos
- 4) Pedir sólo una cosa en una pregunta
- 5) Utilizar un número similar y reducido de preguntas en los casos.

3.2.7 Un chatbot de aprendizaje de refuerzo profundo

En este artículo (Iulian V. Serban, 2017), el autor describe el chatbot llamado MILABOT: un chatbot de aprendizaje de refuerzo profundo desarrollado por el Instituto de Algoritmos de Aprendizaje de Montreal (MILA) para la competencia del Premio Amazon Alexa. MILABOT es capaz de conversar con humanos sobre temas populares de pequeñas charlas a través del habla y el texto. El sistema consiste en un conjunto de modelos de recuperación y generación de lenguaje natural, que incluyen modelos basados en plantillas, modelos de bolsa de palabras, red neuronal secuencia a secuencia y modelos de red neuronal de variable latente.

La arquitectura de su sistema se inspira en el éxito de los sistemas de aprendizaje en conjunto. Estos sistemas consisten en muchos submodelos independientes combinados inteligentemente entre sí. Entre los ejemplos de estos sistemas de conjunto se incluyen el ganador del Premio Netflix (Koren, 2009), citado en (Iulian V. Serban, 2017), pp-2) que utiliza cientos de modelos de aprendizaje automático para predecir las preferencias de películas de los usuarios, e IBM Watson (Ferrucci, 2010), citado en (Iulian V. Serban, 2017), pp-2) el primer sistema de aprendizaje automático que ganó el juego de preguntas y respuestas Jeopardy! en 2011. Más recientemente, Google observó mejoras sustanciales en la construcción de un sistema de traducción automática neural basado en un conjunto (Wu, 2016), citado en (Iulian V. Serban, 2017), pp-2).

El aporte de este artículo para esta investigación fue conocer los diferentes modelos con los que se pueden construir un chatbot, entrenarlo y principalmente como se construyó el chatbot MILABOT, para llegar a concursar por un premio contra Alexa. Cabe mencionar que este chatbot se creó desde cero, sin ningún soporte de alguna herramienta.

3.2.8 Encuesta sobre las técnicas de diseño de Chatbot en los sistemas de conversación del habla

En el artículo de (Sameera, 2015), presenta una encuesta sobre las técnicas utilizadas para diseñar los Chatbots y se hace una comparación entre las diferentes técnicas de diseño de nueve documentos cuidadosamente seleccionados según los principales métodos adoptados.

Estos documentos son representativos de las importantes mejoras de los Chatbots en el último decenio. En el documento se examinan las similitudes y diferencias de las técnicas y se estudia en particular los Chatbots ganadores del premio Loebner (el premio Loebner es una competición de carácter anual que concede premios a un programa de ordenador que esté considerado por el jurado que lo compone, como el más inteligente de los que se han presentado.)

El habla humano-computadora está ganando impulso como técnica de interacción informática. Ha habido un reciente aumento en los motores de búsqueda basados en el habla y asistentes como Siri, Google Chrome y Cortana. Las técnicas de Procesamiento del Lenguaje Natural (PNL) como NLTK para Python pueden aplicarse para analizar el habla, y se pueden encontrar respuestas inteligentes diseñando un motor para proporcionar respuestas apropiadas similares a las humanas. Este tipo de programa se denomina Chatbot, que es el objeto de este estudio.

3.2.9 Tabla comparativa de los trabajos relacionado de Lenguaje Natural

Los trabajos presentados en 3.2 TRABAJOS RELACIONADOS, en la sección de lenguaje natural, y resumidos a continuación en la **Tabla 4**, son trabajos que están relacionados con esta investigación. Se ha estado trabajando con el reconocimiento de voz y todo lo que tiene que ver con el procesamiento del texto, desde limpiar el texto (quitar espacios, signos de puntuación, volver todas las palabras en minúsculas), analizarlo en niveles del lenguaje natural (morfológico, semántico, sintáctico), y buscando la similitud semántica entre palabras. Entre los trabajos sobresalientes están el de (Sameera, 2015), cuyo trabajo explica y estudia específicamente las técnicas de diseño de Chatbot en la última década. Aunque en el mercado aparecen algunos productos comerciales que involucran estas técnicas (Ej. Siri, Alexa, Microsoft Cortana).

Tabla 4.- Trabajos relacionados PLN

Trabajo Relacionado	Técnica o método utilizado	Creación de un chatbot	Reconocimiento del habla	Problema resuelto
A Knowledge-Intensive Method for	Razonamiento basado en los	No aplica	No aplica	Muestra cómo realizar las preguntas correctas

Conversational CBR	casos conversacionales			para generar un chatbot coherente
Procesamiento de lenguaje natural para estudiar completitud de requisitos	Granularidad gruesa y Léxico Extendido del Lenguaje	No aplica	Sí	Se utilizó la granularidad para disminuir las omisiones y trata de evitar las ambigüedades al obtener los requerimientos desde la voz
Procesamiento semántico automático, enfocado en la coherencia textual, para apoyar la producción escrita de noticias	Análisis Semántico Latente	Sí	No aplica	Los autores crearon un chatbot para recolectar noticias sobre política y procesaron los textos para evaluar la coherencia textual
Herramientas Libres para Procesamiento del Lenguaje Natural	Análisis morfo-sintáctico	No aplica	No aplica	Describe todo el procesamiento del lenguaje natural, y todos los niveles del procesamiento
Análisis de la dinámica del contenido semántico de textos	Análisis Semántico Latente y NLTK	No aplica	No aplica	Se detectó por medio del análisis semántico latente y el procesamiento de textos, la aproximación de las palabras que tienen los libros de Harry Potter con la palabra “oscuridad” incluso aunque dentro del entrenamiento no existiera esa palabra.
Razonamiento basado en casos de conversación	Razonamiento basado en casos conversacionales	Sí	No aplica	En este artículo se detallan las problemáticas que presenta la técnica CCBR, y que se tienen que conocer para generar una

				buena guía en la atención al cliente
Un chatbot de aprendizaje de refuerzo profundo	Conjuntos de modelos de respuesta	Sí	No aplica	Los modelos de respuesta toman como entrada un diálogo y como salida una respuesta en texto de lenguaje natural. Además, los modelos de respuesta también pueden producir uno o varios valores escalares, indicando su confianza interna.
Encuesta sobre las técnicas de diseño de Chatbot en los sistemas de conversación del habla	Reconocimiento Automático del Habla (ASR) y NLTK	Sí	Sí	Los Chatbots diseñados para sistemas de diálogo en los estudios seleccionados se limitan, en general, a aplicaciones particulares. Los "chatbots" de propósito general necesitan mejoras mediante el diseño de bases de conocimientos más completas.

SECCIÓN 2 – INGENIERÍA DE SOFTWARE

En esta sección se presentan los trabajos relacionados con esta investigación en el área de Ingeniería de Software.

3.2.10 Structured Requirements Definition

En el libro (Orr K. T., 1981) el autor nos describe el porqué de los diagramas de entidades a nivel usuario, cómo son utilizados y por qué deberían ser una primera aproximación del problema entre el interesado y el analista. Dentro del libro también se utilizan varios ejemplos y problemáticas que el autor va graficando por medio del diagrama. Esto nos ayuda a conocer los elementos principales, y cómo se van relacionando los elementos incrementalmente, paso a paso.

La metodología menciona que primero debemos identificar claramente la entidad principal y colocarla en el centro. Posteriormente, se deben identificar todas aquellas personas, organizaciones o sistemas, con las que interactúa nuestra entidad principal, ellas serán las entidades secundarias y serán colocadas alrededor de la entidad principal mediante óvalos. Y finalmente, se deben colocar las transacciones (mensajes) que se envían entre la entidad primaria y las entidades secundarias. Los mensajes se representarán mediante flechas donde la cabeza de la flecha mostrará la dirección del mensaje.

El libro mencionado es de suma importancia ya que el objetivo de la investigación es modelar e identificar a las personas involucradas en el dominio del problema administrativo. Por lo que, para llevar a cabo ese modelado se seguirá esta metodología, y los pasos descritos en ella.

3.2.11 Representación de la regla formal y verificación de los requisitos del lenguaje natural utilizando una ontología

En el artículo mencionado el autor ((Sadoun, 2014) citado en (Hernández G. S., 2019), pp-12) menciona que el desarrollo y funcionamiento de un sistema de software está basado en los requerimientos, donde estos son escritos en un lenguaje natural, por lo tanto, las personas han buscado formas que complementen esos requerimientos y los ayuden a modelarlos y representados de una manera estandarizada, durante este artículo se describen los requerimientos del comportamiento del sistema, es decir, requerimientos del usuario, y muestra que estos requerimientos fueron estandarizados y representados en una ontología OWL Maude, que es una representación formal basada en términos algebraicos, que permiten describir la dinámica del sistema.

Es por ello que este artículo se pone en esta investigación, como muestra de que, en años anteriores, los requerimientos se obtienen desde el lenguaje natural e informal, y se ha estado trabajando para poder tener una representación formal sin alejarse de las necesidades y descripciones del cliente.

3.2.12 Modelado de dominios con metodología de objetos y procesos

En (Sturm, 2006) plantea que, durante el análisis de los dominios, el dominio se modela de forma reutilizable, y la mayoría de los enfoques de análisis sufren de baja accesibilidad, expresividad limitada y formalidad débil. Por lo que, ellos en su artículo presentan un enfoque formal, accesible y expresivo para este análisis, y lo hacen extendiendo la metodología de objetos y procesos (OPM, por sus siglas en inglés). La ingeniería de dominio incluye varias actividades, entre ellas:

- Análisis del dominio
- Diseño del dominio
- Implementación del dominio.

El artículo se enfoca en el análisis del dominio y en cómo utilizarlo mejor para modelar aplicaciones dentro de un dominio específico, a su vez, hacen un experimento donde verifican la extensión propuesta puede mejorar la calidad del modelo en comparación con la calidad a la que se llega sin la extensión. Sus resultados experimentales muestran que, cuando a las personas se les presentó un conjunto de requisitos, los sujetos que usaron OPM con la extensión de análisis de dominio llegaron a un modelo de sistema que era mejor que el modelo de sistema al que llegaron los sujetos que usaron sólo OPM. La extensión utilizada se llama enfoque de modelado de dominio basado en la aplicación (ADOM, por sus siglas en inglés) se basa en una arquitectura de tres capas: la capa de aplicación, la capa de dominio y la capa de lenguaje de modelado. Influenciado por el Mecanismo de Meta-Objeto (MOF).

El artículo es relevante en esta investigación porque la representación de esta metodología es parecida a la metodología de Warnier Orr, la diferencia entre ellas aparte de la simbología es que OPM, es una metodología ya estandarizada, lo cuál podría ser de gran utilidad para seguir representando el proceso de un dominio del problema, en trabajos futuros.

3.2.13 Las aplicaciones del procesamiento del lenguaje natural (PNL) para la ingeniería de requisitos de software - Una revisión sistemática de la literatura

En este artículo (Nazir, 2017) mencionan que el PLN es una técnica muy utilizada e implementada en la inteligencia artificial para la extracción de elementos de texto plano, y para el procesamiento de requerimientos de software donde ayuda a especificar la solución propuesta, así como su clasificación (en requerimientos funcionales y requerimientos no funcionales). Mencionan un estudio del uso del procesamiento del lenguaje natural en el contexto de la Ingeniería de Requisitos de Software. Dónde la revisión sistemática de literatura estuvo centrada entre los años 2002-2016, y se identificaron seis técnicas de PLN, 2 algoritmos PLN, 14 herramientas PLN (por ejemplo, GATE tool, Open NLP, NLKT, entre otros) y otras nueve herramientas PLN que son aportes de los investigadores. Al final del artículo se concluye que las técnicas de PLN aceleran el proceso de Ingeniería de Requisitos de Software.

Este trabajo al igual que el trabajo mencionado en los antecedentes de (Hernández G. S., 2019) realizó una revisión sistemática de literatura de la aplicación del procesamiento de lenguaje natural, y su objetivo es identificar las mejores prácticas. A diferencia del método realizado en Cenidet, el mapeo sistemático, se basó principalmente en el procesamiento de lenguaje natural para entender el dominio del problema y extracción de requerimientos de usuario (Elicitación de requerimientos), de manera que por medio de un esquema de clasificación se provea un panorama de las técnicas existentes. Así mismo, este artículo sirve para conocer técnicas que ayudan al procesamiento del texto.

3.2.14 Voz a Modelo: Un enfoque para generar modelos de requisitos a partir de mecanismos de reconocimiento de voz

En (Soares, 2015) los autores refieren que usualmente para modelar los requisitos típicamente se usan algunas notaciones gráficas con la ayuda de herramientas CASE. Sin embargo, esas herramientas hacen que la actividad sea exclusiva de los ingenieros de requisitos (analistas) y de la parte involucrada (interesados) sin ningún problema de accesibilidad, pero se

encuentra una problemática para los profesionales con alguna discapacidad, debido a que tienen problemas de accesibilidad para poder utilizar esas herramientas y construir los modelos de requerimientos. Es por ello que, en este artículo se propone el marco VoiceTo Model, para mejorar la accesibilidad del proceso de requisitos mediante la integración efectiva de un ingeniero de requisitos o un interesado con discapacidades a través del reconocimiento de voz.

Este trabajo resulta importante ya que en el artículo se busca lo mismo que en esta investigación, solo que ellos obtienen los requerimientos desde la voz para facilitar la interacción de personas discapacitadas, y en esta investigación es conocer a los actores involucrados en un determinado proceso, que podría también ser utilizado por personas con discapacidad.

3.2.15 Extracting Software Product Line Feature Models from Natural Language Specifications

En (Sree-Kumar, 2018) el autor examina los trabajos anteriores en los que se han presentado instrumentos para extraer el modelo de características (FM, por sus siglas en inglés) de especificaciones textuales y se comparan sus puntos fuertes y sus limitaciones. La especificación de una familia de productos de software puede incluir documentos escritos en lenguaje natural. La extracción automática de conocimientos de esos documentos es un problema difícil que requiere el uso de técnicas de procesamiento del lenguaje natural (PNL). Estos conocimientos pueden formalizarse en un modelo de características (FM), un diagrama que capta las características principales y las relaciones entre ellas. También se propone un marco para la extracción de características y relaciones, que supera las limitaciones identificadas y se basa en herramientas de PNL de código abierto de última generación. Este marco se evalúa con respecto a trabajos anteriores utilizando varios estudios de casos, que muestran resultados mejorados.

Como se ha mencionado este trabajo necesita conocer todas las herramientas de procesamiento natural que existan y en qué caso se adaptan mejor, es por ello que este artículo fue útil para conocer las herramientas de código abierto y saber cómo se utilizaron para la extracción de conocimiento de los documentos hechos mediante el lenguaje natural.

3.2.16 Tabla Comparativa de los trabajos relacionados de Ingeniería de Software

Los artículos presentados en la sección de Ingeniería de Software y resumidos en la **Tabla 5**, son trabajos que han llevado a cabo la representación gráfica de los requerimientos, del dominio del problema y del comportamiento del sistema. Todos estos modelos con diferentes técnicas e incluso con metodologías ya estandarizadas. Como la metodología OPM, donde incluso ya se ha estado trabajando con una extensión que mejora dicha metodología y, por lo tanto, da mejores resultados en la representación y en la obtención de información. La representación gráfica utilizada en esta investigación es la de (Orr K. T., 1981), representa una aproximación inicial del dominio del problema. La técnica de Orr, como la metodología OPM, muestra los procesos, las personas involucradas dentro del dominio y las tareas intercambiadas. Esta representación es importante ya que preferentemente se debe trabajar y modelar desde la perspectiva de las personas que usaran el sistema. Así, es importante diferenciar la perspectiva del jefe.

Tabla 5.- Trabajos Relacionados Ingeniería de Software

Trabajo Relacionado	Técnica o método utilizado	Representación gráfica	Problema resuelto
Structured Requirements Definition	Metodología de Warnier Orr (diagrama de entidades)	Sí	Buscar una representación gráfica que ayude a recabar todas las personas involucradas dentro de un dominio del problema. Identificando entidades, transacciones (mensajes) y la dirección de esas transacciones.
Representación de la regla formal y verificación de los requisitos del lenguaje natural utilizando una ontología	Uso de ontologías para representar el comportamiento de un sistema de software	Sí	El trabajo se enfoca en representar los requerimientos de usuario, que describen el comportamiento del sistema, mediante una ontología OWL, en una representación matemática formal llamada Maude
Modelado de dominios con	OPM y ADOM	Sí	Se ha ampliado la Metodología de Procesos de

metodología de objetos y procesos			Objetos (OPM) para manejar el enfoque de modelado de dominio de aplicaciones (ADOM). La extensión OPM incluye roles, que son estereotipos similares, e indicadores de multiplicidad. Aplicando ADOM-OPM al dominio de sistemas de control de acceso.
Las aplicaciones del procesamiento del lenguaje natural (PNL) para la ingeniería de requisitos de software - Una revisión sistemática de la literatura	GATE tool, Open NLP, NLKT	Sí	Realiza el procesamiento de lenguaje natural de las historias de usuario a diagramas de casos de uso. También se realiza un mapeo sistemático de técnicas para realizar el PLN
Voz a Modelo: Un enfoque para generar modelos de requisitos a partir de mecanismos de reconocimiento de voz	Herramientas Case	Sí	A través del reconocimiento de voz, el ingeniero debe poder expresar los requisitos, que se utilizan automáticamente para generar modelos de requisitos, más específicamente modelos orientados a objetivos, modelos de objetos y modelos de características. Además, el marco automatizado genera modelos de requisitos a partir del reconocimiento de voz, definiendo en primer lugar una gramática donde tienen los comandos para crear y modificar un modelo.
Extracting Software Product Line Feature Models from Natural Language Specifications	Modelo de características	Sí	En el artículo se revisan trabajos que han trabajado con textos para la extracción del modelo de características (FM), comparando sus fortalezas y limitaciones.

3.3 RESUMEN DEL CAPÍTULO

En este como se mencionó al inicio del capítulo se presentaron algunos trabajos que se realizaron en el CENIDET, y en otras instituciones (trabajos relacionados), respecto al trabajo que se ha realizado en la creación de un chatbot, artículos que han estado trabajando con el Procesamiento del Lenguaje Natural y trabajos que muestran las investigaciones que se ha llevado acabo para representar gráficamente el dominio del problema.

Tomando como referencia los trabajos presentados, en el siguiente capítulo se exhibe la metodología que se realizó para llevar acabo la solución propuesta que permite obtener el objetivo general mencionado en el capítulo 1 .

Capítulo 4

Metodología y Resultados

4.1 METODOLOGÍA DE SOLUCIÓN

En la **Figura 10** se muestra el esquema que representa la metodología de solución que se desarrolló para esta investigación. Dicha metodología está dividida en tres fases. La primera fase llamada **Arquitectura del chatbot** es aquella donde se utilizan las herramientas que reconocen la voz y la transforman a texto. Para la fase, se realizó un mapeo sistemático y se seleccionó la mejor herramienta que se adaptó a nuestras necesidades, esta fue la plataforma de Google llamada DialogFlow. También durante esta fase se buscó la secuencia de preguntas que ayudaran a recabar toda la información necesaria para crear el modelado, de una forma sencilla, práctica, y a su vez de fácil entendimiento para el experto del dominio. Por último, se creó una aplicación interactiva para poder realizar la entrevista.

En la segunda fase llamada **Análisis del texto** se realiza todo el Procesamiento del Lenguaje Natural (NLP, por sus siglas en ingles). Para esta fase se buscaron todas las herramientas, librerías o procedimientos que ayudan al reconocimiento de las entidades presentes e involucradas dentro del dominio seleccionado, las entradas o salidas, y la secuencia entre ellas, todo esto a un nivel morfosintáctico, el cual se explica más adelante en este capítulo.

Finalmente, en la tercera fase **Creación del diagrama**, se realiza el diagrama del proceso en base a la metodología de Warnier Orr, este diagrama se realizó de forma manual en la herramienta Visio de Microsoft y con la información procesada en la fase anterior. Las tres fases se describen a más detalle en las secciones siguientes.

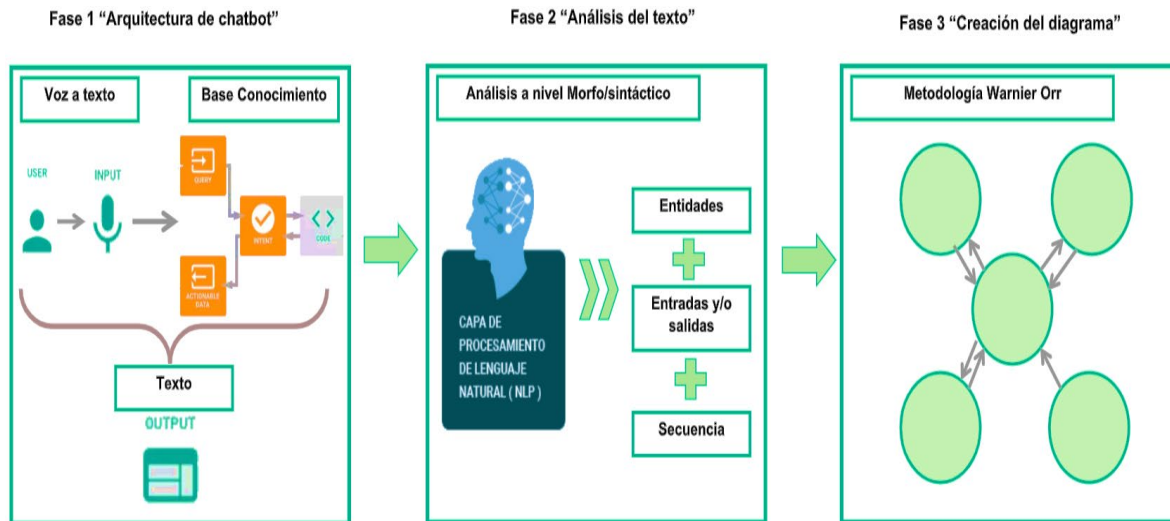


Figura 10.- Metodología de Solución.

SECCIÓN 1

4.1.1 FASE UNO - ARQUITECTURA DEL CHATBOT

Las actividades que necesitan realizarse para llevar a cabo esta primera fase de una forma adecuada y crear el chatbot correctamente se enlistan a continuación:

- Crear el chatbot dentro de la plataforma de Google, el chatbot debe contar con algunos recursos para su funcionamiento. A continuación, en la **Tabla 6** se explican los recursos que se necesitan para la realización y/o utilización del chatbot dentro del framework de DialogFlow.
- Definir la estructura de la conversación que va a guiar al experto del dominio. Esto se realizó con un diagrama adaptado para esta investigación.
- Establecer el vocabulario suficiente para el dominio administrativo dentro de DialogFlow.
- Construir la interfaz que sea amigable con el usuario, y a su vez el usuario pueda llevar la entrevista mediante la voz y el texto.

Las actividades antes mencionadas y enlistadas, se irán explicando a continuación, de acuerdo al orden que se enlistaron. Primero los recursos para utilizar DialogFlow, y así sucesivamente.

RECURSOS PARA EL CHATBOT

Como se mencionó anteriormente, en esta fase se empieza a realizar la **Creación del chatbot**, durante esta fase se definieron los requerimientos que tenía que cumplir el chatbot y la organización que tendrían las preguntas, así como los recursos necesarios para que funcionen adecuadamente los requerimientos. Así, se definieron cuatro requerimientos importantes para el chatbot. En la **Tabla 6** se pueden visualizar los requerimientos y los recursos necesarios para satisfacer cada uno. Estos requerimientos se encuentran descritos en el capítulo 5, en la documentación de las pruebas.

Tabla 6.- Recursos chatbot

ID	Requerimiento	Recurso			
		Aplicación	Artefacto	Archivos	Subproceso
RF1	Visibilidad de chatbot en la página web y/o aplicación	<ul style="list-style-type: none"> ○ Servidor Local ○ Navegador Web ○ DialogFlow 	No aplica	No aplica	No aplica
RF2	Reconocimiento de voz	No aplica	Micrófono	No aplica	No aplica
RF3	Seguimiento de la entrevista (preguntas)	<ul style="list-style-type: none"> ○ DialogFlow 	No aplica	No aplica	<ul style="list-style-type: none"> ○ Entidades ○ Contextos ○ Intenciones
RF4	Creación del historial	<ul style="list-style-type: none"> ○ DialogFlow (Pestaña Historial) 	No aplica	Conversación .txt	No aplica

ESTRUCTURA DE LAS PREGUNTAS

La actividad principal de la etapa fue definir la secuencia adecuada de las preguntas, y cuáles de éstas eran las esenciales para recabar la información inicial (entidades, entradas y transiciones) del dominio del problema a tratar. Cabe mencionar que en el chatbot la parte

más importante son las preguntas, ya que si no se cuenta con las preguntas necesarias puede que estemos omitiendo información o recabemos información innecesaria. Como se mencionó anteriormente, el chatbot se creó en la plataforma DialogFlow de Google, que permite crear agentes inteligentes que reconocen el lenguaje natural mediante voz o texto, de forma sencilla, integrando aprendizaje automático.

Para seleccionar la herramienta que mejor se adaptara a las necesidades de esta investigación y cumpliera el alcance de poder crear agentes conversacionales con reconocimiento de voz, se realizó un estudio de mapeo sistemático de herramientas utilizadas para la creación de chatbots. La revisión de la literatura fue del periodo limitado por los años 2013 y 2018, escrita en idioma inglés, y produjo la identificación de nueve herramientas, pero solo tres de ellas se han utilizado más por la facilidad, tecnología y documentación disponible. Algunas de las características que se tomaron en cuenta para comparar estas herramientas son: la tecnología que aplica, por ejemplo: redes neuronales, procesamiento de lenguaje natural (PLN), aprendizaje automático o aprendizaje basado en reglas; la disponibilidad de la documentación en internet; que procese el idioma español; el lenguaje de programación que soporta; el tipo de licencia (enfocándose a licencias gratuitas); que incluya el reconocimiento de voz; y por último, que no existiera un límite en el número de interacciones (Millán, 2019).

Con la revisión literaria se concluyó que la herramienta que mejor se adapta a las necesidades de esta investigación es la herramienta ofrecida por Google, debido a que cumple con los criterios antes señalados. Adicionalmente, dicha herramienta se ejecuta en una plataforma en la nube de Google, por lo que su ejecución y acceso a la información puede realizarse desde cualquier equipo, además, permite implementarse o desempeñarse en cualquier plataforma externa, es decir, existe una compatibilidad de la plataforma con diferentes aplicaciones y sistemas operativos, por ejemplo, se puede crear una aplicación en el Asistente de Google, Messenger, entre otros. Asimismo, nos brinda la libertad de poder dar seguimiento al proceso de la conversación y el chatbot. En otras palabras, dentro de la plataforma de DialogFlow existe un apartado llamado **Análisis y Validación**, donde se pueden observar todos los errores que tuvo nuestro chatbot (Ej. Cuando no se inició correctamente el chatbot, cómo fue el flujo de las intenciones), incluso se pueden revisar las intenciones dónde el chatbot no supo que preguntar o perdió el seguimiento de la conversación, esto es útil para retroalimentar

(corregir los errores, e incluso aumentar el vocabulario) el chatbot, especialmente donde los errores se presenten del lado del chatbot, por ejemplo: en las intención, entidades, etc., lo cual beneficia en un mantenimiento (vocabulario actualizado) continuo a lo largo del tiempo.

Por otro lado, para poder seleccionar las preguntas que ayudarían a formular la entrevista y cuál iba a ser su secuencia, se creó una simbología especial para identificar las intenciones, contextos, decisiones (preguntas) y entidades que interactuarán dentro de la plataforma de Google, estos conceptos se encuentran descritos en el **Capítulo 2 Marco Teórico**. Esta simbología se puede visualizar en la **Figura 11**.

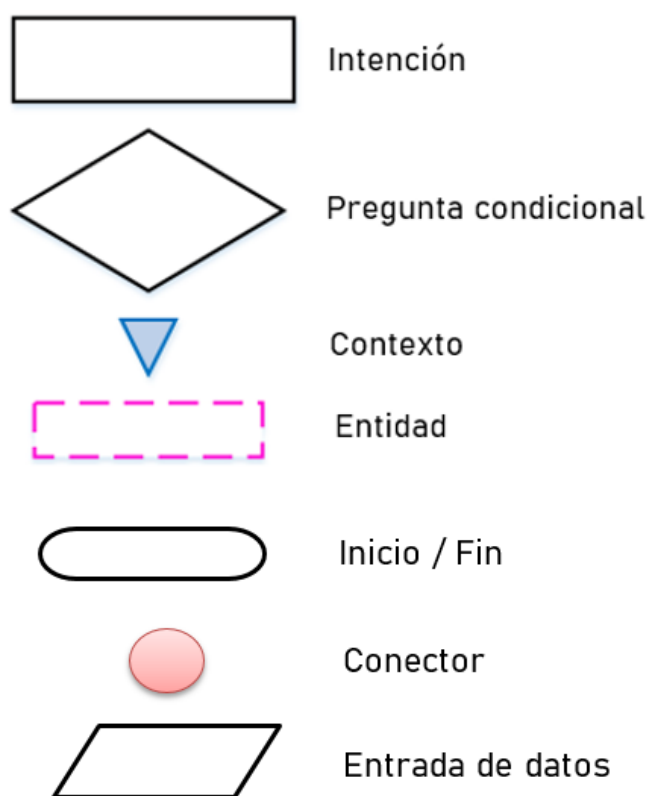


Figura 11.- Simbología de la Secuencia de Preguntas

La simbología fue utilizada para la creación de un diagrama de flujo adaptado para esta investigación, creado para representar la estructura del diálogo que iba a tener el chatbot y facilitar la comprensión y definición del mismo. Así como conocer el objetivo (información que iba a recabar) de cada pregunta y saber la relación entre los contextos, intenciones, etc.

Cabe mencionar que las entidades que se manejan dentro de la plataforma, son diferentes a las entidades que se identifican para crear el diagrama de Warnier Orr. Por ejemplo; Las entidades que se utilizan en la plataforma DialogFlow, como ya se dijo, tienen el objetivo de determinar de forma exacta como se extraen los datos de una expresión de usuario final. En otras palabras, las entidades son todas las palabras o conjunto de palabras o frases que se consideran equivalentes, como se puede visualizar en la **Figura 12**. Por otro lado, las entidades que se reconocen con el diagrama de Warnier Orr, son todas las personas involucradas que intercambien información, dentro de un proceso del dominio del problema, ver la **Figura 13**.

Ejemplo de entidad de DialogFlow:

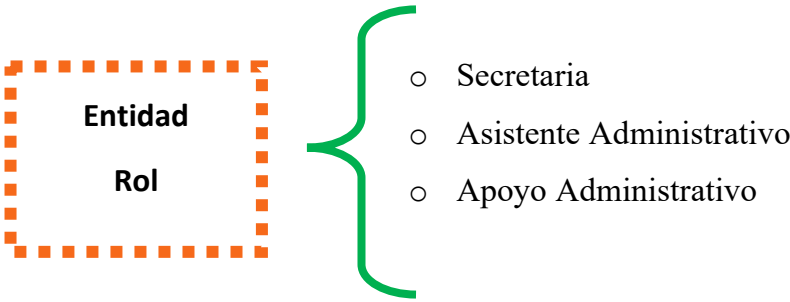


Figura 12.- Ejemplo entidad DialogFlow

Ejemplo de entidad Warnier Orr:

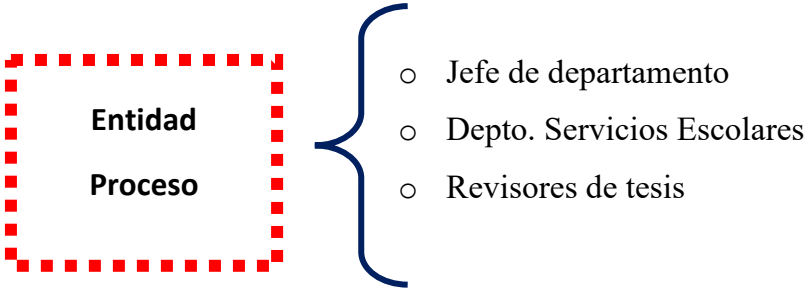


Figura 13.- Ejemplo entidad proceso Warnier Orr

En la **Figura 14** se puede observar la primera parte de la secuencia que siguen las preguntas y cómo interactúan entre todos los componentes en la realización de la entrevista.

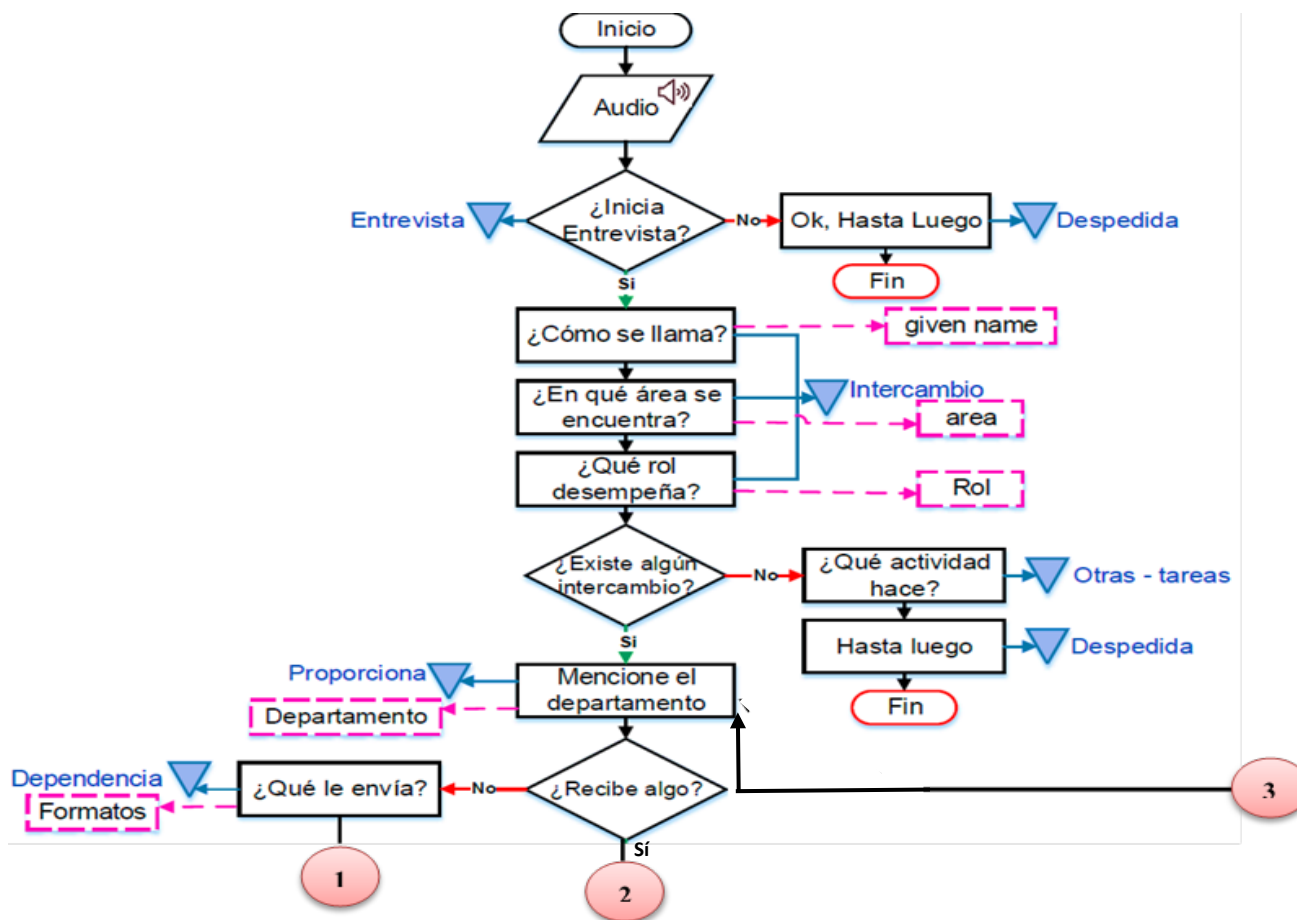


Figura 14.- Diagrama de la secuencia de preguntas Parte 1/2

A continuación, se explica un pedazo del flujo que sigue la primera parte del diagrama:

- 1) Se necesita activar el chatbot, esto se hace con una entrada de audio, esta será la voz del interesado saludando al chatbot.
- 2) Una vez que se haya activado el chatbot, se hace una pregunta condicional ¿se puede iniciar la entrevista?, esta pregunta tiene como contexto *Entrevista* (triángulo invertido azul). En caso de que la respuesta sea negativa, se lanza la intención con la respuesta **ok, será otro día**, con el contexto *Despedida*, y ahí finaliza la entrevista.
- 3) Por el lado contrario, si tenemos una respuesta positiva, se lanzarán tres preguntas de intención consecutivas (se tiene que responder para poder enviar la otra pregunta), **¿Cómo se llama?**, **¿En qué área se encuentra?** y **¿Qué rol desempeña en el área que esta?**, estas comparten el mismo contexto *Intercambio*, pero cada pregunta está ligada a su propia entidad (rectángulo punteado rosa). Por ejemplo, la pregunta ¿cómo se llama? esta enlazada a la entidad *Given Name*, que es una entidad de Google y aparte se buscaron los 100 nombres más comunes en México, para aumentar el vocabulario.
- 4) Después se procede a la preguntar condicional **¿Existe algún intercambio con algún departamento?**, como nos interesa conocer solo a las personas que realicen algún intercambio con la persona involucrada, no nos arroja mucha información que no exista este intercambio, así que, si la respuesta es negativa, se presenta la intención **¿Qué actividad hace?** Con el contexto *Otras_tareas*, después se presenta la intención de **despedida** y se termina la conversación.
- 5) Si de lo contrario se responde afirmativamente, se arroja la pregunta de intención **Mencione al depto. con el que existe este intercambio**, quien tiene como contexto *Proporciona* y la entidad *Departamento*

La segunda parte del diagrama se puede visualizar en la **Figura 15**, en esta las preguntas pueden llegarse a ciclar (conector 3), esto dependerá de la cantidad de personas (entidades) con las que interactúa la entidad principal.

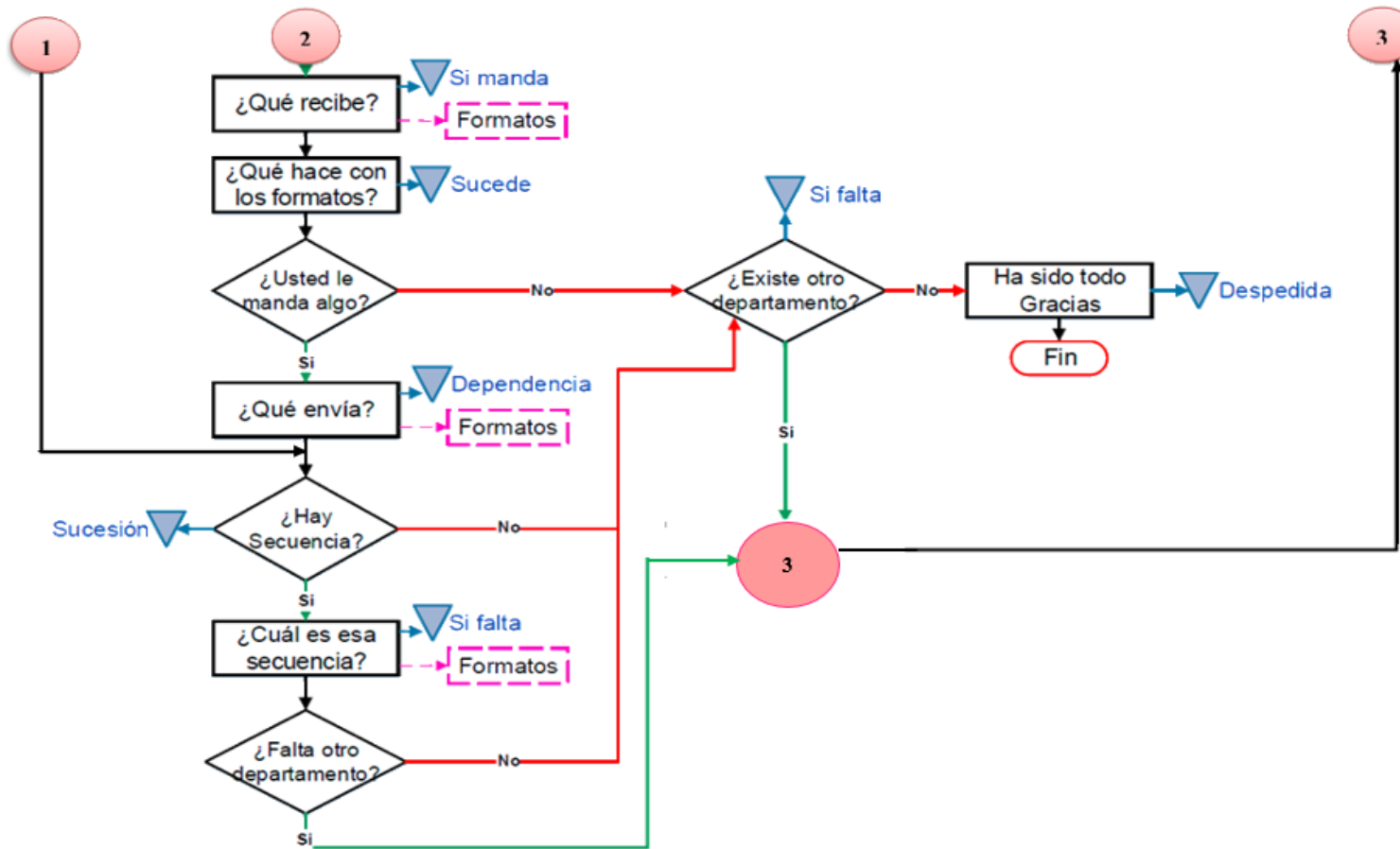


Figura 15.- Diagrama de la secuencia de preguntas Parte 2/2

En seguida, se explica brevemente el flujo que sigue la segunda parte del diagrama:

- 1) Una vez que se confirma que existe un intercambio, por el conector 2, se procede a la preguntar de intención **¿Qué recibe?**, que tiene como contexto **Si manda** y entidad **Formatos**, después se preguntará **¿Qué hace con los formatos recibidos?**
- 2) Seguidamente se arroja la pregunta condicional **¿Sí él o ella manda algo?**, si la respuesta es no, se formula la preguntar condicional **¿Si existe otro depto. con el que se cambie información?**, si la respuesta vuelve a ser negativa, se presenta la intención despedida y se finaliza la entrevista, si es positiva se va al conector 3.
- 3) En cambio, si la respuesta es positiva, se pregunta **¿Qué envía?**, esta pregunta tiene asociado el contexto **Dependencia** y la entidad **Formatos**, luego se preguntará **¿Existe alguna secuencia con lo que se manda?**, nuevamente si la pregunta es negativa se irá hacia el conector 3.
- 4) Al contrario, si existe la respuesta positiva se pregunta **¿Cuál es esa secuencia?** Con contexto **Si falta** y entidad **Formatos**. Y Finalmente se concluye con la pregunta **¿Falta otro depto. por mencionar?**, y al conector 3.

En el diagrama se incluyeron las entidades y los contextos, para tener un mejor panorama y saber cómo se relacionan entre sí. Así mismo se evita el riesgo de que faltara alguno al momento de pasar las preguntas a la plataforma DialogFlow.

Finalmente, fue necesario corroborar si las preguntas eran las correctas y si con ellas se iba a recabar toda la información mínima necesaria. Es por ello que se pasaron a la plataforma de Google, y de acuerdo con las dos partes del diagrama, la entrevista está compuesta por siete preguntas condicionales y 14 preguntas de intención, que pueden repetirse según lo amerite el curso de la conversación, como se vio en la explicación anterior.

VOCABULARIO DEL DOMINIO ADMINISTRATIVO

Ahora bien, al crear el chatbot se buscaba que el usuario se pudiera comunicar con él de una forma más fluida, agradable y como lo haría con una persona normalmente, es por ello que se le da el nombre de Atenea. Como se ha dicho anteriormente, para poder conformar la entrevista dentro de la plataforma se crearon 5 entidades, 20 intenciones, 10 contextos y entre 800 y 1300 frases de entrenamiento. La explicación de estos términos se encuentran en el **Capítulo 2 Marco Teórico**. Para explicar mejor la relación entre todos los elementos, en la **Figura 16** se muestra el esquema de cómo está estructurada la plataforma y la interacción entre los componentes.

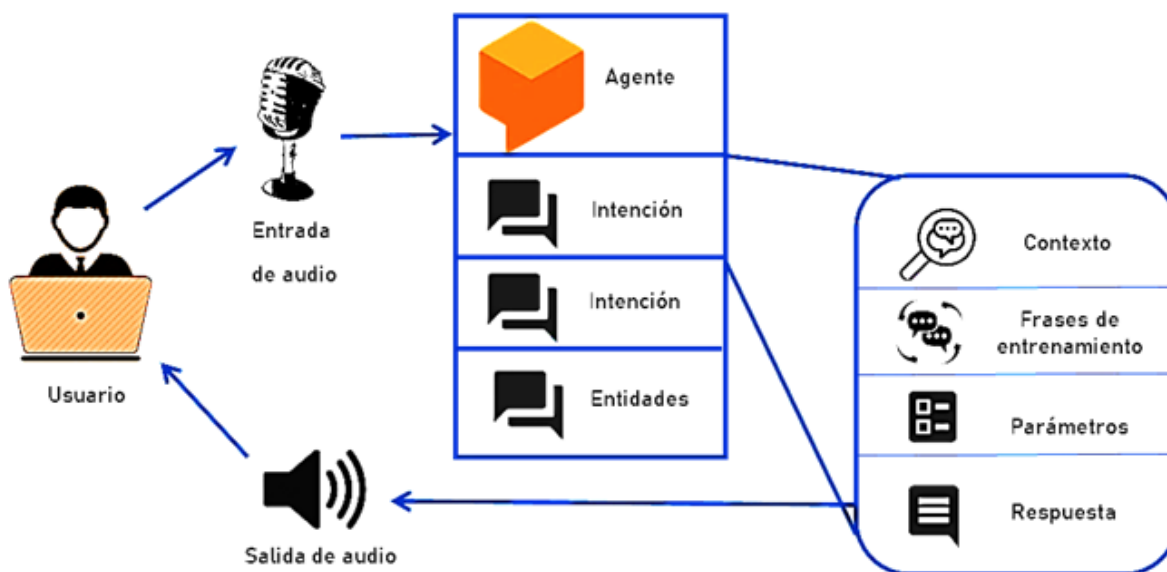


Figura 16- Relación entre componentes dentro de la plataforma de Google

Antes de comenzar a explicar la estructura de DialogFlow, es importante mencionar que la plataforma le da el nombre de agente al chatbot que se crea, es por eso que en la figura se puede observar este nombre.

Como se ha afirmado antes, Atenea funciona por medio de la voz, esto activará al chatbot, éste está conformado por un número indefinido de intenciones y entidades. Para este caso (en el área administrativa, del Departamento de Ciencias Computacionales del Cenidet) cinco y veinte respectivamente, en la figura sólo se pueden observar dos intenciones y una entidad, a su vez las intenciones están compuestas por contextos, frases de entrenamiento, variables

y una respuesta, que se presenta al usuario por medio de audio, más adelante se presentarán ejemplos de estos elementos.

Las entidades se deben de escribir sin espacios, para que DialogFlow pueda reconocerlas. En la **Figura 17** se pueden observar las cinco entidades creadas: **área, departamento, formatos, nombre y rol**, como se mencionó al principio, el trabajo de investigación está enfocado en el dominio administrativo, dentro del Departamento de Ciencias Computacionales del CENIDET. Por consiguiente, para poder crear la entidad departamento, se utilizó un glosario de términos administrativos para tener el mayor vocabulario posible.

@ area
@ departamento
@ formatos
@ nombre
@ rol

Figura 17.- Entidades del chatbot Atenea

También, están las 20 intenciones creadas, donde 17 de ellas son las principales (se representa en color negro), es decir, son las que guiaran al usuario por toda la entrevista y las tres restantes son de excepción (representadas en color azul), mostradas en la **Tabla 7**. Las tres intenciones restantes se activan cuando el chatbot no entendió que le está respondiendo el usuario, o el interesado responde las preguntas con algo fuera de lo esperado. Por ejemplo:

Chatbot: ¿Existe algún intercambio?

Usuario: tal vez, quizá, no sé.

Se activará la intención de excepción *ExisteIntercambio Context: intercambio*, y se mostrará una alerta con el mensaje **Solo puede responder sí o no, ¿Existe algún intercambio?**

Las intenciones utilizadas dentro del chatbot se incluyen en la **Tabla 7**, donde se observa una descripción breve de lo que hace cada intención, o en qué momento se activa dentro de la conversación. Como se mencionó antes las de color azul son las intenciones de excepción. En la parte de Anexos, se explica cómo crear intenciones y entidades en DialogFlow.

Tabla 7.- Intenciones del chatbot

Nombre de la intención	Descripción
<p>Default Fallback Intent</p>	<p>Esta es una intención de excepción, y se activará cuando el usuario en lugar de saludar o responder las preguntas con lo que se pide, responda otras cosas, fuera de contexto. Por ejemplo:</p> <ul style="list-style-type: none"> ➤ ¿Qué hace?, ¿De qué sexo es?, etc. <p>La intención puede mandar un mensaje, como el siguiente:</p> <ul style="list-style-type: none"> ➤ No le entendí, puede repetirlo, por favor, y un mensaje parecido se ciclará hasta que se responda lo que se pide.
<p>Default Welcome Intent</p>	<p>Esta intención es la que se activará por default una vez que se haya saludado al chatbot correctamente. Ésta incluye un mensaje como el siguiente:</p> <ul style="list-style-type: none"> ➤ Buen día, mi nombre es Atenea, quiero conocer mejor el trabajo que desempeña. ¿Puedo realizarle una entrevista?
<p>Despedida</p>	<p>Esta intención será presentada al usuario cuando no pueda responder la entrevista, se podrá visualizar un mensaje como el siguiente:</p> <ul style="list-style-type: none"> ➤ Bye, muy buen día
<p>ExisteDependencia Context:dependencia</p>	<p>Esta intención también es de excepción, y se mostrará cuando el chatbot le pregunte al usuario “¿Existe alguna dependencia entre los formatos?”, y este le responda con “Quién sabe, no estoy seguro, etc.”</p> <p>En esta intención se podrá visualizar un mensaje como el siguiente:</p>

	<ul style="list-style-type: none"> ➤ Únicamente puede responder Sí o No. ¿Entre los formatos existe alguna dependencia?
ExisteIntercambio Context:intercambio	<p>Esta intención es de excepción y se visualizara en el chatbot cuando el chatbot pregunte ¿Existe algún intercambio?, y el usuario responda con: “a lo mejor, no lo sé”, etc. El mensaje que muestra la intención es el siguiente:</p> <ul style="list-style-type: none"> ➤ Solo puede responder si o no. ¿Existe algún intercambio de documentos con algún departamento o persona?
NoDependencia	<p>Como su nombre lo dice, la intención se presentará cuando se diga que no existe una dependencia entre los formatos enviados. El mensaje que se podrá visualizar es:</p> <ul style="list-style-type: none"> ➤ ¿Falta algún departamento por mencionar?
Noentrevista	<p>Esta intención se muestra para empezar la entrevista otro día, en caso de que el usuario no pueda iniciarla en ese momento. Muestra un mensaje como:</p> <ul style="list-style-type: none"> ➤ Ok, será otro día. Qué tenga un excelente día
Nohayintercambio	<p>Como se dijo anteriormente, se necesita conocer solo las entidades que intercambien información, cuando no intercambia información, se activa esta intención y puede presentar un mensaje como el que se visualiza a continuación:</p> <ul style="list-style-type: none"> ➤ Muy bien, me puede explicar ¿cuál es la función de su rol?
Nohaymas	<p>La intención tiene como objetivo presentarse cuando no existan más departamentos con los cuales se intercambia información. El mensaje que se puede observar en esta intención es:</p> <ul style="list-style-type: none"> ➤ Le agradezco mucho su tiempo proporcionado en esta entrevista, ¡tenga un excelente día!

Nomanda	<p>Cuando el usuario no manda ninguna información al depto. mencionado, la intención de Nomanda aparece, y puede mostrar un mensaje como:</p> <ul style="list-style-type: none"> ➤ Ah ok, ¿hay un departamento más con el que se cambie información?
Norecibo	<p>Si por el lado contrario, no se recibe nada del depto. mencionado, se activa esta intención y muestra un mensaje como el de abajo:</p> <ul style="list-style-type: none"> ➤ Entiendo, ¿existe alguna dependencia entre los formatos mencionados?
Orden	<p>Se creó esta intención con la finalidad de que se active cuando existe una dependencia entre los formatos, está recuperará la secuencia que explique el usuario.</p>
Otras_tareas	<p>Como su nombre lo dice, se activará cuando dentro del rol del usuario no exista la actividad de intercambio, la intención captura la información de las actividades que realiza la persona.</p>
Quesucede	<p>La intención de Quesucede se activará cuando el usuario diga que recibe algún formato del depto. mencionado. Después, mandará a la siguiente pregunta (intención).</p>
SiDependencia	<p>Esta intención es lo contrario de la intención NoDependencia, se activa cuando el usuario mencione que sí existe una dependencia entre formatos.</p>
Sientrevista	<p>Será la intención que siga cuando el usuario mencione que puede iniciar la entrevista en ese momento, esta contendrá las preguntas de:</p> <ul style="list-style-type: none"> ➤ ¿Cómo se llama? ➤ ¿En qué área está? ➤ ¿Qué rol cumple en esa área?
Sihayintercambio	<p>Es lanzada por el chatbot cuando el usuario entrevistado, diga que dentro de su rol sí hay un intercambio con personas y deptos.</p>

Sihaymas	El objetivo de la intención es activarse cuando aún falten más departamentos con los que haya un intercambio.
Simanda	Será usada por el chatbot cuando el usuario entrevistado, también le mande algo al depto. que en ese momento se haya mencionado.
Sirecibo	Es la intención que se ocupará cuando el usuario entrevistado, reciba algo del depto. mencionado en ese momento.

Como se puede observar en la tabla anterior, las veinte intenciones creadas se usarán en algún momento de la entrevista, esto no quiere decir que, en todas las entrevistas realizadas por Atenea, se usarán todas las intenciones, o las entrevistas tendrán las mismas intenciones. El flujo de éstas lo define el usuario de acuerdo a la entrevista, como se vio en el diagrama de flujo del dialogo, presentado anteriormente.

Para el correcto funcionamiento de dichas intenciones y comprender lo que el interesado responde, se necesita de contextos (un contexto en DialogFlow sirve para entender de qué se está hablando y así intuir qué intención es la que sigue.). Atenea cuenta con 10 contextos distribuidos en las 20 intenciones. En seguida, se enlistan los contextos utilizados en nuestro chatbot, los contextos nos ayudan a que nuestro chatbot tenga una memoria y sepa cuándo se dispara o utiliza cada intención. Estos se colocan siempre al principio de cada intención y es importante saber si será un contexto de entrada o de salida, y cuánto tiempo tendrá de vida éste. Es importante mencionar que los nombres de los contextos no se pueden poner en mayúscula ni con espacios en blanco.

- | | |
|-----------------|-------------|
| 1) entrevista | 9) sucesión |
| 2) intercambio | 10) simanda |
| 3) despedida | |
| 4) proporciona | |
| 5) otras_tareas | |
| 6) sucede | |
| 7) dependencia | |
| 8) sifalta | |

Por otro lado, las frases de entrenamiento también son importantes para las intenciones. Son colecciones de las declaraciones posibles que los usuarios podrían decir para disparar la intención. Entre más frases de entrenamiento contenga la intención mejor reconocimiento y funcionamiento tendrá nuestro. En la documentación de Google, se recomiendan que sean al menos diez frases de entrenamiento (Google, 2020). En la **Figura 18** se pueden observar algunos ejemplos de las frases de entrenamiento para la intención **Default Welcome Intent**, que es la intención que utiliza Atenea para dar la bienvenida al usuario.

” hola buen día	” buen día
” hola qué tal	” hola
” buenas tardes	” hey
” buenos días	” saludos
” qué tal	” hey ho

Figura 18.- Frases de entrenamiento intención Default Welcome Intent

De la misma manera, los parámetros son parte de las intenciones. Los parámetros son datos estructurados que se pueden usar fácilmente para realizar alguna lógica o generar respuestas, es decir, son elementos que reciben la información que da el usuario y esta ira cambiando de valor dependiendo de la respuesta. Por ejemplo: La variable \$nombre, puede tener el valor de Raúl, Susana, etc. El termino variable y parámetro se utilizan por la plataforma, con el mismo significado.

Las variables o parámetros tienen seis atributos: Obligatorio, Nombre, Entidad, Valor, Lista y Mensajes. En seguida se van a describir cada uno de los atributos, estos se pueden utilizar de acuerdo a lo que se quiera lograr en el flujo de la entrevista, es decir, no todos los parámetros pueden ser obligatorios para continuar con la entrevista, y no en todos se pondrán variables o mensajes, todo eso será a criterio de la persona que crea el chatbot.

- **Obligatorio** (Required): Se marcará esta casilla si se requiere el parámetro para completar el intent.
- **Nombre del parámetro** (Parameter Name): Será el nombre que identifica el parámetro. Ejemplo: fecha.
- **Entidad** (Entity): Es el nombre de la entidad a la que está asociado el parámetro., y para utilizar esa entidad se debe de utilizar el signo de @. Ejemplo: @departamento.
- **Valor** (Value): Este es un identificador que se usa para hacer referencia a los valores extraídos establecidos en el tiempo de ejecución. Ejemplo: \$fecha.
- **Es lista** (Is List): Se marcará esta casilla si los valores extraídos deben mostrarse como una lista. Ejemplo: Naranja y Manzana.
- **Mensajes** (Prompts): Son las preguntas que el chatbot formulará al usuario si no se proporcionó este parámetro.

Una vez explicado los atributos de los parámetros, en la **Figura 19** se puede observar las variables utilizadas para la intención **Sientrevista**, esta intención se ejecuta en manera de formulario, donde cada variable cuenta con 10 a 15 formas diferentes de preguntas para obtener ese valor.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	nombre	@nombre	\$nombre	<input type="checkbox"/>	¿Me proporcion a...
<input checked="" type="checkbox"/>	area	@area	\$area	<input type="checkbox"/>	\$nombre, ¿en q u...
<input checked="" type="checkbox"/>	rol	@rol	\$rol	<input type="checkbox"/>	Mucho gusto \$n o...

Figura 19.- Ejemplo de variables de la entidad Sientrevista

Y para concluir con las intenciones, están las respuestas que es la forma en la que va a responder Atenea al usuario, en la **Figura 20** se muestran las posibles respuestas para la intención de **Sientrevista**. De igual manera, se observa que dentro de las respuestas se

utilizan las variables para captar la respuesta que dio el usuario y mostrársela para crear un dialogo más natural (de persona a persona) entre Atenea y el interesado.

Text or SSML Response		?	🗑
1	\$nombre, de este rol de \$rol.original que me acaba de mencionar. ¿Existe algún intercambio de documentos con algún departamento o persona?		
2	Me dijo que es \$rol.original, ¿quisiera saber si ese rol implica un intercambio de oficios con algún departamento o persona?		🗑
3	Siendo \$rol.original, ¿hay algún intercambio de formatos con algún departamento o persona?		
4	\$nombre.original, conforme a este rol de \$rol.original, ¿Debe haber un intercambio de información con alguna persona o departamento?		
5	De su papel de \$rol.original, ¿hace algún intercambio de formatos con algún departamento o persona dentro de esta institución?		

Figura 20.- Ejemplo de respuesta para la intención Sientrevista

Todas las intenciones antes mencionadas están formadas de la misma forma y con todos los atributos anteriormente descritos, algunas contendrán más variables y respuestas, dependiendo del objetivo de la intención y de qué información se necesita obtener.

Para finalizar esta fase, el interesado tiene que concluir la entrevista realizada por el chatbot Atenea.

Una vez concluida, el chatbot guardará la conversación dentro de la plataforma de Dialogflow en el apartado de historial. Y Así, podremos utilizar la conversación en formato .txt para la siguiente fase.

4.1.2 IMPLEMENTACIÓN DE LA FASE UNO – CREACIÓN DEL CHATBOT

El caso de estudio es un problema administrativo, donde se entrevistaron a tres entidades dentro del Departamento de Ciencias Computacionales (Secretaria, Alumno y Presidenta de Claustro) la entidad principal fue la secretaria de la jefatura. Para poder presentar el chatbot al usuario se implementó la plataforma de Google **DialogFlow** en dos entornos. El primero es en una página web (ver **Figura 21**) y el segundo entorno es Google Assistant (ver **Figura 22**), este último para simular una conversación más natural y humana entre el chatbot y el usuario (interesado).

- Página Web, desde un servidor local



Figura 21.- Interfaz chatbot en página web

- Asistente de Google



Figura 22. Interfaz del chatbot en Google Assistant

4.1.2.1 Página web Atenea (sección 1 página web)

Se creó una página web para que el usuario (interesado) pudiera visualizar el chatbot Atenea, en una interfaz amigable y que pudiera realizar la entrevista dentro de su horario laboral, esta página se encuentra en un servidor local llamado Wampserver.

Dentro de la página se podrán visualizar cinco botones como se observa en la **Figura 23**, cada botón dirigirá al usuario a una sección dentro de la página.

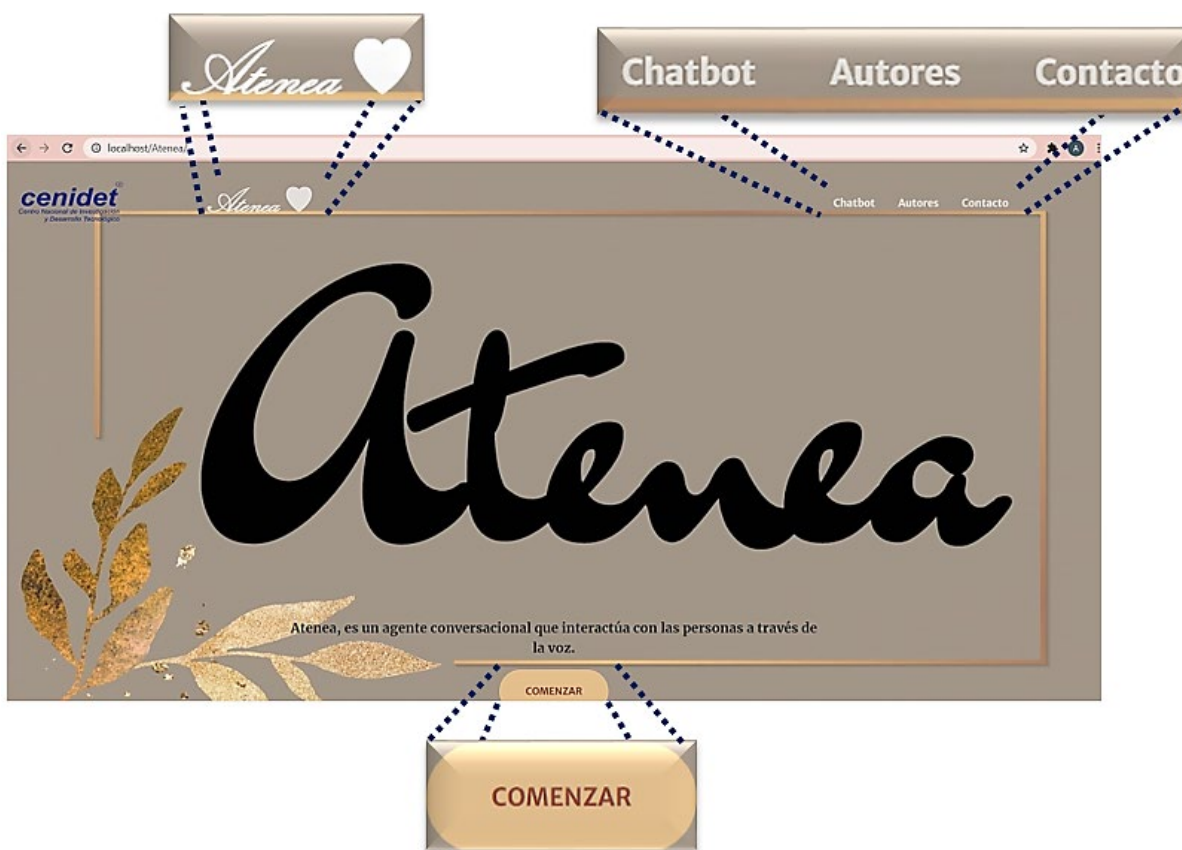


Figura 23.- Botones de la página web Atenea

Las acciones de cada botón se describen a continuación:

- El botón **“Atenea”** funciona como botón de inicio para posicionar la página en su sección principal.
- El botón **“Chatbot”** dirige al usuario a la sección que contiene el chatbot.

- El botón “**Autores**” muestra a las personas que desarrollaron el prototipo y trabajo de investigación.
- El botón “**Contacto**” expone la ubicación del centro donde se realizó el proyecto y los datos de los autores.
- El botón “**Comenzar**” tiene la misma función que el botón chatbot.

4.1.2.2 Botón Chatbot y Botón Comenzar (Sección 2 página web)

Cuando el usuario haya hecho clic el botón “**Chatbot**” o el botón “**Comenzar**”, se dirigirá a la sección dos dentro de la página, en esta sección se visualiza una leyenda “*Por favor, active primero el avatar*”, en la parte izquierda se podrá ver el avatar el cuál se debe activar para conocer las indicaciones para llevar a cabo la entrevista y en el lado derecho de la pantalla se puede ver la interfaz del chatbot. Ver la **Figura 24**.

En esta sección el usuario tiene que ir activando el micrófono de la pantalla (ilustrado con una flecha roja en la **Figura 24**) con cada pregunta presentada en la entrevista. Un ejemplo del flujo de la entrevista se muestra a continuación.

Atenea: Me da gusto conocerte Liliana. ¿Qué papel ejerce en este lugar?.

usuario: secretaria.

Atenea: Liliana, de este rol de secretaria que me acaba de mencionar. ¿Existe algún intercambio de documentos con algún departamento o persona?.

Usuario: sí

Atenea: Por favor, nombre al departamento o persona con él que intercambia formatos.

Usuario: con los alumnos.

Atenea: Me dijo Alumnos, ¿Alumnos le da a usted algún oficio?.

Usuario: no

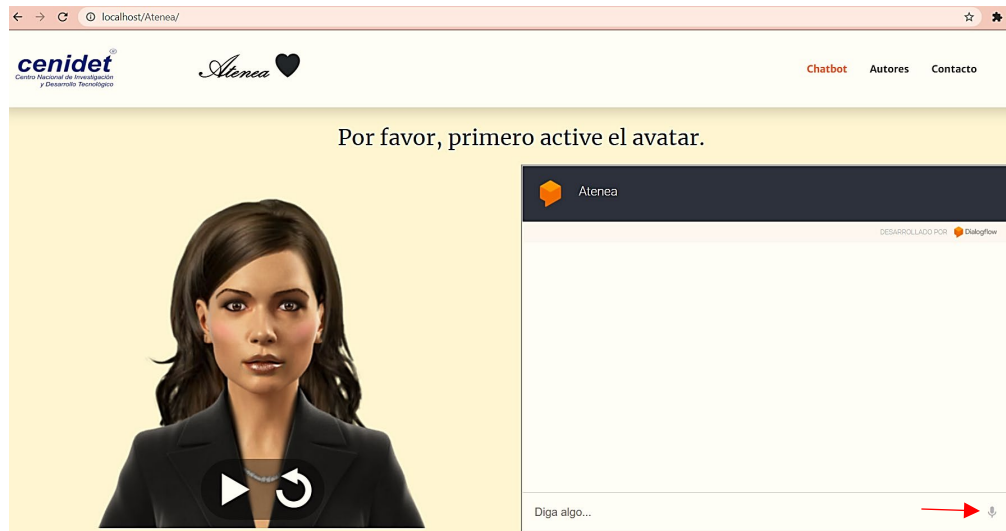


Figura 24.- Botón Chatbot sección dos

La conversación de arriba sigue el flujo del diagrama expuesto en la metodología. Donde se pueden tener diferentes preguntas y respuestas, de acuerdo a lo que vaya respondiendo el interesado. Por ejemplo: Se observa que el rol de Liliana es **Secretaria** y que **sí** existe un intercambio de información, donde la primera entidad secundaria que menciona es **Alumnos**.

4.1.2.3 Botón Autores (Sección 3 página web)

En esta sección se puede visualizar a las personas que desarrollaron el prototipo y el trabajo de investigación. Ver la **Figura 25**.

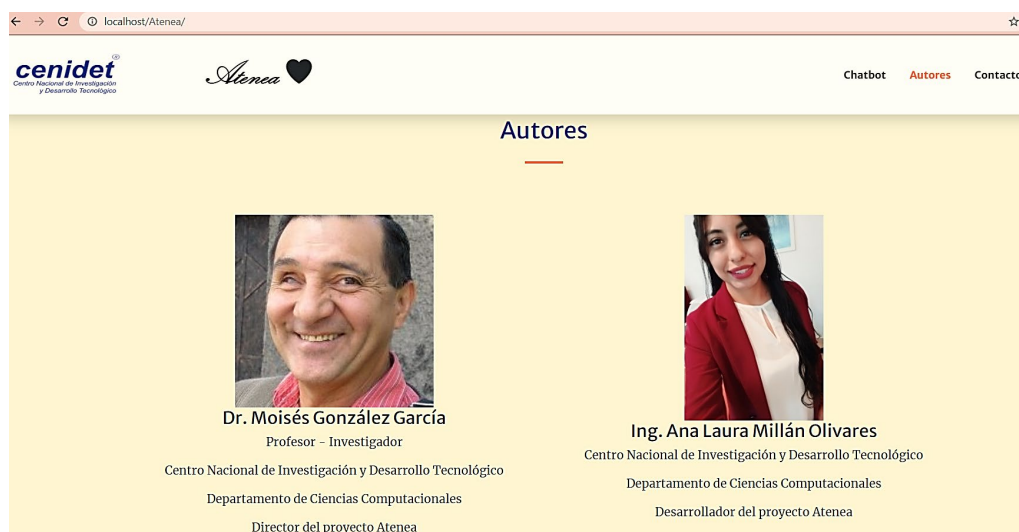


Figura 25.- Botón Autores Sección tres

4.1.2.4 Botón Contacto (Sección 4 página web)

En esta sección se describe lo que se podrá ver si se presiona el botón contacto. Se despliegan los datos de la institución donde se realizó el prototipo, así como el correo para contactar al autor del proyecto Atenea (ver **Figura 26**)

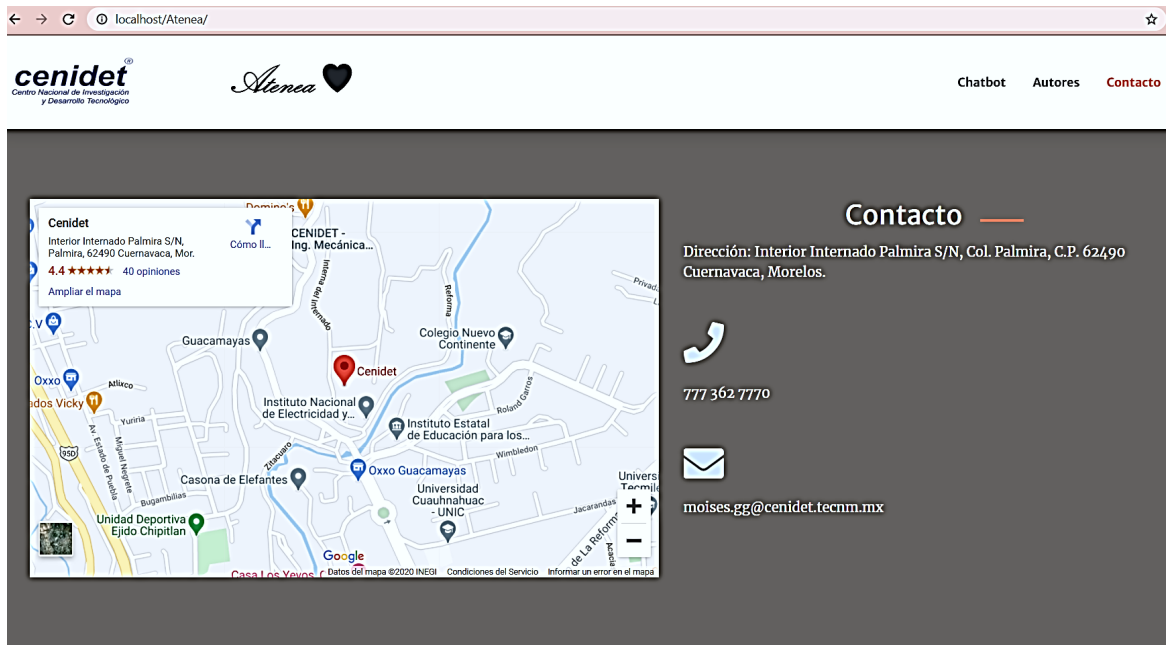


Figura 26.- Botón Contacto Sección cuatro

4.1.2.5 Entorno Google Assistant para presentar Atenea

En este apartado se va a explicar el segundo entorno para mostrar el chatbot al usuario, este entorno se creó para tener una conversación que se asemeje más a una plática de persona a persona, es decir, más fluida, natural y más humana entre el usuario y Atenea. Aquí se utiliza la aplicación del Asistente de Google para llamar el Api de DialogFlow, para poder utilizar esta Api desde el asistente de Google el usuario tiene que tener el correo vinculado con el que se creó el chatbot y tendrá que decir “Hablar con chatbot Atenea” desde la interfaz de Google (ver **Figura 27**)



Figura 27.- Activar chatbot Atenea desde el Asistente de Google

Google responde con “Perfecto, aquí tiene la versión de prueba del chatbot Atenea” y se activará Atenea para poder empezar la entrevista (ver **Figura 28**)

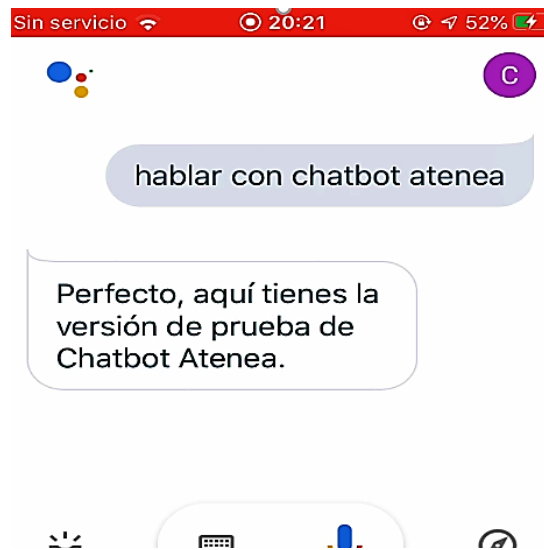


Figura 28.- Google activa el chatbot Atenea

Después de que se activó el chatbot, a diferencia de lo que sucede en la página web, Atenea es quien empieza saludando al usuario para poder empezar con la entrevista. Se empieza a mostrar las preguntas, de acuerdo al flujo que vaya teniendo la entrevista, como se puede ver en la **Figura 29**. Dentro de este entorno no es necesario presionar el micrófono, ya que google tendrá un periodo de escucha después de hacer la pregunta, es por eso que este entorno asemeja más una conversación de persona a persona.

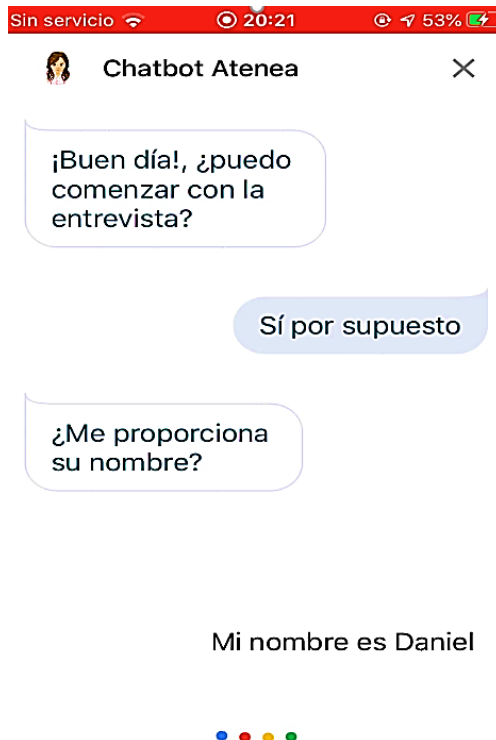


Figura 29.- Entrevista al usuario desde Google Assistant

A continuación, en la **Figura 30** se puede observar una pequeña parte de la conversación de la entrevista que se le realizó a la secretaria de la jefatura del Departamento de Ciencias Computacionales del Centro Nacional de Investigación y Desarrollo Tecnológico. Esta entrevista se realizó desde el entorno gráfico de la Página Web.

Y en la **Figura 31** se observa el historial de dicha conversación dentro de la plataforma de DialogFlow, en la pestaña de historial, que ya se había mencionado.

Por otro lado, en la **Figura 32** se muestra un fragmento de la conversación de la entrevista al Alumno en archivo de texto plano, realizada en la plataforma Google Assistant. Se entrevistó a la entidad Alumno, porque es una persona con la que la secretaria tiene intercambio de información. Así mismo, en la **Figura 33** se observa el historial en la plataforma de DialogFlow, donde se ve que el usuario activa el evento de Google Assistant (señalado en un rectángulo punteado rojo), esto se ve reflejado en el historial tanto de la plataforma como el archivo de texto plano, ya que Google activa el chatbot y Atenea es quién empieza la entrevista, a diferencia de la entrevista realizada mediante la página web, donde es el usuario el que activa el chatbot.



En la **Figura 30** se observa la conversación de la secretaria en archivo de texto plano.

```
1 usuario: Hola.
2 Atenea: Buen día, mi nombre es Atenea, quiero conocer mejor el trabajo que desempeña.¿Puedo realizarle una entrevista?
3 usuario: claro.
4 Atenea: Vamos a conocernos mejor, ¿cuál es su nombre?
5 usuario: me llamo Liliana.
6 Atenea: Liliana, ¿en qué área está?
7 usuario: en ciencias de la computación.
8 Atenea: Me da gusto conocerte Liliana. ¿Qué papel ejerce en este lugar?
9 usuario: soy secretaria.
10 Atenea: Liliana, de este rol de secretaria que me acaba de mencionar. ¿Existe algún intercambio de documentos con algún departamento o persona?
11 usuario: sí.
12 Atenea: Por favor, nombre al departamento o persona con él que intercambia formatos.
13 usuario: con los alumnos.
14 Atenea: Me dijo Alumnos, ¿Alumnos le da a usted algún oficio?
15 usuario: no.
16 Atenea: Voy entendiendo. ¿Qué formatos son los que usted les manda?
17 usuario: les doy el formato m6, m7, m8, m9, m10, m11, m12 y el m13.
18 Atenea: Me mencionó estos formatos m6, m7, m8, m9, m10, m11, m12 y m13, de acuerdo. ¿Estos formatos tienen algún seguimiento entre ellos?
19 usuario: sí.
20 Atenea: Podría explicarme, ¿cuál es la sucesión de dichos formatos?
21 usuario: son en orden numérico.
22 Atenea: ¿Falta algún departamento por mencionar?
23 usuario: sí.
24 Atenea: ¡Ya casi, ya casi! ¿Qué departamento le falta mencionarme?
25 usuario: al jefe de departamento.
26 Atenea: Me dijo Jefe de Departamento, ¿Jefe de Departamento le da a usted algún oficio?
27 usuario: sí.
```

Figura 30.- Historial de la entrevista a la Secretaria archivo .txt

En la **Figura 31**, se observa el historial de la conversación de la secretaria dentro de la plataforma de DialogFlow

History

All platforms ▾ All conversations ▾  Oct 14, 2019 ▾ –  Nov 6, 2020 ▾

Conversation	Date	
USER me llamo liliana	Nov 27, 12:16 PM	
AGENT Liliana, ¿en qué área está?	Nov 27, 12:16 PM	⋮
USER en ciencias de la computación	Nov 27, 12:16 PM	
AGENT Me da gusto conocerte Liliana. ¿Qué papel ejerce en este lugar?	Nov 27, 12:16 PM	⋮
USER secretaria	Nov 27, 12:17 PM	
AGENT Liliana, de este rol de secretaria que me acaba de mencionar. ¿Existe algún intercambio de documentos con algún departamento o persona?	Nov 27, 12:17 PM	⋮
USER si	Nov 27, 12:17 PM	
AGENT Por favor, nombre al departamento o persona con él que intercambia formatos	Nov 27, 12:17 PM	⋮

Figura 31.- Historial de la entrevista a la secretaria en DialogFlow



En la **Figura 32**, se observa el historial de la conversación entre Atenea y el alumno, en archivo de texto plano


```
1 Atenea: ¡Buen día!, ¿puedo comenzar con la entrevista?.
2 Usuario: Sí por supuesto.
3 Atenea: ¿Me proporciona su nombre?.
4 Usuario: Mi nombre es Daniel.
5 Atenea: Daniel, ¿en qué área está?.
6 Usuario: estoy yendo en departamento de ciencias computacionales.
7 Atenea: Un placer Daniel, ¿Qué rol tiene en este departamento?.
8 Usuario: soy estudiante de ingeniería de software.
9 Atenea: Daniel, conforme a este rol de estudiante, ¿Debe haber un intercambio de información con alguna persona o departamento?.
10 Usuario: claro intercambio información con servicios escolares.
11 Atenea: Dijo Departamento de Servicios Escolares, ¿recibe algún formato de Departamento de Servicios Escolares?.
12 Usuario: Sí claro.
13 Atenea: ¿Y qué formatos recibe de ellos?.
14 Usuario: ellos me dan mi boleto de calificaciones Y horario y mi credencial.
15 Atenea: Dígame, ¿qué pasa con esos formatos?.
16 Usuario: solo se guardan.
17 Atenea: ¿Usted le manda algo a ellos?.
18 Usuario: si también.
19 Atenea: Entiendo. Y me podría decir, ¿cuáles son los formatos que le envía?.
20 Usuario: yo les doy el comprobante de pago y mi horario.
21 Atenea: ¡Qué interesante!, tengo otra duda ¿Entre estos formatos comprobante de pago Y mi horario, y los anteriores, hay una dependencia?.
22 Usuario: no, son independientes.
23 Atenea: ¿Hay un departamento más con el que se cambie información?.
24 Usuario: sí, hay más.
25 Atenea: ¿Qué departamento no se ha mencionado?.
26 Usuario: esta recursos financieros.
27 Atenea: Me dijo Departamento de Recursos Financieros, ¿recursos financieros le da a usted algún oficio?.
28 Usuario: ajá.
29 Atenea: Me puede decir, ¿cuáles son esos formatos que les recibe?.
30 Usuario: pues me dan mi comprobante de pago.
```




Figura 32.- Historial en archivo .txt de la entidad Alumno

En la **Figura 33**, se presenta la conversación del alumno dentro de DialogFlow, donde Google Assistant es el evento que lleva la conversación, el cual es invocado por el usuario.

History

All platforms ▾ All conversations ▾  Jun 9, 2020 ▾ —  Nov 6, 2020 ▾

Conversation Date 

25+ interactions   
Duration: 5 min














 USER	GOOGLE_ASSISTANT_WELCOME  Event	Jun 9, 8:21 PM	
 AGENT	¡Buen día!, ¿puedo comenzar con la entrevista?	Jun 9, 8:21 PM	
 USER	Sí por supuesto	Jun 9, 8:21 PM	
 AGENT	¿Me proporciona su nombre?	Jun 9, 8:21 PM	
 USER	Mi nombre es Daniel	Jun 9, 8:21 PM	
 AGENT	Daniel, ¿en qué área está?	Jun 9, 8:21 PM	
 USER	estoy yendo en departamento de ciencias computacionales	Jun 9, 8:21 PM	
 AGENT	Un placer Daniel, ¿Qué rol tiene en este departamento?	Jun 9, 8:21 PM	

Figura 33.- Historial del Alumno en DialogFlow

SECCIÓN 2

4.1.3 FASE DOS - ANÁLISIS DEL TEXTO

Dentro de esta fase se encuentran dos de las sub-fases más importantes dentro de la investigación, ya que de estas depende la correcta identificación de los elementos para la creación del diagrama de entidades (Ver **Figura 34**). Estas sub-fases son: la sub-fase uno **Etiquetado** y la sub-fase dos **Archivo.csv** de la metodología ilustrada anteriormente. Estas son las que realizan el procesamiento del texto que se genera de la conversación entre en chatbot y el interesado.

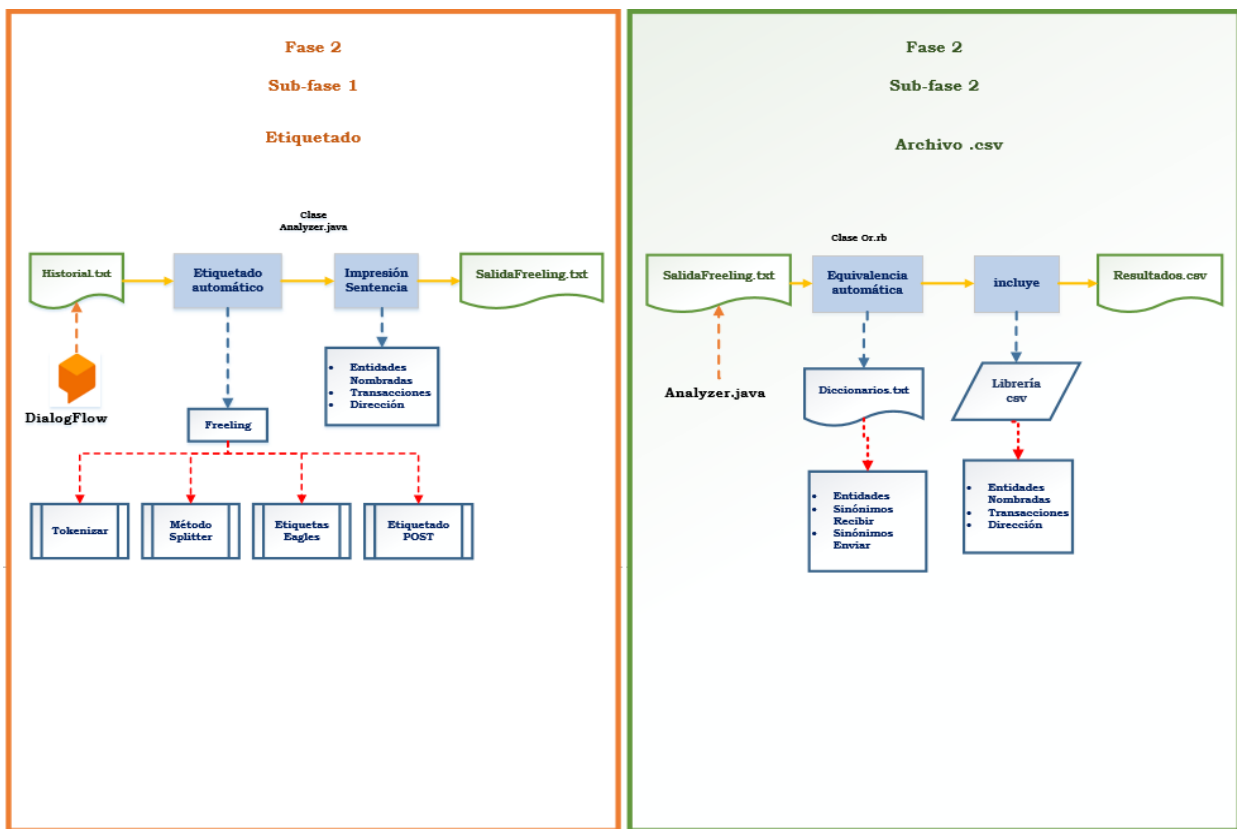


Figura 34.- Sub-fases de la fase Análisis del texto

4.1.3.1 Sub-Fase Uno Etiquetado

El texto obtenido del chatbot Atenea se trabajó a nivel morfosintáctico. Este nivel, es uno de los seis que existen en la disciplina del PLN, con los que se pueden trabajar los textos, en la siguiente **Figura 35** se ilustran los niveles.

En este trabajo de investigación se utilizó el nivel morfológico que estudia la estructura interna de las palabras, para definir, delimitar y clasificar sus unidades y el nivel sintáctico, que como se puede observar en la imagen se encarga del estudio de las reglas de las palabras, es decir, estudia las formas en que se componen las palabras, así como el vínculo sintagmático y paradigmático existentes entre ellas.

La morfología como se mencionó trabaja sobre el análisis de las palabras, este incluye la forma, que es la característica a partir del análisis de sus constituyentes con significado (léxico o gramatical). A estos constituyentes les llamamos morfemas (Ej. gato: **gat** (lexema) + **o** (morfema con significado de género masculino)) y estructura (Ej. número, género, tiempo, etc.). Y la sintaxis trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración (sujeto, objeto directo, adjetivo, etc.) y que sintagmas (Palabra o grupo de palabras que constituyen una unidad sintáctica y que cumplen una función determinada con respecto a otras palabras de la oración) son parte de una oración.



Figura 35.- Niveles del PLN (Limited, 2020)

Un monema es la unidad mínima del lenguaje dotada de significado, donde toda palabra tiene al menos un monema. A su vez, el monema se divide en dos tipos como son:

- **Monema léxico o lexema:** Es aquel conjunto de caracteres que brinda significado (información) con referencia al mundo exterior, es decir, van a hacer referencia a elementos del mundo.
- **Monema gramatical o morfema:** Hace referencia a aquel conjunto de caracteres o palabras que su concepto es únicamente gramatical, es decir, te dan información sobre la categoría gramatical de la palabra correspondiente al mundo de las categorizaciones (género, número, tiempo, etc.)

Las palabras se agrupan en clases de palabras denominadas **categorías gramaticales** de acuerdo a la función, forma y significado. Para determinar la categoría morfológica a la que pertenece el lexema en la palabra analizada, se tiene que unir el morfema al lexema.

Por otro lado, la sintaxis se define como la disposición de palabras en una oración para mostrar su relación. Describe la secuencia de símbolos que constituyen programas válidos. En informática es el conjunto de reglas que indican las secuencias correctas de los elementos propios de un lenguaje de programación. Es el plano en el que se desenvuelve: **oración** (serie o cadena de palabras que transmite un sentido completo), los elementos de análisis son:

- Palabras
- Sintagmas
- Oraciones

La sintaxis muestra al hablante la forma como debe escribir oraciones bien estructuradas, la sintaxis es más útil al usuario del lenguaje que al sistema de PLN, debido a que el usuario es quien debe saber cómo organizar las oraciones, el sistema no necesita esta información, quien le dará la estructura al sistema es el usuario.

En la **Figura 36** se puede observar un ejemplo de sintaxis de dependencia, que nos muestra la relación de las palabras, es decir, se concentra en las relaciones entre palabras, sin indicar cómo se agrupan.

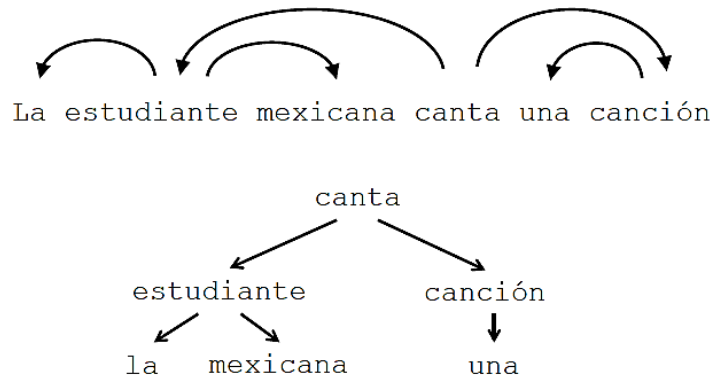


Figura 36.- Ejemplo de sintaxis de dependencia

En la figura anterior, se puede observar una oración, dónde las flechas indican qué palabras se relacionan, además señalan el sentido de la dependencia, y se puede ver que la palabra [estudiante] está relacionada con las palabras [la y mexicana], de igual manera la palabra [canta] está relacionada con las palabras [canción, una, estudiante, la, y mexicana].

Una vez explicado el nivel en el que se trabajó el texto, más adelante se muestra el diagrama adaptado para esta investigación, que explica el comportamiento del etiquetado creado en el lenguaje de Java, junto con la librería de Freeling, que es un analizador morfosintáctico que funciona en 15 idiomas entre ellos el idioma español, que es el idioma en el que se trabajó esta investigación. La simbología ocupada en este diagrama se presenta en la **Figura 37**.

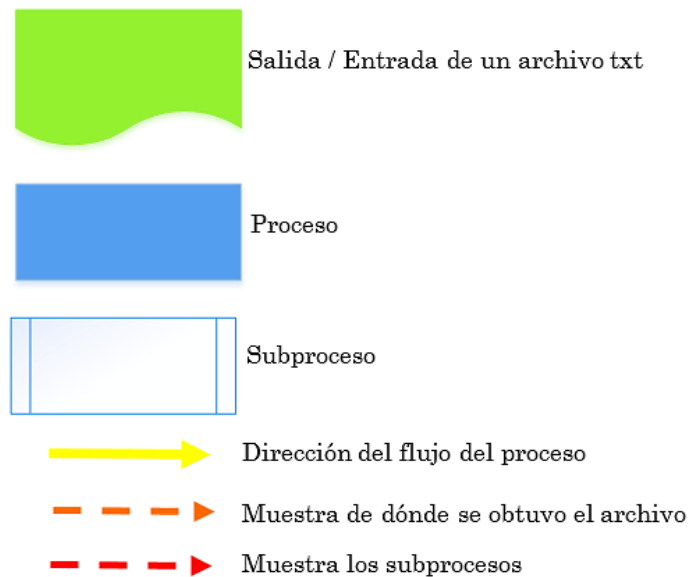


Figura 37.- Simbología diagrama Etiquetado con Freeling

En la **Figura 38** se puede observar el proceso que se realizó para llevar a cabo el etiquetado de la conversación obtenida desde el chatbot.

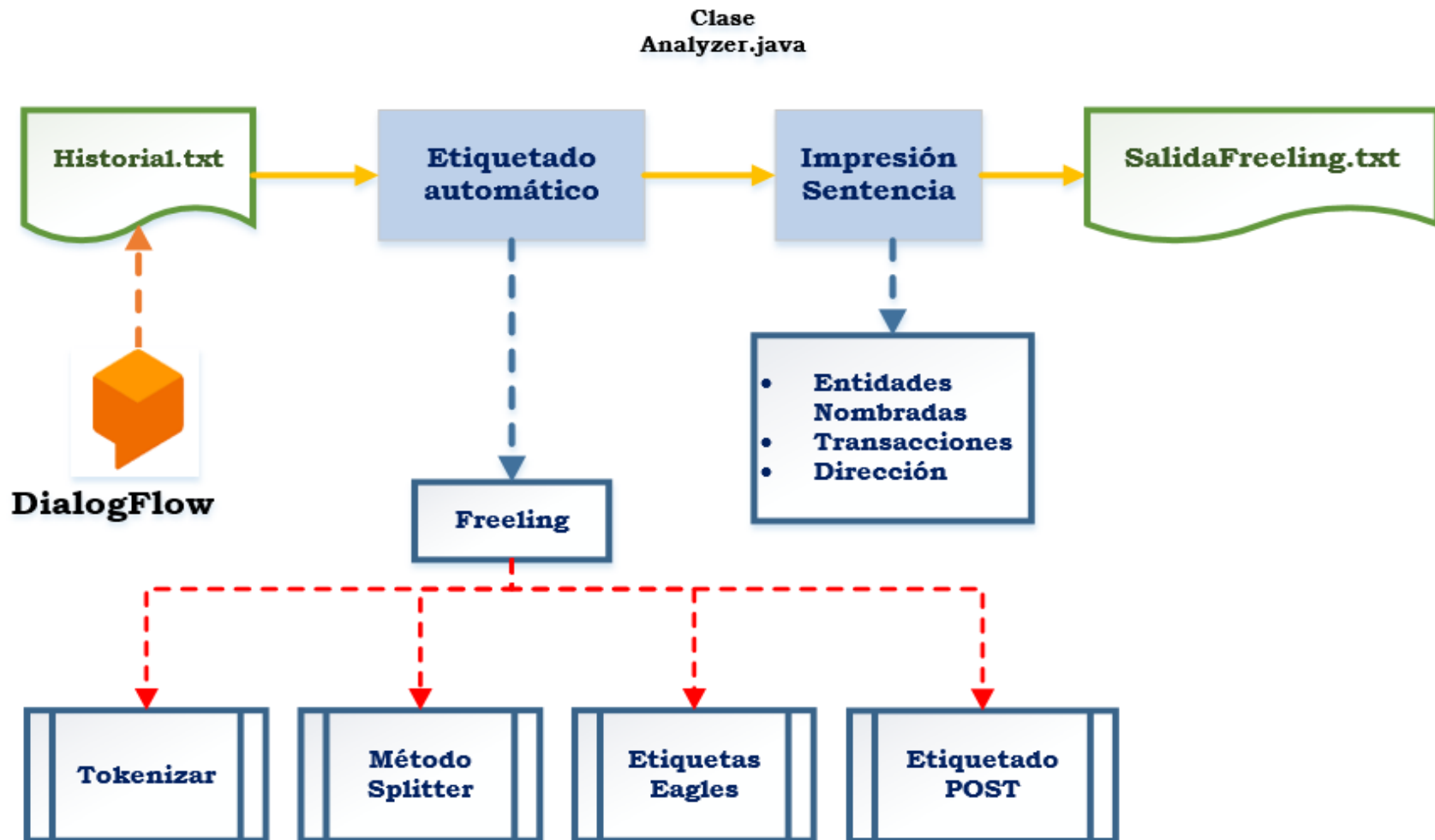


Figura 38.- Diagrama clase Analyzer. java

Seguidamente, se explica el proceso que sigue el diagrama de la **Figura 38**.

- 1) Se necesita tener el historial de la conversación en un archivo de texto plano, este se obtendrá desde la plataforma de DialogFlow, dentro de la plataforma hay un apartado llamado **Historial**, aquí se mostrarán todas las conversaciones que se hayan hecho con el chatbot, también se podrán buscar por fecha de realización, se necesita seleccionar el historial de la conversación deseada y guardarlo en un archivo .txt
- 2) Una vez que se tiene el historial en un archivo de texto plano, se manda a llamar dentro de la clase Analyzer.java, como se muestra en la **Figura 39**, esta es una clase que se creó en el lenguaje de java, el cual contiene el analizador morfosintáctico llamado Freeling.

```
//LEER ARCHIVO  
Path archivo = Paths.get("./", "Entrevista_Presidenta.txt");
```

Figura 39.- Ejemplo de cómo se lee el historial dentro de la clase Analyzer.java

En la figura anterior, se puede observar cómo se manda a llamar el archivo .txt, para este caso el archivo se llama **Entrevista_Presidenta.txt**.

- 3) Después de haber encontrado el archivo a etiquetar, se empieza a ejecutar el analizador morfosintáctico Freeling, el texto se pasará por el proceso **Etiquetado automático**, este proceso necesita de subprocesos como se explica a continuación:
 - a) Primero, el texto se va a separar por tokens (separación de cada palabra del texto), es decir, se separará el texto en cada secuencia de caracteres no separada por espacios en blanco, por ejemplo: la palabra [Hola] es un token, el saludo [buenos días] se conforma por dos tokens, [buenos] es un token y [días] es otro token.
 - b) Segundo, una vez que se tienen los tokens, el texto pasa por el método Splitter, el cual ayudará a separar el texto por oraciones y sirve para saber en dónde termina cada palabra, para este caso la palabra terminará en un espacio en blanco. Por ejemplo: El saludo tokenizado [Buenos] [Días], tiene un espacio entre cada token, lo que hace el método Splitter es eliminar ese espacio en blanco y pasar [Días] al siguiente párrafo.

- c) Y finalmente, el texto se procesa con las etiquetas Eagles y el etiquetado Post (**part-of-speech tagging, POS tagging**, por sus siglas en inglés) estos ayudaran en el proceso de etiquetar cada palabra del texto con su categoría gramatical. Este etiquetado se hace de manera automática y puede tardar unos minutos en terminar, esto dependerá del peso del archivo a etiquetar.
- 4) Una vez que se realizó todo el proceso del etiquetado, se pasa al proceso **Impresión sentencia**, aquí se mostrarán desde la línea de comandos las siguientes sentencias, junto con el texto procesado:
 - a) Las entidades nombradas encontradas (una entidad nombrada es una palabra que se identifica como nombre de persona, organización, lugar, fecha, tiempo), por ejemplo: Liliana, Secretaria o Depto. de Ciencias Computacionales.
 - b) Las transacciones, son los formatos que se intercambian entre las entidades, por ejemplo: Evaluación de seminario, etc.
 - c) La dirección, es de dónde sale cada formato, esta dirección se definirá con los sinónimos de enviar o recibir.
 - 5) Por último, todas las sentencias encontradas se guardan en un archivo de texto plano, y a su vez el archivo se guarda en la misma dirección donde se encuentre la clase Analyzer.java. En el punto **4.1.3.2 Resultados Sub-Fase Uno- Etiquetado**, se pueden encontrar los resultados de este etiquetado.

4.1.3.2 Resultados Sub-Fase Uno- Etiquetado

En este apartado se van a explicar los resultados que se obtuvieron de los historiales de las tres entidades entrevistadas. Como se había mencionado en el punto anterior, el analizador morfosintáctico llamado Freeling se implementó en el lenguaje de Java., este programa se va a ejecutar desde la línea de comandos de Windows, los comandos para ejecutarla se presentan en la **Figura 40** y la **Figura 41**.

También se comentó que este programa necesita como entrada el archivo en texto plano que se analiza. Estas entradas son los historiales mostrados en la sección anterior.

Después de que dentro de la clase en java llamada `Analyzer.java` se especificó el documento que se va analizar, en la línea de comandos se va a dirigir a la carpeta donde se encuentre la clase, en este caso la ruta es `C:\Users\laura\OneDrive\Escritorio\PLN`, delante de esa ruta se va a ejecutar el primer comando `javac -classpath freeling.jar Analyzer.java` como se presenta en la **Figura 40**.

```
C:\Users\laura\OneDrive\Escritorio\PLN> javac -classpath freeling.jar Analyzer.java
```

Figura 40.- Primer comando para ejecutar la clase `Analyzer.java`

Si no se presentó ningún error, se da enter y se procede a ejecutar el segundo comando `javac -classpath freeling.jar;. Analyzer`. Como se presenta en la **Figura 41**.

```
C:\Users\laura\OneDrive\Escritorio\PLN> java -classpath freeling.jar;. Analyzer
```

Figura 41.- Segundo comando para ejecutar la clase `Analyzer.java`

Cuando ya hayan ejecutado los comandos sin ningún error, automáticamente la clase `Analyzer.java` empezará con el procesamiento del texto, este proceso es interno y no se puede visualizar en la línea de comandos. Como ya se mencionó este proceso se divide en distintos procesos, como primer punto el texto pasará por la sentencia tokenizar, que procesa cada palabra dentro de texto definido de la **Figura 42** de la clase `Analyzer.java`, donde “`I`” es la variable que guarda el texto tokenizado, para utilizar esta variable posteriormente.

```
86 // Extract the tokens from the line of text.  
87 ListWord l = tk.tokenize( texto );
```

Figura 42.- Tokenización del historial de conversación

En segundo lugar, se pasa a la sentencia `Splitter` que se guarda en la variable “`ls`”, esta sentencia se ejecuta con la siguiente línea como se puede observar en la **Figura 43**, aquí se ocupa la variable “`I`” de la sentencia tokenizar anteriormente explicada, y se pone “`false`”

porque no se determina un carácter en especial para separar las oraciones, sino que se toma como separación el espacio en blanco.

```
89 // Split the tokens into distinct sentences.  
90 ListSentence ls = sp.split( 1, false );
```

Figura 43.- Sentencia Splitter para procesar el historial

El tercer paso es, pasar el texto por el análisis morfológico, por el método POST y finalmente el texto pasará por las etiquetas Eagles que será la que nombra las entidades. Todas utilizando la variable “ls” del método Splitter. Estas sentencias se muestran en la **Figura 44**.

```
92 // Perform morphological analysis  
93 mf.analyze( ls );  
94  
95 // Perform part-of-speech tagging.  
96 tg.analyze( ls );  
97  
98 // Perform named entity (NE) classification.  
99 neclass.analyze( ls );
```

Figura 44.- Sentencia para nombrar las entidades dentro del texto

Finalmente, cuando este proceso se haya ejecutado, en la línea de comandos, se mostrará el resultado final. Es decir, se mostrará cada palabra con su respectiva categoría gramatical de acuerdo a las etiquetas Eagles, explicadas en el capítulo 2 en la sección **2.2.12 Etiquetas Eagles**, como se observa en la **Figura 45**.

```
----- LANG_IDENT results -----
Language detected (from first line in text): es
----- TAGGER results -----

Usuario usuario NP00SP0
: : Fd
no no RN
de de SPS00
que que CS
, , Fc
gracias gracia NCFP000
. . Fp

Atenea atenea NP00SP0
: : Fd
Muchas mucho DI0FP0
gracias gracia NCFP000
, , Fc
hemos haber VAIP1P0
acabado acabar VMP00SM
la el DA0FS0
entrevista entrevista NCFS000
. . Fp

Usuario usuario NCMS000
: : Fd
bye bye VMIP3S0
. . Fp

Atenea atenea NP00SP0
: : Fd
bye bye VMIP3S0
. . Fp
```

Figura 45.- Impresión en línea de comandos de las palabras con su categorización correspondiente

En la figura anterior se puede observar la forma, el lema y el etiquetado, respectivamente del texto de la entrevista. Por ejemplo, en el verbo **[hemos]** (señalado con una flecha roja), es el verbo en la forma que lo dijo el interesado, el etiquetado muestra el lema **[haber]** el cuál es el verbo sin conjugación, y la etiqueta correspondiente a ese verbo es **[VAIP1P0]** que se lee como Verbo, Auxiliar, Indicativo, Presente, Primera persona (representada con el número 1), Plural, y como no cuenta con un género se agrega un cero (0), esto de acuerdo con las etiquetas Eagles. Así pasa con cada una de las palabras que contenta el texto a etiquetar.

Por otro lado, conforme se imprime en la línea de comandos, se guarda un archivo en texto plano que contiene las palabras y su categoría para que se pueda consultar o utilizar posteriormente, este se guarda en la misma carpeta en donde está la clase, como se puede ver en la **Figura 46**, en este caso el archivo de salida se llama **SalidaFreeling1**, el numero cambia según sea el número de la entrevista que se esté procesando, este nombre es editable.

```

SalidaFreeling1.txt
82 Liliana liliana NP00V00
83 . . Fp
84 ¿ ¿ Fia
85 Qué qué DT0CN0
86 papel papel NCMS000
87 ejerce ejercer VMIP3S0
88 en en SPS00
89 este este DD0MS0
90 lugar lugar NCMS000
91 ? ? Fit
92 . . Fp
93 usuario usuario NCMS000
94 : : Fd
95 secretaria secretario NCFS000
96 . . Fp
97 Aniabot aniabot NP00SP0
98 : : Fd
99 Liliana liliana NP00SP0
100 , , Fc
101 de de SPS00
102 este este DD0MS0
103 rol rol NCMS000
104 de de SPS00
105 secretaria secretario NCFS000

149 alumnos alumno NCMP000
150 . . Fp
151 Atenea atenea NP00SP0
152 : : Fd
153 Me me PP1CS000
154 dijo decir VMIS3S0
155 Alumnos alumnos NP00SP0
156 , , Fc
157 ¿ ¿ Fia
158 Alumnos alumnos NP00SP0
159 le le PP3CSD00
160 da dar VMIP3S0
161 a a SPS00
162 usted usted PP2CS00P
163 algún alguno DI0MS0
164 oficio oficio NCMS000

```

Figura 46.- Historial con su categoría gramatical

En la **Figura 46** se puede observar que el texto de entrada ahora está dividido por palabras y cada palabra contiene su respectiva categoría gramatical, como se explicó en la **Figura 45**.

En la **Figura 46** están marcadas las entidades nombradas (con rectángulos punteados color rosa), que muestran algunas entidades que serán de utilidad para realizar la representación del diagrama de entidades a nivel usuario, así mismo se puede observar el sinónimo del verbo enviar (rectángulo punteado color azul), dicho por el usuario como “dar”.

SECCIÓN 3

4.1.3.3 Sub-Fase Dos Archivo .csv

Esta sub-fase es la última de la **fase dos- Análisis del texto**, aquí es cuando se crea un archivo que pueda leer la computadora y al mismo tiempo muestre la información mínima necesaria para crear el diagrama de entidades a nivel usuario. Dentro de esta etapa también se creó un diagrama adaptado para esta investigación, el cual muestra el comportamiento de la clase creada en ruby llamada Or. rb (Ver **Anexo E**) que realiza una comparación del texto resultante de la etapa anterior hasta crear el archivo .csv el cual es un formato legible por la

computadora y que será de utilidad para el analista, quien creará el diagrama de entidades a nivel usuario. Por el momento este diagrama se realiza de forma manual. La simbología que se utilizó en este diagrama es la misma del diagrama anterior, y en la **Figura 47** se presenta la simbología.

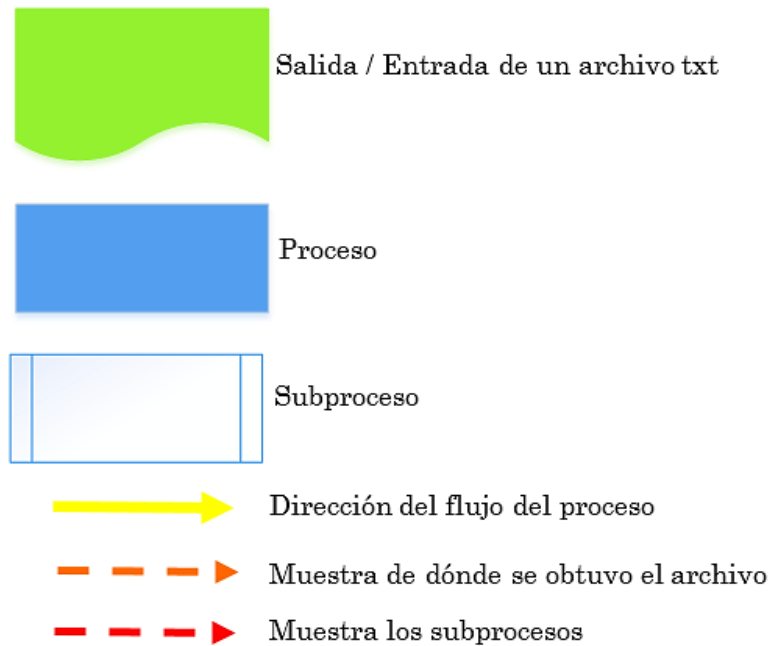


Figura 47.- Simbología del diagrama para la creación del archivo .csv

La **Figura 48** expone el proceso que realiza la clase Or.rb donde se definirá la información mínima requerida para la creación del diagrama de entidades a nivel usuario.

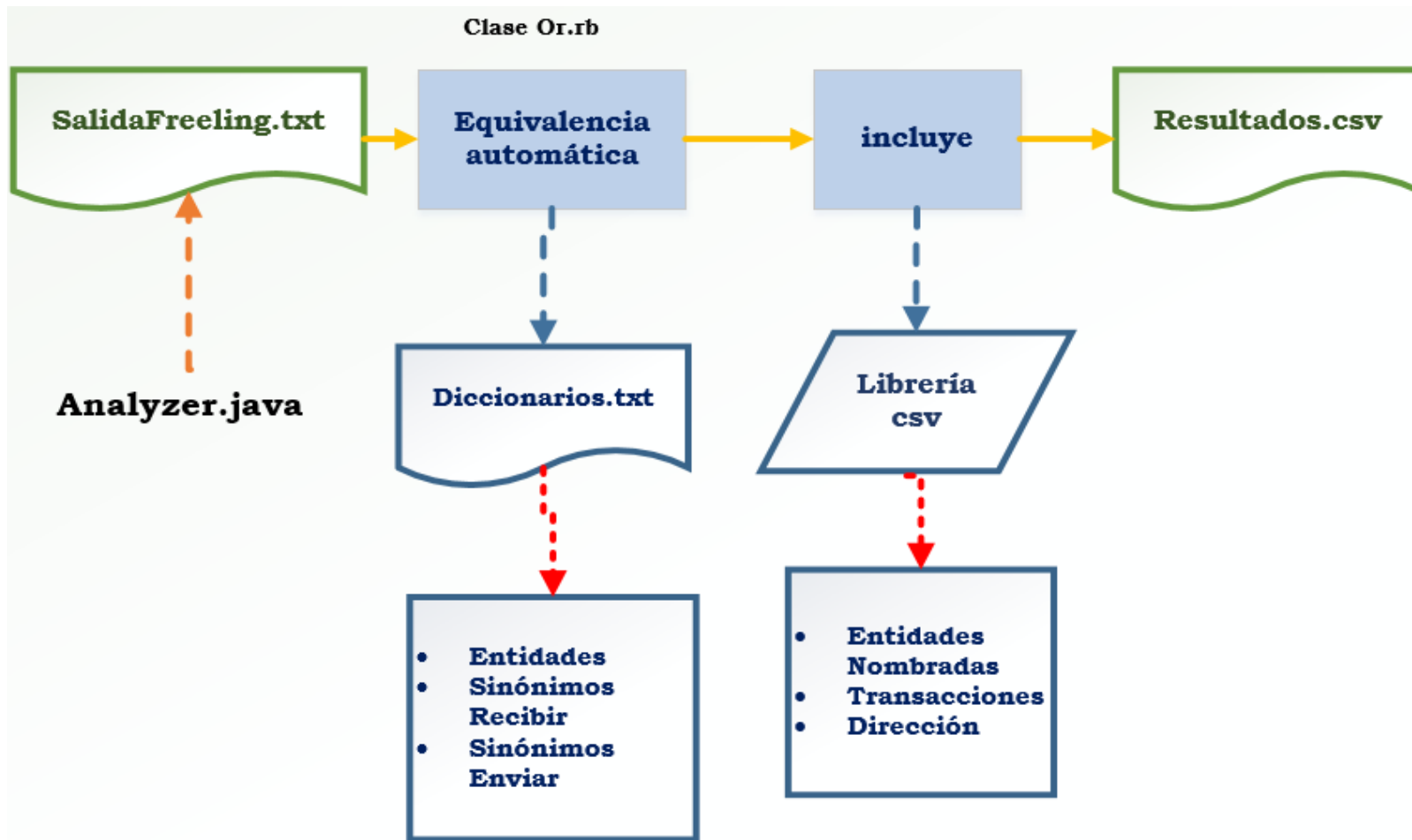


Figura 48.- Diagrama Clase Or.rb

Posteriormente, se describen los pasos que se realizan en la clase Or desde la entrada del archivo hasta la salida del mismo.

Es importante mencionar que la clase fue creada en el lenguaje de ruby, el cuál es un lenguaje de programación dinámico *Open Source* (de código abierto) y el cual se puede modificar sin necesidad de tener una licencia. Se escogió este lenguaje de programación principalmente porque es de código abierto, y también porque no tiene una estructura en la programación por lo que se puede usar criterios simples (Ej. Las variables no necesitan declararse con algún valor o tipo de dato como en java) para determinar y mencionar a las variables. Esta característica mejora su legibilidad frente a otros lenguajes. Dicho lo anterior, los pasos que sigue esta clase son los siguientes:

- 1) Esta clase necesita como entrada un archivo de texto plano. En este caso es el archivo que se creó en la etapa anterior, producido por la clase **Analyzer.java**. Para este caso el archivo de salida que creó la etapa anterior se llama **SalidaFreeling.txt**.
- 2) Este archivo lo lee el proceso **Equivalencia automática**. El proceso compara el archivo de entrada con cada uno de los diccionarios predefinidos para ir reconociendo las entidades (origen/destino), las transacciones y la dirección que siguen esas transacciones. Estos diccionarios son cuatro y también se encuentran en archivos de texto plano, los cuales son:
 - **Diccionario de entidades**, este diccionario tiene todas las entidades que se pudieran utilizar dentro de la entrevista, aquí también se utilizó el glosario de términos administrativos, el cual se empleó para crear el vocabulario del chatbot, esto para tener el mismo entrenamiento tanto del chatbot como de la clase. Sin correr el riesgo de que falte algún termino o departamento.
 - **Diccionario de sinónimos del verbo enviar**, aquí se encuentran todas las variaciones del verbo enviar, por ejemplo: **mandar, dirigir, remitir**, estos verbos se encuentran en su forma simple, es decir, sin conjugar, porque recordemos que eso hace el etiquetado, pone las palabras en su forma simple, y por lo tanto es más fácil de reconocer, a tener un verbo conjugado. El

objetivo de este diccionario es definir la dirección del formato solo con el verbo enviar, sin importar que sinónimo utilice el usuario (interesado). De este modo el analista sabrá que la dirección de salida siempre se definirá con el verbo **enviar**.

- **Diccionario de sinónimos del verbo recibir**, al igual que el diccionario anterior, este archivo de texto plano tiene los posibles sinónimos del verbo recibir, por ejemplo: **aceptar, tomar, acoger, agarrar, etc.**, estos también irán en su forma simple. Y el objetivo de este archivo es dar también la dirección de la transacción mencionada por el usuario. Así el analista sabrá que el otro verbo para definir la dirección de la transacción estará marcado por el verbo **recibir**.
- **Diccionario de formatos**, este es un archivo que contendrá todos los formatos utilizados dentro del dominio a trabajar, para este caso solo se encontrarán los formatos utilizados para el departamento de Ciencias Computacionales en el proceso de maestría.

El vocabulario de los diccionarios se puede incrementar de acuerdo al dominio que se vaya a trabajar, e incluso para los diferentes departamentos que se encuentren dentro de la misma institución.

4.1.3.4 Resultados Sub-Fase Dos – Archivo .csv

Esta es la sub-fase final para procesar el texto, hasta este punto todo el proceso desde la entrevista entre Atenea y el usuario y el etiquetado del historial debe hacerse correctamente y sin ningún inconveniente, debido a que la salida generada de las dos fases anteriores es la entrada para esta. En donde se generan los datos necesarios para representar el dominio del problema.

Para lograrlo se creó un programa en el lenguaje de Ruby, que cuenta con cuatro diccionarios que ayudan a la **clase Or.rb** a ir comparando cada línea del archivo entrante, los cuales ya se describieron en la parte de arriba.

Para hacer funcionar la clase se tiene que importar la librería del archivo a crear, en este caso se importó la librería del archivo .csv, esta sentencia se puede ver en la **Figura 49**.

```
3 require 'csv'
```

Figura 49.- Librería .csv en ruby

Después se le tiene que dar una estructura (nombrar las columnas) del archivo que se va a crear, este encabezado se puede observar en la **Figura 50**.

```
or.rb x
5 # Declara el encabezado para el archivo que se creará
6 encabezado = ['Entidad-Origen', 'Direccion', 'Formato', 'Entidad-Destino']
```

Figura 50.- Encabezado del archivo .csv

Finalmente, una vez que se dio una estructura al documento a crear, se empezará a comparar el archivo con los diccionarios establecidos, es decir, se buscará línea por línea alguna palabra que coincida con la de los diccionarios, si se encuentra la coincidencia se agregará a la columna correspondiente, esto se hará con los cuatro diccionarios.

Esta comparación se hace con el método **include** de ruby. El método include, como su nombre lo dice, da información sobre sí se incluye o no la palabra buscada, este método se resalta (con un rectángulo punteado rojo) en la **Figura 51**.

```
if linea2.include? linea.strip
  puts "El origen "+linea.strip
```

Figura 51.- Método include en ruby

Por último, cuando se terminó de comparar cada línea que contiene el archivo etiquetado, las entidades que se hayan reconocido, así como la dirección de cada transacción serán agregadas en las columnas del archivo .csv respectivamente, este archivo se guardará en la misma carpeta en donde se encuentra la clase Or. rb y se llama **resultado.csv**, en la **Figura 52** se muestra la salida para la entrevista de la entidad principal secretaria.

A	B	C	D
Entidad_Origen	Direccion	Formato	Entidad_Destino
secretario	enviar	m6	alumno
secretario	enviar	m7	alumno
secretario	enviar	m8	alumno
secretario	enviar	m9	alumno
secretario	enviar	m10	alumno
secretario	enviar	m11	alumno
secretario	enviar	m12	alumno
secretario	enviar	m13	alumno
secretario	enviar	m4	directores_de_tesis
secretario	enviar	m5	directores_de_tesis
secretario	recibir	m4	directores_de_tesis
secretario	recibir	m5	directores_de_tesis
secretario	recibir	m6	directores_de_tesis
secretario	recibir	m9	directores_de_tesis
secretario	recibir	m10	directores_de_tesis
secretario	recibir	m11	directores_de_tesis
secretario	recibir	m12	directores_de_tesis
secretario	enviar	m3	jefe_de_departamento
secretario	enviar	m4	jefe_de_departamento
secretario	enviar	m5	jefe_de_departamento
secretario	enviar	m14	jefe_de_departamento
secretario	enviar	m17	jefe_de_departamento
secretario	enviar	m18	jefe_de_departamento
secretario	enviar	m20	jefe_de_departamento
secretario	recibir	m3	jefe de departamento

Figura 52.- Archivo resultado.csv para la entrevista a la secretaria

Como se puede observar en la figura anterior, se reconocieron adecuadamente las **entidades** las cuales están señaladas con un rectángulo punteado rosa, se reconoció la dirección de las transacciones, es decir, cuando se manda un archivo desde la entidad principal se define con el verbo **enviar** (rectángulo punteado naranja), cuando la entidad principal recibe algún documento de las entidades secundarias se define por el verbo **recibir** (rectángulo punteado azul), y también se reconocieron los **formatos** (rectángulo punteado verde) que se intercambia por las entidades. Los nombres de los formatos se forman por la letra “m” y un

número, que representa el avance del alumno, en el procedimiento de la maestría en el Departamento de Ciencias de la Computación.

En la **Figura 53** se puede observar el archivo resultante de la entrevista al alumno. Que es similar al mostrado en la Figura anterior.

A	B	C	D
Entidad_Origen	Direccion	Formato	Entidad_Destino
alumno	enviar	voucher	recursos_financieros
alumno	recibir	comprobante_pago	recursos_financieros
alumno	dar	comprobante_pago	departamento_de_servicios_escolares
alumno	recibir	boleta_calificaciones	departamento_de_servicios_escolares
alumno	recibir	credencial	departamento_de_servicios_escolares
alumno	enviar	horario	departamento_de_servicios_escolares
alumno	enviar	m7	jefe_de_departamento
alumno	recibir	m6	secretario
alumno	recibir	m7	secretario
alumno	recibir	m8	secretario
alumno	recibir	m9	secretario
alumno	recibir	m10	secretario
alumno	recibir	m11	secretario
alumno	recibir	m12	secretario
alumno	recibir	m13	secretario
alumno	enviar	m6	director
alumno	enviar	m7	director
alumno	enviar	m8	director
alumno	enviar	m9	director
alumno	enviar	m10	director
alumno	enviar	m11	director
alumno	enviar	m12	director

Figura 53.- Archivo resultado2.csv para la entrevista al alumno

Y en la **Figura 54** se puede observar el resultado de la entrevista a la presidenta de claustro, la cual es una entidad con la que se intercambia información.

A	B	C	D
Entidad_Origen	Direccion	Formato	Entidad_Destino
presidenta_de_el_claustro	enviar	habilitación de los profesores	jefe_de_departamento
presidenta_de_el_claustro	enviar	asignación de comité	jefe_de_departamento

Figura 54.- Archivo resultado3.csv para la entrevista a la presidenta de claustro

SECCIÓN 4

4.1.4 FASE TRES - CREACIÓN DEL DIAGRAMA DE ENTIDADES

Esta fase es la última fase por realizar, una vez que se haya realizado la entrevista, y el historial haya sido procesado con las herramientas del procesamiento del lenguaje natural, lo último que resta es corroborar si se obtuvo la información mínima necesaria para crear el diagrama de entidades a nivel usuario propuesto por Warnier Orr. Esto se hará con la ayuda del archivo .csv creado en la fase **Análisis del texto** en la sub-fase dos **Creación del archivo .csv**

Justo después de contar con el archivo, se procedió a buscar un editor que nos ayudara a crear el diagrama de entidades, para este caso se utilizó el programa Visio de Microsoft, Visio es un programa gratuito que ya viene en los programas dentro de cualquier paquete Microsoft que tenga nuestra computadora. Por otro lado, lo que se tiene que hacer es ir siguiendo el archivo .csv y la estructura descrita en la metodología, para ir poniendo dentro del editor los elementos necesarios para crear el diagrama. La simbología que se utilizó para realizar el diagrama se puede visualizar en la **Figura 55**.

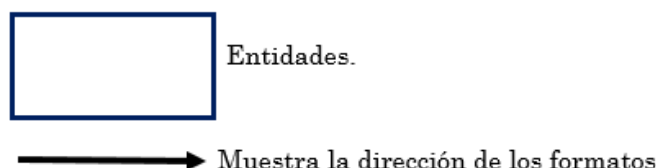


Figura 55.- Simbología para representar el diagrama de entidades a nivel usuario en Visio Microsoft

4.1.5 RESULTADOS CREACIÓN DEL DIAGRAMA DE ENTIDADES

Con la información mínima necesaria recabada en el archivo .csv, se puede proceder a crear manualmente el diagrama en el editor que se prefiera. En la **Figura 56** se puede observar el primer diagrama de entidades a nivel usuario de la entidad principal **Secretaría**, donde se ve que intercambia información (escrita sobre las líneas), la dirección de la información es representada con flechas (donde la cabeza de la flecha representa la dirección), y las entidades con las que intercambia información son las siguientes:

- **Jefe de Departamento**
- **Alumnos**
- **Director de tesis**
- **Comité**
- **Presidenta del Consejo de Postgrado**
- **Subdirección Académica**
- **Desarrollo Académico**
- **Servicios Escolares**

La **Figura 56** representa el diagrama de entidades a nivel usuario del dominio del problema administrativo, dónde la entidad principal es la **secretaria** de la jefatura del Depto. de Ciencias Computacionales.

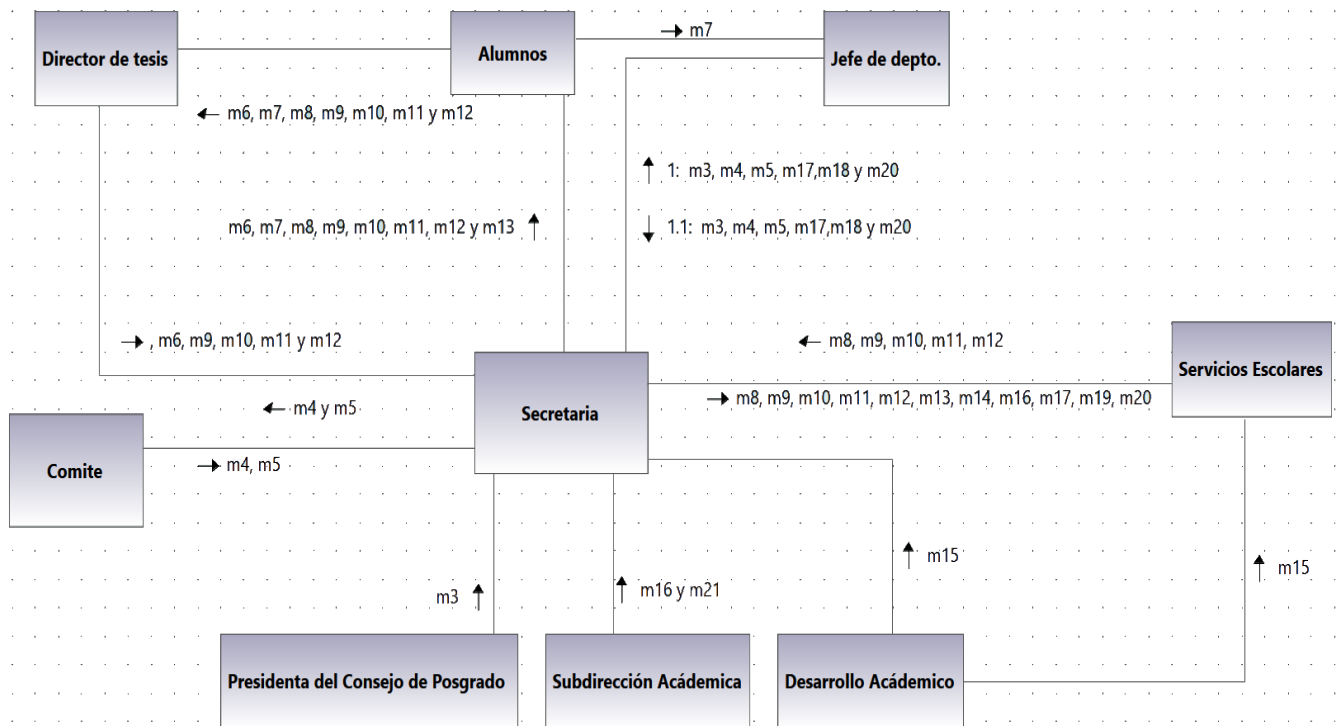


Figura 56.- Diagrama de entidades a nivel usuario - Secretaria (entidad principal)

La **Figura 57** muestra el diagrama de entidades a nivel usuario del alumno, el cual tiene intercambio de información con: **La secretaria, Jefe de Depto. Recursos Financieros, Director de tesis y Servicios Escolares**, la dirección de los formatos está señalada por la punta de la flecha.

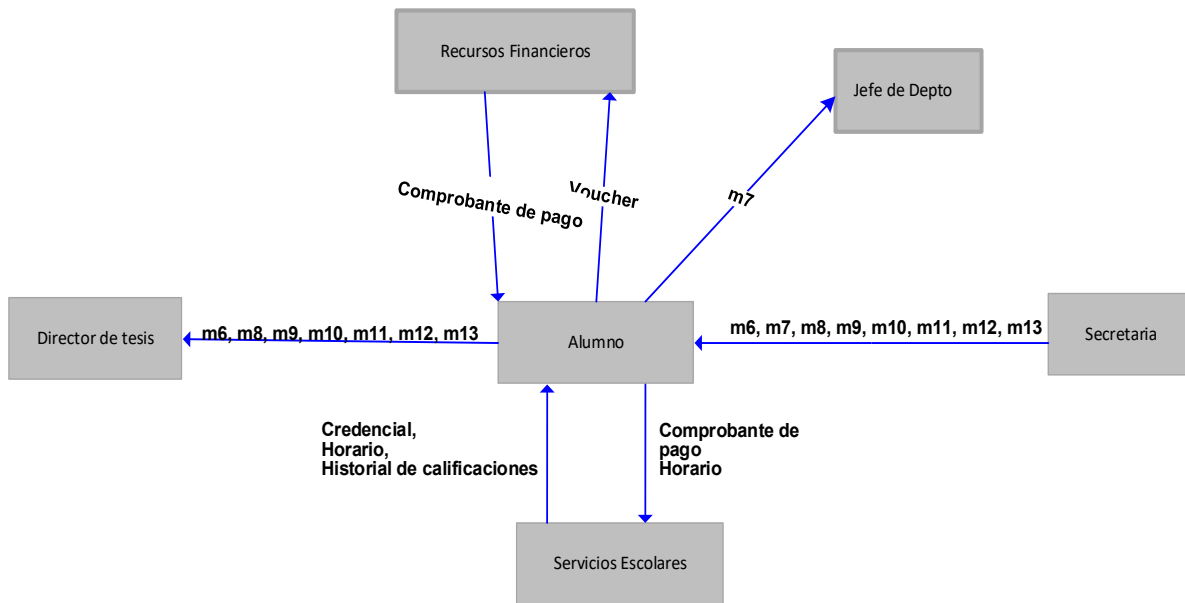


Figura 57.- Diagrama de entidades a nivel usuario Alumno (entidad secundaria)

La **Figura 58** representa el diagrama de entidades a nivel usuario de la entidad secundaria **Presidenta del Consejo de Postgrado**, quien tiene intercambio de información solo con el **Jefe de Departamento**, la dirección se va a definir por la punta de la flecha y los documentos intercambiados se encuentran arriba de la flecha.

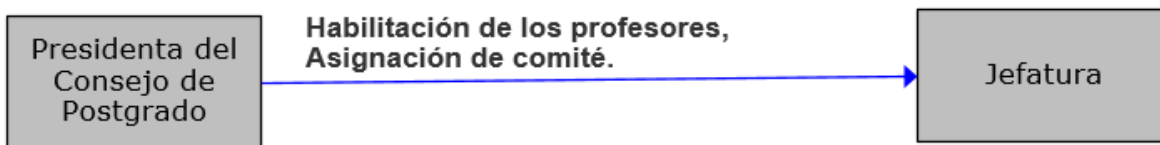


Figura 58.- Diagrama de entidades a nivel usuario Presidenta de Claustro (entidad secundaria)

4.2 RESUMEN DEL CAPÍTULO

En el capítulo presentado, se mostró y explicó todo el proceso que se llevó a cabo para realizar adecuadamente cada una de las etapas de la metodología y así poder llegar al objetivo esperado. Evidenciando las librerías, Framework, elementos y componentes necesarios para poder construir el chatbot Atenea y generar el archivo que pueda leer la computadora y contuviera la información mínima necesaria para crear el diagrama de entidades de Warnier Orr. Así mismo, también se evidenciaron los resultados finales de cada etapa de la metodología. El siguiente capítulo presenta el plan de pruebas del estudio de caso de la Jefatura del Departamento de Ciencias Computacionales del Cenidet.

Capítulo 5

Pruebas

5.1 PRUEBAS

En este capítulo se presenta un resumen del plan de pruebas que se realizó para validar el desarrollo del chatbot Atenea, descrito en el **capítulo 4 Metodología y Resultados**. El plan de pruebas fue diseñado con base al estándar IEEE (2008) 829-2008- IEEE Standard for Software and System Test Documentation.

Para llevar a cabo la fase de pruebas se desarrollaron seis documentos:

- 1) **Plan de Pruebas:** En este documento se define el alcance del plan de pruebas, los criterios de aceptación, los elementos y los casos de pruebas que se requieren para llevarlo a cabo.
- 2) **Diseño de Pruebas:** En este documento se detallan las características a probar, y la forma para realizar las pruebas conforme a los casos de pruebas mencionados en el plan de pruebas.
- 3) **Casos de Pruebas:** En este documento se va a especificar el avance, entradas y salidas de cada prueba mencionados en el documento **Plan de pruebas** y acorde a los casos.
- 4) **Procedimiento de Pruebas:** En este documento se describen los pasos del procedimiento que se va a seguir para efectuar el conjunto de pruebas descrito en el documento Diseño de pruebas. Este procedimiento se apega a la metodología.
- 5) **Registro de Pruebas:** En este documento se registran los detalles de la ejecución de las pruebas.
- 6) **Reporte de pruebas:** En este documento se detallan los resultados generales de las pruebas.

5.1.1. Plan de Pruebas

En este documento se define el alcance del plan de pruebas, los criterios de aceptación, los elementos y los casos de pruebas que se requieren para realizarlo, en la **Tabla 8** se pueden observar los casos establecidos.

Tabla 8.- Casos de prueba

Requerimiento	Caso de Prueba
Visibilidad del chatbot en la página web	TC-01
Reconocimiento de la voz	TC-02
Correcto seguimiento de la entrevista	TC-03
Creación del historial de conversaciones	TC-04
Etiquetado del historial	TC-05
Creación del archivo csv	TC-06
Elaboración del diagrama de entidades	TC-07

5.1.2 Diseño de pruebas

Aquí se van a detallar las características a probar, y la forma para realizar las pruebas conforme a los casos de pruebas descritos en el punto anterior. Las estrategias de pruebas propuestas se describen en la **Tabla 9**.

Tabla 9.- Pruebas a realizar

Requerimiento	Característica	Caso de Prueba	Procedimiento de Prueba
1. Visibilidad del chatbot en la página web	<ol style="list-style-type: none">1. Mostrar interfaz del chatbot.2. Sincronizar con las preguntas de DialogFlow.3. Mostrar entrevista.	TC-01	TPr-01
2. Reconocimiento de la voz	<ol style="list-style-type: none">1. Se reconocerá la voz del personal entrevistado.2. El chatbot tendrá que llevar la entrevista mediante la voz.	TC-02	TPr-02

3. Correcto seguimiento de la entrevista	1. El chatbot tendrá que mostrar una secuencia correcta entre las preguntas que hace al entrevistado.	TC-03	TPr-03
4. Creación del historial de conversaciones	1. Generar el historial de cada conversación. 2. Mostrar datos del historial.	TC-04	TPr-04
5. Etiquetado del historial	1. Mandar a llamar el historial en archivo plano. 2. Ejecutar la librería. 3. Generar el archivo Salidafreeling.txt.	TC-05	TPr-05
6. Creación del archivo csv	1. Mandar a llamar el archivo Salidafreeling.txt. 2. Ejecutar programa. 3. Generar el archivo Resultados.csv.	TC-06	TPr-06
7. Elaboración del diagrama de entidades	1. Revisar el archivo Resultados.csv 2. Generar manualmente el diagrama de entidades nivel uno, con ayuda del archivo Resultados.	TC-07	TPr-07

5.1.3 Casos de pruebas

En este documento se especifica la entrada y salida de cada de prueba que se establecen en el documento Plan de Pruebas y acorde a los casos de pruebas. En la **Tabla 10** se muestra el ejemplo del caso de pruebas TC-01.

Para cada caso de prueba se describe:

- **Identificador**
- **Título del caso de prueba**
- **Descripción de la prueba**
- **Objetivo de la prueba**
- **Resultado esperado**
- **Procedimiento de la prueba**
- **Especificaciones de entrada**
- **Especificaciones de salida**
- **Necesidades ambientales en caso que lo requiera**
- **Resultados obtenidos**
- **Resultado de la prueba**

Tabla 10.- Muestra del caso de prueba TC-01

Identificador	TC-01
Título de caso de prueba	“Visualización del chatbot en la página web”
Descripción de la prueba	
El usuario podrá visualizar dentro de la página web, una pestaña llamada “chatbot”, que mostrará la interfaz del chatbot para poder comenzar con la entrevista. O también se podrá visualizar la interfaz presionando un botón con la leyenda “Comenzar”.	
1.1 objetivo de la prueba:	
El objetivo de la prueba de TC-01 es asegurarse de que la interfaz del chatbot provee al usuario el acceso y la navegación apropiada a las preguntas para la correcta ejecución de la entrevista.	
Resultado Esperado	
a) verificar que se tiene acceso a la interfaz del chatbot.	
1.2 Procedimiento de pruebas	
1.2.1 Instrucciones	
a) solicitar la página web http://localhost/atenea/	
b) verificar que exista la interfaz del chatbot, listo para iniciar la entrevista	
1.3 Especificaciones de entrada	
Se necesita tener un servidor local instalado y tener conexión a internet.	
1.4 Especificaciones de salida	
Se mostrará la página web donde se encontrará la interfaz del chatbot listo para iniciar la entrevista.	

1.5 Necesidades ambientales

Ninguna

1.6 Resultado obtenido

Se visualizó correctamente la interfaz del chatbot Atenea, el cual estaba listo para empezar la entrevista.

1.7 Resultado de la prueba

La prueba fue exitosa SI NO

En el caso de que no se obtenga el resultado esperado, se tendrá que revisar que se esté ejecutando correctamente el servidor, que la pagina este en el directorio corrector del servidor, y que se esté llamando correctamente el Framework de DialogFlow. En cualquiera de estos casos, la prueba se registraría como no exitosa.

5.1.4 Procedimiento de Pruebas

En este documento se describen los pasos para llevar a cabo el conjunto de pruebas descritos en los documentos anteriores.

Los puntos en general que se especifican para cada proceso son:

- **Registro**
- **Configuración**
- **Inicio**
- **Proceso**
- **Medición**
- **Suspender**
- **Reiniciar**
- **Cierre**
- **Contingencia**

En la **Tabla 11** se puede observar el procedimiento para el caso de prueba TC-01.

Tabla 11.- Muestra del procedimiento de pruebas TC-01

Etapa	Descripción
Registro	El reporte de las pruebas se va a ejecutar de forma manual mediante el registro de los eventos ocurridos durante la ejecución de la prueba, esto incluye cualquier circunstancia que pase dentro de la misma.
Configuración	Se tendrá que iniciar el servidor WampServer y abrir un navegador, cuando esté abierto el navegador se tendrá que escribir la siguiente liga: http://localhost/atenea/
Inicio	Dentro de la página se encontrarán las pestañas de Chatbot, Autores y Contacto, nos dirigiremos a la pestaña de Chatbot o solo le damos clic en el botón Comenzar desde la vista principal.
Proceso	En el apartado de Chatbot, se visualizará un avatar y la interfaz del chatbot. Para conocer las instrucciones a seguir, se tendrá que activar primero el avatar, y en segundo, se podrá saludar al chatbot presionado el micrófono de la ventana.
Medición	Dentro de la interfaz el tiempo dependerá del usuario, de cuanto demore para responder cada pregunta.
Suspender	La plática se puede suspender si el usuario se tarda más de cinco minutos en responder.
Reiniciar	Para poder reiniciar la plática se tendrá que iniciar desde cero, y se volverá a tener que saludar al chatbot.
Cierre	La plática terminará cuando el usuario haya respondido todas las preguntas lanzadas por el chatbot y una vez que este se haya despedido
Contingencia	Si se llegará a presentar alguna anomalía durante la prueba, se suspenderá la prueba para poder corregir los errores y se reiniciará la prueba cuando estos se hayan corregido.

5.1.5 Registro de pruebas

En este documento se registran los detalles de la ejecución de cada prueba.

Los puntos que se presentan en cada caso son:

- **Registro de la prueba**
- **Descripción**
- **Las actividades y eventos de la prueba**
- **Las anomalías que se hayan presentado en las pruebas.**

En la **Tabla 12** se muestran las actividades y eventos del caso de prueba TC-01.

Tabla 12.- Muestra de las actividades y eventos del caso de prueba TC-01

Paso	Actor	Descripción	Verificado
1	Usuario	Se inició el servidor WampServer	Sí
2	Usuario	Se abrió el navegador de preferencia y se escribió la url http://localhost/atenea/ necesaria.	Sí
3	Página	Se muestra la interfaz y avatar	Sí
4	Usuario	Se accedió a la interfaz del chatbot Atenea.	Sí

5.1.6 Reporte de Pruebas

En la **Tabla 13** que se muestra en seguida se detallan los resultados generales de las pruebas.

Tabla 13.- Resultados generales de las pruebas

	Requerimiento	Caso de Prueba	Resultado de la Prueba
1.	Visibilidad del chatbot en la página web	TC-01	Exitoso
2.	Reconocimiento de la voz	TC-02	Exitoso
3.	Correcto seguimiento de la entrevista	TC-03	Exitoso
4.	Creación del historial de conversaciones	TC-04	Exitoso
5.	Etiquetado del historial	TC-05	Exitoso
6.	Creación del archivo csv	TC-06	Exitoso
7.	Elaboración del diagrama de entidades	TC-07	Exitoso

5.2 ANÁLISIS DE RESULTADOS

Dentro del Procesamiento del Lenguaje Natural (PLN), existen métricas que ayudan a evaluar los modelos una vez implementados, de acuerdo a su desempeño y a la validez de los resultados devueltos. Estas métricas ayudan a evaluar los errores y aciertos de los modelos mediante ecuaciones. Para esto, existe una gran variedad de métricas, las cuales se usarán de acuerdo al objetivo que se desea evaluar. Las métricas que se utilizaron en esta investigación son: la Precisión (*precision* en inglés), cuya ecuación ayuda a validar que cantidad de los datos que se obtienen del modelo son verdaderamente correctos, y la Cobertura (*recall* en inglés), evalúa la cantidad de datos que el modelo es capaz de identificar como verdadero.

Estas métricas se aplicaron al modelo hecho en Java (el cual realiza el etiquetado del historial de la entrevista realizada entre el chatbot y el interesado), con el objetivo de evaluar la identificación de las entidades nombradas y transacciones (mensajes) que intervienen en el proceso administrativo. Esto se hizo para cada entrevista realizada y para cada dato a evaluar, es decir, se aplicó dos veces la métrica de precisión y la de cobertura, una para la evaluación de las entidades y la otra para la evaluación de las transacciones.

Estos datos procesados por el modelo y a su vez analizados manualmente, se presentan en una tabla llamada Matriz de confusión (Heras, 2020).

La matriz de confusión trabaja con cuatro valores y se explican a continuación:

- Verdaderos positivos (**VP**)
- Verdaderos Negativos (**VN**)
- Falsos Positivos (**FP**)
- Falsos Negativos (**FN**)

Se considera que los datos son Verdaderos Positivos (VP) cuando el modelo creado detecta los mismos datos positivos, que el experto identificó manualmente como datos correctos o positivos. Lo mismo sucede con los Verdaderos Negativos (VN), se considera así sí el modelo detecta los mismos datos negativos que el experto identificó manualmente. De lo contrario, se considera que los datos son Falsos Positivos (FP) cuando el modelo evalúa o

identifica que el dato es positivo, cuando en realidad no lo es. Del mismo modo, ocurre con los Falsos Negativos (FN) se clasificará así cuando el modelo cree o identifica un dato como negativo y en realidad es positivo.

Después de haber explicado los datos que se utilizan en la matriz de confusión, y como se van aplicar en la investigación, en seguida se explican las ecuaciones de las métricas a utilizar.

La Precisión se evalúa con la siguiente ecuación:

$$\text{Precisión} = \frac{\text{Verdaderos positivos (VP)}}{\text{Verdaderos positivos (VP)} + \text{Falsos positivos (FP)}}$$

La ecuación que mide la Cobertura es la siguiente:

$$\text{Cobertura} = \frac{\text{Verdaderos positivos (VP)}}{\text{Verdaderos positivos (VP)} + \text{Falsos negativos (FN)}}$$

Evaluación de las entidades de la entrevista uno (Secretaria)

De acuerdo con la primera entrevista realizada a la secretaria, la cual se procesó con el modelo de etiquetado, se obtuvo un corpus de datos de 1,312 tokens, de los cuales 9 tokens son etiquetados como entidades. Los 1,312 tokens se utilizaron para evaluar el modelo y los resultados se compararon con los resultados obtenidos en el conteo manual. Obteniendo como resultado los siguientes datos:

- Tokens identificados como entidades: 8
- Tokens identificados como entidades y sí lo son: 7
- Tokens identificados como entidades que no lo son: 2
- Tokens no identificados como entidades pero que sí lo son: 1
- Tokens no identificados como entidades y que no lo son: 1,302

Los datos anteriores se presentan en la matriz de confusión (ver **Tabla 14**), para calcular la precisión:

Tabla 14.- Matriz de confusión de Precisión para entidades de la entrevista uno

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	7	2
	Negativos	1	1,302

Los datos señalados con rectángulo punteado rojo, se reemplazaron en la ecuación de precisión, dando como resultado un 0.87 multiplicamos el valor por 100, nos da un valor del **87% de precisión.**

Lo mismo se hace con la medida de cobertura, y en la **Tabla 15** se muestran los datos anteriores:

Tabla 15.- Matriz de confusión de Cobertura para entidades de la entrevista uno

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	7	2
	Negativos	1	1,302

Los datos señalados con rectángulo punteado azul, se reemplazaron en la ecuación de cobertura, dando como resultado un 0.77 multiplicamos el valor por 100, nos da un valor del **77% de cobertura.**

De acuerdo a los resultados obtenidos, podemos concluir que el modelo para la identificación de las entidades en la entrevista uno mostró un **87%** de precisión y un **77%** de cobertura, estos porcentajes nos dicen que, para esta entrevista, todos los datos que el modelo identifico como entidades son realmente correctas, aunque no logró identificar todas las entidades.

Evaluación de las transacciones de la entrevista uno (Secretaria)

A continuación, se muestran los resultados obtenidos de la segunda matriz de confusión aplicada a la primera entrevista, para evaluar la identificación de las transacciones (mensajes). En esta matriz de confusión se utilizó el mismo corpus de 1,312 tokens, de los cuales 19 tokens son etiquetados como transacciones. Los 1,312 tokens se utilizaron para

evaluar el modelo y los resultados se compararon con los resultados obtenidos en el conteo manual. Obteniendo como resultado los siguientes datos:

- Tokens identificados como transacciones: 18
- Tokens identificados como transacciones y sí lo son: 16
- Tokens identificados como transacciones que no lo son: 5
- Tokens no identificados como transacciones pero que sí lo son: 2
- Tokens no identificados como transacciones y que no lo son: 1,289

Los datos anteriores se presentan en la matriz de confusión como se puede ver en la **Tabla 16**, para calcular la precisión:

Tabla 16.- Matriz de confusión de Precisión para transacciones de la entrevista uno

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	16	5
	Negativos	2	1,289

Los datos señalados con rectángulo punteado rojo, se reemplazaron en la ecuación de precisión, dando como resultado un 0.88 multiplicamos el valor por 100, nos da un valor del **88% de precisión.**

Lo mismo se hace con la medida de cobertura, como se muestra en la **Tabla 17**:

Tabla 17.- Matriz de confusión de Cobertura para transacciones de la entrevista uno

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	16	5
	Negativos	2	1,289

Los datos señalados con rectángulo punteado azul, se reemplazaron en la ecuación de cobertura, dando como resultado un 0.76 multiplicamos el valor por 100, nos da un valor del **76% de cobertura.**

Al igual que en los resultados de las entidades, las evaluaciones de las transacciones arrojaron un resultado de **88%** de precisión y **76%** de cobertura, lo cual nos demuestra que todos los datos identificados como transacciones por el modelo son correctos y coinciden con los datos que se identificaron como correctos manualmente.

Evaluación de las entidades de la entrevista dos (Alumno)

Al igual que la entrevista uno, la segunda entrevista realizada al alumno se le aplicaron las medidas de evaluación, tanto para las entidades como para las transacciones. A continuación, se muestran los resultados obtenidos de la matriz de confusión para evaluar la identificación de las entidades. Para esta matriz de confusión se utilizó un corpus de 848 tokens, de los cuales 6 tokens son etiquetados como entidades. Los 848 tokens se utilizaron para evaluar el modelo y los resultados se compararon con los resultados obtenidos en el conteo manual. Obteniendo como resultado los siguientes datos:

- Tokens identificados como entidades: 5
- Tokens identificados como entidades y sí lo son: 4
- Tokens identificados como entidades que no lo son: 2
- Tokens no identificados como entidades pero que sí lo son: 1
- Tokens no identificados como entidades y que no lo son: 841

Los datos anteriores se presentan en la matriz de confusión como se puede ver en la **Tabla 18**, para calcular la precisión:

Tabla 18.- Matriz de confusión de Precisión para entidades de la entrevista dos

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	4	2
	Negativos	1	841

Los datos señalados con rectángulo punteado rojo, se reemplazaron en la ecuación de precisión, dando como resultado un 0.8 multiplicamos el valor por 100, nos da un valor del **80% de precisión**.

Lo mismo se hace con la medida de cobertura, como se puede observar en la **Tabla 19**:

Tabla 19.- Matriz de confusión de Cobertura para entidades de la entrevista dos

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	4	2
	Negativos	1	871

Los datos señalados con rectángulo punteado azul, se reemplazaron en la ecuación de cobertura, dando como resultado un 0.66 multiplicamos el valor por 100, nos da un valor del **66% de cobertura.**

Para la segunda entrevista se obtienen resultados similares a la primera entrevista, se obtiene una precisión del **80%** y una cobertura del **66%**, concluyendo que las entidades identificadas son correctas y concuerdan con los datos identificados manualmente.

Evaluación de las transacciones de la entrevista dos (Alumno)

Para esta matriz de confusión se utilizó el mismo corpus de las entidades, el cual tiene un total de 848 tokens, de los cuales 13 tokens son etiquetados como transacciones. Los 848 tokens se utilizaron para evaluar el modelo y los resultados se compararon con los resultados obtenidos en el conteo manual. Obteniendo como resultado los siguientes datos:

- Tokens identificados como transacciones: 12
- Tokens identificados como transacciones y sí lo son: 11
- Tokens identificados como transacciones que no lo son: 5
- Tokens no identificados como transacciones pero que sí lo son: 1
- Tokens no identificados como transacciones y que no lo son: 831

Los datos anteriores se presentan en la matriz de confusión como se puede ver en la **Tabla 20**, para calcular la precisión:

Tabla 20.- Matriz de confusión de Precisión para transacciones de la entrevista dos

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	11	5

	Negativos	1	831
--	-----------	---	-----

Los datos señalados con rectángulo punteado rojo, se reemplazaron en la ecuación de precisión, dando como resultado un 0.91 multiplicamos el valor por 100, nos da un valor del **91% de precisión.**

En la **Tabla 21** se muestra la tabla de confusión para medir la cobertura:

Tabla 21.- Matriz de confusión de Cobertura para transacciones de la entrevista dos

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	11	5
	Negativos	1	831

Los datos señalados con rectángulo punteado azul, se reemplazaron en la ecuación de cobertura, dando como resultado un 0.68 multiplicamos el valor por 100, nos da un valor del **68% de cobertura.**

En este caso los resultados nos muestran una precisión del **91%** y una cobertura del **68%**, con los cuales estamos disminuyendo los datos que puedan entrar en la clasificación de los falsos positivos. En este caso se aumentó la precisión debido a que el corpus contaba con menos tokens a evaluar, ya que se obtuvo el mismo error que la entrevista uno.

Evaluación de las entidades de la entrevista tres (Presidenta del Consejo de Postgrado)

El corpus para evaluar esta entrevista tiene un total de 511 tokens, de los cuales 2 tokens son etiquetados como entidades. Aquí también se utilizó el corpus de 511 tokens para evaluar el modelo y los resultados se compararon con los resultados obtenidos en el conteo manual. Obteniendo como resultado los siguientes datos:

- Tokens identificados como entidades: 2
- Tokens identificados como entidades y sí lo son: 1
- Tokens identificados como entidades que no lo son: 2
- Tokens no identificados como entidades pero que sí lo son: 1
- Tokens no identificados como entidades y que no lo son: 507

Los datos anteriores se presentan en la matriz de confusión como se puede ver en la **Tabla 22**, para calcular la precisión:

Tabla 22.- Matriz de confusión de Precisión para entidades de la entrevista tres

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	1	2
	Negativos	1	507

Los datos señalados con rectángulo punteado rojo, se reemplazaron en la ecuación de precisión, dando como resultado un 0.5 multiplicamos el valor por 100, nos da un valor del **50% de precisión.**

En la **Tabla 23** se muestra la tabla de confusión para medir la cobertura:

Tabla 23.- Matriz de confusión de Cobertura para entidades de la entrevista tres

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	1	2
	Negativos	1	507

Los datos señalados con rectángulo punteado azul, se reemplazaron en la ecuación de cobertura, dando como resultado un 0.33 multiplicamos el valor por 100, nos da un valor del **33% de cobertura.**

En este caso de la entrevista tres, los resultados muestran una precisión del **50%** y una cobertura del **33%**, debido a que se tiene un corpus pequeño y la cantidad de entidades a identificar es mínima. En consecuencia, de las dos entidades a identificar el modelo solo pudo identificar una, dando la mitad de la precisión.

Evaluación de las transacciones de la entrevista tres (Presidenta del Consejo de Postgrado)

Finalmente, se realizó la tercera evaluación a la última entrevista realizada por el chatbot para medir las transacciones. Se utilizó el mismo corpus de las entidades, como se mencionó el

corpus tiene un total de 511 tokens, de los cuales se tiene que reconocer 2 tokens de transacción. Obteniendo como resultado los siguientes datos:

- Tokens identificados como entidades: 2
- Tokens identificados como entidades y sí lo son: 1
- Tokens identificados como entidades que no lo son: 3
- Tokens no identificados como entidades pero que sí lo son: 1
- Tokens no identificados como entidades y que no lo son: 506

Los datos anteriores se presentan en la matriz de confusión como se puede ver en la **Tabla 24**, para calcular la precisión:

Tabla 24.- Matriz de confusión de Precisión para transacciones de la entrevista tres

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	1	3
	Negativos	1	506

Los datos señalados con rectángulo punteado rojo, se reemplazaron en la ecuación de precisión, dando como resultado un 0.50 multiplicamos el valor por 100, nos da un valor del **50% de precisión.**

En la **Tabla 25** se muestra la tabla de confusión para medir la cobertura:

Tabla 25.-Matriz de confusión de Cobertura para transacciones de la entrevista tres

Datos Manuales		Datos de estudio (Modelo)	
		Positivos	Negativos
Datos Objetivo	Positivos	1	3
	Negativos	1	506

Los datos señalados con rectángulo punteado azul, se reemplazaron en la ecuación de cobertura, dando como resultado un 0.25 multiplicamos el valor por 100, nos da un valor del **25% de cobertura.**

Del mismo modo, pasa con la identificación de las transacciones, el modelo solo pudo identificar una transacción de las dos identificadas manualmente, esto no da una precisión del **50%** y una cobertura del **33%**.

Para concluir con esta sección, en la **Tabla 26** se muestran las dos medidas de evaluación obtenidas de cada una de las entrevistas, donde se evalúan las entidades involucradas en el proceso administrativo. Y en la **Tabla 27**, se muestran los resultados obtenidos en la evaluación de las transacciones que intervienen en el proceso administrativo. Cabe mencionar, que no existe ningún modelo que muestre como resultado un 100% en las medidas de evaluación, debido a que no hay un modelo perfecto.

Tabla 26.- Resultados de las medidas de evaluación para las entidades

	Entrevista 1	Entrevista 2	Entrevista 3
Precisión	87%	80%	50%
Cobertura	77%	66%	33%

Tabla 27.- Resultados de las medidas de evaluación para las transacciones

	Entrevista 1	Entrevista 2	Entrevista 3
Precisión	88%	91%	50%
Cobertura	76%	68%	25%

Concluyendo con la evaluación del modelo, los resultados obtenidos con estas métricas nos señalan que el modelo realizado para el etiquetado de historial de conversaciones, tiene la capacidad de detectar la mayor cantidad de las entidades y transacciones correctamente, donde la mayoría de ellas son verdaderas y coinciden con los datos manuales. Aunque puede haber ciertos errores debido a que se tiene una baja cobertura, ya que pudiera detectar datos como entidades o transacciones y no lo son. Por consiguiente, los datos no nos proporcionan alguna información en la creación del diagrama de entidades a nivel usuario.

5.3 DISCUSIÓN DE RESULTADOS

Como se puede observar en la **Tabla 13** todo los casos de prueba presentaron un resultado exitoso, lo que conlleva al cumplimiento de los requerimientos funcionales antes mencionados en su totalidad. Por otro lado, el reconocimiento de las entidades, la

identificación de las transacciones y el flujo de las transacciones, ayudaron a crear de forma correcta el diagrama de entidades a nivel usuario de Warnier Orr, de cada una de las entidades entrevistadas en el dominio administrativo. Esto ayudó a conocer todo el proceso de la maestría dentro del CENIDET en el Departamento de Ciencias Computacionales, las entidades involucradas en el proceso y el flujo que llevan los documentos (mensajes/transacciones).

La utilización de las dos interfaces para la creación del chatbot nos permitió generar un ambiente amigable para el usuario, aunque la conclusión a la que se puede llegar es que la herramienta que más se asemeja a una entrevista de persona a persona es la interfaz de Google Assistant, la cual activa automáticamente el periodo de escucha del micrófono por un periodo de tiempo, lo que ayuda al interesado (usuario) a responder como lo haría con una persona física sin la necesidad de presionar algo extra, como es el caso de la interfaz de la página web. Aunque también tiene su desventaja ya que el periodo de escucha es un poco corto.

Las entidades que se reconocieron con el programa en Java y Ruby, también fueron identificadas manualmente, esto para tener un mejor resultado y corroborar que todas las entidades, transacciones y el flujo de las mismas fueran correctas y corroborar si se habían obtenido todos los datos mínimos necesarios.

5.4 RESUMEN DEL CAPÍTULO

En este capítulo se presentó un ejemplo del plan de pruebas que se llevó a cabo para cada parte de la metodología, donde se puede concluir que todos los casos fueron exitosos, como se puede observar en el capítulo y también se presenta la evaluación del modelo realizado para la etiquetación, se evalúa este modelo debido a que se necesita identificar correctamente las transacciones para garantizar el modelado del dominio del problema.

Capítulo 6

Conclusiones

6.1 CONCLUSIONES

En este capítulo se exponen las conclusiones obtenidas como resultado de la presente investigación, también se incluyen las aportaciones y los trabajos futuros.

De acuerdo con los resultados presentados en el capítulo 4 se concluye que se cumplió el objetivo general , el cuál es mencionado en la sección **1.2.1 OBJETIVO GENERAL** en el capítulo 1.

“Realizar por medio de la voz entrevistas estructuradas para identificar entidades y flujo de transacciones en un dominio administrativo.”

Se concluye que los objetivos específicos se cumplieron satisfactoriamente de acuerdo a lo siguiente:

- Se obtuvo la plataforma de Google llamada DialogFlow el cuál procesa la voz y el texto para la creación del chatbot, el cual llevó acabo la entrevista con las entidades involucradas en el dominio administrativo mediante la voz satisfactoriamente.
- Se Identificaron correctamente las entidades principales, entidades secundarias y el flujo de las transacciones del dominio administrativo por medio de técnicas de Procesamiento de Lenguaje Natural.
- Y finalmente, se realizó el modelado del dominio administrativo mediante el diagrama de Warnier Orr.

Con los resultados obtenidos se puede aseverar lo siguiente:

- Las pruebas reportaron un **resultado exitoso**, lo que conlleva al **cumplimiento de los requerimientos** funcionales en su totalidad, y al **objetivo establecido** en el trabajo de investigación como ya se mencionó.
- Se postula como hipótesis, que el chatbot facilita la interacción con los usuarios, e involucra al interesado en el proceso del dominio del problema. Aunque se requiere

experimentación para comprobarla. Aunque en nuestra experiencia se logró una comunicación adecuada y un entendimiento por ambas partes (analista y cliente). Aparentemente, debido a que el analista amplió el vocabulario del chatbot y conoce mejor el vocabulario del dominio. Y se espera que para dominios similares se disminuya esta problemática.

- El chatbot puede emplearse en otros dominios siempre y cuando se actualice el vocabulario del mismo con la terminología pertinente. En consecuencia, esto es más fácil que hacer que una persona se aprenda todo el vocabulario de un dominio que no conoce ni domina.
- El diagrama utilizado para representar la estructura del diálogo, se encuentra en un nivel de abstracción alto, el cuál ayuda a empezar a detectar los actores involucrados dentro del proceso. Y a comprender el espacio del problema en el que se encuentra ubicado el dominio, ayudando a representar la problemática que se desea resolver

Adicionalmente se lograron los siguientes resultados durante esta investigación:

- El trabajo de investigación realizado derivó en la publicación del artículo ***Mapeo Sistemático de herramientas web para la creación de un chatbot*** publicado en la 3era Jornada de la Ciencia y la Tecnología del Centro Nacional de Investigación y Desarrollo Tecnológico realizada por el CENIDET. Se incluye la portada del artículo en el Anexo A.
- Se participó con una ponencia llamada ***Entrevista por medio de un chatbot con reconocimiento de voz enfocada a un modelo de negocio para la identificación de entidades***, realizada por el Congreso Internacional de Lingüística y de Corpus CILCC 2020 en Medellín Colombia. Se incluye el resumen y constancia en el Anexo B, así mismo se tiene la presentación disponible utilizada en el congreso.

6.2 APORTACIONES

El proporcionar una herramienta que ayude al analista a llevar a cabo la entrevista es de utilidad ya que como se mencionó entrenar a un agente conversacional (chatbot) es más fácil

que tratar que un analista se aprenda el vocabulario y jerga del dominio del problema, además de ayudar a resolver la problemática mencionadas en el capítulo 1.

Las aportaciones de este trabajo de investigación son:

- Un proceso para lograr un primer acercamiento en la identificación de las entidades y transacciones que están involucradas en un dominio de problema administrativo, en dos plataformas y/o entornos diferentes para mostrarlas al usuario.
- Una clase en java con herramientas de lenguaje natural donde se incluye un analizador morfosintáctico que ayuda al etiquetado del texto, extrayendo de cada palabra su lema y categoría gramatical correspondiente. Se incluye como anexo D.
- Un modelo preestablecido de alto nivel de abstracción donde se detectan los actores y los tipos de transacción involucrados en un proceso administrativo. En este caso el proceso de maestría en el departamento de Ciencias Computacionales.

6.3 TRABAJOS FUTUROS

Para los trabajos futuros se propone lo siguiente:

- Ampliar el módulo de las preguntas para obtener un diagrama menos abstracto, conforme se vaya trabajando los módulos se recomienda incrementar los vocabularios tanto del chatbot como de los diccionarios del PLN.
- Explorar otras plataformas de creación de chatbot para verificar el comportamiento de las preguntas y obtener más información importante para generar modelos menos abstractos.
- Experimentar con otros entornos que incluyan el reconocimiento de voz para presentar el chatbot al usuario.
- Experimentar con las distintas herramientas de Procesamiento del Lenguaje Natural para el tratamiento del texto.

REFERENCIAS BIBLIOGRÁFICAS

- Aamodt, A. (2013). 'A Knowledge-Intensive Method for Conversational CBR'. Norwegian University of Science and Technology.
- Alemany, L. A. (20 de noviembre de 2005). 'Herramientas Libres para Procesamiento del Lenguaje Natural'. Córdoba, Argentina.
- Altszyler., E. y. (2015). 'Análisis de la dinámica del contenido semántico de textos'. Simposio Argentino de Inteligencia Artificial., pp. 256-263.
- Breslow, D. W. (2014). 'Conversational Case-Based Reasoning'. Kluwer Academic Publishers. Manufactured in The Netherlands, pp. 1-24.
- Cabrera, S. H. (2018). 'Procesamiento semántico automático, enfocado en la coherencia textual'. Revistas Electrónicas UACH, pp. 58 - 122.
- Catalunya, U. P. (20 de Julio de 2020). Freeling. Obtenido de Freeling: <http://nlp.lsi.upc.edu/freeling/demo/demo.php>
- Cervantinos, C. E. (17 de 08 de 2020). Centro de Estudios Cervantinos. Obtenido de Centro de Estudios Cervantinos: <https://www.centroestudioscervantinos.es/analisis-morfologico/>
- Cortez, V. &. (2009). 'Procesamiento de lenguaje natural'. Ingeniería de Sistemas e Informática, pp. 46.
- Czarnecki, K. &. (2000). 'Generative Programming—Methods, Tools, and Applications. Addison-Wesley'.
- Diaz, M. J. (2019). 'Análisis social aplicando técnicas de lenguaje'. Scientia et Technica Año XXIV, pp. 496-503.
- Ferrucci, D. B.-C. (2010). 'Building Watson: An overview of the DeepQA project'. AI magazine, pp. 31.
- Fuentes, M. d. (2013). 'Análisis de Requerimientos' . Cdmx.
- Gaikwad, S. K. (2010). 'Review on Speech Recognition Technique. International Journal of Computer Applications'.
- Gelbukh, A. &. (2010). 'Procesamiento automático del español con enfoque'. 2da edición.

- Gelbukh., A. (2010). 'Procesamiento de Lenguaje Natural y sus aplicaciones'. Komputer Sapiens, pp. 6-32.
- Google, c. (2020). Documentación. Recuperado el 03 de marzo de 2020, de <https://cloud.google.com/dialogflow/docs/basics?hl=es-419>
- González, G.M. Agosto (2006). 'Método de Desarrollo Arquitectónico en Grupo'. DF, México: Instituto Politécnico Nacional.
- Heras, J., 2020. IArtificial.net. [En línea] Available at: [https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/#Precision_\(Precision\)](https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/#Precision_(Precision))
- Hernández, G. S. (2019). 'Mapeo sistemático del reconocimiento del habla ,proceso del lenguaje natural y uso de ontologías para identificar el dominio del problema y los requerimientos de solución'. Cuernavaca, Morelos.
- Hull, E. D. (2017). 'Requirements Engineering in the Solution Domain. Requirements Engineering'. https://doi.org/10.1007/978-3-319-61073-3_6, 135–158. Obtenido de https://doi.org/10.1007/978-3-319-61073-3_6
- Iulian V. Serban, C. S. (2017). 'A Deep Reinforcement Learning Chatbot'. pp 1-40.
- Jurafsky, D. &. (2008). 'Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition'. Prentice Hall.
- Koren, Y. B. (2009). 'Matrix factorization techniques for recommender systems'. Computer , pp. 42.
- Limited, F. (2020, Diciembre 09). 'FindGlocal Limited'. Retrieved from FindGlocal Limited: <http://www.findglocal.com/PE/Arequipa/428709821205648/Joyce-Salinas-Psicoterapeuta>
- Litvak, C. S. (2016). 'Procesamiento de Lenguaje Natural para Estudiar Completitud de requisitos'. San Justo. Buenos Aires: Departamento de Ingeniería e Investigaciones Tecnológicas, UNLaM.
- Madeja. (14 de 08 de 2020). 'Marco de Desarrollo de Software de la Junta de Andalucía'. Tomado de 'Marco de Desarrollo de Software de la Junta de Andalucía': <http://www.juntadeandalucia.es/servicios/madeja/sites/all/themes/madeja/libros/966.pdf>

- Millán, O. A., González, G. M., González, F. N, & Castro, S. N. A. (2 de Julio-Diciembre de 2019). 'Mapeo Sistemático de herramientas web para la creación'. Jornada de Ciencia y Tecnología Aplicada, 2, pp. 38-42.
- Nazir, F. B. (2017). 'The Applications of Natural Language Processing (NLP) for Software Requirement Engineering - A Systematic Literature Review.'. Information Science and Applications, pp. 424.
- Orr, K. T. (1981). 'Structured Requirements Definition'. En K. T. Orr, Structured Requirements Definition (pp. 61-135). United States of America.
- Padró, L. (2011). 'Analizadores Multilingües en FreeLing'. Linguamática, pp. 13-20.
- Raffino, M. (15 de 08 de 2020). 'Lenguaje'. Obtenido de 'Lenguaje': <https://concepto.de/lenguaje/>.
- Reinhartz, B. I. (2013). "Domain Engineering- Product Lines". Iris Reinhartz-Berger; Arnon Sturm; Tony Clark; Sholom Cohen; Jorn Bettin,.
- Sadoun, D. D.-D. (2014). 'Formal rule representation and verification from natural language requirements using an ontology'. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 226-235.
- Sameera, A. A. (2015). 'Survey on Chatbot Design Techniques in Speech Conversation Systems'. International Journal of Advanced Computer Science and Applications.
- Saussure, F. (1945). 'Curso de linguística General'. Buenos Aires: Losada.
- Soares, F. A. (2015). 'VoiceToModel: An Approach to Generate Requirements Models from Speech Recognition Mechanisms'.
- Sree-Kumar, A. P. (2018). 'Extracting Software Product Line Feature Models from Natural Language Specifications'. Actas de la 22a Conferencia Internacional.
- Sturm, A. D. (2006). 'Domain Modeling with Object-Process Methodology'. pp. 1-17.
- Taylor, M. &. (2008). 'Software Architecture: Foundations, Theory, and Practice. John Wiley & Sons'.
- Wu, Y. S. (2016). 'Google's neural machine translation system: Bridging the gap between human and machine translation'. arXiv preprint arXiv:1609.08144.
- Zaragoza K. (2019). Editor para el Lenguaje de Dominio Especifico Diagramas de Entidad. Cuernavaca.

Anexos

ANEXO A

En este anexo se presenta el artículo presentado en la 3era Jornada de la Ciencia y la Tecnología Aplicada del Cenidet, llamado Mapeo Sistemático de herramientas web para la creación de un chatbot.

Mapeo Sistemático de herramientas web para la creación de un chatbot

Millán-Olivares Laura*,
González-García Moisés **,
Nimrod González-Franco ***,
Noé Alejandro Castro-Sánchez****

CENIDET, Cuernavaca, Morelos;

Interior Internado Palmira S/N, Col. Palmira, C.P. 62490

email: [@ana.millan18ca](mailto:ana.millan18ca), [@moises](mailto:moises), [@nimrod](mailto:nimrod), [@ncastro](mailto:ncastro) @cenidet.edu.mx

Resumen: En el mundo, las empresas invierten en mecanismos que les ayuden a mejorar servicios basados en TICs para incrementar su productividad. Uno de estos mecanismos son los chatbots, que son agentes conversacionales con el objetivo de mantener una conversación de manera natural. En este artículo se reporta la búsqueda y hallazgos acerca de herramientas que faciliten la creación de estos agentes conversacionales con reconocimiento de voz, la revisión de la literatura estuvo centrada en artículos del 2013 al 2018, escritos en el idioma inglés y produjo la identificación de nueve herramientas más populares dentro de la literatura, pero solo tres de ellas son las más utilizadas por la facilidad, tecnología y documentación disponible dentro de la red.

Palabras Claves: Chatbot, Mapeo sistemático, Reconocimiento de Voz, Elicitación de Requerimientos, modelo de dominio

1. INTRODUCCIÓN

El avance de Internet y de las nuevas Tecnologías de la Información y la Comunicación, ha sido el origen de importantes cambios que impactaron no sólo en la vida cotidiana de los individuos, sino también en organizaciones e instituciones sociales (Litwin, 2005). En consecuencia, estos cambios desembocaron en un paradigma que produjo modificaciones significativas en la forma de relacionarnos, de trabajar e incluso de enseñar y aprender.

En el mundo, el sector salud y computacional ofrecen sus servicios a una gran cantidad de clientes diariamente y a medida que un servicio es requerido por más clientes, las empresas invierten en mecanismos de inteligencia artificial (IA) que les ayuden a mejorar estos servicios. El potencial que tiene la IA, debido a su exactitud y rapidez al procesar una tarea. Por esta razón, los robots que llevan conversaciones (chatbots), soportados por técnicas de IA, pueden facilitar que las empresas mejoren la eficiencia (dado que realizan su tarea con la menor cantidad de recursos) y eficacia (logran el objetivo esperado) de su proceso de atención al cliente (Quintana, 2018).

Debido al incremento de las necesidades de los clientes, la industria de la computación ha incentivado la creación y exploración de diferentes entornos y plataformas web que pueden involucrar la IA, en particular a los chatbots.

Por ello, se requiere la investigación de plataformas web que ayuden a la creación de chatbots. Según (Sameera, 2015), un chatbot es un programa informático que tiene la capacidad de mantener una conversación con humanos usando el lenguaje natural.

En este reporte se exponen los resultados obtenidos de la elaboración de un estudio de mapeo sistemático sobre las plataformas web más populares para la creación de un chatbot. Con el fin de conocer qué herramientas se utilizan más y clasificar la información más importante de cada una de ellas. Y de esta manera, seleccionar la herramienta que posibilite su aplicación en investigaciones de la ingeniería de software, específicamente en la etapa de análisis dentro del ciclo de desarrollo de software.

El artículo se divide en las siguientes secciones: Metodología de Investigación (sección 2), donde

se describe los pasos que se llevaron a cabo durante el estudio; Resultados (sección 3), se presentan los resultados obtenidos en la aplicación del estudio, representados en una tabla; a continuación, se presentan las Conclusiones (sección 3), y finalmente la Bibliografía (sección 4).

2. METODOLOGÍA DE INVESTIGACIÓN

De acuerdo con (Petersen, 2008) un Estudio de Mapeo Sistemático (EMS), es un estudio secundario que tiene como objetivo construir un esquema de clasificación y estructurar un campo de interés de la ingeniería de software. La metodología para realizar un EMS, es la más adecuada para llevar a cabo el estudio requerido ya que se cuenta con tópicos de ingeniería de software con un espectro bastante amplio. Estos tópicos requieren estudiarse de forma combinada con el fin de evidenciar, su uso y los posibles trabajos futuros que puedan identificarse a partir de la información resultante.

Para llevar a cabo el estudio de mapeo sistemático, es necesario definir un objetivo, que se convertirá en las preguntas de investigación, las cuales guiarán la siguiente parte del estudio. Se tomó en cuenta la metodología descrita en (Petersen, 2008), la cual consiste en las actividades siguientes: (1) Definir preguntas de investigación, (2) Selección de fuentes, (3) Selección de artículos, (4) Inclusión y Exclusión de los artículos.

Es importante mencionar que se tiene que seguir el protocolo para evitar cualquier atraso que pueda presentarse durante la aplicación del estudio. En este caso, se fijaron objetivos en términos de alcance y tiempo. Donde el alcance se establece mediante las preguntas de investigación. A continuación, se describe cada proceso del mapeo sistemático.

A) Definir Preguntas de Investigación

En la ingeniería de software la etapa de Análisis es una de las más importantes dentro de un proyecto para obtener un resultado exitoso, puesto que en ella se elabora el modelo del problema (negocio) y a partir de este se especifican los requerimientos del software (solución).

De este modo, el modelo se obtiene mediante entrevistas entre los profesionales de la computación y los clientes (expertos en su área). De acuerdo a lo mencionado, se pretende automatizar la parte de la entrevista entre el experto en computación y el cliente mediante un

chatbot con reconocimiento de voz, para mantener la comunicación en un lenguaje natural entre ambas personas, sin importar los tecnicismos de cada individuo.

Según lo anterior, la pregunta principal que se plantea es:

¿Existe alguna plataforma o marco (Framework) que ayude a automatizar el dominio de “creación de agentes inteligentes conversacionales mediante el habla” a través de alguna plataforma que integre el reconocimiento de voz?

Esta pregunta busca obtener la mayor evidencia posible sobre la implementación o aplicación de cualquier plataforma o *Framework* relacionada con la creación de un chatbot para conocer sus características y así elegir el que mejor se adapte a nuestras necesidades. La idea principal es encontrar todas las plataformas posibles, conocer sus tendencias, ventajas y desventajas, compararlas entre ellas, en el contexto del dominio del problema y en el dominio de requerimientos.

A su vez, esta pregunta principal, para realizar la búsqueda, se dividió en tres preguntas secundarias. Estas preguntas se representan en la Tabla 17.

Tabla 28. Preguntas de Investigación

ID	PREGUNTA DE INVESTIGACIÓN	EXPLICACIÓN
RQ1	¿Qué herramientas se utilizan para reconocimiento de voz?	Se requiere adquirir las principales herramientas que integren el reconocimiento de voz.
RQ2	¿Qué metodología se utiliza para la creación de chatbots?	Se requiere evidenciar las metodologías a seguir para generar un chatbot
RQ3	¿Qué metodología se utiliza para realizar la Elicitación de requerimientos?	Se requiere conocer las metodologías que ayuden a empezar la etapa de análisis, para crear las preguntas adecuadas para el chatbot

B) Selección de Fuentes

Las fuentes utilizadas en el estudio fueron las siguientes bibliotecas digitales: *IEEE*, *ACM Digital Library*, *Science Direct* y *JSTOR*. El criterio de selección de estas bases de datos, fue básicamente la disponibilidad para el acceso, ya que son de acceso libre, para la búsqueda y revisión del resumen.

C) Selección de Artículos

En este apartado se muestra la estrategia de búsqueda que se utilizó para llevar a cabo este EMS.

1. Palabras claves

A partir de las preguntas de investigación se obtuvieron las siguientes palabras claves: Reconocimiento de voz, Elicitación de requerimientos, Chatbot, Framework, Herramienta y Metodología.

2. Sinónimos

Se buscaron y generaron sinónimos a las palabras claves, en idioma inglés, pues la mayoría de la bibliografía se redacta en dicho idioma. Los sinónimos se presentan en la Tabla 18.

Tabla 29.- Palabras claves y sus sinónimos

PALABRAS CLAVES	SINÓNIMOS
Speech Recognition	Voice, Communication, conversation, dialogue, talk
Elicitation of requirements	Processing, N/A
Chatbot	N/A
Framework	Frame
Tool	IDE
Methodology	Method

3. Cadenas de Búsqueda

Se construyeron las siguientes cadenas (SQ, *Search Question*) de búsqueda con las palabras clave obtenidas y se utilizaron algunos de los

operadores lógicos: OR para sinónimos o alternativas y AND para combinar las palabras:

SQ principal: ((platform OR Framework) AND (create) AND ((conversational agent) OR (chatbot) AND (recognition OR processing) AND (voice OR speak)))

SQ1: ((tools OR (IDE's)) AND ((recognition OR processing) AND (voice OR speak)))

SQ2: ((methodology OR method) AND (create) AND (chatbot))

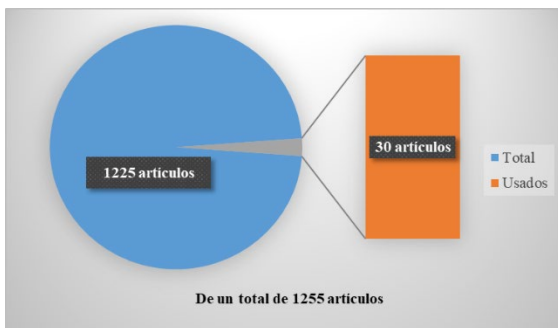
SQ3: (methodology AND (elicitation of requirements))

4. Generación de la búsqueda en las librerías digitales seleccionadas

A continuación, se muestran los resultados que arrojó cada cadena de búsqueda, en cada una de ellas se aplicó el filtrado de inclusión y exclusión que se describen en la sección D.

El resultado que arrojó la **cadena uno** fue un total de **1255 artículos (100%)**, a todos estos se les aplicaron los criterios dejando un total de **30 artículos (2.39%)** útiles para esta investigación. Esto se evidencia con la Gráfica 1.

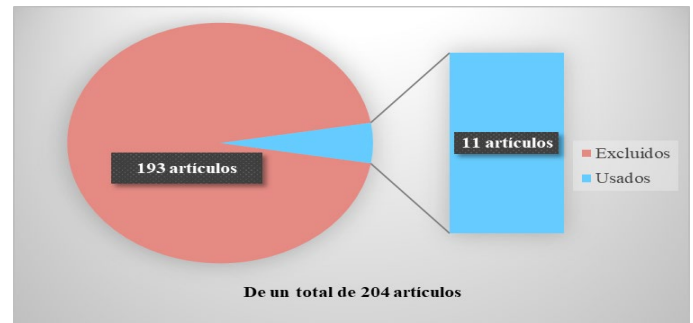
Gráfica 1.- Artículos que mencionan una metodología para el reconocimiento de voz



En segundo lugar, tenemos el resultado de la **cadena dos** con un total de **204 artículos (100%)** de los cuales se le aplicaron los criterios de inclusión y exclusión y solo **11 artículos** fueron útiles (**4.9%**). Donde nueve artículos hablan de metodología (4.41%) y dos tratan acerca de la creación de chatbot (0.98%). Y los resultados se representan en la

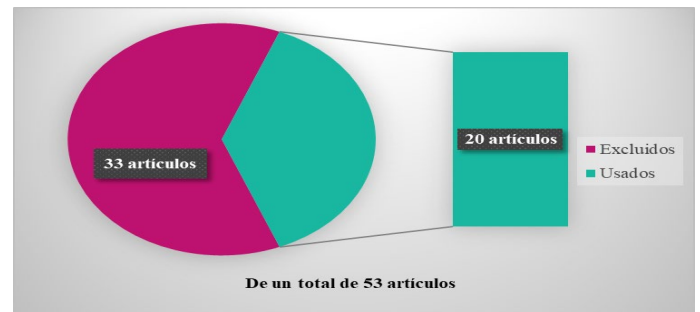
Gráfica 2.

Gráfica 2.- Artículos que mencionan una metodología para la creación de chatbot



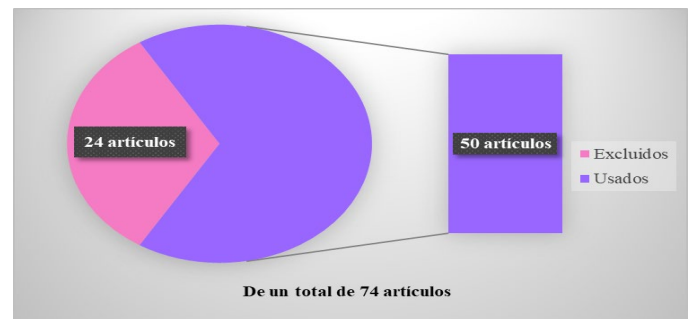
En tercer lugar, se muestran los resultados que arrojó la **cadena tres**, se encontraron **53 artículos (100%)** de los cuales con los criterios solo **20 artículos** fueron útiles (37%). Se puede visualizar los resultados en la Gráfica 3.

Gráfica 3.- Artículos que mencionan la Elicitación de Requerimientos



Y, por último, la **cadena principal** arrojó un total de **74 artículos (100%)**, a los cuales con los criterios solo fueron útiles **50 artículos (67.56%)**. En la Gráfica 4 se pueden observar los artículos obtenidos.

Gráfica 4.- Artículos que mencionan plataformas para la creación de chatbot con reconocimiento de voz



D) Inclusión y Exclusión de los artículos

Como último punto de la selección de los artículos, se fijaron criterios de inclusión que se muestran a continuación en la Tabla 19, y los criterios de exclusión se muestran en la Tabla 20, con lo que se descartaron aquellos artículos no relevantes y los que no respondían ninguna pregunta de investigación.

Tabla 30.- Criterios de Inclusión

TIPO	CRITERIOS
INCLUSIÓN	<ul style="list-style-type: none"> ○ Artículos publicados entre los años 2013 y 2018. ○ Artículos que estén escritos en el idioma inglés. ○ Artículos publicados en revistas y conferencias. ○ Artículos que tengan acceso al resumen, palabras claves y conclusiones. ○ Artículos que evidencien las herramientas para la creación de chatbot. ○ Artículos que hablen del reconocimiento de voz en aplicaciones para chatbot.

Tabla 31.- Criterios de exclusión

TIPO	CRITERIOS
EXCLUSIÓN	<ul style="list-style-type: none"> ○ Artículos incompletos. ○ Artículos que no estén accesibles por las bases de datos seleccionadas. ○ Artículos que aborden alguna herramienta o metodología, pero sin ninguna relación o una relación poca clara. ○ Artículos que no presenten un objetivo claro que sea identificable. ○ Artículos que no incluyen ninguna característica del reconocimiento de voz en herramientas para chatbot.

3. RESULTADOS

A continuación, se muestran los resultados obtenidos del estudio de mapeo sistemático, el cual se enfocó a dar una aproximación de todas las herramientas existentes para la creación de un

asistente inteligente conversacional, basado en las preguntas de investigación.

Se obtuvieron nueve aplicaciones más utilizadas y mencionadas dentro de la literatura, algunas de las características que se pueden tomar en cuenta para comparar las herramientas de creación de chatbots son: la tecnología que aplica, por ejemplo: redes neuronales, procesamiento de lenguaje natural (PLN), aprendizaje automático o aprendizaje basado en reglas por mencionar algunas; la disponibilidad de la documentación en internet; el idioma; el lenguaje de programación que soporta, licencia etc. Para este caso se tomaron en cuenta las siguientes características y se muestran en la Tabla 21, donde la “x” muestra la característica con la que cuenta la herramienta y “—” es aquella característica que no incluye.

Tabla 32.- Herramientas Web

Criterios Nombre	Voz	Idioma Español	Licencia Gratuita	Interoperabilidad	Sin límite de interacciones
	IBM Watson	X	X	X	X
AgentBot (aivo)	X	X	X	--	--
Twyla	--	--	--	X	--
Pypestream	--	X	--	--	X
Live Agent	X	X	--	X	--
Wit.ai	X	X	X	--	--
Chatfuel	--	X	X	--	--
Dialogflow	X	X	X	X	X
Microsoft Language Understanding Intelligent Service (LUIS)	X	X	--	--	--

4. CONCLUSIONES

Este artículo aporta un panorama general sobre el estudio de las herramientas más utilizadas que

ayudan a la creación de un chatbot. Para proporcionar esta visión general se realizó un estudio de mapeo sistemático, con el cual se logró obtener una clasificación de herramientas web basadas en una serie de preguntas de investigación, con las cuales se pudo conocer la diversidad de herramientas que soportan un objetivo único, crear un chatbot. Cualquier herramienta se puede utilizar, pero todo dependerá de las necesidades de cada desarrollador y las características que se esté buscando cumplir con la herramienta. Esta clasificación proporcionó datos importantes obtenidos de la búsqueda, dejándonos conocimientos y demostrando que el tema de los chatbots es muy prometedor y con grandes posibilidades de utilización en la industria de la computación y cualquier área donde se emplee. Por lo tanto, se debe seguir estudiando.

Respondiendo a la pregunta principal de este estudio, la herramienta que mejor se adapta a las necesidades de esta investigación es la herramienta ofrecida por Google llamada DialogFlow, la cual nos da la disponibilidad de crear chatbots mediante el reconocimiento de voz, entre sus ventajas se encuentra que no se debe pagar por usar sus servicios, no limita el número de interacciones lo que se adecua muy bien a esta investigación para la etapa de pruebas y despliegue. También, se ejecuta en una plataforma en la nube de Google, además, permite conectarse a cualquier plataforma externa. Por ejemplo: Facebook, Asistente de Google, Messenger, entre otros. Asimismo, nos brinda la libertad de poder monitorear el proceso de nuestra conversación y así lograr retroalimentar nuestro chatbot, especialmente donde el chatbot no sabe que responder o preguntar en este caso, lo cual nos beneficia mucho para un continuo mantenimiento a lo largo del tiempo.

Dentro de los artículos más relevantes que arrojó el estudio se encontró el de (V. Serban I, 2017) el cual nos habla de un chatbot de aprendizaje de refuerzo profundo llamado MILABOT, fue creado para competir con Alexa de Amazon. MILABOT es capaz de conversar con humanos mediante pláticas pequeñas, este chatbot está creado con redes neuronales, PLN y aprendizaje automático entre otras tecnologías. También se encuentra el artículo de (Litvak, C, 2016) que nos habla de que el desarrollo de modelos de requisitos en lenguaje natural facilita la interacción entre todos los involucrados, aunque contribuye a generar ambigüedades, pero que al

saber usarlo se puede mejorar la calidad de los mismos. Por otro lado, encontramos a (A. Lezcano, L. 2015) y a (Orr, K. 1981) el cual son diagramas y metodologías respectivamente para lograr la calidad que debe cumplir el sistema que se va a desarrollar, por eso, es importante conocerlas.

REFERENCIAS

- Litwin, E., 2005. *'Tecnologías educativas en tiempos de Internet'*: Buenos Aires: Amorroutu, pp.113-138.
- Petersen, K. F. R. M. S. a. M. M., 2008. *'Systematic mapping studies in software engineering'*. Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering, Vol. 2, n° 1, pp. 68-77.
- Quintana E, R. J., 2018. *'Chatbot para consultas sobre trámites administrativos en la Municipalidad de Surco'*. Repositorio Institucional – UTP, Vol. 1, n° 1, pp. 2-15.
- Sameera A, A.-K., 2015. *'Survey on Chatbot Design Techniques in Speech Conversation Systems'*. International Journal of Advanced Computer Science and Applications, Vol. 6, n° 7, pp. 1-9.
- V. Serban I, 2017. *'A Deep Reinforcement Learning Chatbot'*. Institute for Learning Algorithms: Montreal, pp. 1-40
- Litvak, C. S. |. H. G. D. S. |. D. J. H., 2016. *'Procesamiento de Lenguaje Natural para Estudiar Completitud de requisitos'*, San Justo. Buenos Aires: Departamento de Ingeniería e Investigaciones Tecnológicas, UNLAM. pp. 1-5.
- A. Lezcano, L.* , J. A. G. y. C. A. V., 2015. *'A Model for the Identification of KAOS Elements (Knowledge Acquisition Automated Specification) from the Specification of Requirements in Natural Language'*. vol. 26, n°. 6, pp. 5-10.
- Orr, K. T., 1981. *'Structured Requirements Definition. In: Structured Requirements Definition'*. United States of America, pp. 61-135.
- Alemany, L. A. i., 2005. *'Herramientas Libres para Procesamiento del Lenguaje Natural'*, Córdoba, pp.4-12.

ANEXO B

En este apartado se anexa el resumen del artículo presentado en el Congreso Internacional de Lingüística Computacional y de Corpus CILCC 2020 en Colombia.

Entrevista por medio de un chatbot con reconocimiento de voz enfocada a un modelo de negocio para la identificación de entidades

Millán-Olivares Laura *,
González-García Moisés **,
Noé Alejandro Castro-Sánchez***

CENIDET, Cuernavaca, Morelos;

*email: *ana.millan18ca@cenidet.edu.mx,*

*{**moises.gg, *** noe.cs} @cenidet.tecnm.mx*

Resumen: En el mundo, las empresas invierten en mecanismos que les ayuden a mejorar servicios basados en TICs para incrementar su productividad. Uno de estos mecanismos son los chatbots, que son agentes conversacionales con el objetivo de mantener una conversación de manera natural.

Alan Turing se puede considerar el padre de la Inteligencia Artificial (IA). Siendo el autor del concepto de computadora, predijo que la máquina podría llegar a adquirir una capacidad comparable con la inteligencia humana. Para poner fin a las posibles discusiones sobre qué es la inteligencia y si es plausible o no atribuirla a una máquina, en el año 1950 propuso la llamada Prueba de Turing (o también «juego de imitación») para examinar la inteligencia de una máquina. La prueba se basa en la idea de que la interacción verbal constituye un medio en el que la inteligencia se hace más patente. (Gomez, 2000).

Además, que los chatbots debe contener características que apoyen en la comprensión de la propuesta para alcanzar los resultados esperados y que sean satisfactorios (Paredes, 2019). Entonces, el potencial que tiene la IA, es debido a su exactitud y rapidez al procesar una tarea.

Por esta razón, los robots que llevan conversaciones (chatbots), soportados por técnicas de IA, pueden facilitar que las empresas mejoren la eficiencia (dado que realizan su tarea con la menor cantidad de recursos) y eficacia (logran el objetivo esperado) de su proceso de atención al cliente.

El chatbot Atenea es un proyecto a largo plazo que se está llevando a cabo en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), con el objetivo final de crear un agente capaz de interactuar con cualquier ser humano y este obtenga toda la información necesaria de un dominio de problema y pueda producirse el modelo de negocio correspondiente a ese dominio.

De esta manera, la idea del chatbot Atenea es que aprenda de cada interacción que tenga con las personas entrevistadas y esas interacciones lo ayuden a obtener el mayor vocabulario posible para poder generar las entrevistas en base a este conocimiento. El proyecto está todavía en su fase inicial, cabe mencionar que todas las entrevistas se realizarán por medio de reconocimiento de voz.

Lo que se ha realizado de este proyecto es la creación del chatbot que ya cuenta con reconocimiento de voz, este fue creado en la plataforma de Google llamado DialogFlow, es una herramienta que permite crear agentes inteligentes que reconozcan el lenguaje natural mediante voz o texto de forma sencilla e integra machine learning.

Este agente (Atenea) realiza una entrevista a un personal administrativo, con el objetivo de conocer su rol dentro de la empresa donde trabaja y saber con quién o quienes interactúa durante todo su proceso, también se requiere saber si intercambia alguna información (oficios o documentos) con estas entidades con las que interactúa.

En cuanto al agente se sabe que es igual a un ser humano y al igual que a él, se necesita llenar de vocabulario y entrenamiento para que sepa manejar las situaciones dentro de una conversación, es por ello que el chatbot cuenta con intents y entidades que ayudarán a guiar la entrevista. Un intent clasifica la intención del usuario final para un turno de conversación y cada parámetro de intent pertenece a un tipo, el cual se denomina tipo de entidad. Esto determina de forma exacta cómo se extraen los datos de una expresión de usuario final. (Cloud, 2020).

En base a esto el chatbot Atenea está compuesto de 12 preguntas cicladas que guiaran la entrevista con el personal administrativo, éstas a su vez están divididas en cinco entidades, 20 intenciones y aproximadamente 400 frases de entrenamiento. Esto con el fin de proporcionarle el mayor

conocimiento posible al agente y este pueda guiar al entrevistado de una forma correcta y fácil de entender.

Por otro lado, concluida la entrevista y recabada toda la información necesaria del dominio del problema, se analizará el historial de dicha conversación con una biblioteca de Procesamiento del lenguaje Natural (*NLP*, por sus siglas en inglés), esta biblioteca es conocida como Freeling, y nos ayuda para el procesamiento multilingüe automático, que proporciona un análisis lingüístico. Este análisis es etiquetado por medio de lemas, es decir, la forma base de una palabra, el análisis se realiza en el lenguaje de java.

Después de extraer el texto lematizado, se pasa a un segundo filtro donde se compara cada lema con los diccionarios dentro de este, existe tres diccionarios, uno con sinónimos de enviar, el segundo con sinónimos de recibir y por último el tercer diccionario incluyen las diferentes entidades posibles con las que un personal administrativo puede tener un intercambio de información, ya que estos conceptos son los mínimos necesarios para poder realizar el modelo de negocios correcto a ese dominio.

Palabras claves: Chatbot, Modelo de negocio, Lenguaje Natural, Reconocimiento de voz, Procesamiento de Lenguaje Natural, Modelo de dominio, Entidades, Inteligencia Artificial, Intent, Entidades.

Referencias

- Cloud, G., 2020. *Documentación*. [En línea] Available at: <https://cloud.google.com/dialogflow/docs/basics?hl=es-419> [Último acceso: 03 marzo 2020].
- García, A. E. E., 2009. 'Vida e Inteligencia Artificial'. *ACIMED*, 19(1), pp. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352009000100006.
- Gomez, E. & T., 2000. 'La conciencia: Capitulo 2'. En: '*El rompecabezas del cerebro*'. s.l.: s.n., p. 5.
- Lisboa, L. & G. V. & L. D. & A. E. & M. S. & F. R., 2010. '*A systematic review of domain analysis tools*'. ResearchGate. *Journal*, Volumen 52, pp. 1-13.
- Litwin, E., 2005. *Tecnologías educativas en tiempos de Internet*. En: Buenos Aires: Amorroutu.
- Orr, K. T., 1981. '*Structured Requeriments Definition. In: Structured Requeriments Definition*'. United States of America: s.n.

- Paredes, C. T. M., 2019. *Impacto de los chatbots en la atención al cliente en la cooperativa de ahorro y crédito El sagrario*, s.l.: Uniandes.

ANEXO C

En este anexo se va a explicar cómo se crean las entidades e intenciones dentro de la plataforma de Google llamada DialogFlow.

- Como primer punto, dentro de la plataforma de DialogFlow, se debe crear un agente, definir el idioma en el que se va a trabajar, y tener un nombre para dicho agente. Ver

Figura 59

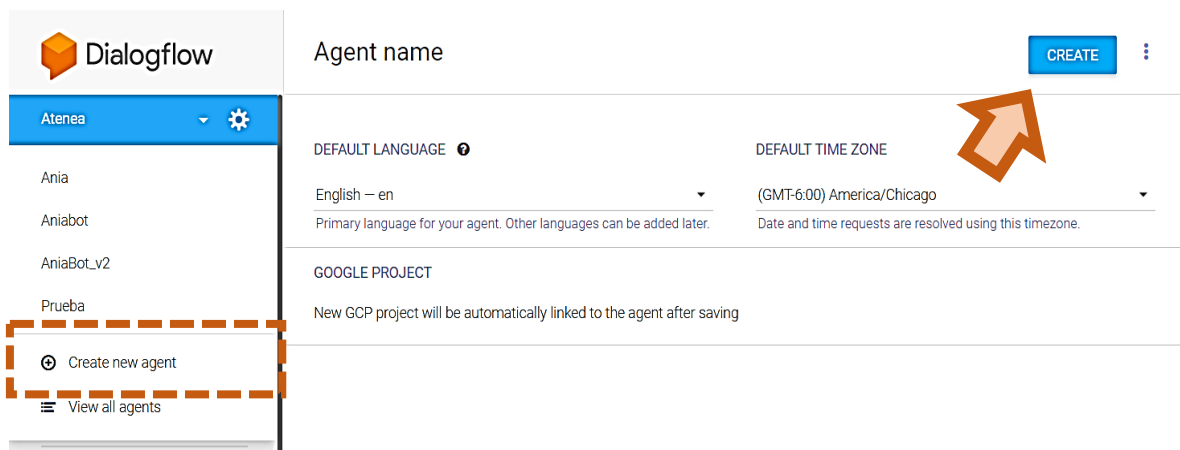


Figura 59.- Crear un agente (Chatbot) en DialogFlow

- Después de crear nuestro chatbot, tenemos que ir empezando a crear las entidades que utilizará nuestro chatbot, esto se hace posicionándose en la pestaña de entidades, darle un nombre y presionar el botón de crear. Ver **Figura 60**

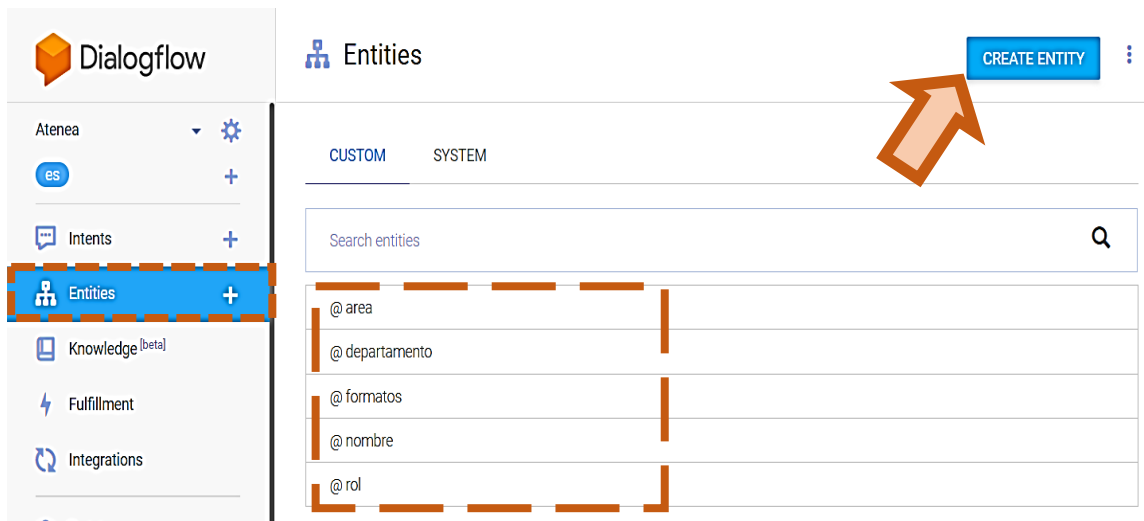


Figura 60.- Crear una entidad en DialogFlow

- Para poder crear las intenciones se tiene que hacer el mismo proceso que se realizó con las entidades, posicionarse en la pestaña intención, darle un nombre y finalmente dar clic en el botón de crear intención. Ver **Figura 61**



Figura 61.- Crear una intención en DialogFlow

ANEXO D

En este anexo se muestra el código que se utilizó en la Clase Analyzer.java, donde se implementa el analizador morfosintáctico Freeling, para el procesamiento del texto.

```
public class Analyzer {
// Directorio de instalación de Freeling
private static final String DATA = "C:/freeling-3.1-win/data/";
```


//Idioma con el que se trabajará Freling

```
private static final String LANG = "es";
public static void main( String argv[] ) throws IOException {
    System.loadLibrary( "freeling_javaAPI" );

// Crear Analizadores.
    LangIdent lgid = new LangIdent(DATA + "/common/lang_ident/ident-few.dat");
    Tokenizer tk = new Tokenizer( DATA + LANG + "/tokenizer.dat" );
    Splitter sp = new Splitter( DATA + LANG + "/splitter.dat" );
    Maco mf = new Maco( op );
    HmmTagger tg = new HmmTagger( DATA + LANG + "/tagger.dat", true, 2 );
    ChartParser parser = new ChartParser(
        DATA + LANG + "/chunker/grammar-chunk.dat" );
    DepTxala dep = new DepTxala( DATA + LANG + "/dep/dependences.dat",
        parser.getStartSymbol() );
    Nec neclase = new Nec( DATA + LANG + "/nec/nec/nec-ab-poor1.dat" );
    Senses sen = new Senses(DATA + LANG + "/senses.dat" ); // sense dictionary
    Ukb dis = new Ukb( DATA + LANG + "/ukb.dat" ); // sense disambiguator

//Leer archivo
    Path archivo = Paths.get("./", "Entrevista_Presidenta.txt"); //NOMBRE O DIRECCIÓN DEL ARCHIVO

    Charset charset = Charset.forName("utf-8");
        List <String> lines = Files.readAllLines(archivo, charset);
        String texto = "";
        for (String line : lines) {
            texto += " " + line; // Almacena todas las líneas del archivo en una sola variable
                //System.out.println(tex)
        }
    // while (texto != null ) {
// Extrae los tokens de la línea de texto.
    ListWord l = tk.tokenize( texto );

// Divide los tokens en frases distintas.
```

```

ListSentence ls = sp.split( l, false );

// Análisis morfológico
mf.analyze( ls );

// Part-of-speech tagging.
tg.analyze( ls );

// Clasificación de entidades nombradas (NE)
neclass.analyze( ls );
sen.analyze( ls );
dis.analyze( ls );
printResults(ls, "tagged" );
}

private static void printSenses( Word w ) {
    String ss = w.getSensesString();
private static void printResults( ListSentence ls, String format ) {
BufferedWriter writer = null;
try {
writer = new BufferedWriter(new FileWriter("./SalidaEntrevista3.txt")); //escribe archivo
    if (format == "parsed") {
        System.out.println( "----- CHUNKER results -----" );
        ListSentenceIterator sIt = new ListSentenceIterator(ls);
        while (sIt.hasNext()) {
            Sentence s = sIt.next();
            TreeNode tree = s.getParseTree();
            printParseTree( 0, tree );
        }
    }
    else if (format == "dep") {
        System.out.println( "----- DEPENDENCY PARSER results -----" );

        ListSentenceIterator sIt = new ListSentenceIterator(ls);

```

```

        while (sIt.hasNext()) {
Sentence s = sIt.next();

            TreeDepnode tree = s.getDepTree();
            printDepTree( 0, tree);

        }
    }
else
{
    System.out.println( "----- TAGGER results -----" );
    // saca las palabras analizadas de ls..
    ListSentenceIterator sIt = new ListSentenceIterator(ls);
    while (sIt.hasNext()) {
        Sentence s = sIt.next();
        ListWordIterator wIt = new ListWordIterator(s);
        while (wIt.hasNext()) {
            Word w = wIt.next();

// imprimir en consola
            System.out.print(
                w.getForm() + " " + w.getLemma() + " " + w.getTag() );
                writer.write(w.getForm() + " " + w.getLemma() + " " +
w.getTag() + "\n"); //almacena los datos en el archivo
            // printSenses( w );
            System.out.println();
        }
        System.out.println();
    }
}
} catch(Exception e){
    e.printStackTrace();
} finally {

```

```

        if (writer != null) {
            try {
                writer.close();
            } catch(Exception e2){}
        }
    }
}

```

ANEXO E

En este anexo se muestra el código que se utilizó en la Clase Or.rb de Ruby, donde se implementa el programa que compara el archivo etiquetado con los diccionarios establecidos.

```

#encoding: UTF-8
require 'csv'
# Declara el encabezado para el archivo que se creará
encabezado = ['Entidad-Origen', 'Direccion', 'Formato', 'Entidad-Destino']
#-----Variables-----
origen = nil
direccion = nil
formato= nil
destino= nil
num_linea = 0
num_linea1 = 0
num_linea2 = 0
num_linea3 = 0
num_linea4 = 0
#Creación del archivo .CSV
CSV.open('resultado2.csv', 'w', write_headers: true, headers: encabezado) do |archivo_nuevo|
##-----Reconocer el origen-----
# Abre el archivo que contiene cada una de las líneas que se van a buscar en otro archivo
File.open("Entidades.txt") do |archivo|

```

```

archivo.each do |linea|
  num_linea +=1
  puts "Linea #{num_linea} del archivo Entidades.txt = "+linea.strip
      # Abre un archivo delimitado por comas
  # En el archivo, se busca la línea que se leyó del primer archivo
File.open("SalidaEntrevista2.txt").each do |linea2|
  if linea2.include? linea.strip
    puts "El origen "+linea.strip
    origen = linea.strip
    archivo_nuevo << [origen]
    break
  end
end
end
end

end

## ----- Reconocer Dirección-----

# Abre el archivo que contiene cada una de las líneas que se van a buscar en otro archivo
File.open("SinonimosRecibir.txt") do |archivo|
  archivo.each do |linea1|
    num_linea1 +=1
    puts "Linea #{num_linea1} del archivo Recibir.txt = "+linea1.strip
      # Abre un archivo delimitado por comas
  # En el archivo, se busca la línea que se leyó del primer archivo
File.open("SalidaEntrevista2.txt").each do |linea3|
  if linea3.include? linea1.strip
    puts "Acción "+linea1.strip
    direccion = linea1.strip
    archivo_nuevo << [origen,direccion]
    break
  end
end

```

```

        end
    end
end
## ----- Reconocer dirección-----
# Abre el archivo que contiene cada una de las líneas que se van a buscar en otro archivo
File.open("SinonimosEnviar.txt") do |archivo|
    archivo.each do |linea4|
        num_linea2 +=1
        puts "Linea #{num_linea2} del archivo Entidades.txt = "+linea4.strip
            # Abre un archivo delimitado por comas
        # En el archivo, se busca la línea que se leyó del primer archivo
        File.open("SalidaEntrevista2.txt").each do |linea5|
            if linea5.include? linea4.strip
                puts "Acción "+linea4.strip
                direccion = linea4.strip
                archivo_nuevo << [origen,direccion]
                break
            end
        end
    end
end
end
## ----- Reconocer formato-----
# Abre el archivo que contiene cada una de las líneas que se van a buscar en otro archivo
File.open("Formatos.txt") do |archivo|
    archivo.each do |linea8|
        num_linea4 +=1
        puts "Linea #{num_linea4} del archivo Formatos.txt = "+linea8.strip
            # Abre un archivo delimitado por comas
        # En el archivo, se busca la línea que se leyó del primer archivo
        File.open("SalidaEntrevista2.txt").each do |linea9|

```

```

    if linea9.include? linea8.strip
      puts "Formato "+linea8.strip
      formato = linea8.strip
      archivo_nuevo << [origen,direccion,formato]
      break
    end
  end
end

end

end

end

## ----- Reconocer Destino-----

# Abre el archivo que contiene cada una de las líneas que se van a buscar en otro archivo
File.open("Entidades.txt") do |archivo|
  archivo.each do |linea6|
    num_linea3 +=1
    puts "Linea #{num_linea3} del archivo Entidades.txt = "+linea6.strip
      # Abre un archivo delimitado por comas
    # En el archivo, se busca la línea que se leyó del primer archivo
    File.open("SalidaEntrevista2.txt").each do |linea7|
      if linea7.include? linea6.strip
        puts "Destino "+linea6.strip
        destino = linea6.strip
        archivo_nuevo << [origen,direccion,formato,destino]
        break
      end
    end
  end
end
end
end
end

```