



**UNIVERSIDAD DE GUAYAQUIL  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA DE INGENIERÍA EN TELEINFORMÁTICA**

**TRABAJO DE TITULACIÓN  
PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN TELEINFORMÁTICA**

**ÁREA  
TECNOLOGÍA DE LOS ORDENADORES**

**TEMA  
MODELO BASADO EN WEB SCRAPING Y CHATBOT  
PARA CONOCER LAS HABILIDADES DIGITALES MÁS  
DEMANDADAS EN SECTOR DE TI**

**AUTOR  
VERA VERA DAYANNA MELINA**

**DIRECTORA DEL TRABAJO  
ING. COMP. CASTILLO LEÓN ROSA ELIZABETH, MG.**

**GUAYAQUIL, SEPTIEMBRE 2022**



## ANEXO XI.- FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN



### FACULTAD DE INGENIERÍA INDUSTRIAL CARRERA INGENIERÍA EN TELEINFORMÁTICA

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN			
TÍTULO:	“MODELO BASADO EN WEB SCRAPING Y CHATBOT PARA CONOCER LAS HABILIDADES DIGITALES MÁS DEMANDADAS EN SECTOR DE TI.”		
AUTOR (apellidos y nombres):	VERA VERA DAYANNA MELINA		
TUTOR y REVISOR (apellidos y nombres):	ING. COMP. CASTILLO LEÓN ROSA ELIZABETH, MG ING. SIST. GARCÍA TORRES INGRID ANGÉLICA, MG		
INSTITUCIÓN:	UNIVERSIDAD DE GUAYAQUIL		
UNIDAD/FACULTAD:	FACULTAD DE INGENIERÍA INDUSTRIAL		
MAESTRÍA/ESPECIALIDAD:			
GRADO OBTENIDO:	INGENIERA EN TELEINFORMÁTICA		
FECHA DE PUBLICACIÓN:	28/SEPTIEMBRE/2022	No. DE PÁGINAS:	109
ÁREAS TEMÁTICAS:	TECNOLOGÍA DE LOS ORDENADORES		
PALABRAS CLAVES/ KEYWORDS:	WEB SCRAPING, HABILIDADES DIGITALES, CHATBOT, PYTHON, FIREBASE, DIALOGFLOW.		
<p><b>RESUMEN:</b> En este proyecto se plantea el desarrollo de un modelo de sistema compuesto por un chatbot que funciona con datos extraídos mediante web scraping y cuyo propósito es facilitar el conocimiento de las habilidades digitales más requeridas por empresas que ofertan empleos del área de desarrollo de software. Además de una metodología no experimental con enfoque cualitativo, se empleó también una metodología para la implementación del modelo, la cual está compuesta por las fases de web scraping, almacenamiento de datos y entrenamiento del Lenguaje Natural. Durante el estudio se ratificó a través de una encuesta a 89 estudiantes de la carrera Telemática, que su conocimiento sobre las habilidades digitales más demandadas es básico. Finalmente se obtuvo como resultado una herramienta de tipo chatbot basada en el modelo propuesto, que facilita el conocimiento de las habilidades tecnológicas más demandadas y que cumple con los parámetros especificados por los estudiantes a través de la encuesta.</p> <p><b>ABSTRACT:</b> This project presents the development of a system model consisting of a chatbot that works with data extracted through web scraping and whose purpose is to facilitate understanding of the digital skills most required by companies that offer jobs in the software development area. In addition to a non-experimental methodology with a qualitative approach, a methodology was also used for the implementation of the model, which is composed of the phases of web scraping, data storage and Natural Language training. During the study, it was confirmed through a survey to 89 students of the Telematics career, that the understanding about the most demanded digital skills is basic. Finally, a chatbot-type tool based on the proposed model was obtained as a result, which facilitates the understanding of</p>			

the most demanded technological skills and that meets the parameters specified by the students through the survey.

ADJUNTO PDF:	SI (X)	NO
CONTACTO CON AUTOR/ES:	Teléfono: 0982076583	E-mail: dayanna-vera10@hotmail.com
CONTACTO CON LA INSTITUCIÓN:	Nombre: Ing. Ind. Ramón Maquilón Nicola, Mg	
	Teléfono: 042-658128	
	E-mail: direccioni@ug.edu.ec	



**ANEXO XII.- DECLARACIÓN DE AUTORÍA Y DE  
AUTORIZACIÓN DE LICENCIA GRATUITA  
INTRANSFERIBLE Y NO EXCLUSIVA PARA EL USO NO  
COMERCIAL DE LA OBRA CON FINES NO  
ACADÉMICOS**



---

**FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA INGENIERÍA EN TELEINFORMÁTICA**

---

LICENCIA GRATUITA INTRANSFERIBLE Y NO COMERCIAL DE LA OBRA CON  
FINES NO ACADÉMICOS

Yo **VERA VERA DAYANNA MELINA** con C.C. No. **0944158849**, certifico que los contenidos desarrollados en este trabajo de titulación, cuyo título es **“MODELO BASADO EN WEB SCRAPING Y CHATBOT PARA CONOCER LAS HABILIDADES DIGITALES MÁS DEMANDADAS EN SECTOR DE TI.”** son de mi absoluta propiedad y responsabilidad, en conformidad al Artículo 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN\*, autorizo la utilización de una licencia gratuita intransferible, para el uso no comercial de la presente obra a favor de la Universidad de Guayaquil.

A handwritten signature in blue ink that reads "Dayanna Vera V".

---

**VERA VERA DAYANNA MELINA**  
**C.C.: 0944158849**



**ANEXO VII.- CERTIFICADO PORCENTAJE DE  
SIMILITUD  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Habiendo sido nombrado **ING. COMP. CASTILLO LEÓN ROSA ELIZABETH, MG**, tutor del trabajo de titulación certifico que el presente trabajo de titulación ha sido elaborado por **VERA VERA DAYANNA MELINA**, con mi respectiva supervisión como requerimiento parcial para la obtención del título de **INGENIERO EN TELEINFORMÁTICA**.

Se informa que el trabajo de titulación: **MODELO BASADO EN WEB SCRAPING Y CHATBOT PARA CONOCER LAS HABILIDADES DIGITALES MÁS DEMANDADAS EN SECTOR DE TI**, ha sido orientado durante todo el periodo de ejecución en el programa antiplagio TURNITIN quedando el 5% de coincidencia.

<https://ev.turnitin.com/app/carta/es/?u=1133714545&s=1&o=1896318465&lang=es>



**ING. COMP. CASTILLO LEÓN ROSA ELIZABETH, MG.**  
**DOCENTE TUTOR**  
**C.C.: 0922372610**  
**FECHA: 12 DE SEPTIEMBRE DEL 2022**



**ANEXO VI. - CERTIFICADO DEL DOCENTE-TUTOR  
DEL TRABAJO DE TITULACIÓN  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 12 de septiembre de 2022

Sr (a).

**Ing. Annabelle Lizarzaburu Mora, MG.**

Director (a) de Carrera Ingeniería en Teleinformática/Telemática

FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE  
GUAYAQUIL

Ciudad.-

De mis consideraciones:

Envío a Ud. el Informe correspondiente a la tutoría realizada al Trabajo de Titulación **MODELO BASADO EN WEB SCRAPING Y CHATBOT PARA CONOCER LAS HABILIDADES DIGITALES MÁS DEMANDADAS EN SECTOR DE TI**, indicando que ha cumplido con todos los parámetros establecidos en la normativa vigente:

- El trabajo es el resultado de una investigación.
- El estudiante demuestra conocimiento profesional integral.
- El trabajo presenta una propuesta en el área de conocimiento.
- El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se adjunta el certificado de porcentaje de similitud y la valoración del trabajo de titulación con la respectiva calificación.

Dando por concluida esta tutoría de trabajo de titulación, **CERTIFICO**, para los fines pertinentes, que el estudiante está apto para continuar con el proceso de revisión final.

Atentamente,



Firmado electrónicamente por:

**ROSA  
ELIZABETH  
CASTILLO LEÓN**

**ING. COMP. CASTILLO LEÓN ROSA ELIZABETH, MG.  
DOCENTE TUTOR  
C.C.: 0922372610  
FECHA: 12 DE SEPTIEMBRE DEL 2022**



**ANEXO VIII.- INFORME DEL DOCENTE REVISOR  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 22 de septiembre de 2022.

Sr (a).

**Ing. Annabelle Lizarzaburu Mora, MG.**

Director (a) de Carrera Ingeniería en Teleinformática / Telemática

**FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE  
GUAYAQUIL**

Ciudad. -

De mis consideraciones:

Envío a usted el informe correspondiente a la REVISIÓN FINAL del Trabajo de Titulación **“MODELO BASADO EN WEB SCRAPING Y CHATBOT PARA CONOCER LAS HABILIDADES DIGITALES MÁS DEMANDADAS EN SECTOR DE TI”** del estudiante **VERA VERA DAYANNA MELINA**. Las gestiones realizadas me permiten indicar que el trabajo fue revisado considerando todos los parámetros establecidos en las normativas vigentes, en el cumplimiento de los siguientes aspectos:

Cumplimiento de requisitos de forma:

El título tiene un máximo de 18 palabras.

La memoria escrita se ajusta a la estructura establecida.

El documento se ajusta a las normas de escritura científica seleccionadas por la Facultad.

La investigación es pertinente con la línea y sublíneas de investigación de la carrera.

Los soportes teóricos son de máximo 5 años.

La propuesta presentada es pertinente.

Cumplimiento con el Reglamento de Régimen Académico:

El trabajo es el resultado de una investigación.

El estudiante demuestra conocimiento profesional integral.

El trabajo presenta una propuesta en el área de conocimiento.

El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se indica que fue revisado, el certificado de porcentaje de similitud, la valoración del tutor, así como de las páginas preliminares solicitadas, lo cual indica el que el trabajo de investigación cumple con los requisitos exigidos.

Una vez concluida esta revisión, considero que el estudiante está apto para continuar el proceso de titulación. Particular que comunicamos a usted para los fines pertinentes.

Atentamente,



Firmado electrónicamente por:  
**INGRID  
ANGELICA  
GARCIA TORRES**

**ING. SIST. GARCÍA TORRES INGRID ANGÉLICA, MG**

**C.C: 1308497682**

**FECHA: 23 DE SEPTIEMBRE DEL 2022**

### **Dedicatoria**

Este trabajo es dedicado principalmente a Dios, quien desde el día uno en nivelación hasta el día de hoy ha sido mi fiel sustento y fuente de toda sabiduría, la gloria sea para Él. En segundo lugar dedico este trabajo a mis padres Mayra Vera Figueroa y Miguel Vera González, quienes con amor me han brindado todo su apoyo en cada día de estudio, han sabido comprender cuando estuve muy ocupada tratando de lograr esta meta y siempre me han provisto en mis necesidades a pesar de sus limitaciones.



## **Agradecimientos**

A mi tutora ing. Rosa Castillo por la guía brindada durante todo este proceso y por la excelencia de sus enseñanzas impartidas desde el primer semestre, a mis compañeros de estudio por colaborar con las encuestas, a mis compañeros de trabajo David y Jhonier por sugerirme buenas ideas en la programación de este proyecto, a mi familia y a mi amigo Nelson por alentarme a llegar hasta el final.

## Índice general

N°	Descripción	Pág.
	Introducción	1

### Capítulo I

#### El problema

N°	Descripción	Pág.
1.1	Problema de investigación	2
1.1.1	Planteamiento del problema	2
1.2	Formulación del problema de investigación	3
1.2.1	Formulación de hipótesis	3
1.3	Justificación e importancia	3
1.4	Objetivos	4
1.4.1	Objetivo general	4
1.4.2	Objetivos específicos	4
1.5	Alcance	5

### Capítulo II

#### Marco Teórico

N°	Descripción	Pág.
2.1	Antecedentes de la investigación	6
2.2	Marco Teórico	8
2.2.1	Web Scraping	8
2.2.1.1	Herramientas para usuarios en general	8
2.2.1.2	Herramientas para usuarios con conocimientos en programación	8
2.2.2	Inteligencia Artificial	10
2.2.3	Aprendizaje Automático	10
2.2.3.1	Aprendizaje Supervisado	11
2.2.3.2	Aprendizaje No Supervisado	11
2.2.3.3	Aprendizaje Por Refuerzo	11
2.2.4	Aprendizaje Profundo	11
2.2.5	Procesamiento del Lenguaje Natural (PLN)	11
2.2.5.1	Arquitectura de un sistema PLN	12

<b>Nº</b>	<b>Descripción</b>	<b>Pág.</b>
2.2.5.2	Sintaxis y Gramática	13
2.2.6	Agente conversacional (Chatbot)	15
2.2.7	DialogFlow	16
2.2.7.1	Agentes	16
2.2.7.2	Intents	16
2.2.7.3	Entidades	17
2.2.7.4	Contextos	17
2.2.7.5	Fulfillment	18
2.2.7.6	Integración	19
2.2.7.7	Modos de entrada	19
2.2.7.8	Modos de salida	19
2.2.8	Facebook Messenger	19
2.2.8.1	Facebook Messenger Bot	20
2.2.9	Firebase	20
2.2.9.1	Cloud Firestore	20
2.2.10	Webhooks	21
2.2.11	Habilidades digitales en desarrollo de software	22
2.2.11.1	Lenguajes de Programación	23
2.2.11.1	Frameworks	23
2.2.11.1	Bases de Datos	23
2.2.11.1	Control de Código Fuente	23
2.2.11.1	Entornos de desarrollo integrado (IDE)	23
2.3	Marco Conceptual	23
2.3.1	Web Scraping	23
2.3.2	Selenium	24
2.3.3	Chatbot	24
2.3.4	Dialogflow	24
2.3.5	Firestore	24
2.3.6	Meta Developers	24
2.3.7	Habilidades Digitales	24
2.3.8	Webhook	24
2.3.9	Requerimientos Funcionales	25

<b>N°</b>	<b>Descripción</b>	<b>Pág.</b>
2.3.10	Requerimientos no funcionales	25
2.3.11	Xpath	25
2.4	Marco Legal	25
2.4.1	Web Scraping	25

### **Capítulo III**

#### **Diseño de la investigación**

<b>N°</b>	<b>Descripción</b>	<b>Pág.</b>
2.5	Enfoque de la investigación	27
2.5.1	Enfoque cualitativo	27
2.6	Operacionalización de variables	27
2.7	Técnica de recolección de datos	27
2.8	Población	27
2.9	Muestra	28
2.10	Desarrollo del cuestionario	28
2.11	Interpretación y análisis de resultados	28
2.12	Desarrollo de la propuesta	36
2.12.1	Requerimientos funcionales	36
2.12.2	Requerimientos no funcionales	38
2.12.2.1	Accesibilidad	38
2.12.2.2	Funcionalidad	38
2.12.3	Requerimientos de software	39
2.12.4	Diagramas de caso de uso	39
2.12.4.1	Diagramas de caso de uso consulta de habilidades	39
2.12.4.2	Diagramas de caso de uso saludar	40
2.12.4.3	Diagramas de caso de uso extraer información	40
2.12.5	Arquitectura del modelo	41
2.12.5.1	Capa de presentación	41
2.12.5.2	Capa de servicio PLN	41
2.12.5.3	Capa de almacenamiento de datos	41
2.12.5.4	Capa de Web Scraping	41
2.12.16	Diseño de interfaces	42

<b>Nº</b>	<b>Descripción</b>	<b>Pág.</b>
2.12.17	Diseño de base de datos	47
2.13	Implementación	48
2.13.1	Extracción de datos del portal Computrabajo	48
2.13.1.1	Exploración del sitio web	48
2.13.1.2	Reconocimiento de la estructura HTML	49
2.13.1.3	Desarrollo del código web scraping	51
2.13.1.4	Almacenamiento en la base de datos	52
2.13.2	Extracción de datos de LinkedIn	53
2.13.2.1	Exploración del sitio web	53
2.13.2.2	Reconocimiento de la estructura HTML	54
2.13.2.3	Desarrollo del código web scraping	56
2.13.2.4	Almacenamiento en la base de datos	56
2.13.3	Integración entre servicios	56
2.13.3.1	Integración Dialogflow con Messenger	57
2.13.3.1	Integración Firebase con Dialogflow	57
2.13.4	Entrenamiento PLN	58
2.14	Modelo final	59
2.14.1	Funcionalidades principales	59
2.14.1.1	Saludar	59
2.14.1.2	Consultar lo más demandado	60
2.14.1.3	Consultar significado	61
2.14.1.4	Consultar ofertas	62
2.14.1.5	Visitar página de la oferta	63
2.14.1.6	Despedida	64
2.15	Escenario de prueba	64
2.15.1	No existe significado	65
2.15.2	No se encontraron ofertas	65
2.15.3	Fallo al conectar con la base de datos	66
2.16	Conclusiones	68
2.17	Recomendaciones	69
	<b>Anexos</b>	70
	<b>Bibliografía</b>	87

## Índice de Tablas

<b>Nº</b>	<b>Descripción</b>	<b>Pág.</b>
1	Requerimiento funcional 1	36
2	Requerimiento funcional 2	37
3	Requerimiento funcional 3	37
4	Requerimiento funcional 4	37
5	Requerimiento funcional 5	38
6	Requerimiento de software para el desarrollo del proyecto	39
7	Entrenamiento PLN en Dialogflow	58

## Índice de Figuras

<b>N°</b>	<b>Descripción</b>	<b>Pág.</b>
1	Industrias consideradas según sectores	3
2	Componentes PLN	12
3	Arquitectura de un sistema PLN	13
4	Análisis gramático de una oración	15
5	Flujo básico para detectar coincidencias de intents	17
6	Ejemplo de contexto para un agente bancario	18
7	Integración de Dialogflow con otros servicios	19
8	Webhooks	21
9	Lenguajes de programación más populares	22
10	Gráfico estadístico pregunta 1: qué semestre está cursando actualmente?	29
11	Gráfico estadístico pregunta 2: ¿cuánto conoce acerca de las tecnologías para el desarrollo de software más solicitadas por las empresas en Ecuador actualmente?	30
12	Gráfico estadístico pregunta 3: seleccione los cargos que conoce o ha escuchado anteriormente	31
13	Gráfico estadístico pregunta 4: ¿cuánto conoce acerca de las tecnologías para el desarrollo de software más solicitadas por las empresas en Ecuador actualmente?	31
14	Gráfico estadístico pregunta 5: ¿ha interactuado alguna vez con un chatbot?	32
15	Gráfico estadístico pregunta 6: ¿cómo le resulta más cómodo acceder a un chatbot?	32
16	Gráfico estadístico pregunta 7: ¿cuánto tiempo estaría dispuesto a dedicar para interactuar con un chatbot informativo?	33
17	Gráfico estadístico pregunta 8: ¿cómo preferiría que fueran las respuestas de un chatbot?	34
18	Gráfico estadístico pregunta 9: en cuanto al contenido de las respuestas de un chatbot, dhl1	34

<b>N°</b>	<b>Descripción</b>	<b>Pág.</b>
19	Gráfico estadístico pregunta 10: ¿qué acciones le gustaría que realizara el chatbot para que su conocimiento acerca del mercado laboral del desarrollo de software mejore?	35
20	Diagrama de caso de uso consultar habilidades	40
21	Caso de uso saludo del usuario	40
22	Caso de uso extraer información	40
23	Arquitectura del modelo	42
24	Interfaz de saludo	43
25	Interfaz habilidades más demandadas	44
26	Interfaz mostrar significado	45
27	Interfaz ver ofertas	46
28	Diseño de bases de datos	47
29	Entorno del portal Computrabajo	48
30	Detalles de la oferta laboral	49
31	Estructura HTML Computrabajo	50
32	Estructura HTML del detalle de la oferta	50
33	Código identificadores Xpath	51
34	Librerías de Python a utilizar	51
35	Bucle for para recorrer ofertas	52
36	Credenciales de acceso firebase	53
37	Almacenando datos en Firestore	53
38	Entorno de LinkedIn	54
39	Estructura HTML de LinkedIn	55
40	HTML descripción de la oferta en LinkedIn	55
41	Función saludar	59
42	Función consultar habilidades más demandadas	60
43	Función consultar significado	61
44	Función consultar ofertas laborales	62
45	Función visitar página de la oferta	63
46	Función de despedida	64
47	Escenario “No existe significado”	65
48	Escenario “No se encontraron ofertas”	66



<b>N°</b>	<b>Descripción</b>	<b>Pag.</b>
49	Escenario “Fallo en la conexión con la base de datos	67
50	Código en Python del web scraping Computrabajo	73
51	Código en Python del web scraping LinkedIn	74
52	Instalación firebase-tools y CLI de firebase	75
53	Inicio de sesión en la CLI de firebase	75
54	Creación del proyecto local en la CLI de Firebase	76
55	Código JavaScript del chatbot en Visual Studio Code	76
56	Apartado integraciones, en Dialogflow	77
57	Apartado Webhooks, en Meta For Developers	77
58	Frases de entrenamiento intent saludo	78
59	Frases de entrenamiento intent más demandado	78
60	Frases de entrenamiento intent consultar significado	79
61	Frases de entrenamiento intent consultar ofertas	79
62	Frases de entrenamiento intent despedida	80
63	Entity habilidad	80
64	Envío de solicitud de revisión de aplicación, Meta Developers	86



**ANEXO XIII.- RESUMEN DEL  
TRABAJO DE TITULACIÓN (ESPAÑOL)  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA INGENIERÍA EN TELEINFORMÁTICA**

---



**“MODELO BASADO EN WEB SCRAPING Y CHATBOT PARA CONOCER LAS  
HABILIDADES DIGITALES MÁS DEMANDADAS EN SECTOR DE TI”**

**Autor:** Vera Vera Dayanna Melina

**Tutor:** Ing. Comp. Castillo León Rosa Elizabeth, Mg.

**Resumen**

En este proyecto se plantea el desarrollo de un modelo de sistema compuesto por un chatbot que funciona con datos extraídos mediante web scraping y cuyo propósito es facilitar el conocimiento de las habilidades digitales más requeridas por empresas que ofertan empleos del área de desarrollo de software. Además de una metodología no experimental con enfoque cualitativo, se empleó también una metodología para la implementación del modelo, la cual está compuesta por las fases de web scraping, almacenamiento de datos y entrenamiento del Lenguaje Natural. Durante el estudio se ratificó a través de una encuesta a 89 estudiantes de la carrera Telemática, que su conocimiento sobre las habilidades digitales más demandadas es básico. Finalmente se obtuvo como resultado una herramienta de tipo chatbot basada en el modelo propuesto, que facilita el conocimiento de las habilidades tecnológicas más demandadas y que cumple con los parámetros especificados por los estudiantes a través de la encuesta.

**Palabras Clave:** Web Scraping, Habilidades Digitales, Chatbot, Python, Firebase, Dialogflow.



**ANEXO XIV.- RESUMEN DEL  
TRABAJO DE TITULACIÓN (INGLÉS)  
FACULTAD DE INGENIERÍA INDUSTRIAL  
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



---

**“MODEL BASED ON WEB SCRAPING AND CHATBOT TO LEARN THE MOST  
DEMANDED DIGITAL SKILLS IN THE IT SECTOR”**

**Author:** Vera Vera Dayanna Melina

**Advisor:** SE Castillo León Rosa Elizabeth, Mg.

**Abstract**

This project presents the development of a system model consisting of a chatbot that works with data extracted through web scraping and whose purpose is to facilitate understanding of the digital skills most required by companies that offer jobs in the software development area. In addition to a non-experimental methodology with a qualitative approach, a methodology was also used for the implementation of the model, which is composed of the phases of web scraping, data storage and Natural Language training. During the study, it was confirmed through a survey to 89 students of the Telematics career, that the understanding about the most demanded digital skills is basic. Finally, a chatbot-type tool based on the proposed model was obtained as a result, which facilitates the understanding of the most demanded technological skills and that meets the parameters specified by the students through the survey.

**Key Words:** Web Scraping, Digital Skills, Chatbot, Python, Firebase, Dialogflow.

## **Introducción**

La necesidad de aplicar nuevas tecnologías en prácticamente todas las industrias ha provocado que el campo del desarrollo de software se amplíe enormemente, tal es así que actualmente existen cientos de lenguajes de programación, frameworks, bases de datos, etc., todos con sus características propias y utilizados para diferentes necesidades. Del mismo modo la demanda de profesionales en software ha crecido alrededor de todo el mundo y en Latinoamérica la situación se agudiza debido a la escasez de talento capacitado, por ejemplo, en Ecuador se estimó un déficit de 9000 programadores durante el año 2021.

Por ello el presente proyecto pretende aportar al conocimiento de la demanda laboral en el sector del desarrollo de software, enfocándose en las habilidades o requerimientos más demandados entre las ofertas que se publican en portales de empleo online. Todo esto con el fin de ayudar a los estudiantes a conocer con exactitud las habilidades digitales que más se requieren para obtener un empleo en el sector del software, despertando el interés a capacitarse en dichas habilidades.

Para esto se propone el desarrollo de un chatbot que funcione con una base de datos, la cual se alimentará con data extraída mediante web scraping o raspado web realizado sobre los portales de empleo Computrabajo y LinkedIn, para que de esta forma el chatbot muestre entre otras cosas un resumen de las habilidades más demandadas en base a datos de ofertas laborales reales y vigentes a nivel de Ecuador.

A lo largo del presente documento se encontrarán tres capítulos, el primero que trata acerca del diseño de la investigación en donde se detallan el planteamiento del problema, la justificación del proyecto, su objetivo y su alcance. En el capítulo dos se detalla toda la fundamentación teórica necesaria para el entendimiento de los componentes del modelo a implementar, así como sus implicaciones legales. Finalmente, el capítulo tres contiene los detalles de la metodología empleada para el estudio y la implementación del modelo, así como los resultados obtenidos después de aplicar dichas metodologías.

Entre los resultados obtenidos cabe mencionar el cumplimiento de los objetivos ya que se logró implementar la herramienta propuesta juntamente con todas funcionalidades previstas y las que fueron resultado de la encuesta realizada a 89 estudiantes de la carrera Ingeniería en Telemática de la Universidad de Guayaquil

# **Capítulo I**

## **El Problema**

### **1.1. Problema de investigación**

#### **1.1.1. Planteamiento del problema.**

Actualmente se vive la época de lo que se ha denominado la Cuarta Revolución Industrial o Industria 4.0, los países más desarrollados invierten grandes sumas de dinero en innovación e implementación de tecnologías y en Latinoamérica se evidencia la misma situación, pues según las predicciones de la IDC (2021) para el presente año en curso estimaba un crecimiento de la inversión del 9.4% por parte del sector empresarial, una cifra alentadora para los profesionales en la materia.

Del mismo modo en que crece la inversión, el mercado laboral de TI en Ecuador se vuelve cada año más amplio y competitivo, actualmente existen quince nuevos cargos que hasta hace poco no existían, entre los cuales se encuentran Analista de datos, User experience, Marketing digital, Product owner, etc... esto fue lo que reveló el estudio denominado “El mercado salarial y laboral ecuatoriano en la era de la transformación digital” realizado por Deloitte en el cual participaron alrededor de 200 empresas entre nacionales y extranjeras. Pero a pesar de la creciente demanda de profesionales en tecnologías, el país sigue teniendo una escasez de talento capacitado para estos cargos, para citar un ejemplo en concreto del área de desarrollo de software, la Cámara de Innovación y Tecnologías (2021) declara que el país tiene un déficit de 9.000 programadores al año.

Esta escasez de talento capacitado tiene su origen principalmente en el ámbito educativo universitario, sin embargo también puede atribuirse a otra situación que acontece entre los jóvenes universitarios que a pesar de estudiar carreras relacionadas al desarrollo de software, no conocen con exactitud cuales son las habilidades o conocimientos que las empresas están solicitando, es decir, no conocen los requerimientos del mercado laboral de sus profesiones o aún conociéndolos no se sienten preparados y por ello apuntan hacia trabajos más conocidos y con menos requisitos, como asesores de call center, ventas, entre otros. frente a esta situación la problemática evidente es el desconocimiento de las habilidades o requerimientos que las empresas solicitan de las carreras del área tecnológica, y es precisamente esta problemática la que aborda el presente proyecto de investigación, con el fin de aportar a la divulgación de estas habilidades o conocimientos que un profesional del desarrollo de software debe tener para alcanzar un trabajo de su área en Ecuador

## 1.2. Formulación del problema de investigación.

El desconocimiento de las habilidades digitales más demandadas en el sector del desarrollo de software es un factor influyente en el déficit de profesionales capacitados que se necesita para suplir la creciente demanda de estos.

### 1.2.1. Formulación de hipótesis

La implementación de una herramienta amigable permitirá mejorar el conocimiento de las habilidades tecnológicas que requiere el mercado laboral del desarrollo de software en Ecuador.

## 1.3. Justificación e importancia

Desde 2019 el Ecuador apunta hacia un cambio en la matriz productiva basada en la construcción de una sociedad del conocimiento, y entre sus distintos ejes se encuentra sustitución de bienes y servicios que se importan, por bienes y servicios que ya existen nacionalmente, y entre ellos resalta el campo tecnológico en sus distintas áreas como software, hardware y servicios informáticos. La sustitución mencionada tiene lugar debido a la contratación de profesionales del exterior, como se explicó en el planteamiento del problema, la escasez de talento capacitado en tecnologías también deriva en la búsqueda de profesionales fuera del país, y esto es algo que gubernamentalmente se pretende cambiar, por ello y según la Secretaría Nacional de Planificación y Desarrollo (SENPLADES), la industria tecnológica tiene un papel fundamental en el logro de estos cambios.

SECTORES	INDUSTRIAS
BIENES	1) Alimentos frescos y procesados
	2) Biotecnología (bioquímica y biomedicina)
	3) Confecciones y calzado
	4) Energías renovables
	5) Industria Farmacéutica
	6) Metalmecánica
	7) Petroquímica
	8) Productos forestales de madera
SERVICIOS	9) Servicios ambientales
	10) Tecnología (software, hardware y servicios informáticos)
	11) Vehículos, automotores, carrocerías y partes
	12) Construcción
	13) Transporte y logística
	14) Turismo

Figura 1 Industrias consideradas, según sectores. Fuente: SENPLADES. Elaboración: Equipo Técnico del Observatorio Económico y Social

Es aquí donde toma relevancia la necesidad de ofrecer herramientas que contribuyan al conocimiento del campo laboral TI en Ecuador, puesto que conocer de manera fácil datos reales y actualizados sobre los conocimientos tecnológicos que las empresas están solicitando a través de los portales web de empleos, puede resultar en la orientación que tanto estudiantes de carreras tecnológicas como los bachilleres recién graduados necesitan conocer, además de servir como una herramienta estratégica a la hora de buscar empleo, ya que al tener una idea clara de lo que las empresas demandan, los estudiantes pueden acceder a la autoeducación en dichas habilidades y tener así un perfil más competitivo.

Algo importante a resaltar es el enfoque del presente proyecto hacia las habilidades digitales más demandadas, ya que normalmente el mercado laboral hace énfasis en ofertar cargos con determinado nombre, sin embargo, lo que realmente vale la pena saber son las habilidades o conocimientos que se requieren para ejercer dicho cargo, por ello el modelo que se propone en este trabajo de investigación es el uso de la técnica web scraping (en español llamado raspado web) para la extracción de datos de las vacantes publicadas en bolsas de trabajo online, de modo que el código determine las habilidades digitales que aparecen con mayor frecuencia para presentarlas como las más demandadas, mostrándolas al usuario final a través de un asistente virtual chatbot, con lo cual se asegura que el acceso a estos datos sea fácil para el usuario final. Como complemento para la solución del problema antes mencionado, se propone también la posibilidad de direccionamiento a las vacantes desde el chatbot hacia el portal web para facilitar la postulación del usuario a determinada oferta laboral en la que se crea apto.

## **1.4. Objetivos**

### **1.4.1. Objetivo general.**

Desarrollar un modelo que facilite el conocimiento de las habilidades digitales más demandadas en el área de TI en Ecuador, empleando la técnica de web scraping para la extracción de datos y un chatbot para la presentación interactiva de los mismos.

### **1.4.2. Objetivos específicos.**

- Recopilar información teórica para la conformación de antecedentes que aporten al desarrollo del modelo propuesto
- Diseñar un modelo que contenga y relacione las partes de extracción, almacenamiento y presentación de datos
- Implementar un chatbot en Facebook Messenger sobre una página de prueba

- Evaluar el funcionamiento del modelo.

### **1.5. Alcance**

Lo que se espera entregar del proyecto es el chatbot, el cual servirá como medio para presentar de la información resultante del web scraping.

Para la realización del web scraping se evaluarán sitios web que publiquen ofertas laborales para determinar los dos sitios más favorables para realizar la extracción de datos. En cuanto a la base de datos se determinará la herramienta más adecuada, de modo que facilite la conexión de la data almacenada para el envío de respuestas al chatbot.

Respecto al chatbot, se implementará como un prototipo sobre una página de prueba en Facebook Messenger, el mismo presentará únicamente vacantes en territorio ecuatoriano correspondientes al área de desarrollo de software, por ende, el raspado web también será aplicado sobre ofertas de empleo de esta área.

En cuanto a limitaciones se puede mencionar que el chatbot no realizará el envío de hojas de vida a las ofertas laborales, sino que direccionará al usuario al portal de origen de la vacante, puesto que el fin del proyecto no es ser un buscador de empleo si no una herramienta que facilite el entendimiento del mercado laboral TI con datos reales y actualizados.



## **Capítulo II**

### **Marco Teórico**

#### **3.1 Antecedentes de la investigación**

Gómez (2019) con su trabajo sobre el desarrollo de un asistente chatbot para la búsqueda de ofertas laborales en Sevilla-España, asienta un precedente en el uso de asistentes virtuales en el sector laboral. El objetivo de dicha investigación es implementar una herramienta intuitiva capaz de recolectar toda la información necesaria para crear un perfil del usuario, para así enlazarlo con las ofertas laborales que más le puedan interesar y facilitar la experiencia de buscar empleo. La metodología empleada se basa en fases, las cuales comprenden la extracción de datos, entrenamiento, evaluación y creación de la interfaz de usuario, con lo cual se obtuvieron resultados positivos al lograr que los usuarios encuentren ofertas bien segmentadas según sus competencias y perfil, todo esto en un menor tiempo y con una mayor fluidez.

Otro trabajo para mencionar es el de Tigreros (2022), quien desarrolló una arquitectura que utiliza un chatbot como medio para interactuar con los colaboradores de un hospital, el objetivo de su propuesta fue implementar una herramienta tecnológica que mejore la comunicación entre jefes y empleados de manera que contribuya al fortalecimiento de un mejor clima laboral. La metodología empleada se basó en una investigación exploratoria, utilizando como herramienta de recopilación de datos la encuesta aplicada a una muestra de 139 empleados de distintos roles y cargos dentro del hospital. Para el desarrollo de la herramienta se utilizaron plataformas como Dialogflow, Firebase, Google Sheets y Google Calendar. Luego de hacer las respectivas verificaciones del correcto funcionamiento del chatbot se identificó que la herramienta cumplía con su objetivo de fortalecer la comunicación entre jefes y empleados del hospital, siendo un medio confiable en donde los empleados pueden hacer conocer sus necesidades de forma oportuna y con total libertad.

Del mismo modo se puede evidenciar la eficacia de aplicar asistentes virtuales de tipo chatbot al servicio de comunidades educativas, como lo demuestra el estudio inicial realizado por Múnera (2022) cuyo objetivo fue diseñar una propuesta de asistente virtual que atendiera las dudas de los estudiantes de bibliotecología de la Universidad de Antioquia modalidad virtual. Entre otros trabajos de este tipo se encuentran el realizado por Castro (2021) con su chatbot que resuelve dudas sobre el uso de la plataforma Moodle utilizando

como medio la aplicación Whatsapp y el proyecto de Murgu (2021) para la asistencia de los estudiantes de la Universidad de Madrid implementado sobre Telegram.

Por otra parte, es importante indagar también sobre el estado del arte en lo referente al web scraping, como resultado de la investigación se obtuvo que esta técnica tiene diversas e interesantes aplicaciones tales como búsqueda de productos, ofertas laborales, documentación científica, entre otras que se detallan a continuación.

Huaman (2019) realizó un nuevo modelo para la búsqueda de productos alimenticios utilizando web scraping, su objetivo fue obtener una herramienta que ayude al ciudadano común a buscar productos alimenticios existentes dentro de la tienda web de dos supermercados de Perú, permitiendo así mismo la comparación de precios para que el usuario pueda ahorrar en sus compras. La metodología utilizada fue una investigación de tipo aplicada, de nivel correlacional y descriptivo con un diseño experimental, realizando encuestas a una muestra escogida de 32 consumidores en supermercados de la ciudad de Arequipa. Como resultado del proyecto se obtuvo un asistente virtual que interactúa con el usuario para mostrarle los productos existentes en el stock de dos supermercados conocidos de la ciudad y a través de las encuestas realizadas se conoció la satisfacción de los usuarios por la facilidad y rapidez con la que ahora podían tomar decisiones sobre sus compras.

Otra aplicación que se encontró fue el trabajo realizado por Llorens (2021) en donde desarrolló una aplicación web para obtener de manera automática el estado del arte de un determinado tema de investigación. Con su trabajo buscaba ofrecer a estudiantes y profesionales una aplicación que mediante web scraping realizara una búsqueda automática de documentación sobre la base de datos científica Scopus. Su metodología empleaba diversas tecnologías como Node.js, Angular, MongoDB, Selenium y arquitectura API REST, con las cuales logró desarrollar la herramienta propuesta que realiza el estado del arte y muestra entre otros datos, los años de mayor publicación de trabajos con el tema en cuestión, los documentos más citados, autores con mayor presencia, etc.

Un tercer trabajo que se puede citar es el de Sánchez (2020) quien propuso una plataforma web que permitiera la visualización de habilidades técnicas que el mercado laboral tecnológico requiere para la contratación de nuevos talentos. La metodología empleada fue desing thinking y a lo largo de su proyecto se describe el proceso de la formación del negocio ya que su trabajo lo orientó hacia la generación de ingresos mediante el modelo freemium, apoyándose del web scraping y procesamiento de lenguaje natural (NLP) obtuvo una herramienta que detecta las habilidades que los empleadores demandan con mayor

frecuencia, a la vez que le brinda al usuario sugerencias de fuentes online donde puede adquirir dichas habilidades y ser así más competente.

## 2.2. Marco teórico

### 2.2.1. Web Scraping.

“También conocido como Web harvesting o Web data extraction, es el proceso de rastreo y descarga de información desde sitios web para la extracción de datos no estructurados o poco estructurados a un formato estructurado” (Tadeo, 2015). En otras palabras, es obtener datos desde una o varias páginas web, similar al trabajo que hace un motor de búsqueda en internet. Por lo general el Web Scraping se realiza a través de programas o algoritmos que ejecutan consultas a un servidor web para extraer datos que en principio se encuentran en formato HTML. Este procedimiento es acompañado luego por el tratamiento de dichos datos que finalmente da como resultado información útil para el usuario.

*2.2.1.1. Herramientas para usuarios en general. Existen dos tipos de herramientas para realizar Web Scraping, que se las puede dividir dependiendo del conocimiento que posee el usuario. Así, para usuarios sin conocimientos de programación se tienen las siguientes:*

- a) **Scraper Api:** Herramienta basada en web que ofrece una interfaz agradable a la vista del usuario, permite la extracción del HTML puro de las paginas web, entre sus ventajas se tiene que ayuda a renderizar código escrito en JavaScript, sin embargo, su principal desventaja es la necesidad de implementar librerías para el parseo y descomposición del HTML para obtener solo los datos deseados. Tiene una versión de prueba gratuita con limitación en el numero de consultas a los servidores.
- b) **Octoparse:** ofrece un mejor control sobre el proceso del Web Scraping, además de evitar el captcha debido a su extracción en la nube con rotaciones de IP; trae RegEx incorporado y utiliza una API para guardar los datos directamente en una base de datos. Es compatible con Windows, Mac y tiene su versión gratuita mas dos planes de pago adicional.
- c) **ParseHub:** permite la creación de scrapers y exporta los resultados en formato JSON o csv. Entre sus principales cualidades está la limpieza de texto y HTML previo a la descarga de datos y su almacenado en servidores de forma automática.

- d) **ScrapingBee:** facilita la ejecución de lenguaje JavaScript de las páginas, a la vez que realiza un cambio de proxies por cada consulta. Lo que la diferencia de las anteriores es su API que puede scrapear resultados de búsquedas de Google.

*2.2.1.2. Herramientas para usuarios con conocimientos en programación. Para ciertos casos se requieren herramientas que sean más robustas y más completas en cuanto al tratamiento de datos, para ello se desarrollan programas específicos mayormente en lenguaje Python, aunque también se puede mencionar R, ambos son lenguajes especializados en el tratamiento de grandes volúmenes de datos; a continuación, una lista de las herramientas más populares para realizar Web Scraping.*

- a) **Scrapy:** es un framework que permite la administración de peticiones al servidor y preservación de sesión de usuario. La ventaja de utilizar Scrapy radica en su eficiencia, es decir, extraer datos en mayor cantidad, en menos tiempo y con un menor consumo de recursos. Sin embargo, puede ser poco recomendada para principiantes en la programación debido a su documentación dirigida a desarrolladores con conocimientos previos.
- b) **BeautifulSoup:** “es una biblioteca de Python para extraer datos de archivos HTML y XML” (Richardson, 2017). Resulta fácil de usar debido a su lenguaje simple, extensa documentación y gran apoyo de la comunidad. Entre sus desventajas resalta la necesidad de apoyarse en otras librerías como Request (para realizar solicitudes al servidor) y HTML5lib (para el parseo de los datos obtenidos).
- c) **Selenium:** es un entorno de pruebas de software para aplicaciones basadas en la web, el cual provee de una herramienta de grabar / reproducir para crear pruebas sin usar un lenguaje de scripting. Dentro de Selenium, se encuentra Selenium WebDriver, el cual permite abrir un navegador e interactuar con él mediante comandos, de esta forma, mediante el uso de lenguajes de programación como Python, se pueden realizar scripts que permitan extraer información de diferentes webs de forma automatizada. (Fernández, 2019)

### **2.2.2. Inteligencia artificial.**

Para definirla de un modo sencillo, Rouhiainen (2018) se refiere a la Inteligencia Artificial (IA) como “la habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana”. En términos más técnicos, se la puede definir como los algoritmos que permiten a las computadoras aprender de los datos que recibe para tomar decisiones como lo hace un humano, con la diferencia de no necesitar descanso y con una posibilidad mínima de cometer errores. Aunque su uso se extiende a un sin número de aplicaciones, entre los mas comunes se pueden listar:

- Detección y clasificación de objetos
- Reconocimiento y clasificación de imágenes estáticas
- Procesamiento eficiente y escalable de datos
- Distribución de contenido en redes sociales

### **2.2.3. Aprendizaje automático.**

Se trata de la rama de la inteligencia artificial en la que los ordenadores tienen la capacidad de aprender mediante el uso de algoritmos generales y en base a patrones en el flujo de datos, lo cual también le permite acumular experiencia para la futura toma de decisiones. Así por ejemplo se tienen los filtros de spam de correo electrónico, que son capaces de discriminar entre los correos basura y correos verificados.

**2.2.3.1. Aprendizaje Supervisado.** *El algoritmo utiliza datos que han sido procesados previamente para determinar la forma en que debe ser categorizada la nueva información. Además, en este tipo de aprendizaje se requiere de la intervención humana a la hora de realizar una retroalimentación sobre el funcionamiento del algoritmo, por ejemplo, cuando se le proporciona al programa una cierta cantidad de imágenes para que luego pueda reconocer imágenes similares.*

**2.2.3.2. Aprendizaje No Supervisado.** *En este tipo de aprendizaje no se requieren datos previamente procesados o clasificados para que el programa sepa qué hacer, sino que de forma autónoma encuentra la manera de clasificar los datos, por ende, aquí no se requiere de la intervención humana para su funcionamiento.*

**2.2.3.3. Aprendizaje Por Refuerzo.** *Implica que los algoritmos puedan aprender de la experiencia, en especial cuando se da un “refuerzo positivo” cada vez que acierta. Este tipo de aprendizaje puede compararse con el entrenamiento a las mascotas, en el sentido de que cada vez que aciertan son recompensados con un dulce.*

#### **2.2.4. Aprendizaje profundo.**

Dentro del machine learning se encuentra la subcategoría del Deep Learning o aprendizaje profundo, el cual consiste en entrenar una computadora no con ecuaciones predefinidas que procesan los datos siempre del mismo modo, sino más bien estableciendo parámetros acerca de los mismos para que la computadora aprenda de forma autónoma mediante el reconocimiento de patrones, todo esto basado en muchas capas de procesamiento.

#### **2.2.5. Procesamiento del lenguaje natural (PLN)**

Según Cortez (2009) consiste en la utilización de un lenguaje natural para comunicarnos con la computadora, debiendo ésta entender las oraciones que le sean proporcionadas, 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.

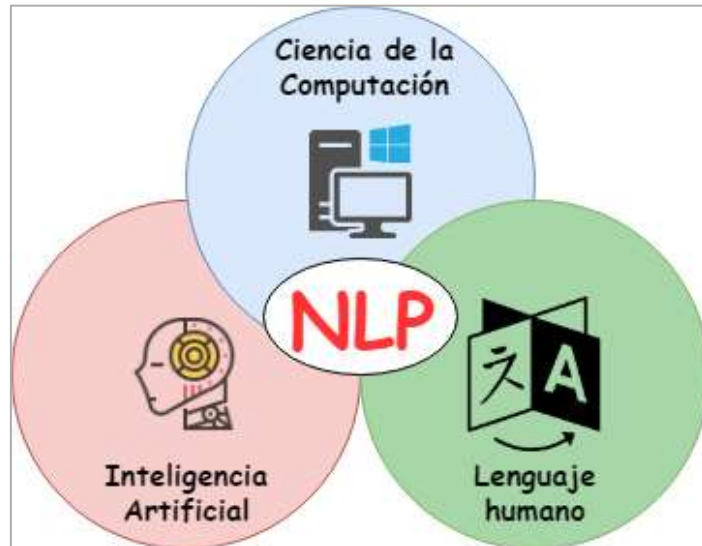


Figura 2 Componentes PLN. Obtenida de <https://www.tecnoblog.org/desarrollo/procesamiento-lenguaje-natural/>. Elaborada por Toledano (2022)

#### 2.2.5.1. Arquitectura de un sistema de PLN

La arquitectura de un sistema de PLN se sustenta en una definición del Lenguaje Natural por niveles (Cortez,2009), estos son:

- a. **Nivel Fonológico:** trata de cómo las palabras se relacionan con los sonidos que representan.
- b. **Nivel Morfológico:** trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas.
- c. **Nivel Sintáctico:** trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración y que sintagmas son parte de otros sintagmas.
- d. **Nivel Semántico:** trata del significado de las palabras y de cómo los significados se unen para dar significado a una oración, también se refiere al significado independiente del contexto, es decir de la oración aislada.
- e. **Nivel Pragmático:** trata de cómo las oraciones se usan en distintas situaciones y de cómo el uso afecta al significado de las oraciones. Se reconoce un subnivel recursivo: discursivo, que trata de cómo el significado de una oración se ve afectado por las oraciones inmediatamente anteriores.

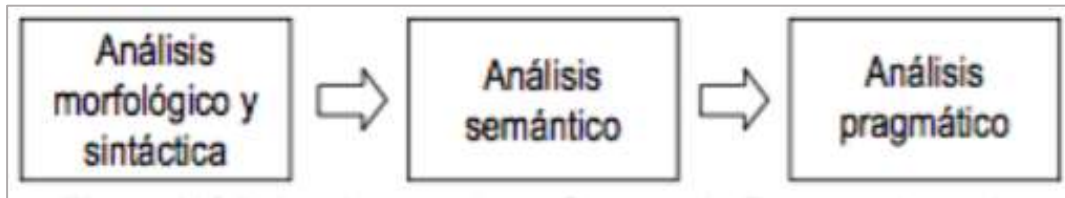


Figura 3 Arquitectura de un sistema PLN. Tomado de <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5923/5121>. Elaborado por Cortez52009)

El esquema presentado muestra el proceso mediante el cual la computadora interpreta y analiza las frases que el usuario proporciona, en una secuencia de pasos que se pueden describir a continuación:

1. El usuario manifiesta su petición a la computadora
2. La computadora analiza las oraciones proporcionadas, en el sentido morfológico y sintáctico, es decir, si las frases contienen palabras compuestas por morfemas y si la estructura de las oraciones es correcta. En esta etapa juegan un papel importante el analizador lexicográfico y el analizador sintáctico. El primero denominado scanner se encarga de identificar los componentes léxicos definidos a priori, el segundo denominado parser se encarga de verificar si se cumple un orden gramatical entre los elementos identificados por el escáner (Glenn,1993)
3. Paso siguiente se analiza semánticamente, determinando el significado de la oración y asignándole una expresión lógica (verdadero o falso)
4. Luego se realiza el análisis pragmático de la instrucción, considerando todas las oraciones que intervienen y determinando un significado y sentido en conjunto. Con este paso, la computadora ha entendido la instrucción y tiene lista una respuesta
5. Lo que resta por hacer es la ejecución de la expresión final, para lograr el resultado que el usuario espera.

#### **2.2.5.2. Sintaxis y gramática**

Se entiende como sintaxis al orden de las palabras dentro de una oración que le dan una relación lógica. En programación, es toda aquella secuencia de símbolos establecida como correcta para que un programa funcione, por ejemplo, la expresión  $r = a + b$  es una expresión con sintaxis correcta mientras que  $r = a \ b +$  tiene sintaxis errónea.



Por su parte, una gramática  $G$  es un modelo lingüístico-matemático que representa el orden en que deben ir las palabras para cumplir con el orden sintáctico establecido. Una gramática se define formalmente de la siguiente forma:

$G = (V_t, V_N, P, S)$  donde:

$V_T$ : conjunto finito de símbolos terminales del lenguaje

$V_N$ : conjunto finito de símbolos no terminales

$P$ : conjunto finito de reglas de producción

$S$ : Símbolo distinguido o axioma inicial a partir del cual se reconocerán las secuencias de  $L$  aplicando sucesivamente las reglas de producción

Considerando la siguiente gramática  $G (V_N, V_T, S, P)$  donde

$V_N = \{S, NOMBRE, VERBO, ADJ, NOMB-SING, VERBO-SING, ADVERBIO\}$

$V_T = \{El, La, Los, Las, Pequeño, traviesa, niño, niña, estudia, corre, juega, salta\}$

$P = \{S \rightarrow NOMBRE VERBO$

$NOMBRE \rightarrow ADJ NOMBRE$

$NOMBRE \rightarrow ADJ NOMB-SING$

$VERBO \rightarrow VERB-SING ADVERBIO$

$ADJ \rightarrow El / La / Los / Las / Ellos / Ellas$

$ADJ \rightarrow Pequeño / traviesa / inquieto$

$NOMB-SING \rightarrow niño / niña$

$VERB-SING \rightarrow estudia / corre / juega / salta$

$ADVERBIO \rightarrow rápidamente / despacio / mucho$

$\}$

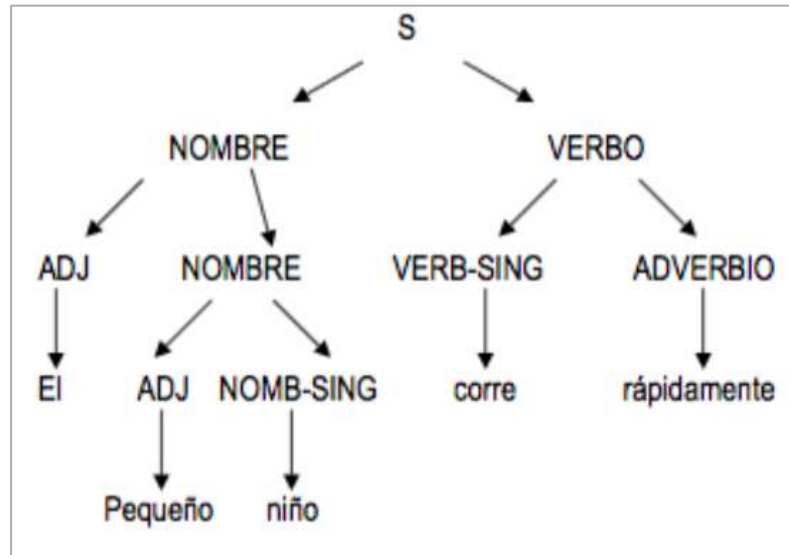


Figura 4 análisis gramatical de una oración. Tomado de <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5923/5121>. Elaborado por Cortez(2009)

Siendo el resultado  $w = \text{'El Pequeño niño corre rápidamente'} \in L(G)$

#### 2.2.6. Agente conversacional (Chatbot)

Un chatbot es definido como el sistema de diálogo entre humanos y computadores que hace uso del PLN (Jia, 2003). Las funciones elementales que componen un chatbot son:

- a) **Dialógica:** debe comprender al usuario, es decir, proporcionar la función de comprensión al recibir una entrada ya sea textual u oral (Jia, 2003), valiéndose de las herramientas de procesamiento de lenguaje.
- b) **Racional:** debe tener acceso a una base externa de conocimiento y sentido común (por ejemplo, a través de corpus de datos) de manera que pueda proporcionar la función de competencia, respondiendo preguntas del usuario. (Jia, 2003)
- c) **Presencia:** debe generar sensación de acompañamiento de una persona real, lo cual es clave para el usuario común. Para esto los desarrolladores se valen de frases típicas del lenguaje natural empleadas a lo largo del chat y la asignación de un nombre propio para el bot.

### 2.2.7. DialogFlow

Dialogflow es una plataforma de comprensión del lenguaje natural de Google que ayuda a los desarrolladores a diseñar interfaces conversacionales e integrarlas en sus aplicaciones. Dialogflow se puede integrar con otros servicios cognitivos, como análisis de sentimientos, servicios de base de conocimiento, etc. Dialogflow también puede aprovechar los servicios cognitivos, de IA y otros disponibles de otros proveedores de la nube a través de la integración basada en API. (Sabharwal, 2020). En palabras simples, es una plataforma que permite la implementación de chatbots basados en PLN.

#### 2.2.7.1. Agentes

Dentro de Dialogflow se conoce como un agente al agente virtual capaz de mantener una conversacion con el usuario final. Técnicamente hablando, un agente es un módulo de comprensión del lenguaje natural que toma las entradas del usuario, ya sean mediante texto o audio, y las traduce en datos estructurados que las aplicaciones o sistemas pueden manejar. Al igual que un ser humano debe ser capacitado para atender a un cliente, un agente necesita ser entrenado para saber cómo responder ante las distintas entradas de un usuario, sin embargo, este entrenamiento no requiere ser tan explícito.

#### 2.2.7.2. Intents

Se puede entender como la intención que el agente debe reconocer ante una determinada expresión de usuario final para saber qué responder, por ello se puede decir que una conversación esta compuesta y clasificada por muchos intents. Dialogflow enlaza la expresion de usuario final y la hace coincidir con el intent que más se relacione a esta. Un intent está compuesto por los siguientes elementos:

Frases de entrenamiento: son las posibles frases que el usuario puede escribir o decir, cuando una *expresion de usuario final* es igual o similar a una de las frases de entrenamiento entonces Dialogflow hace una coincidencia con el intent. En esta lista de frases de entrenamiento basta con escribir las palabras mas representativas, ya que mediante aprendizaje automático el agente puede reconocer muchas frases similares a las de entrenamiento.

Acción: el método que se desencadena cuando ocurre una *coincidencia de intent*, puede definirse una accion por cada intent y estas pueden a su vez desencadenar otras acciones.

Parámetros: pueden establecerse o no para cada intent, dependiendo de las necesidades del sistema, los parámetros son valores específicos que se extraen de la expresión de usuario

final, se identifican por su *tipo de identidad* y son un tipo de dato estructurado que puede ser directamente usado por la lógica del sistema.

Respuestas: el texto, audio o imagen con que responderá el agente, estas pueden solicitar datos al usuario, dar continuidad a la conversación o finalizarla.

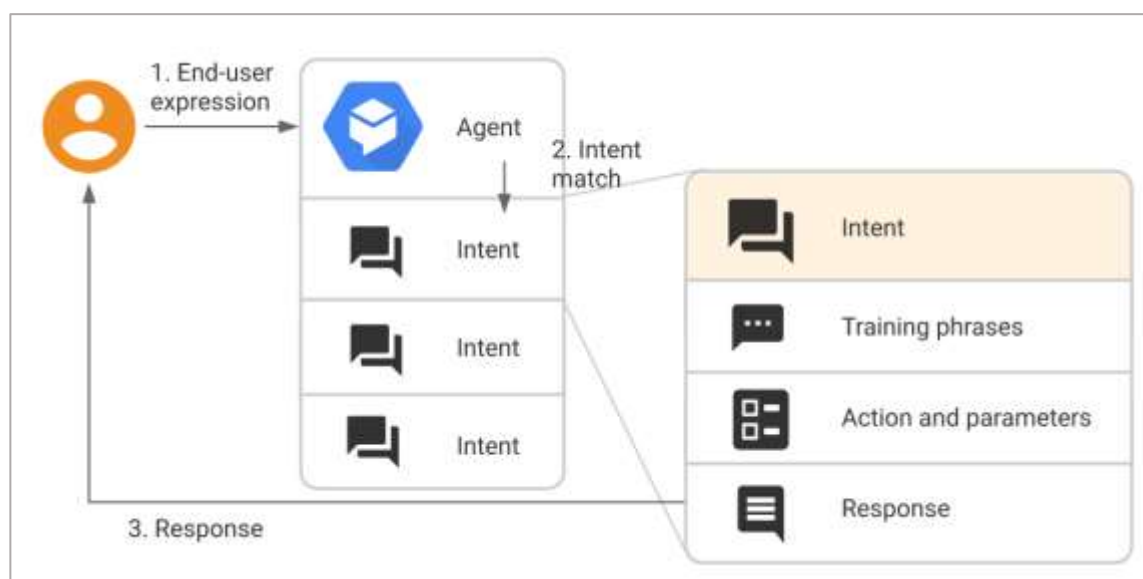


Figura 5 Flujo básico para detectar coincidencias de intents. Tomado de <https://cloud.google.com/dialogflow/es/docs/intents-overview?hl=es-419>. Elaborado por Equipo de Google Cloud (2021)

### 2.2.7.3. Entidades

Están estrechamente relacionadas con los parámetros, las entidades son el tipo de entidad de los parámetros. Se pueden definir entidades personalizadas para identificar algún tipo de dato específico dentro de la expresión de usuario final, por ejemplo, si se quiere definir la entidad *TipoCafe* que coincida con todos los tipos de café tales como cappuccino, americano, express, etc. También existen entidades que Dialogflow tiene predefinidas y que coinciden con algunos de los tipos de datos más reconocidos como fecha, hora, correo electrónico, colores, entre otros.

### 2.2.7.4. Contextos

Al igual que en el lenguaje natural, los contextos en Dialogflow le permiten al agente entender de mejor manera lo que el usuario quiere decir, por ejemplo, en la frase “reserva dos por favor” hace falta tener contexto para entender qué es lo que debe reservar, y posteriormente coincidirlo con el intent correcto. Los contextos son los que dan continuidad lógica a la conversación.

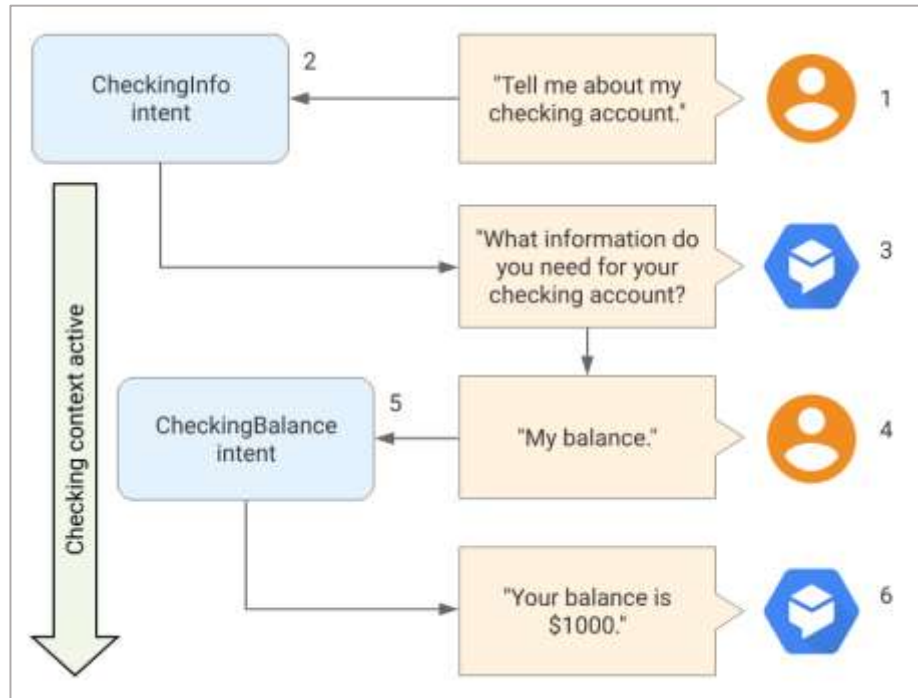


Figura 6 Ejemplo de contexto para un agente bancario. Tomado de <https://cloud.google.com/dialogflow/es/docs/contexts-overview?hl=es-419>. Elaborado por Equipo de Google Cloud (2021)

#### 2.2.7.5. Fulfillment

Es un código implementado como un webhook que va a entrar en acción si la respuesta del webhook falla, o en su defecto si no está habilitado ningún webhook el fulfillment responderá. Este código es el contendrá toda la lógica establecida para cada intent, generando respuestas dinámicas o desencadenando acciones basadas en los datos extraídos del texto ingresado por el usuario. El fulfillment está escrito en java y puede realizar conexiones a bases de datos SQL y NoSQL.

### 2.2.7.6. Integración

Dialogflow permite la integración con diferentes servicios de conversación como Facebook Messenger, en la figura puede verse el modo en que realiza estas integraciones.

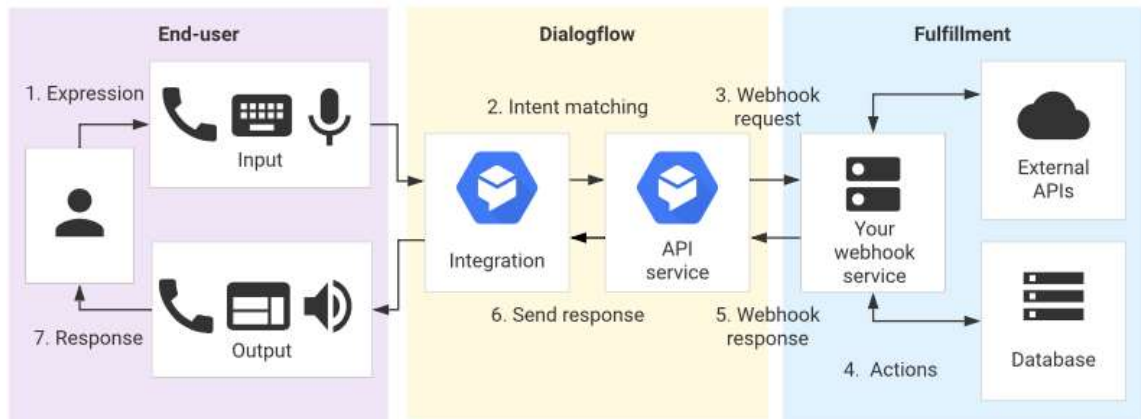


Figura 7 Integración de Dialogflow con otros servicios. Tomado de <https://cloud.google.com/dialogflow/es/docs/contexts-overview?hl=es-419>. Elaborado por Equipo de Google Cloud.(2021)

Se puede observar tres capas, la capa del dispositivo que servirá como interfaz entre el sistema y el usuario, la parte coloreada en verde hace referencia al código y las consultas que realizará a su base de datos, en esta capa se encuentran también los métodos de entrada y salida de datos; posteriormente en la última capa se encuentra la base de datos que puede ser relacional o no relacional.

### 2.2.7.7. Modos de entrada

Para la recepción de datos, Dialogflow admite cadenas de texto, sin embargo si se requiere el ingreso de un archivo de sonido, lo más recomendable es emplear herramientas de la web que conviertan archivos de sonido a un texto

### 2.2.7.8. Modos de salida

Al designar las respuestas que dictará un agente de Dialogflow, estas pueden ser configuradas desde un archivo en formato JSON que contenga los metadatos. Para ofrecer una respuesta de voz es necesario el uso de herramientas de terceros.

## 2.2.8. Facebook Messenger

Es la aplicación de mensajería lanzada en 2008 como parte interna de la aplicación de Facebook, bajo el nombre de Facebook Chat. A partir de 2010 se lanzó al mercado como

una aplicación independiente con el nombre actual de Facebook Messenger, que ofrece posibilidades de realizar videollamadas grupales, llamadas de voz, mensajes de voz, texto, imágenes, envío de documentos, entre otras muchas opciones por explorar. En 2017 añadió al entorno los llamados “Messenger Day”, que son publicaciones en formato historias donde el usuario puede compartir fotos y videos que desaparecen luego de 24 horas de publicado. En el año 2015 implementa opciones que son de interés para las cuentas que simbolizan un canal de atención al cliente, opciones tales como seguimiento de compras, lectura de notificaciones y la posibilidad de integrar aplicaciones dentro del propio Facebook Messenger. Ya en 2016, lanzó una API para desarrolladores que permite la implementación de Chatbots en Messenger.

#### **2.2.8.1. Facebook Messenger bot**

Facebook Messenger es una de las plataformas de mensajería instantánea que permite la integración de bots con el objetivo de promover la interacción de las empresas o startups con sus clientes. De este modo un bot en Facebook Messenger es un asistente virtual que facilita principalmente el servicio al cliente de las empresas, ya que al tratarse de un bot es posible mantener ese servicio las 24 horas del día y así generar mayor cercanía al cliente.

#### **2.2.9. Firebase**

Se trata de una plataforma en la nube que ofrece un conjunto de herramientas que hacen posible desarrollar aplicaciones web y móviles, en menor tiempo, pero con igual calidad. Presenta compatibilidad con Android, iOS y web. La gama de productos que ofrece Firebase satisface distintas necesidades de un proyecto de desarrollo de software, así se distingue entre las herramientas para el desarrollo, crecimiento, monetización y análisis.

**2.2.9.1. Cloud Firestore.** *Es el producto mas reciente del catálogo de Firebase, se trata de una base de datos no SQL que presenta grandes ventajas en contraposición a Realtime Database, entre las principales se encuentran las siguientes:*

- **Almacenado de datos en forma de colecciones:** esto representa un mayor beneficio a medida que la base de datos va creciendo en cantidad y en jerarquía.
- **Realizar consultas:** Firestore permite la ejecución de consultas compuestas con ordenamiento y filtrado, a diferencia de Realtime Database que solo admite consultas bien sea con ordenamiento o con filtrado, pero no con ambas en ambos lados.

- **Multiregional:** aloja datos en los servidores más cercanos a la zona en que se quiere levantar la aplicación
- **Escalamiento automático:** con un límite de 1 millón de colecciones simultáneas.
- **Precio:** se determina según la cantidad de consultas a la base de datos que se realice, entre las consultas mas tradicionales se tienen las de lectura, escritura y eliminación de registros.

### 2.2.10. Webhooks

Se trata de una devolución de llamada HTTP, un evento tipo POST que se envía a una Url y le notifica eventos sucedidos en la aplicación emisora, caracterizado por ser en tiempo real. En términos simples, un webhook sirve como un notificador que comunica un determinado evento sucedido a otra aplicación en la que se encuentra su punto final de Webhook. Esta comunicación corta transmite datos en formato JSON o XML, los cuales pueden ser usados por el usuario según crea conveniente dentro de su aplicación. Entre sus principales usos se tiene:

- Sincronizar datos automaticamente entre aplicaciones, por ejemplo, mantener sincronizada la aplicación móvil de un banco para que el sistema web actualice el saldo de un usuario en el mismo instante en que realiza una transacción.
- Actualizar funcionalidades de una aplicación dependiendo de la necesidad del usuario
- Conectar aplicaciones para que cierto evento sucedido en una desencadene acciones puntuales en otra.

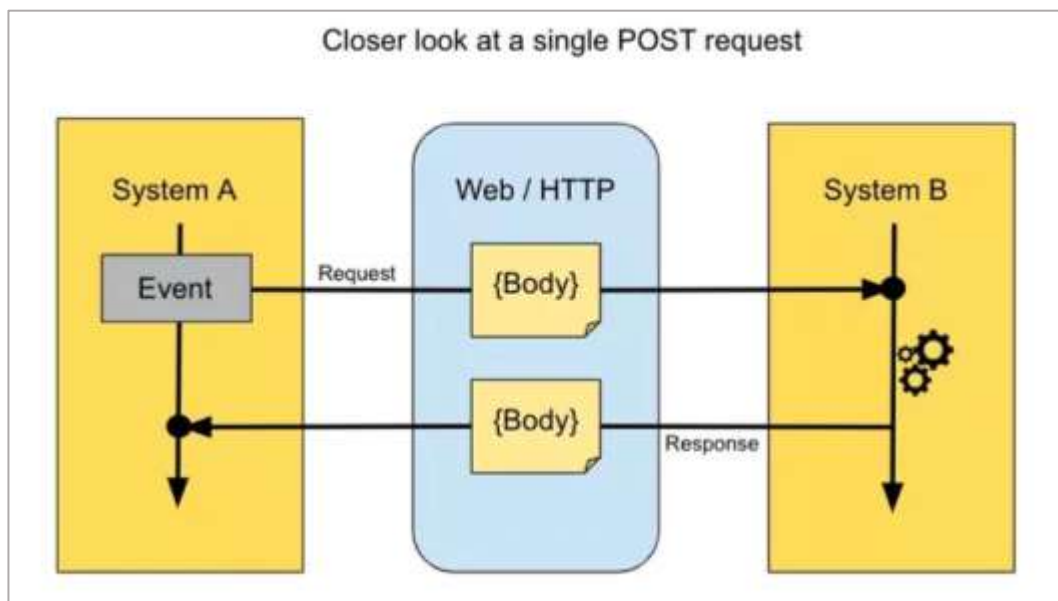




Figura 8 Webhooks. Tomado de <https://www.mdirector.com/blog/que-es-un-webhook/>.

### 2.2.11. Habilidades digitales en desarrollo de software

Habilidad es la capacidad de llevar a cabo una acción o tarea con destreza, se diferencia de la aptitud ya que las aptitudes son naturales en una persona, mientras que las habilidades requieren de conocimiento y se aprenden a lo largo del tiempo.

Es así como se puede hablar de una amplia gama de habilidades digitales en el campo del desarrollo de software, los cuales pueden ser agrupados como sigue a continuación:

#### 2.2.11.1. Lenguajes de programación.

Un lenguaje de programación es el conjunto de instrucciones que tienen una sintaxis y que permiten escribir nuevos programas informáticos. Para citar los lenguajes de programación más populares se tienen los resultados de dos encuestas realizadas por el reconocido sitio Stack Over Flow y PYPL Index respectivamente.

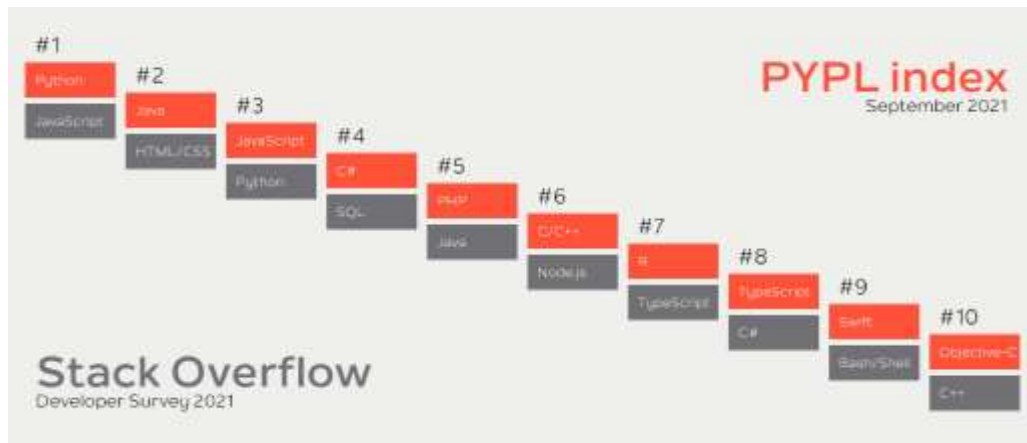


Figura 9 Lenguajes de programación más populares. Tomado de <https://www.stackoverflow.com/es/blog/lenguajes-programacion-populares-2021/>. Elaborado por Stack Overflow(2021)

Entre los lenguajes menos conocidos se tiene Swift, Kotlin, Ruby, Perl, Go, Elixir, entre otros.

#### 2.2.11.2. Frameworks.

Son un marco de trabajo que estandariza conceptos, prácticas y criterios para contribuir al desarrollo ágil de grandes aplicaciones evitando tener que hacerlas desde cero. Estos frameworks ofrecen una especie de plantilla que sirve como punto

de partida. Entre los frameworks más reconocidos están: Symfony, Laravel, CakePHP, CodeIgniter

#### ***2.2.11.3. Bases de datos.***

Se denomina base de datos a un conjunto de datos dispuesto con el objetivo de proporcionar información a los usuarios y permitir transacciones como inserción, eliminación y actualización de datos (Arias, 2014).

Las bases de datos constituyen un elemento esencial para los sistemas de las empresas, por tanto, el trabajo de un desarrollador además de poder realizar consultas a la base de datos, es también mantenerlas protegidas con buenas prácticas en el desarrollo del código y con copias de seguridad. La base de datos más utilizada es SQL, seguida de otras como Oracle, PostgreSQL, MongoDB, Cassandra, Redis, IBM DB2, Microsoft SQL Server, Riak.

#### ***2.2.11.4. Control de código fuente.***

Se refiere al control de versionamiento que se realiza sobre el código de un sistema o programa, lo cual ayuda a la administración, almacenamiento y trabajo en equipo sobre un mismo código simultáneamente. La herramienta por excelencia para el control de versiones es Git, pero el abanico de opciones se extiende a CVS, Apache Subversion (SVN), Mercurial, Monotone, Bazaar, TFS, Helix Core, etc.

#### ***2.2.11.5. Entornos de Desarrollo Integrado (IDE).***

Son programas robustos que facilitan el diseño y desarrollo de aplicaciones, combinando herramientas que son comúnmente utilizadas por los desarrolladores, todas ellas en una sola interfaz de usuario. Existen varios entornos de desarrollo tales como Eclipse, Android Studio, NetBeans, IntelliJ Idea, Xcode, BlueJ, JetBrains, PyCharm, Rstudio, etc.

### **2.3. Marco conceptual**

#### **2.3.1. Web Scraping**

Técnica para la extracción de datos desde sitios web, usualmente consiste en un código de programación que realiza una búsqueda automática en la World Wide Web y extrae data identificando los elementos mediante el código HTML del sitio.

### **2.3.2. Selenium**

Es un framework o entorno de trabajo cuyo fin es realizar pruebas automatizadas de navegación a través de sitios web, permite simular las acciones que un usuario común realiza como dar click en botones, llenar campos de un formulario, hacer zoom, desplazarse hacia abajo, etc.

### **2.3.3. Chatbot**

Software capaz de mantener una conversación en lenguaje natural con una persona, gracias a que cuenta con respuestas automáticas previamente definidas por el desarrollador. Son ampliamente utilizados en atención al cliente ya que constituyen un asistente capaz de brindar información las 24 horas del día.

### **2.3.4. Dialogflow**

Plataforma de comprensión y procesamiento del lenguaje natural que permite el desarrollo de interfaces conversacionales sobre aplicaciones móviles y web, muy reconocido por su facilidad para implementar bots integrados en redes sociales.

### **2.3.5. Firestore**

Base de datos NoSQL, es decir no relacional; que funciona basada en la nube y puede comunicarse con aplicaciones iOS, Android y Web, facilitando la ejecución de consultas a gran escala.

### **2.3.6. Meta Developers**

Plataforma proporcionada por la empresa Meta Inc. (anteriormente Facebook Inc.) que permite el desarrollo de aplicaciones dentro de las redes sociales perteneciente a la propia compañía. Incluye aplicaciones que utilizan Inteligencia Artificial (IA).

### **2.3.7. Habilidades digitales**

Aquellos conocimientos en programación que se requieren de un perfil de desarrollador de software, incluyen lenguajes de programación, bases de datos, frameworks, herramientas de control de versiones, entornos de desarrollo, entre otros. Ejemplo: Python, C++, .NET, SQL, Django, etc.

### **2.3.8. Webhook**

Sistema de comunicación automático que envía notificaciones a otras aplicaciones a través de solicitudes HTTP. Sirven principalmente para notificar un evento sucedido en un sistema a otro, se caracteriza por ser en tiempo real.

### 2.3.9. Requerimientos funcionales

Son una especificación de lo que debe ser implementado. Son descripciones de cómo el sistema debe comportarse, o de un sistema de propiedad o atributo. Ellos pueden ser una limitación en el proceso de desarrollo del sistema (Sommerville, 2009)

Se refiere a toda funcionalidad requerida por el usuario y que conforma la razón de ser del proyecto o sistema a implementar.

### 2.3.10. Requerimientos no funcionales

Todos aquellos requerimientos que no son realizados por el usuario pero que están implícitos en el cumplimiento de los requerimientos funcionales, los requerimientos no funcionales son establecidos y previstos por un arquitecto del software o un entendido en el campo del desarrollo de software.

### 2.3.11. Xpath

Es una expresión regular que permite identificar un elemento dentro de una estructura XML, haciendo referencia a toda la jerarquía que se recorre hasta llegar al elemento en cuestión.

## 2.4. Marco legal

### 2.4.1. Web Scraping

Por tratarse de la extracción y manipulación de datos, el Web Scraping se ve relacionado de alguna forma con el campo legal. A pesar de que no existe una ley que aborde el web scraping directamente, se puede hacer referencia a un marco legal en el sentido del cumplimiento de las leyes y teorías jurídicas relacionadas al manejo de los datos, es así como se puede hablar de una serie de principios que se deben respetar a la hora de realizar Web Scraping, entre estos se tiene:

- a) **Términos de uso:** el propietario de un sitio web puede prohibir el acceso programático a su sitio si lo así lo declara en la política de “términos de uso” publicada en el sitio web. Legalmente, incumplir con estos términos implica un “incumplimiento de contrato” por parte del usuario, sin embargo, para poder iniciar un juicio por violar los “términos de uso”, debe existir primeramente un acuerdo por parte del usuario, que por lo general se lo registra solicitando marcar una casilla de verificación al finalizar la lectura de la política. Por lo tanto, prohibir el Web Scraping de forma explícita en los “términos de uso” no es

suficiente para considerar un escrapear como ilegal ni emprender acciones legales contra el scraper.

- b) Material con copyright:** Extraer y volver a publicar datos o información que es propiedad y tiene derechos de autor explícitos del propietario del sitio web puede conducir a un caso de “infracción de derechos de autor” (Dryer, 2013). Sin embargo, la data generada en un sitio web no es pertenencia del propietario del sitio web, por ejemplo, cuando se tiene una página de reseñas de empresas, los comentarios escritos por los usuarios no son propiedad del dueño del sitio web.
- c) Propósito del web scraping:** como bien se dijo antes, el Web Scraping es legal en todo aspecto, mas cuando a los datos se les da un uso fraudulento, entonces puede tener implicaciones legales, entendiéndose como uso fraudulento a todo aquel que infrinja las garantías establecidas en la vigente Ley Organica de Protección de Datos Personales en Ecuador.
- d) Daño al sitio web:** si la práctica del Web Scraping deriva en una sobrecarga, inhabilitación u otro daño al sitio web, entonces la persona responsable puede ser procesada bajo lo dictado en la Ley Orgánica Integral Penal artículo 232, donde se menciona:

“La persona que destruya, dañe, borre, deteriore, altere, suspenda, trabe, cause mal funcionamiento, comportamiento no deseado o suprima datos informáticos, mensajes de correo electrónico, de sistemas de tratamiento de información, telemático o de telecomunicaciones a todo o partes de sus componentes lógicos que lo rigen, será sancionada con pena privativa de libertad de tres a cinco años.”

### **Capítulo III**

#### **Diseño de la investigación**

La presente investigación tendrá un diseño no experimental, debido a que no se hará manipulación de la variable independiente, sino que se limitará a la observación de su comportamiento en el flujo normal de su ejecución. Por consiguiente, no se establecerá ninguna relación de causalidad entre la variable independiente chatbot y la variable dependiente conocimiento del mercado laboral, pero sí tendrán una relación de crecimiento directa.

#### **2.5.Enfoque de la investigación**

##### **2.5.1. Enfoque cualitativo**

Se aplicará la metodología cualitativa ya que se pretende estudiar a las personas y su grado de conocimiento acerca de las habilidades tecnológicas más demandadas en el campo laboral del desarrollo de software, entre otros aspectos como su punto de vista acerca de la propuesta de un chatbot que funciona con Web Scraping, de tal forma que se puedan identificar y recolectar nuevas funcionalidades para ser aplicadas sobre el modelo a desarrollar en este trabajo de tesis, para que así cumpla con su finalidad de contribuir a la difusión de la oferta laboral para desarrolladores de software en Ecuador.

#### **2.6.Operacionalización de variables**

En base a la formulación de la hipótesis se tienen las siguientes variables:

Variable independiente: asistente virtual chatbot.

Variable dependiente: conocimiento del mercado laboral del desarrollo de software.

#### **2.7.Técnica de recolección de datos**

La técnica utilizada será la encuesta y el instrumento a usar será el cuestionario, con ello se busca recolectar información provechosa para la implementación de funcionalidades que no hayan sido previstas antes y que puedan servir para hacer del modelo una herramienta más útil y eficaz.

#### **2.8.Población**

El estudio se realizará sobre el grupo de estudiantes pertenecientes a la carrera de Ingeniería en Telemática de la facultad de Ingeniería Industrial, Universidad de Guayaquil,

específicamente en los estudiantes de séptimo, octavo y noveno semestre, teniendo así una población finita de 177 personas.

## 2.9.Muestra

Dado que la población objetivo es finita, se debe aplicar la siguiente fórmula:

$$n = \frac{N * Z_{\alpha}^2 * p * q}{e^2 * (N - 1) + Z_{\alpha}^2 * p * q}$$

Donde:

n = Tamaño de muestra buscado

N = Tamaño de la población

Z = Nivel de confianza

e = Error de estimación máximo aceptado

p = probabilidad de éxito

q = (1-p) probabilidad de fracaso

Así se tiene:

$$n = \frac{167 * (1.645)^2 * 50 * 50}{(6)^2 * (177 - 1) + (1.645)^2 * 50 * 50}$$

$$n = 88.42 \cong 89 \text{ personas}$$

## 2.10. Desarrollo del cuestionario

Esta serie de preguntas tendrán como objetivo recolectar información que fortalezca el prototipo a implementar, receptando sugerencias sobre funcionalidades para el chatbot e indicadores que aporten a la construcción de este. La estructura bajo la cual se elaboró la encuesta se la puede ver a detalle en el Anexo 1.

## 2.11. Interpretación y análisis de datos

### Pregunta 1: Qué semestre está cursando actualmente?

En esta pregunta introductorio se establece un perfil del encuestado para determinar el su nivel de estudios. Así se tiene que, el 47.2 % de los que respondieron a la encuesta pertenecen al 9no semestre, el cual es el último semestre de estudios aperturado hasta ahora en la carrera de Telemática. El resto de las respuestas se distribuyen en 28 estudiantes de 7mo semestre y 19 de 8vo.

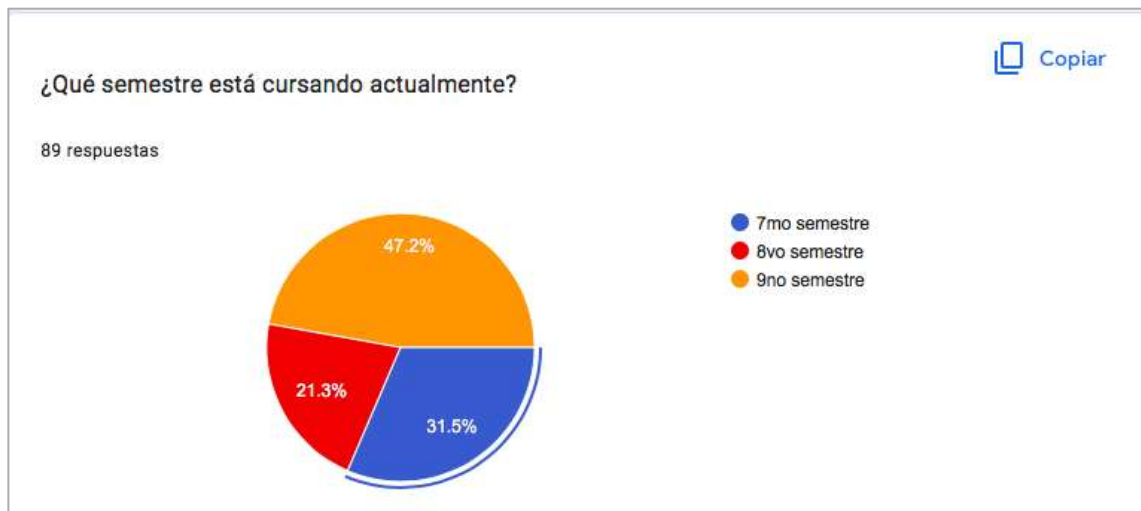


Figura 10 Gráfico estadístico pregunta 1: ¿Qué semestre está cursando actualmente?. Elaborado por Dayanna Vera (2022)

**Pregunta 2: ¿Cuánto conoce acerca de las tecnologías para el desarrollo de software más solicitadas por las empresas en Ecuador actualmente?**

48 de los encuestados afirma tener un conocimiento promedio acerca los cargos que existen para profesionales en el desarrollo de software que actualmente solicitan las empresas, mediante el reporte resultante de esta encuesta generado en Excel se observó que de este grupo de 48 personas 12 pertenecen al 7mo semestre, 9 son de 8vo semestre y 27 pertenecen al 9no semestre, por otra parte, un 40,5% conforman los estudiantes que afirman conocer poco o nada acerca de los cargos de su rama. Esto evidencia una desorientación entre los estudiantes de la carrera de Telemática y ratifica la necesidad de ejecutar un proyecto como este.



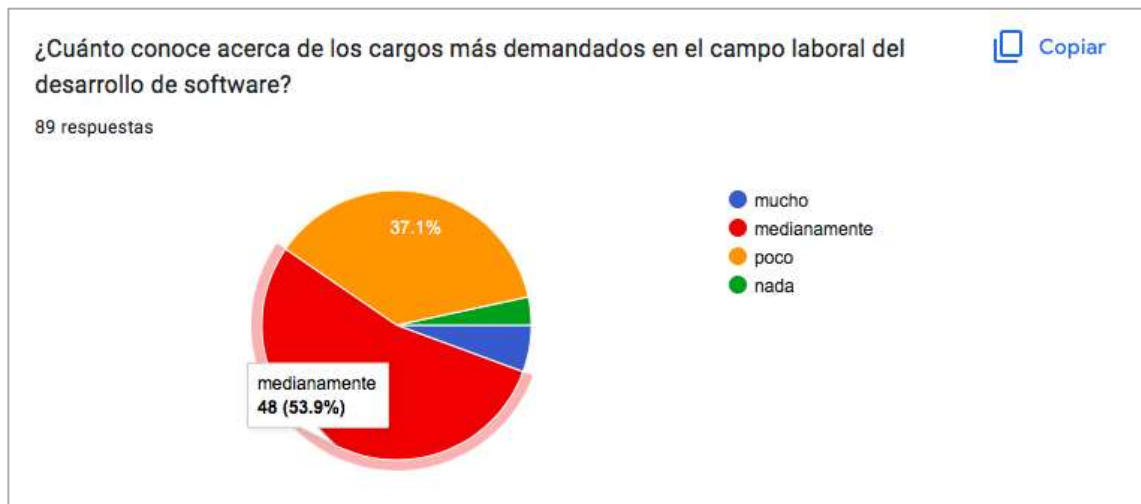


Figura 11 Gráfico estadístico Pregunta 2: ¿Cuánto conoce acerca de las tecnologías para el desarrollo de software más solicitadas por las empresas en Ecuador actualmente?. Elaborado por Dayanna Vera (2022).

### Pregunta 3: Seleccione los cargos que conoce o ha escuchado anteriormente:

El cargo con el que están más familiarizados los estudiantes es el de Programador Web (85,4%), seguido de Ing. de Desarrollo de Software con un 68,5% y Programador de Apps Móviles con un 62,9%. Como se observa, los estudiantes tienen mayor conocimiento de los cargos más generales y conocen poco acerca de otros cargos más especializados como los son Programador Front End, Back End y Full Stack. Este dato resulta importante para el desarrollo de este proyecto ya que se los cargos más conocidos se pueden considerar para la función de buscar ofertas en el chatbot, de tal forma que dichas ofertas se puedan filtrar según los cargos mas conocidos por los estudiantes.

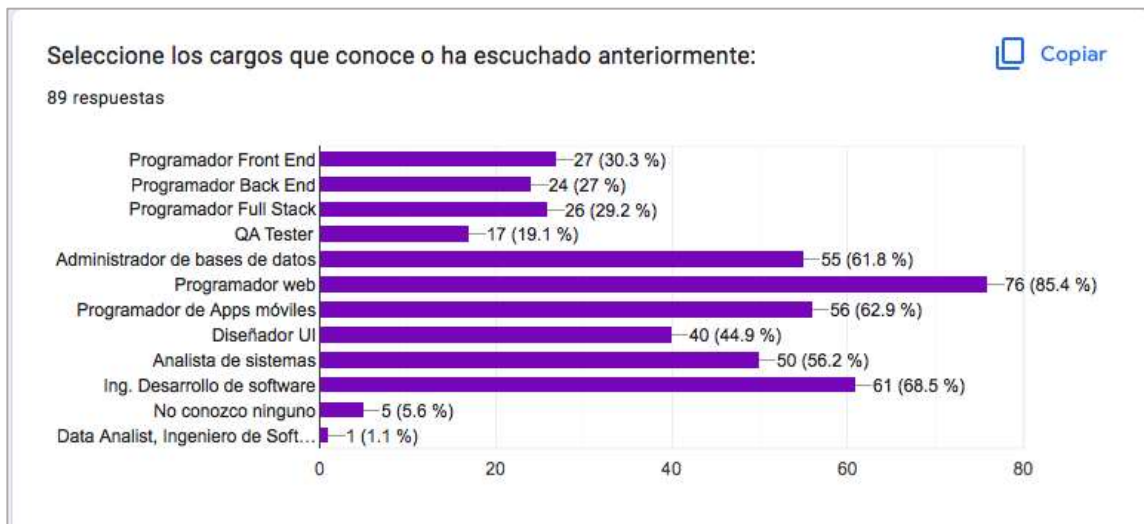


Figura 12 Gráfico estadístico Pregunta 3: Seleccione los cargos que conoce o ha escuchado anteriormente. Elaborado Dayanna Vera (2022).

**Pregunta 4: ¿Cuánto conoce acerca de las tecnologías para el desarrollo de software más demandadas por las empresas en Ecuador actualmente?**

Como se observa, el conocimiento de las tecnologías más demandadas es promedio, ya que el 51,7% dice conocer medianamente acerca de este tema, un 36% de los encuestados afirma conocer poco e incluso un 6,7% dice no conocer nada sobre estas tecnologías. Esto fortalece la necesidad de implementar este proyecto ya que resultaría de gran ayuda para este grupo de personas.

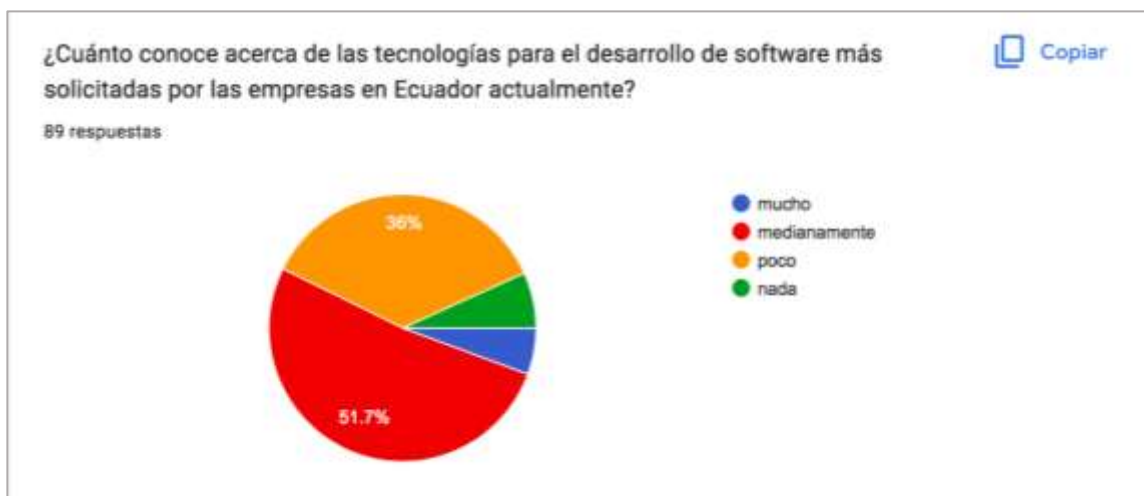


Figura 13 Gráfico estadístico pregunta 4: ¿Cuánto conoce acerca de las tecnologías para el desarrollo de software más solicitadas por las empresas en Ecuador actualmente?. Elaborado Dayanna Vera (2022).

**Pregunta 5: ¿Ha interactuado alguna vez con un chatbot?**

Dado que el 83% de la muestra encuestada conoce lo que es un chatbot y ha interactuado al menos una vez con él, se puede determinar que la herramienta propuesta en este proyecto puede ser aceptada fácilmente por los usuarios.

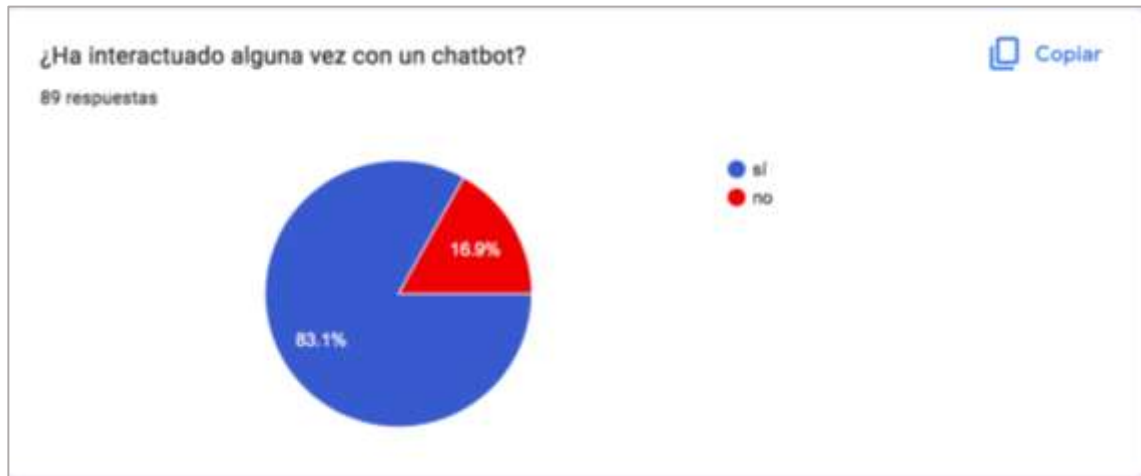


Figura 14 Gráfico estadístico Pregunta 5: ¿Ha interactuado alguna vez con un chatbot?. Elaborado por Dayanna Vera (2022).

#### Pregunta 6: ¿Cómo le resulta más cómodo acceder a un chatbot?

Las opciones más elegidas fueron mediante redes sociales y mediante plataformas web, siendo el 41,6% que prefieren las redes sociales como medio para interactuar de manera más cómoda con un chatbot, esto contribuye a continuar con la propuesta de implementar el chatbot en la plataforma de Messenger.

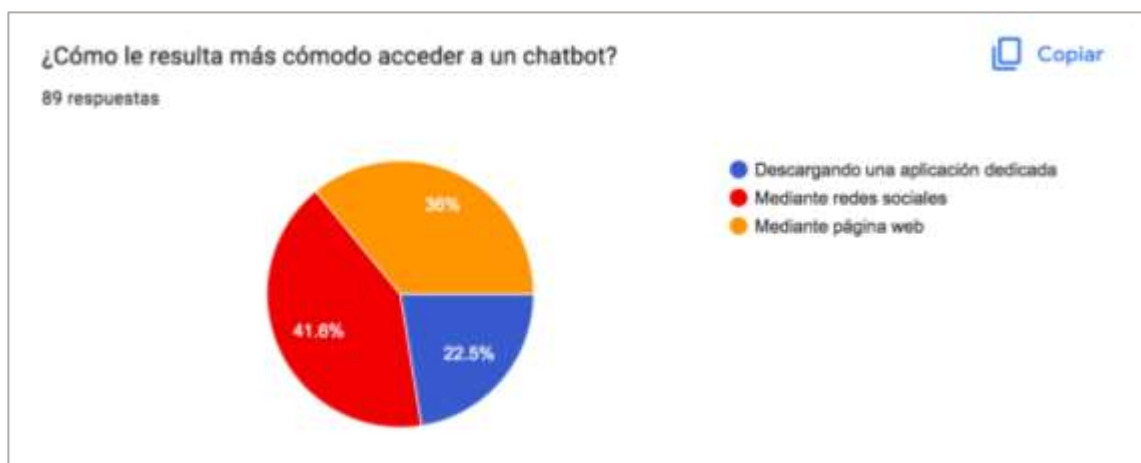


Figura 15 Gráfico estadístico Pregunta 6: ¿Cómo le resulta más cómodo acceder a un chatbot?. Elaborado por Dayanna Vera (2022).

**Pregunta 7: ¿Cuánto tiempo estaría dispuesto a dedicar para interactuar con un chatbot informativo?**

Los resultados demuestran que los estudiantes solo estarían dispuestos a interactuar con un chatbot entre 1 a 5 minutos y como máximo un total de 10 min continuos. Esto nos da una pauta para definir en el chatbot un flujo de conversación corto pero preciso, en el cual los usuarios puedan interactuar con todas las funciones y recibir la información deseada en un tiempo promedio de 5 minutos.

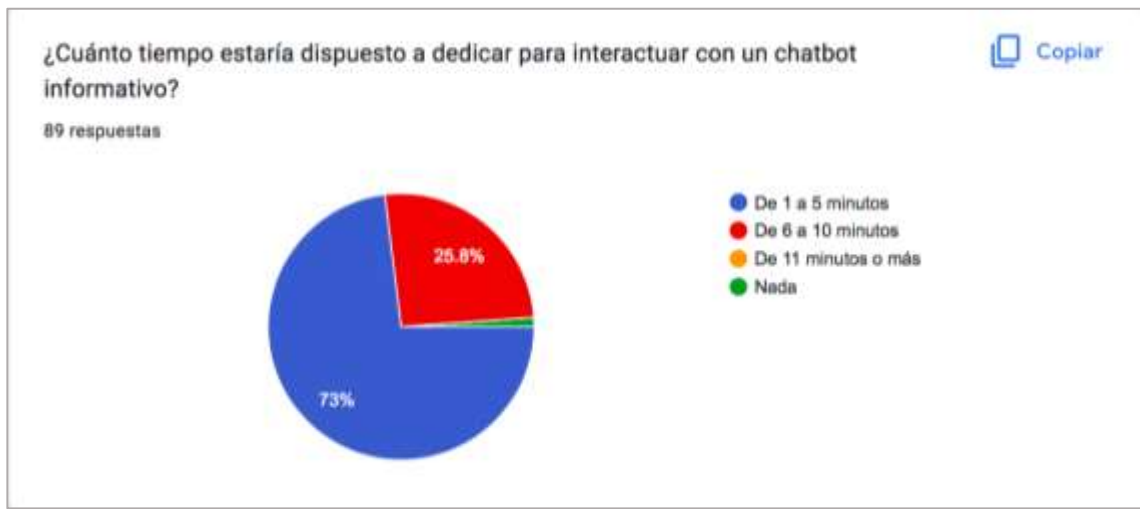


Figura 16 Gráfico estadístico Pregunta 7: ¿Cuánto tiempo estaría dispuesto a dedicar para interactuar con un chatbot informativo?. Elaborado por Dayanna Vera (2022).

**Pregunta 8: ¿Cómo preferiría que fueran las respuestas de un chatbot?**

En cuanto a la interfaz de presentación, el 81,3% de los encuestados manifiesta que preferiría que un chatbot le muestre mensajes que no solo incluyan texto sino también imágenes, esto es importante para el desarrollo del modelo ya que al momento de diseñar las interfaces se debe procurar que la información se manifieste también a través de imágenes.

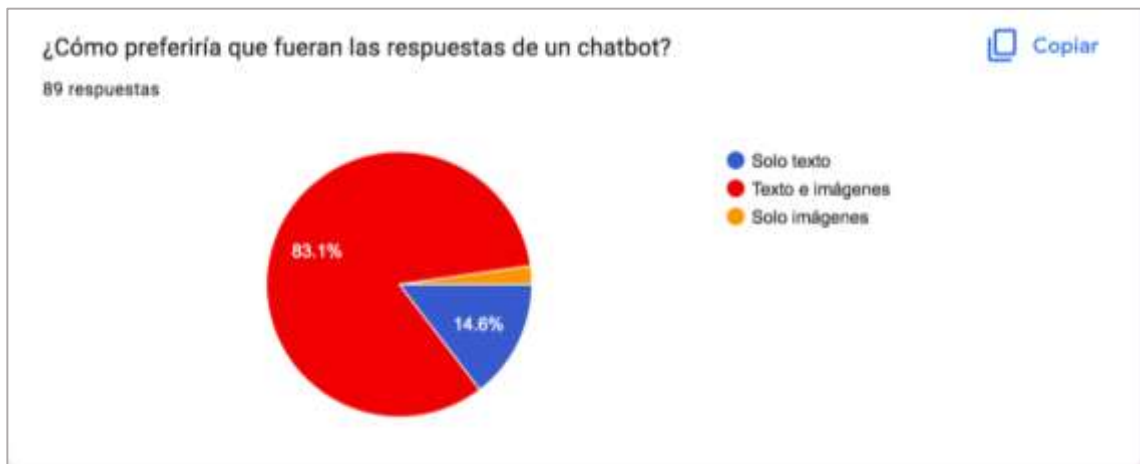


Figura 17 Gráfico estadístico Pregunta 8: ¿Cómo preferiría que fueran las respuestas de un chatbot?. Elaborado por Dayanna Vera (2022).

**Pregunta 9: En cuanto al contenido de las respuestas de un chatbot, preferiría:**

El 64% de las personas preferiría que el chatbot le muestre información con detalles del tema consultado, esto indica que los usuarios prefieren información de calidad, aunque esto implique la presentación de mensajes con mayor cantidad de texto. Esto es importante para el desarrollo del modelo ya que al definir qué información sobre ofertas laborales se va a mostrar debe ser lo mas especifica posible, sin descuidar el aspecto del tiempo que los usuarios están dispuestos a dedicar al chatbot.

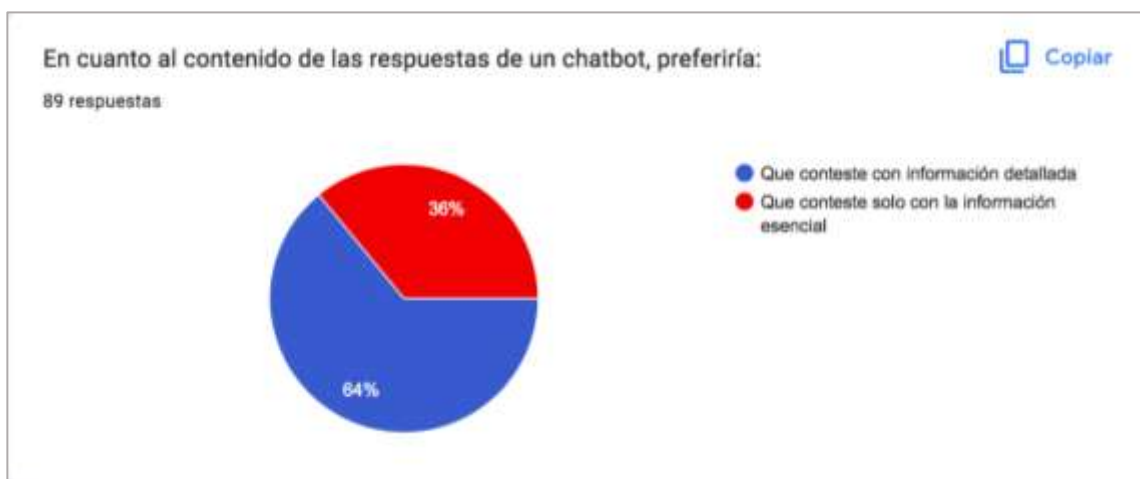


Figura 18 Gráfico estadístico Pregunta 9: En cuanto al contenido de las respuestas de un chatbot, preferiría. Elaborado por Dayanna Vera (2022).

**Pregunta 10: ¿Qué acciones le gustaría que realizara el chatbot para que su conocimiento acerca del mercado laboral del desarrollo de software mejore?**

Entre las funcionalidades más requeridas por los futuros usuarios se obtuvo que para el 74,2% de ellos es muy importante saber la cantidad total de ofertas laborales publicadas en los portales de empleo en cuestión, así mismo al 59,6% les interesa que muestre las tecnologías que más demandan las empresas en Ecuador y como tercer requerimiento más importante está la función de buscar ofertas según una tecnología específica, con un 55,1% de los resultados. Otras funcionalidades a la par son redireccionar a la pagina original de la empresa y brindar el significado de una tecnología desconocida.

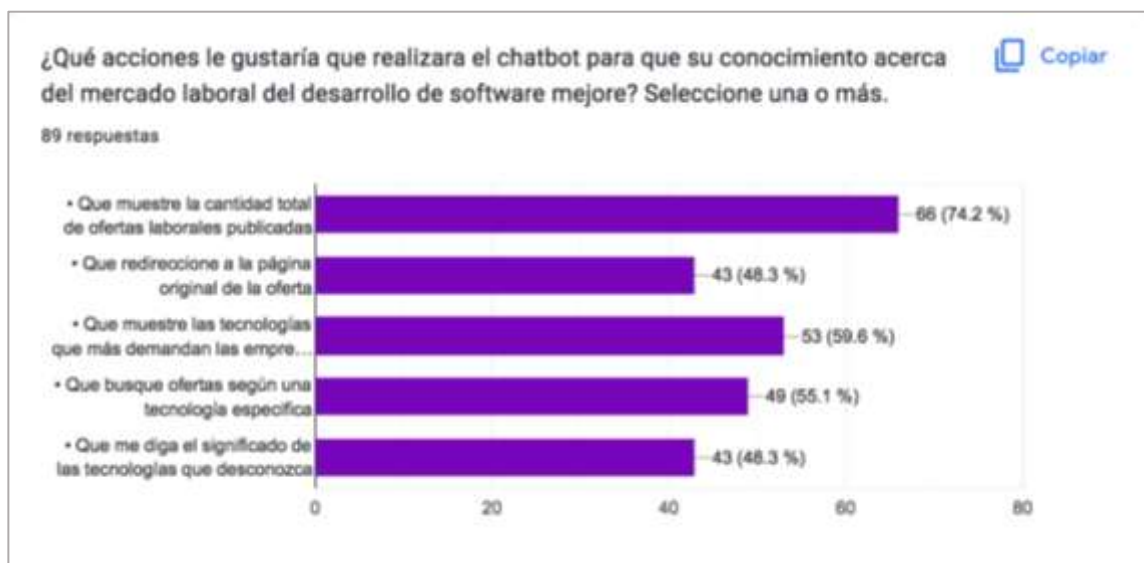


Figura 19 Gráfico estadístico Pregunta 10: ¿Qué acciones le gustaría que realizara el chatbot para que su conocimiento acerca del mercado laboral del desarrollo de software mejore?. Elaborado por Dayanna Vera (2022).

### Analisis general

Con la encuesta realizada se pudo observar que más del 50% de los estudiantes tienen un conocimiento medio acerca de las habilidades digitales más demandadas en sector del desarrollo de software, mientras que en promedio un 36.55% afirma que tiene poco conocimiento acerca de las mismas. Por otra parte los resultados manifiestan que el 83,1% de los encuestados ha interactuado con un chatbot anteriormente, así mismo se obtuvo que una gran mayoría prefiere ser atendido por un chatbot mediante redes sociales y el tiempo que le dedicarían es un máximo de 5 minutos, desean que las respuestas proporcionen información detallada y que a la vez incluya imágenes. Por último los futuros usuarios

desean que el chatbot muestre principalmente la cantidad de ofertas totales encontradas y las habilidades o requerimientos más demandados.

## 2.12. Desarrollo de la propuesta

### 2.12.1. Requerimientos funcionales

En este apartado se especifican las funcionalidades principales que se espera que tenga el modelo propuesto, se detallan en un formato de tabla para un mejor entendimiento de la descripción, los datos de entrada y los datos que se tendrán a la salida de cada proceso ejecutado en la función.

*Tabla 1 “Requerimiento funcional 1”*

<b>Id</b>	RQF-01
<b>Nombre</b>	Extracción de datos (ofertas laborales)
<b>Tipo</b>	Requerimiento funcional
<b>Prioridad</b>	Alta
<b>Pre-rrequisito</b>	N/A
<b>Descripción</b>	El sistema deberá realizar una búsqueda automatizada de todas las ofertas laborales correspondientes al área de desarrollo de software, y almacenarlas en una base de datos.
<b>Entrada</b>	Url's de portales de empleo online.
<b>Proceso</b>	Web scraping.
<b>Salida</b>	Registro en base de datos por cada oferta laboral.

*Información proveniente del desarrollo de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

*Tabla 2 “Requerimiento funcional 2”*

<b>Id</b>	RQF-02
<b>Nombre</b>	Presentar las habilidades más demandadas
<b>Tipo</b>	Requerimiento funcional
<b>Prioridad</b>	Alta
<b>Pre-rrequisito</b>	RQF-01

<b>Descripción</b>	El sistema deberá determinar las habilidades que más se repiten entre las ofertas es decir, las más demandadas y mostrarlas en un texto ordenado, claro y preciso al usuario.
<b>Entrada</b>	Texto ingresado por el usuario.
<b>Proceso</b>	Consulta a la base de datos.
<b>Salida</b>	Listado de habilidades más demandadas.

*Información proveniente del desarrollo de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

*Tabla 3 “Requerimiento funcional 3”*

<b>Id</b>	RQF-03
<b>Nombre</b>	Buscar ofertas por habilidad
<b>Tipo</b>	Requerimiento funcional
<b>Prioridad</b>	Alta
<b>Pre-requisito</b>	RQF-01
<b>Descripción</b>	El sistema deberá consultar en la base cinco registros que coincidan con el nombre de la habilidad que el usuario escribió, para posteriormente presentarlas en la interfaz del chat de forma simple y organizada. Se sugiere el uso de la herramienta carrusel para la presentación de las ofertas como tarjetas individuales que se puedan deslizar hacia los lados.
<b>Entrada</b>	Texto ingresado por el usuario.
<b>Proceso</b>	Consultar ofertas en la base de datos.
<b>Salida</b>	Carrusel con 5 ofertas

*Información proveniente del desarrollo de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

*Tabla 4 “Requerimiento funcional 4”*

<b>Id</b>	RQF-04
<b>Nombre</b>	Redireccionar al sitio de la oferta
<b>Tipo</b>	Requerimiento funcional
<b>Prioridad</b>	Alta
<b>Pre-requisito</b>	RQF-01, RQF-03
<b>Descripción</b>	Luego de mostrar algunas ofertas, el sistema deberá permitir abrir dicha oferta en el navegador al dar click sobre ella.



<b>Entrada</b>	Click del usuario
<b>Proceso</b>	Abrir la página web de la oferta
<b>Salida</b>	Oferta mostrada en el navegador

*Información proveniente del desarrollo de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

*Tabla 5 “Requerimiento funcional 5”*

<b>Id</b>	RQF-05
<b>Nombre</b>	Consultar significado
<b>Tipo</b>	Requerimiento funcional
<b>Prioridad</b>	Alta
<b>Pre-requisito</b>	N/A
<b>Descripción</b>	Para los usuarios que no poseen ningún conocimiento en herramientas de programación, el sistema proporcionará un breve significado del término que el usuario ingrese en forma de pregunta.
<b>Entrada</b>	Texto del usuario
<b>Proceso</b>	Consultar definición en base
<b>Salida</b>	Breve concepto del término

*Información proveniente del desarrollo de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

## **2.12.2. Requerimientos no funcionales**

### ***2.12.2.1. Accesibilidad***

El asistente virtual deberá ser accesible desde el entorno web y el entorno móvil, indiferentemente del hardware que lo ejecute, requiriendo únicamente acceso a internet y una cuenta en la aplicación de Facebook Messenger.

### ***2.12.2.2. Funcionalidad***

Para la presentación de resultados al ejecutar la función de consultar ofertas, el asistente virtual solo podrá mostrar por pantalla un máximo de diez ofertas, esto debido a la limitación establecida por la plataforma de Meta Developers respecto al diseño de mensajes en Messenger.

### 2.12.3. Requerimientos de software

Tabla 6 “Requerimientos de software para el desarrollo del proyecto”

Requerimientos de software	
Nombre	Descripción
Selenium	Para la extracción de datos se requiere la herramienta Selenium, que permite la interacción dinámica con sitios web que contienen código Javascript.
Firebase	Base de datos no relacional que permite el envío de datos en formato JSON para la comunicación con otras aplicaciones.
Dialogflow	Plataforma de machine learning para el desarrollo de agentes virtuales chatbots, basada en la nube.
Facebook Developers	Versión de Facebook dedicada a desarrolladores, permite la integración con Api's externas.
Lucidchart	Herramienta gratuita basada en la web, para el modelado de diagramas UML.
Pycharm	El entorno de desarrollo integrado de Python, ideal para la ejecución de código Python por su intérprete incluido.
Canva	Software de diseño basado en la web, se utilizará como apoyo para el diseño de los bosquejos de las interfaces.
Google Forms	Plataforma para la gestión de encuestas y sus resultados.

*Información proveniente de la investigación de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

### 2.12.4. Diagramas de casos de uso

#### 2.12.4.1. Diagrama de caso de uso consultas de habilidades

En la figura 5 se puede observar tres casos de uso donde el actor usuario puede:

Consultar habilidades más demandadas: donde se muestra un listado de las tres habilidades más demandadas entre todas las ofertas encontradas.

Buscar ofertas por habilidad: donde el usuario ingresa una habilidad y el chatbot muestra una oferta que contenga dicha habilidad.

Consultar información de habilidad: donde se muestra un breve concepto de la habilidad que ingrese el usuario

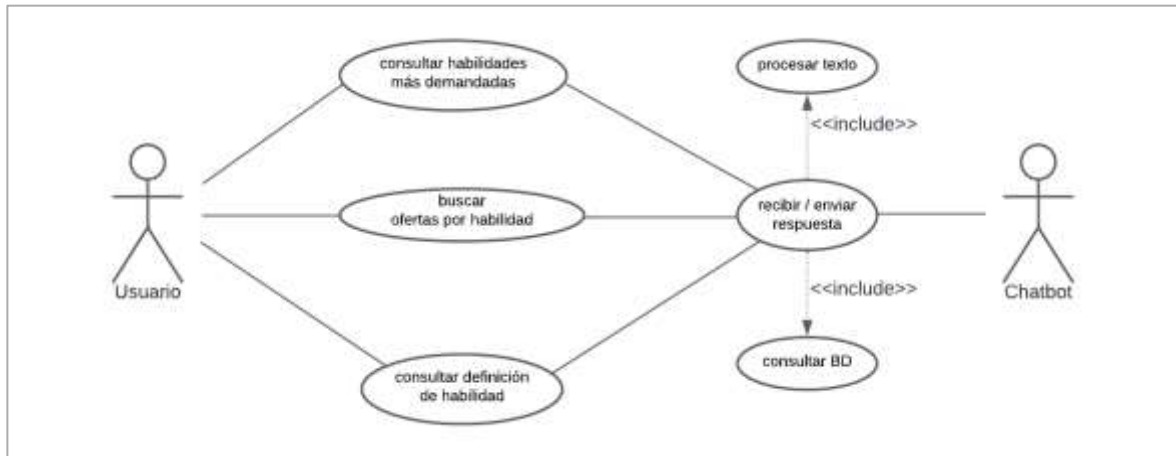


Figura 20 Diagrama de caso de uso consultar habilidades. Elaborado por Dayanna Vera (2022)

#### 2.12.4.2. Diagrama de caso de uso saludar

En la figura 6 se observa el caso de uso dado cuando el usuario ingresa al chat y envía un mensaje de saludo, donde se muestra información acerca de la finalidad y funcionalidad que tiene el chatbot.

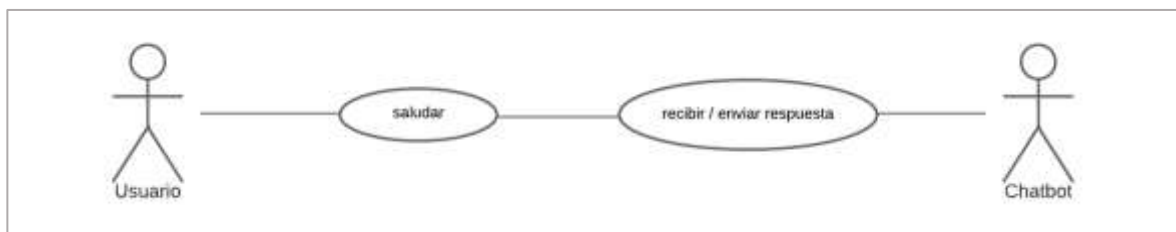


Figura 21 Caso de uso saludo del usuario. Elaborado por Dayanna Vera (2022)

#### 2.12.4.3. Diagrama de caso de uso extraer información

En la figura 7 se aprecia el caso de uso extraer información, donde el actor scraper realiza el raspado web y posteriormente almacena los datos recolectados en la respectiva base de datos.

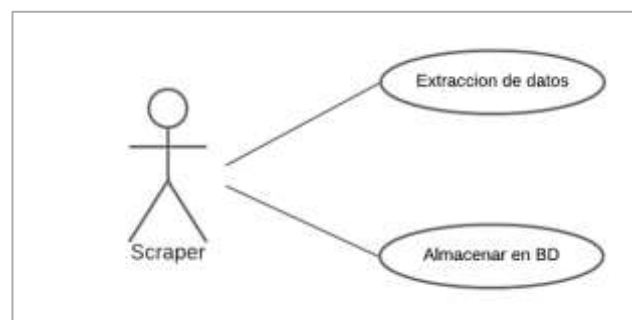


Figura 22 Caso de uso extraer información. Elaborado por Dayanna Vera (2022)

### **2.12.5. Arquitectura del modelo**

En el presente proyecto se propuso un modelo basado en web scraping y chatbot, las cuales son tecnologías distintas que serán conectadas en una misma arquitectura compuesta de tres capas que son:

#### ***2.12.5.1. Capa de presentación***

Esta es la capa que interactúa con el usuario final, se usará Facebook Messenger como plataforma para implementar el chatbot y una cuenta en Facebook Developers para acceder a las opciones de integración con APIs externas, sin embargo, el chatbot se encontrará dentro de una fan page común, por lo que estará disponible al público en general requiriendo únicamente acceso a internet y una cuenta en Facebook. La capa de presentación estará conectada la capa de procesamiento de lenguaje natural ejecutada por Dialogflow.

#### ***2.12.5.2. Capa de servicio PLN***

Esta es la capa que se encargará de la interpretación de las peticiones del usuario, para esto se usará la herramienta Dialogflow de Google, en donde se crearán todas las frases de entrenamiento, intenciones, entidades y respuestas que permitirán al agente virtual entender y mantener el flujo de la conversación con el usuario, respondiendo a todas sus solicitudes. Esta capa es fundamental ya que es la intermediaria entre las capas de presentación y almacenamiento de datos basado en Firebase.

#### ***2.12.5.3. Capa de almacenamiento de datos***

Esta capa usará el servicio de Firebase de Google, la base de dato no relacional ideal para esta arquitectura ya que permite el envío de datos en formato JSON, facilitando el proceso de consultas desde Dialogflow hacia los registros resultantes del Web Scraping realizado sobre el portal Computrabajo y LinkedIn.

#### ***2.12.5.4. Capa de Web Scraping***

En esta capa se realiza la extracción de datos, consiste en un código escrito en lenguaje Python, debido a la estructura de las páginas antes mencionadas, se trabaja también con el framework Selenium que facilita la interacción con páginas que contienen código JavaScript. Todos los datos extraídos serán almacenados en la base de datos Firebasebase.



Figura 23 Arquitectura del modelo. Elaborado por Dayanna Vera (2022).

### 2.12.6. Diseño de interfaces

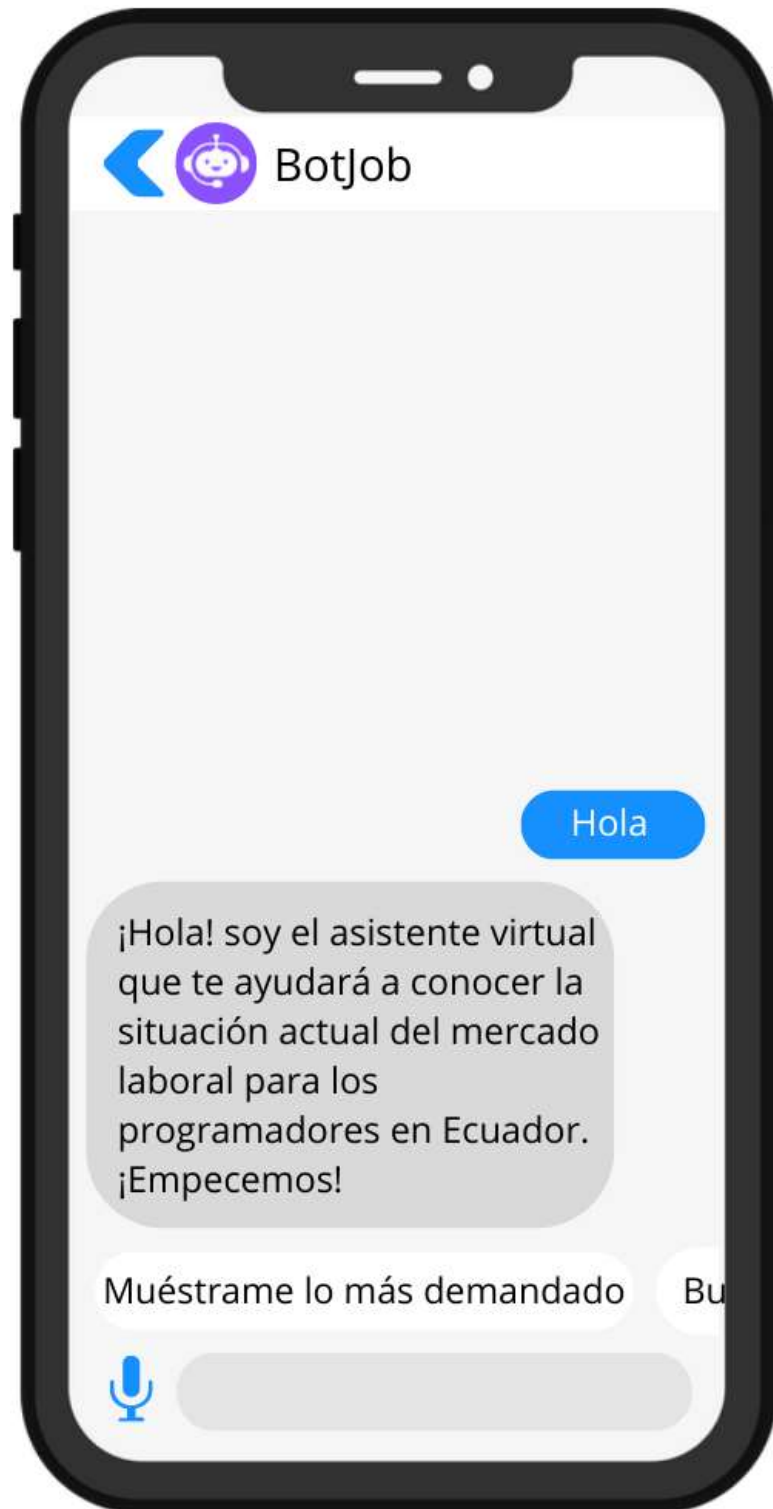


Figura 24 Interfaz de saludo. Elaborado por Dayanna Vera (2022).

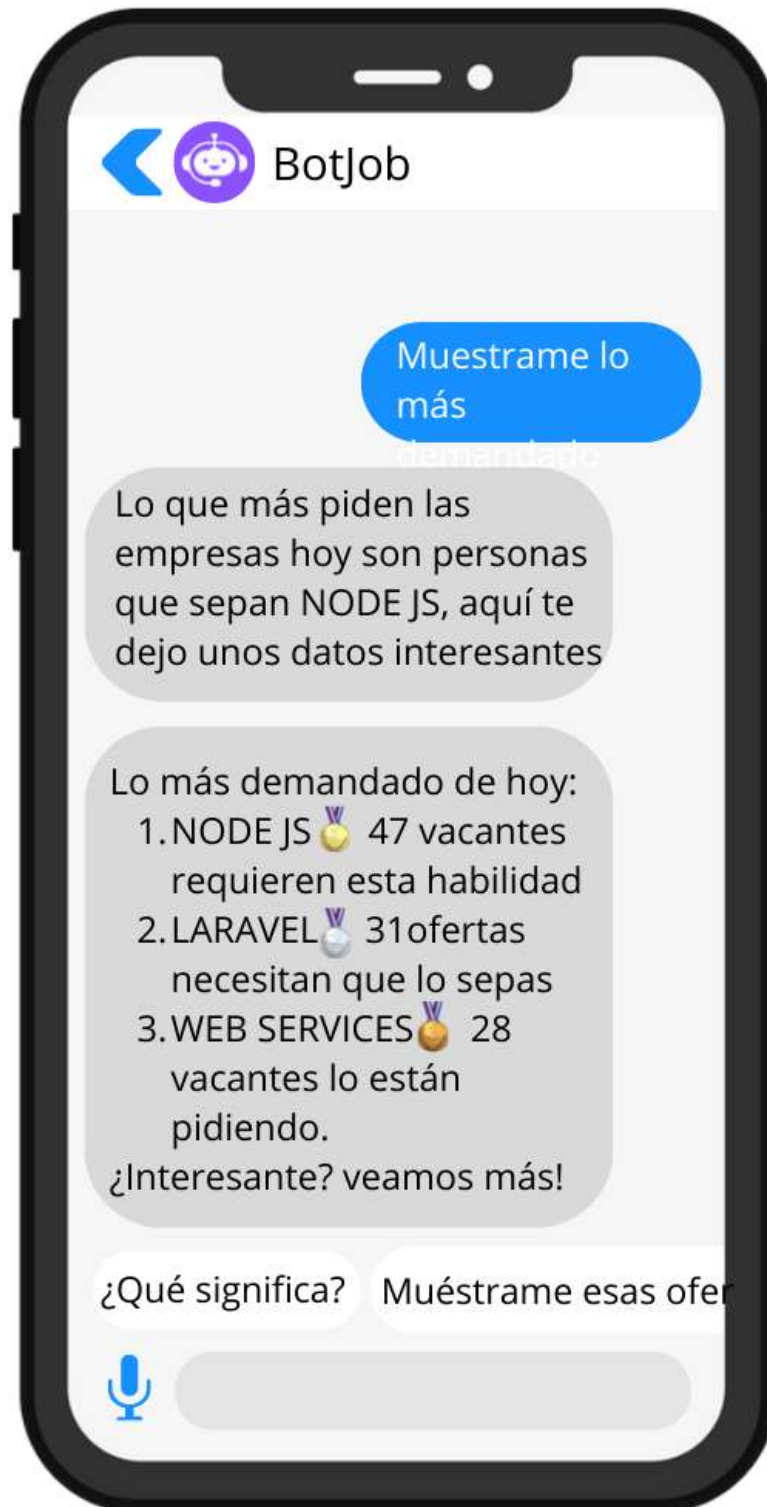


Figura 25 Interfaz habilidades más demandadas. Elaborado por Dayanna Vera (2022).



Figura 26 Interfaz mostrar significado. Elaborado por Dayanna Vera (2022).





Figura 27 Interfaz ver ofertas. Elaborado por Dayanna Vera (2022).

### 2.12.7. Diseño base de datos

Para almacenar y extraer datos de forma flexible, se eligió la base de datos no relacional Firestore de Firebase, por tanto el diseño de la base de datos consiste en tres colecciones que contienen documentos, y estos a su vez están compuestos por campos que contienen un valor. De forma general la estructura de la base de datos está desplegada en nodos que no se limitan a una estructura específica y pueden contener varios tipos de datos, como por ejemplo arrays, clave:valor, gráficos, entres otros.

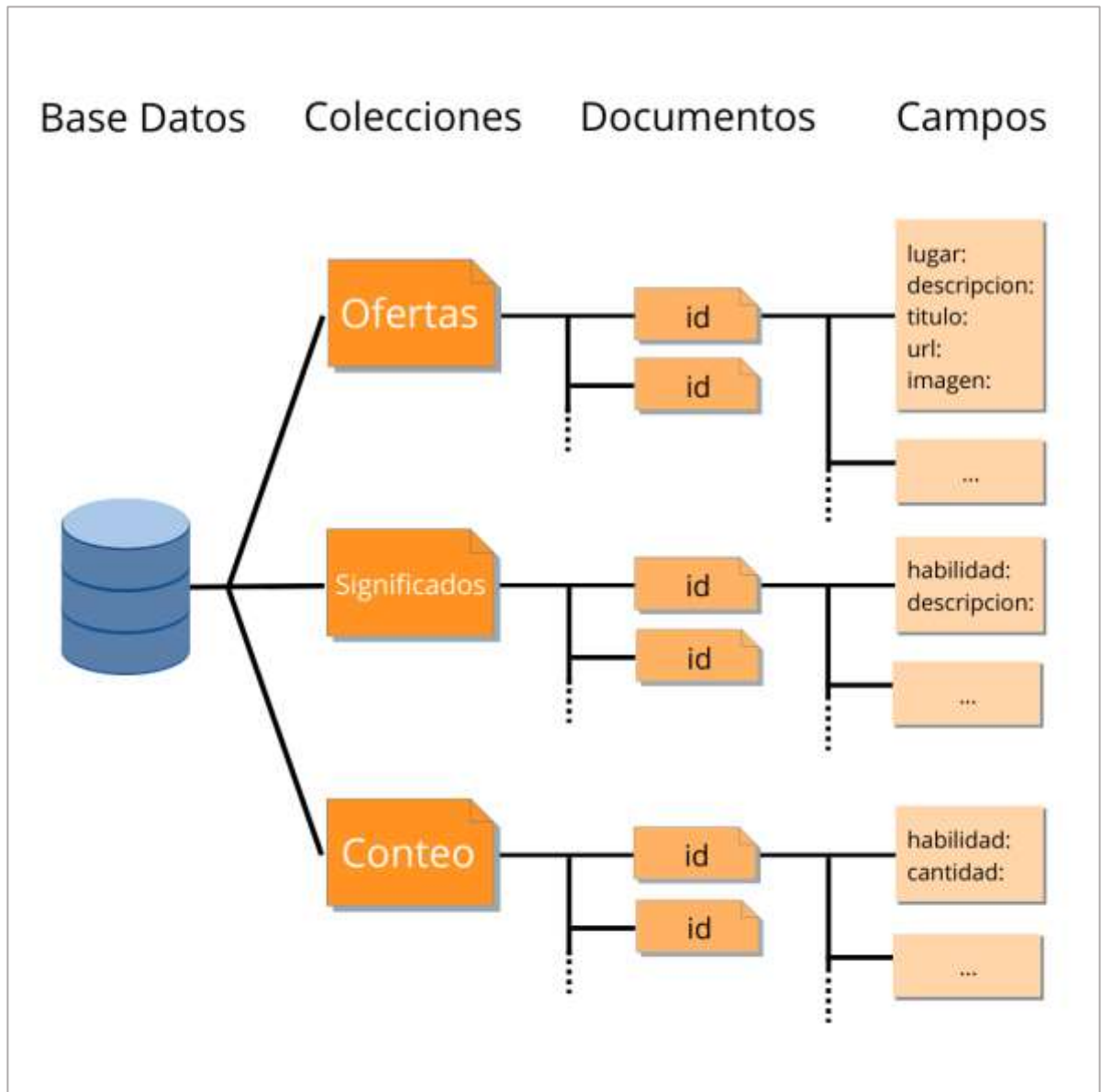


Figura 28 Diseño Base de Datos. Elaborado por Dayanna Vera (2022).

## 2.13. Implementación

La implementación del proyecto estará dividida en tres etapas:

- Extracción y almacenamiento de datos
- Integración entre servicios
- Entrenamiento PLN

Para la fase de extracción de datos se realizarán una serie de pasos que serán llevados a cabo individualmente por cada sitio web, esto es Computrabajo y LinkedIn, debido a su diferente estructura que obliga a tratarlos de forma distinta.

### 2.13.1. Extracción de datos del portal Computrabajo

#### 2.13.1.1. Exploración del sitio web

Para un primer acercamiento con el sitio web, se visita la página para conocer su estructura de forma superficial, es decir, viendo cómo está compuesta de forma general. En el portal Computrabajo se observa que al ingresar a su página principal solo se obtiene acceso a un filtro de “cargo” y “lugar” para realizar búsqueda de ofertas, en este caso se escribe la palabra clave “desarrollo de software” en el campo cargo y el campo lugar se lo deja vacío para así poder obtener las ofertas de todo Ecuador. En la figura 19 se observa la pantalla mostrada al realizar la búsqueda filtrada.

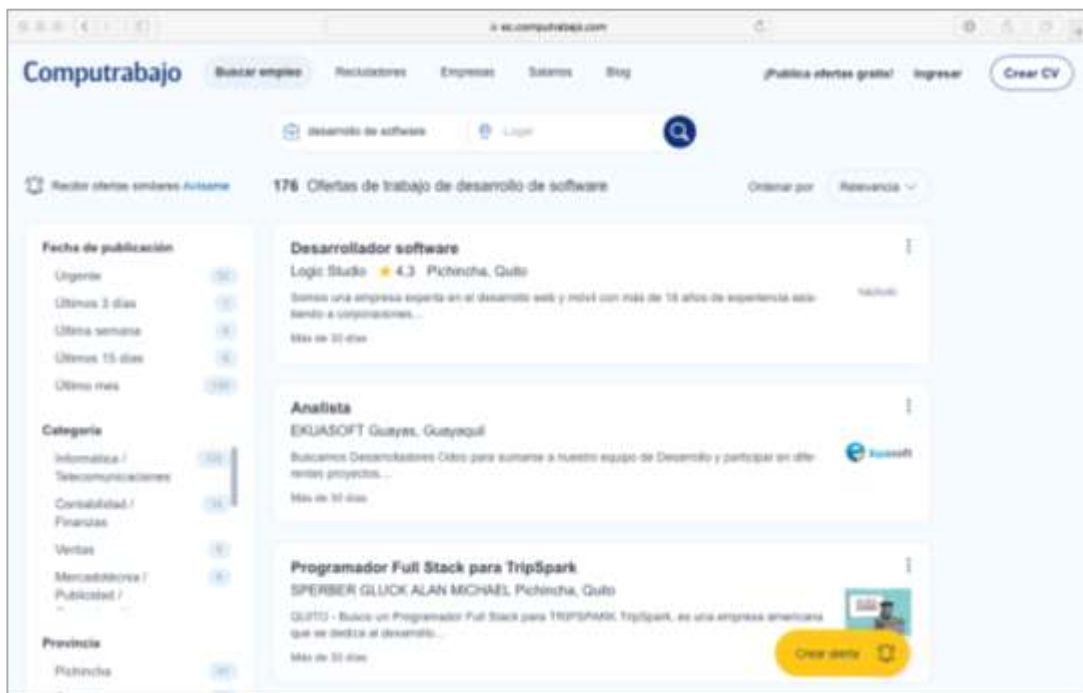


Figura 29 Entorno del portal Computrabajo. Elaborado por Dayanna Vera (2022).

Se observa que el portal contiene una lista en su parte central, donde se muestran todas las ofertas, sin embargo para poder acceder al desglose de los requerimientos de las mismas es necesario dar click en el título para que se cargue la página con todos los detalles, como se muestra en la figura 20, por lo tanto se hace necesaria la utilización de una herramienta capaz de simular el evento del click de un usuario para poder extraer el detalle de los requerimientos, para ello se utiliza Selenium.



Figura 30 Detalles de la oferta laboral. Elaborado por Dayanna Vera (2022).

### 2.13.1.2.Reconocimiento de la estructura HTML

Para este paso se utiliza la herramienta “inspeccionar” que viene integrada en algunos navegadores, en este caso Google Chrome. Aquí se puede revisar la estructura Html de la página como se muestra en la figura 21, donde se observa una jerarquía de etiquetas <div>, <article>, <h1>, <a>, <p>, entre otras; las cuales están en su gran mayoría identificadas por un nombre de clase o un id. En este punto, el elemento que interesa identificar es el título donde el usuario puede dar click y abrir la oferta, dicho elemento está en la etiqueta <h1 class=“fs18 fwB”> que a su vez contiene la etiqueta <a> con nombre de clase “js-o-link fc\_base” donde se encuentra el link clickable. Para indicarle a Selenium el elemento en que debe dar click, se lo identificará mediante su xpath, el cual se obtiene dando click derecho sobre la etiqueta objetivo, esto se hace en la ventana inspeccionar.

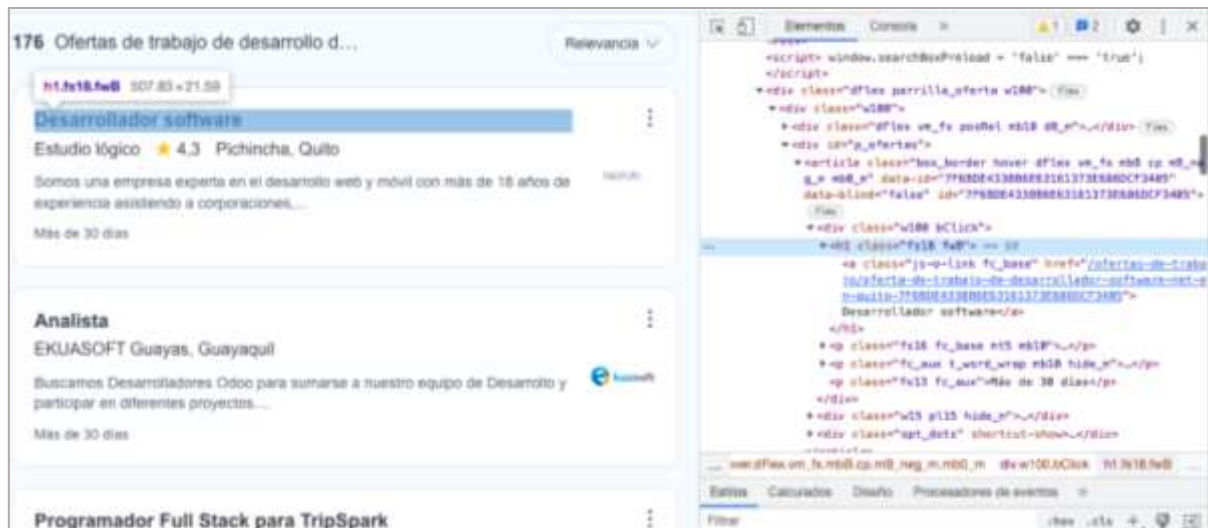


Figura 31 Estructura Html Computrabajo. Elaborado por Dayanna Vera (2022).

Luego de abrirse la página con los detalles de la oferta, se identifican los elementos a extraer, estos son el título de la vacante, nombre y lugar de la empresa y la descripción, del mismo modo se extrae la url de la página con el método `driver.getCurrentUrl()` de Selenium, ya que por cada oferta seleccionada la url cambia. Se prefirió extraer todo el contenido de la descripción puesto que no todas las ofertas detallan los conocimientos necesarios en el apartado “conocimiento” pero sí los tienen en su descripción. El título de la vacante se encuentra en la etiqueta `<h1>` `class="fwB fs24 mb5 box detail w100_m"`, el nombre y lugar de la empresa en la etiqueta `<p>` `class="fs16">` y la descripción en `<p>` `class="mbB">`. Para la identificación de los elementos se utilizará siempre el xpath de dicho elemento, los mismos que se muestran en el fragmento de código observado en la figura 23.

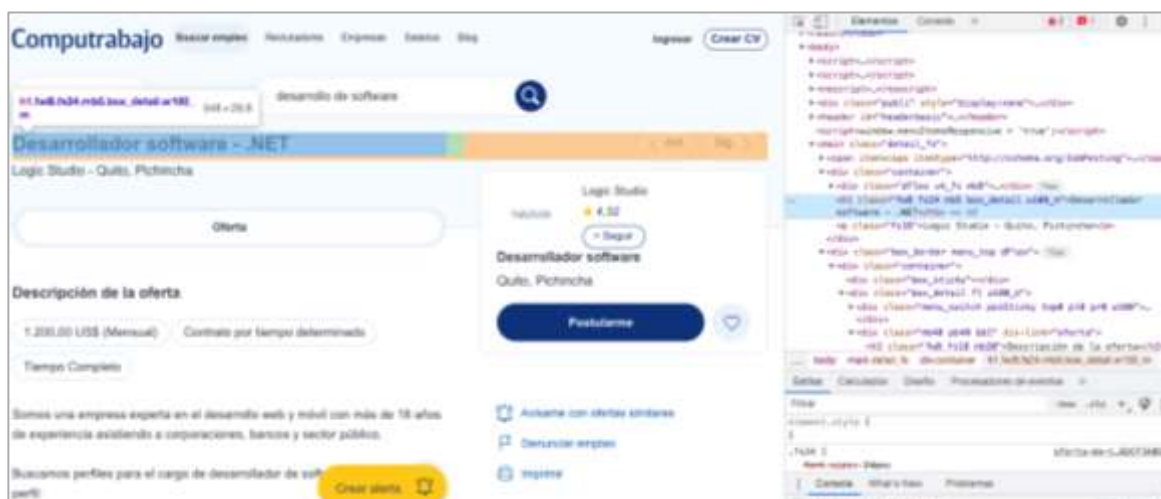


Figura 32 estructura Html del detalle de la oferta. Elaborado por Dayanna Vera (2022).

```

xpath_titulo = '/html/body/main/div[1]/h1'
xpath_lugar = '/html/body/main/div[1]/p'
xpath_descripcion = '/html/body/main/div[2]/div/div[2]/div[2]/p[1]'

```

Figura 33 Código identificadores xpath. Elaborado por Dayanna Vera (2022).

### 2.13.1.3.Desarrollo del código web scraping

Para el desarrollo del código es necesario en primera instancia tener instalado Selenium y el IDE de Python llamado Pycharm, ya que este facilita la ejecución de código gracias a su intérprete integrado. Además, se utilizarán algunas librerías como las que se muestran en la figura 22.

```

from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.common.by import By
import time
import pandas as pd

```

Figura 34 librerías de python a utilizar. Elaborado por Dayanna Vera (2022).

Como se mencionó anteriormente, Selenium realiza pruebas o simulaciones de navegación mediante el uso de un WebDriver que permite la ejecución de una instancia de navegador sencilla y que puede ser manipulada por los comandos de Selenium, es como abrir un pequeño navegador independiente dentro de Selenium. Para efectos de este proyecto se utilizará el driver de Google Chrome versión 104.0.5112.101.

En el apartado anterior se observó que las ofertas laborales se encuentran dentro de una lista ubicada en la parte central de la página; para poder acceder a los requerimientos se necesita seleccionar cada oferta de la lista hasta llegar al final de la página en donde hay botones que direccionan a la siguiente página que contiene más ofertas. Por consiguiente, para poder extraer la descripción de todas las ofertas disponibles es necesario identificar dos elementos claves, estos se muestran a continuación junto con el nombre de la variable que representa sus identificadores xpath dentro del código:

- Nombre de la oferta = ofert\_xpath
- Botón siguiente 1 = next\_btn\_xpath1
- Botón siguiente = next\_btn\_xpath



El botón siguiente tiene una variante debido a que en la primera página de ofertas en Computrabajo este botón tiene un xpath ligeramente distinto al botón que se encuentra en la página número dos en adelante.

Una vez identificados los elementos claves, se analiza la forma de recorrer las ofertas de todas las páginas de resultados, esto se logró guardando en un array todas las ofertas enlistadas, como se ve en la figura 23 la variable `all_oferts` contiene todos los nombres de ofertas encontradas según su xpath. Teniendo guardados todos los resultados, se los recorre en un bucle `for`, en cada iteración ingresa a la página de la oferta extrae el texto de los elementos descripción, lugar, título, url de la oferta y luego regresa a la página anterior para continuar con el siguiente item, así hasta llegar al final de la página. Una vez ahí, Selenium ubica el botón siguiente para ir a la siguiente página y realizar el mismo proceso de extracción; el código termina de ejecutarse cuando haya llegado a la última oferta de la última página.

```
all_oferts = driver.find_elements_by_xpath(ofert_xpath)

for ofert in all_oferts:
    ofert.click()
    description1 = driver.find_element_by_xpath(description_xpath).text
    lugar1 = driver.find_element_by_xpath(lugar_xpath).text
    titulo1 = driver.find_element_by_xpath(titulo_xpath).text
    url_ofert1 = driver.current_url
    driver.back()

next_page1 = driver.find_element_by_xpath(next_btn_xpath1)
next_page1.click()
```

Figura 35 bucle `for` para recorrer las ofertas. Elaborado por Dayanna Vera (2022).

#### **2.13.1.4. Almacenamiento en la base de datos**

Para establecer conexión con la base de datos en la nube es necesario en primer lugar proporcionar las credenciales de acceso, las mismas que se encuentran en un archivo guardado de forma local en la computadora que se utilizó para el desarrollo de este proyecto; Luego se invocan los métodos para guardar datos en Firestore, en este caso se intenta almacenar datos bajo dos situaciones, la primera es cuando el raspado web no encuentra la

imagen porque no está definida, entonces intenta el segundo caso en donde se guarda una imagen por defecto, como se ve en la figura 27.

```
# conexion a firebase
cred = credentials.Certificate("appmaga-6d0cd-firebase-adminsdk-ou4dq-8d81d2d546.json")
firebase_admin.initialize_app(cred)
db = firestore.client()
```

Figura 36 Credenciales de acceso FirebaBase. Elaborado por Dayanna Vera (2022).

```
try:
    imagen1 = driver.find_element_by_xpath(imagen_xpath).get_attribute('src')
    # guardando en firebase
    db.collection('ofertas').add({
        'titulo': titulo1,
        'descripcion': descripcion,
        'lugar': lugar1,
        'url': url_ofert1,
        'imagen': imagen1
    })
except:
    # guardando en firebase
    db.collection('ofertas').add({
        'titulo': titulo1,
        'descripcion': descripcion,
        'lugar': lugar1,
        'url': url_ofert1,
        'imagen': 'https://bdrinformatica.com/wp-content/uploads/2022/03/desarrollo-de-software.jpg'
    })
```

Figura 37 Almacenando datos en Firestore. Elaborado por Dayanna Vera (2022).

## 2.13.2. Extracción de datos de LinkedIn

### 2.13.2.1. Exploración del sitio web

Al observar a interfaz de LinkedIn se puede notar que las oferta de empleo están enlistadas en la parte izquierda de la pantalla y a diferencia de Computrabajo el detalle de los requerimientos se los muestra en el lado derecho una vez que el usuario selecciona una oferta de la lista, esta parte de la pantalla está programada con JavaScript para la carga dinámica del contenido cuando se realiza un click, por ende puede ser tratado con Selenium para la realización de las pruebas. Otro detalle importante para notar es que al deslizar hacia abajo en la pantalla no se muestra un boton de siguiente página si no que se cargan automáticamente nuevas ofertas, así hasta que después de varios deslices hacia abajo se



muestra un botón para ver más ofertas. Para la realización de esta prueba se llegará hasta el final de la página pero no se dará click en el botón “ver más ofertas”.

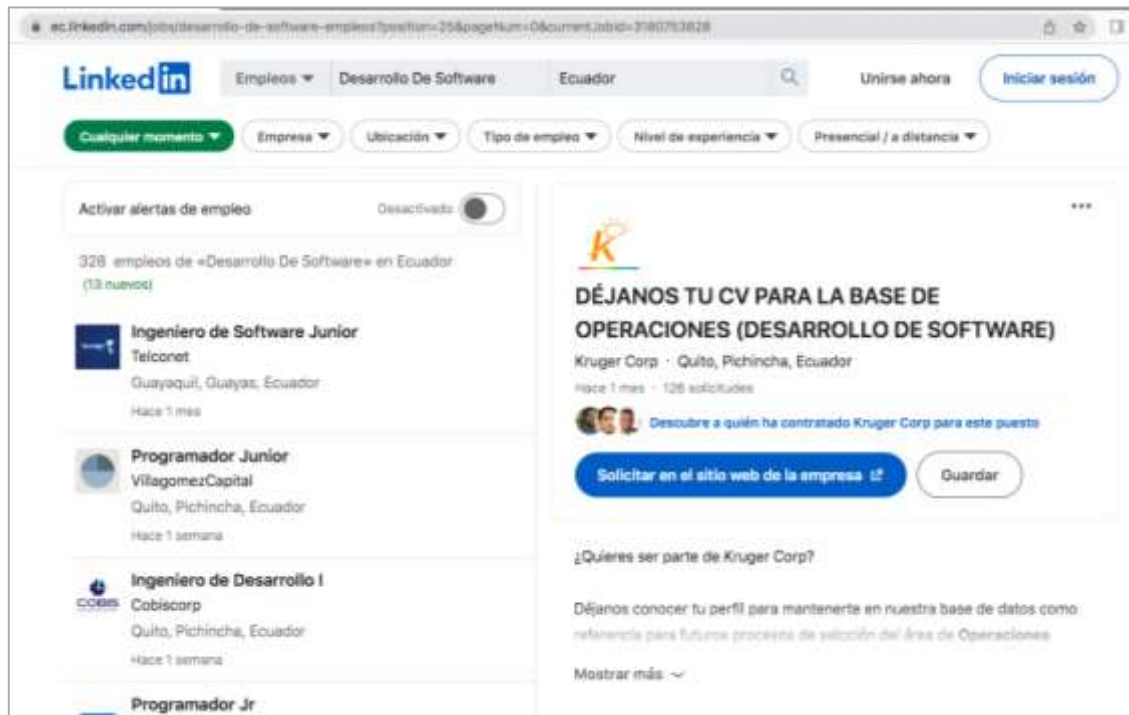


Figura 38 Entorno de LinkedIn. Elaborado por Dayanna Vera (2022).

### 2.13.2.2.Reconocimiento de la estructura HTML

Se observa que el listado de ofertas laborales se encuentra dentro de una etiqueta <ul> con la clase “jobs-search\_results-list”, esta a su vez contiene muchas etiquetas <li> que no tienen nombre de clase. Al revisar el contenido de las etiquetas <li>, se notó no todas tienen la misma estructura, pues algunas contienen todo dentro de una etiqueta <div> y otras dentro de una etiqueta <a>, como se puede ver en la figura 29. Esta variación es aleatoria, por ello para hacer click sobre una oferta se necesitarán dos identificadores xpath, uno para cada caso.



Figura 39 Estructura Html LinkedIn. Elaborado por Dayanna Vera (2022).

Con respecto a la descripción de las ofertas, estas se cargan dinámicamente cuando el usuario selecciona un ítem, lo cual hace que también cambie la url al insertarse la variable “JobId”, como está señalado en la figura 30. También se observa que la descripción de la oferta puede ser recuperada completamente ya que a nivel de código está completa, aunque visualmente no se muestre en su totalidad.

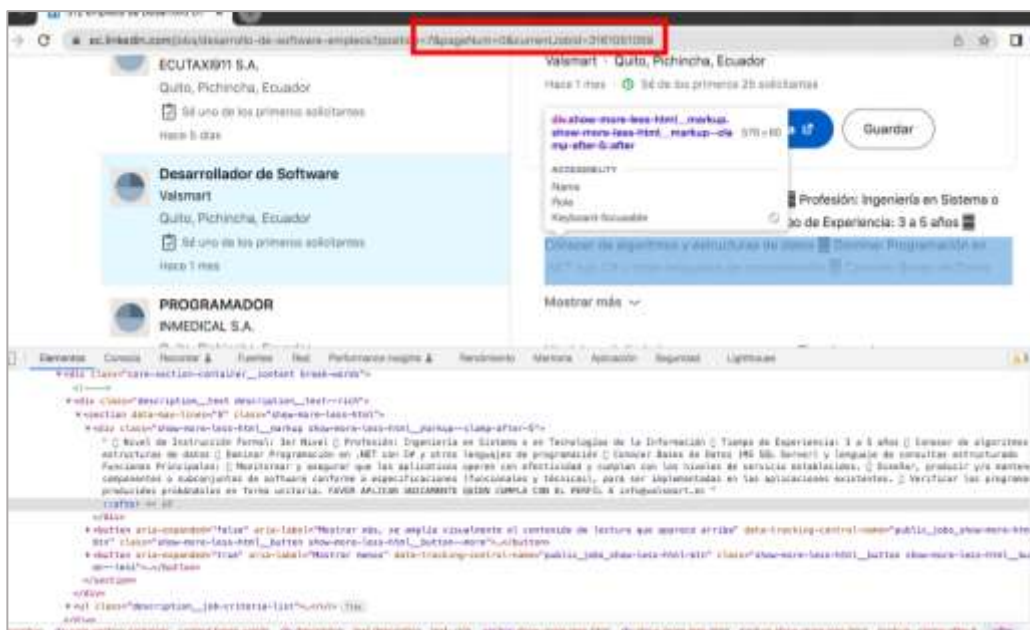


Figura 40 Html descripción de la oferta en LinkedIn. Elaborado por Dayanna Vera (2022).

### 2.13.2.3. Desarrollo del código web scraping

El código de Web Scraping aquí es muy similar al de Computrabajo, difiere únicamente en que LinkedIn tiene un marco extra de seguridad que detecta el tráfico de bots y bloquea las paginas, por ello, para evitar que el programa se detenga por un error, se codificaron las sentencias “try” y “except”.

```
while item < len(allOferts):
    try:
        # oferta dentro de un <div>
        xpathOp1 = '//*[@id="main-content"]/section[2]/ul/li[' + str(item) + ']/div/a'
        time.sleep(2)
        op1 = driver.find_element_by_xpath(xpathOp1)
        driver.execute_script("arguments[0].click();", op1)
        time.sleep(1)
        desc = driver.find_element_by_xpath(xpath_desc1).text
        cant += 1
        item += 1
    except:
        try:
            # oferta dentro de etiqueta <a>
            xpathOp2 = '//*[@id="main-content"]/section[2]/ul/li[' + str(item) + ']/a/div[2]/h3'
            time.sleep(2)
            op2 = driver.find_element_by_xpath(xpathOp2)
            op2.click()
            driver.get(driver.current_url)
            time.sleep(1)
            desc = driver.find_element_by_xpath(xpath_desc2).text
            cant += 1
            item += 1
```

Imagen 1 Código Web Scraping para LinkedIn. Elaborado por Dayanna Vera (2022).

### 2.13.2.4. Almacenamiento en la base de datos

Para guardar datos en la base de Firestore se sigue el mismo procedimiento que se realizó en Computrabajo, tomando en cuenta que los campos del JSON deben tener exactamente el mismo nombre ya que se guardarán en una misma base los resultados de Computrabajo y LinkedIn

## 2.13.3. Integración entre servicios

Como se ha mencionado, el presente proyecto hace uso de varios servicios en la nube tanto de Google como de Facebook, por ello la siguiente etapa de la implementación consiste en la integración de todos estos servicios bajo el esquema que se mostró en la figura 13.

### ***2.13.3.1. Integración Dialogflow con Messenger***

Dado que el asistente virtual propuesto en este proyecto funcionará en la Plataforma Messenger, fue necesario el registro como usuario en la plataforma Meta Developers, la cual permite la creación de aplicaciones en Whatsapp, Instagram y Messenger, y además facilita la integración de las mismas con servicios externos, como por ejemplo con Dialogflow.

Por su parte, Dialogflow dispone del apartado integraciones para conectarse con otros servicios como Messenger, Line, Telegram, Twitter y muchos más; para conectar con estas aplicaciones se registra un token proporcionado por la plataforma deseada, en este caso Messenger from Facebook, de la misma manera Meta for Developers requiere de una devolución de llamada o “Callback URL” que la proporciona Dialogflow, esto es parte del Webhook de la plataforma. El procedimiento realizado se encuentra a detalle en el Anexo 4.

### ***2.13.3.2. Integración Firebase con Dialogflow***

El Webhook de Dialogflow permite la comunicación en tiempo real con otras aplicaciones, así cada vez que el asistente virtual requiera un dato de la base, realizará esta llamada HTTP a Firebase para que le devuelva los datos en formato JSON para presentarlos por mensaje. Para lograr esta integración se requiere la URL de la aplicación en Firebase a la que realizará la llamada, del mismo modo Firebase requiere identificar ese Webhook de Dialogflow en el código del asistente virtual.

### **Desarrollo del código del asistente virtual**

Para realizar la programación del asistente virtual se optó por utilizar la consola de Firebase (CLI) para así utilizar un editor de código más cómodo en el entorno local de la máquina, en este caso se eligió Visual Studio Code, en lugar de utilizar el editor en línea de Dialogflow.

Los demás detalles de la integración se especifican en el Anexo 4

### **2.13.4. Entrenamiento PLN**

En la plataforma de Dialogflow se establecen los intents que permitirán al chatbot identificar la intención de los mensajes del usuario es decir entender lo que le están solicitando. Para realizar el entrenamiento se ingresan una serie de posibles frases que el usuario puede escribir en determinada situación y se establece una respuesta por defecto

para dicha situación. En el desarrollo de este proyecto se configuraron seis intents, uno para cada función del asistente virtual, juntamente con sus frases de entrenamiento como se muestra a continuación:

*Tabla 7 Entrenamiento PLN en Dialogflow*

Nombre del intent	Frases de entrenamiento
Saludo	Hola, empecemos, empezar, holi, hi, hello
ConsultarOfertas	Buscar ofertas, Buscar ofertas con \$habilidad Muéstrame esas ofertas
ConsutlarSignificado	Consultar significado, Qué significa \$habilidad? , Que es \$habilidad? Qué significa eso?
MasDemandado	Ver lo más demandado, Lo más demandado, Habilidades más demandadas, Muestrame lo más demandado.
Despedida	Chao, adiós, hasta luego, bye
Default Fallback Intent	N/A

*Información proveniente del desarrollo de la propia tesis. Elaborado por el autor Dayanna Vera (2022)*

Cabe mencionar que el “Default Fallback Intent” es el intent para responder a cualquier mensaje que el chatbot no pueda entender. También la palabra “habilidad” precedida por el signo de dólar representa que dicha palabra es una variable reconocida por dialogflow que contiene el conjunto de palabras con las habilidades más comunes, esto se puede ver con mayor detalle en el Anexo 5

## 2.14. Modelo final

En este apartado se presentan los resultados de la implementación del modelo propuesto, se destacan las funcionalidades principales con que cuenta el asistente virtual al finalizar el proyecto.

## 2.14.1. funcionalidades principales

### 2.14.1.1. Saludar



Figura 41 Función saludar. Elaborado por Dayanna Vera (2022)

Es la función con la que inicia el flujo de la interacción con el usuario, aquí el asistente virtual expresa una descripción de su finalidad, la cual es dar a conocer las habilidades más demandadas en el sector laboral del desarrollo de software, además brinda instrucciones al usuario sobre cómo interactuar con él.

### 2.14.1.2. Consultar lo más demandado

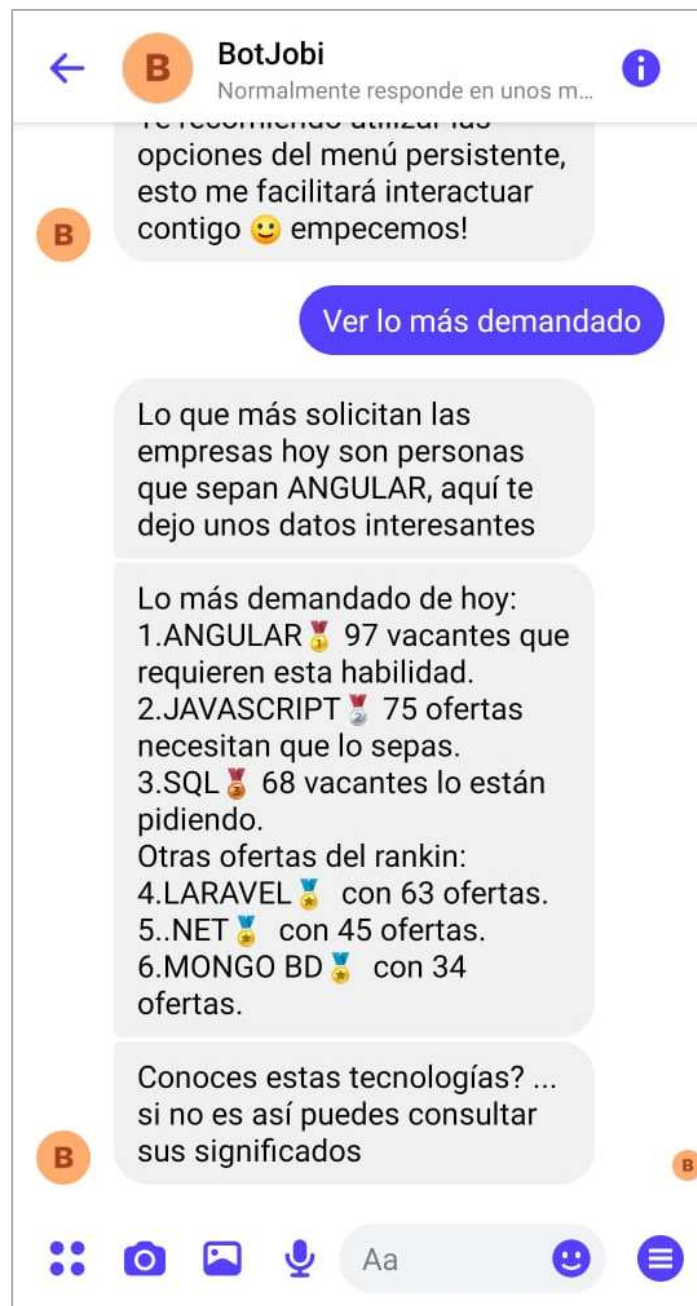


Figura 42 Función consultar habilidades más demandadas. Elaborado por Dayanna Vera (2022)

Aquí el asistente virtual responde con datos obtenidos de la base en Firebase, muestra el resumen de las habilidades más demandadas de entre todas las ofertas encontradas durante el proceso de web scraping. Un aspecto importante para resaltar es la confiabilidad de los datos obtenidos, puesto que son resultado de ofertas reales y vigentes.



### 2.14.1.3. Consultar significado



Figura 43 Función consultar significado. Elaborado por Dayanna Vera (2022)

Esta función muestra el significado de las habilidades que el usuario ingrese, al igual que la función anterior esta también extrae el significado de la colección “significados” de la base de datos en Firebase. Además muestra un mensaje que invita a continuar con la interacción guiándolo hacia la función siguiente.



#### 2.14.1.4. Consultar ofertas

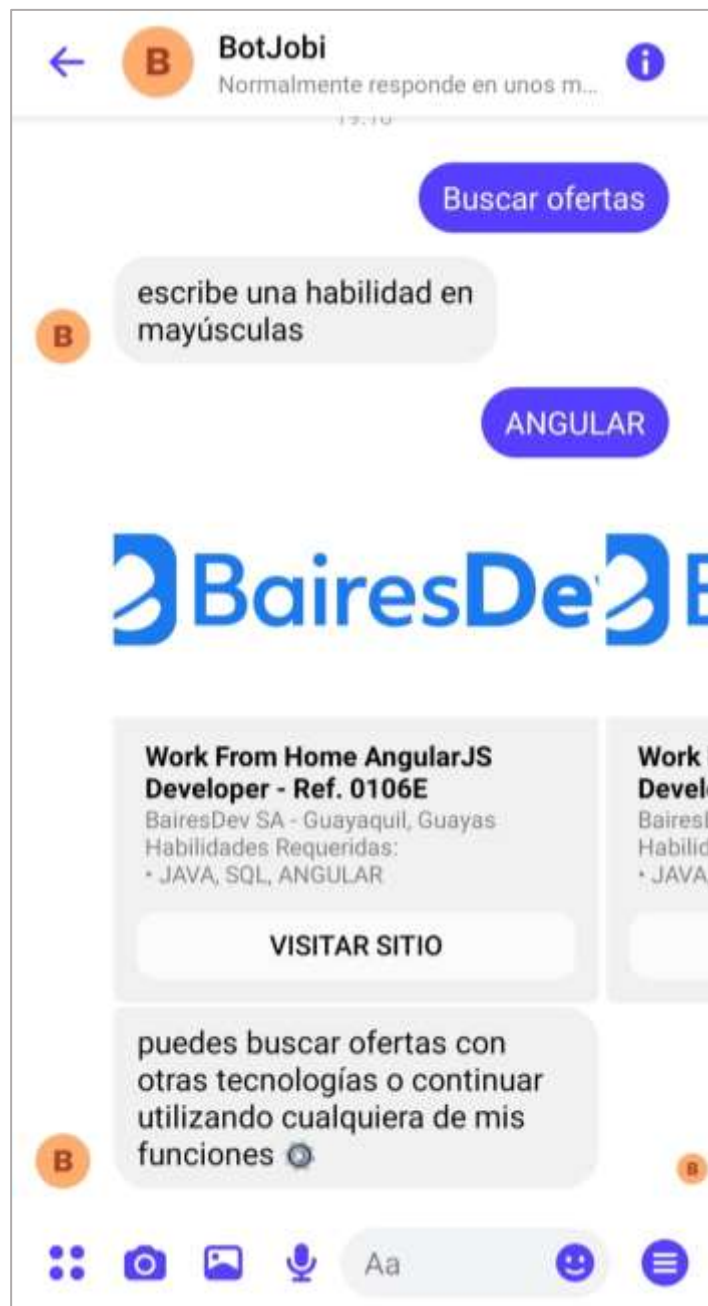


Figura 44 Función consultar ofertas laborales. Elaborado por Dayanna Vera (2022)

Esta función muestra las ofertas que contengan como requerimiento la habilidad ingresada por el usuario, esta es una forma distinta de la manera tradicional de buscar trabajo ya que no se centra en el nombre del cargo sino en las habilidades que este requiere. Las ofertas se presentan en un formato tarjeta que contiene la imagen de la empresa, el nombre del cargo, y las habilidades requeridas y un botón para visitar el sitio original de la oferta. Por ser varias tarjetas estas se presentan en formato carrusel.

#### 2.14.1.5. Visitar página de la oferta



Figura 45 Función visitar página de la oferta. Elaborado por Dayanna Vera (2022)

Esta es la función que se ejecuta cuando el usuario presiona el botón de visitar sitio, se abre una nueva vista dentro de la misma aplicación de Messenger que corresponde al sitio web donde se encuentra la oferta.

#### 2.14.1.6. Despedida



Figura 46 Función de despedida. Elaborado por Dayanna Vera (2022)

Finalmente el asistente virtual en su función de despedida muestra un mensaje que anima al usuario a tomar acciones con la información que recibió y se muestra dispuesto a una próxima interacción.

#### 2.15. Escenarios de pruebas

En esta sección se describen los resultados obtenidos al probar el modelo basado en chatbot y Web Scraping, de forma precisa se detallan aquellos escenarios en los cuales cualquiera de sus componentes o procesos no se comporta de la manera ideal.

### 2.15.1. No existe significado

En la función consultar significado puede darse el caso de no contar con el significado de la habilidad que el usuario ingresa, dado que la base de datos contiene los significados únicamente de las habilidades definidas por el desarrollador. Cuando el asistente virtual busca el significado de una habilidad y no lo encuentra, automáticamente responde con el mensaje mostrado en la figura 37, indicando que no cuenta con él e invitando al usuario a probar con otra.



Figura 47 Escenario “no existe significado”. Elaborado por Dayanna Vera (2022)

### 2.15.2. No se encontraron ofertas

Otro escenario de inexistencia de datos se da en la función de buscar ofertas, cuando el asistente virtual intenta buscar ofertas con determinada habilidad pero no existen ofertas que la requieran, en este caso el asistente virtual muestra un mensaje explicando la situación e invitando al usuario a probar con otra habilidad.

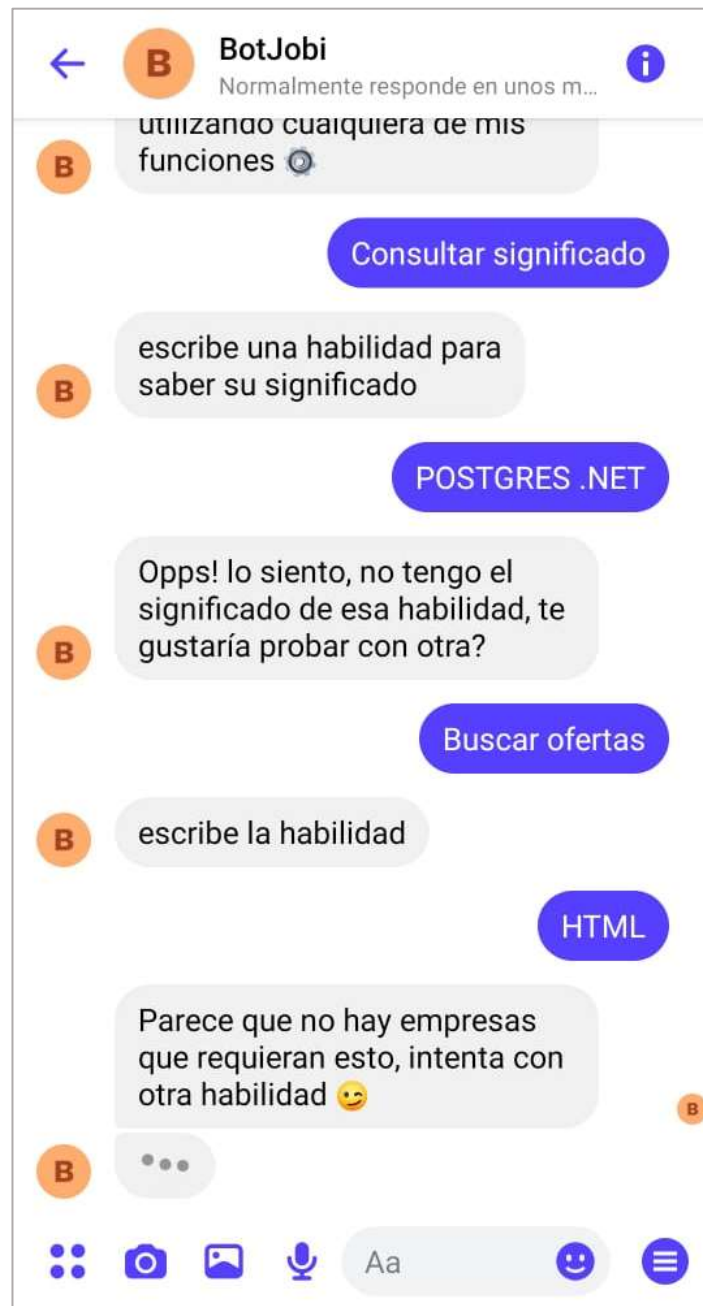


Figura 48 Escenario "no se encontraron ofertas". Elaborado por Dayanna Vera (2022)

### 2.15.3. Fallo al conectar con la base de datos

Como se sabe, Firebase es una base de datos que funciona en la nube, por ello es susceptible de fallar y no estar disponible si sucede alguna caída en la conexión o por haber un error en las reglas de seguridad y acceso, para estos casos el asistente virtual está entrenado para responder con el mensaje mostrado en la figura

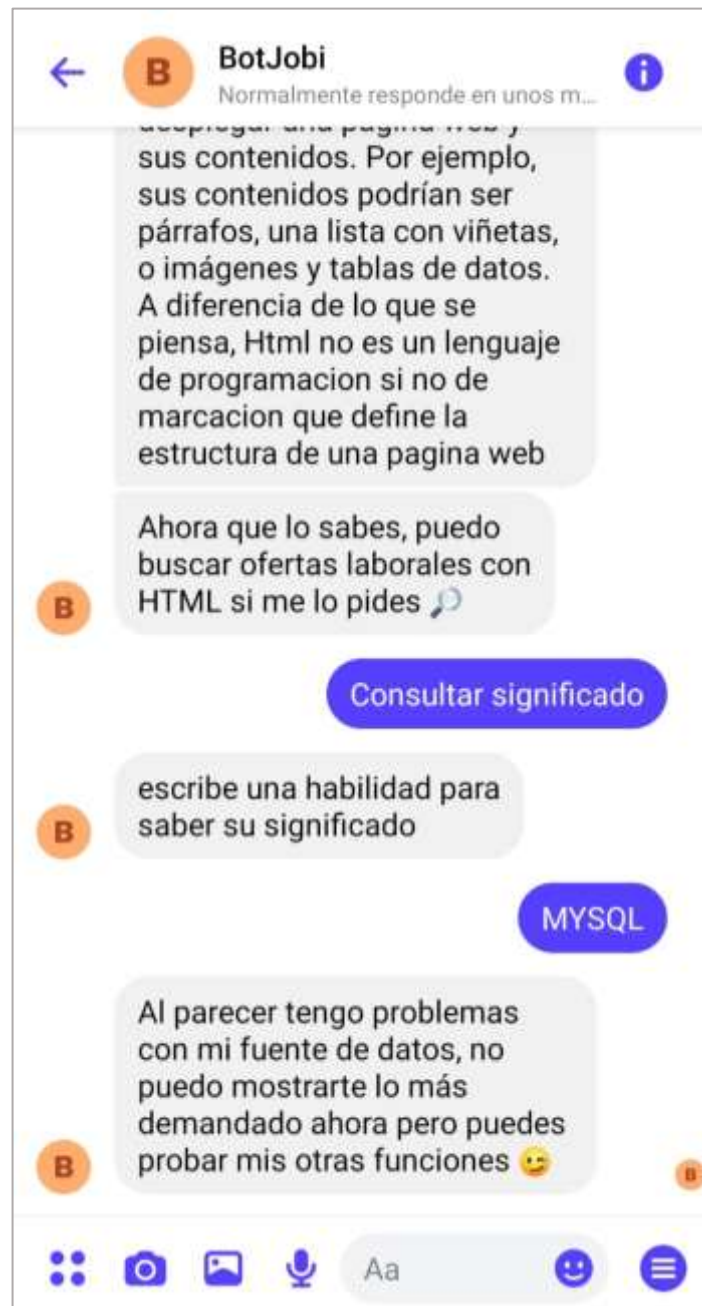


Figura 49 Escenario "fallo en la conexión con la base de datos". Elaborado por Dayanna Vera (2022)

## 2.16. Conclusiones

En cumplimiento a los objetivos específicos planteados en el presente proyecto, se puede concluir que, al realizar la investigación de los antecedentes, se obtuvieron grandes aportes para el desarrollo del modelo propuesto, tal fue el caso del aporte brindado por el trabajo de Huaman (2019), que proporcionó el esquema de una arquitectura basada tres elementos claves que son Web Scraping, Dialogflow y Firebase. Dicha arquitectura sirvió de guía ya que se ajustaba perfectamente a las necesidades de este proyecto.

Del mismo modo, se logró diseñar un modelo que relaciona las capas de extracción, almacenamiento y presentación de datos, proceso que fue detallado en el capítulo 3 correspondiente a la metodología, siendo la capa de extracción aquella realizada mediante Selenium de Python, la capa de almacenamiento implementada sobre Firestore Database de Firebase y la capa de presentación desarrollada sobre Messenger.

Luego de cumplir con el objetivo de implementar un chatbot en Facebook Messenger, se concluye que el ecosistema actualmente llamado “Meta” es una de las herramientas más potentes al alcance de todo tipo de usuarios, ya que con solo registrarse en su plataforma “Meta Developers” nos brinda un abanico de posibilidades para el desarrollo de aplicaciones que se pueden integrar con servicios externos, como en este caso, se logró crear una aplicación sobre una página de prueba en Messenger llamada BotJobi, que consiste en un chatbot que se conecta a los servicios de Firebase y Dialogflow.

Después de evaluar el modelo implementado, se concluye que las funcionalidades previstas y las que fueron resultado de la encuesta se implementaron satisfactoriamente y cumplen con su objetivo de dar a conocer las habilidades más demandadas entre los empleos para desarrolladores de software en Ecuador.

De forma general, se puede concluir que el desconocimiento de las habilidades digitales o tecnologías más demandadas por las empresas es un problema que influye en el déficit de profesionales capacitados para el área de desarrollo de software, por ello la ejecución de este proyecto contribuye a que los estudiantes de carreras relacionadas al desarrollo de software tengan una guía tanto para decidir en qué capacitarse como para encontrar ofertas laborales que se ajusten a las habilidades que ya posee.

## **2.17. Recomendaciones**

Para futuras implementaciones de este mismo tipo se recomienda la adición de nuevas funcionalidades tales como entrenar al chatbot para que pueda filtrar ofertas laborales según el nombre del cargo al que se desea aplicar.

Como otra mejora al proyecto se recomienda hacer que el chatbot ofrezca sugerencias de cursos online para aprender sobre las habilidades mostradas, respondiendo con mensajes que contengan links hacia los sitios de dichos cursos.

Se recomienda extender el alcance del Web Scraping para recolectar datos de ofertas laborales de toda el área de TI u otras profesiones que demanden habilidades específicas, para así extender la funcionalidad del chatbot hacia estudiantes de otras carreras.

Se recomienda también la implementación de este proyecto sobre otras plataformas de Meta tales como WhatsApp o Instagram, ya que son otras plataformas en las que Meta Developers permite desarrollar aplicaciones y además son preferidas por muchos usuarios.

Para el desarrollo de propuestas similares se recomienda el uso de bases de datos no relacionales ya que son más flexibles a la hora de almacenar de datos y realizar consultas en menor tiempo que las bases de datos relacionales.

Así mismo se recomienda el uso de Selenium como entorno para la ejecución del Web Scraping ya que posibilita funciones dinámicas como dar click, desplazarse a través de las páginas, etc. Lo cual resulta útil para obtener datos de páginas cuya estructura Html es más compleja o hacen uso de JavaScript.



ANEXOS

## Anexo N°1: Planificación de la encuesta

### Problema

El desconocimiento del mercado laboral para el sector de las tecnologías es un factor influyente en el déficit de profesionales capacitados que se necesita para suplir la creciente demanda de los mismos.

### Hipotesis

La implementación de una herramienta amigable permitirá mejorar el conocimiento de la situación del mercado laboral tecnológico en Ecuador.

### Variables

Variable independiente: asistente virtual chatbot.

Variable dependiente: conocimiento del mercado laboral del desarrollo de software.

### Objetivos

Recolectar información provechosa para la implementación de funcionalidades que no hayan sido previstas antes y que puedan servir para hacer del modelo una herramienta más útil y eficaz.

### Desarrollo de la encuesta

#### 1. ¿Qué semestre está cursando actualmente?

Variable	Indicador	#	Pregunta	Alternativas
dependiente	Perfil de navegación del usuario	2	¿Ha visitado alguna vez alguno de los siguientes portales de empleo? Seleccione todos los que apliquen	<ul style="list-style-type: none"> <li>• Computrabajo</li> <li>• Multitrabajo</li> <li>• Red Social LinkedIn</li> <li>• Encuentra Empleo</li> <li>• Acción Trabajo</li> <li>• Porfinempleo</li> <li>• Indeed ec</li> </ul>
	Nivel de conocimiento del mercado laboral	3	¿Cuánto conoce acerca de los cargos más demandados en el campo laboral del desarrollo de software?	<ul style="list-style-type: none"> <li>• Mucho</li> <li>• Medianamente</li> <li>• Poco</li> <li>• Nada</li> </ul>
		4	Seleccione los cargos que conoce o ha escuchado anteriormente:	<ul style="list-style-type: none"> <li>• Programador Front End</li> <li>• Programador Back End</li> <li>• Programador Full Stack</li> <li>• Administrador de bases de datos</li> <li>• QA Tester</li> <li>• Programador web</li> <li>• Programador de Apps móviles</li> <li>• Diseñador UI</li> </ul>

Independiente				<ul style="list-style-type: none"> <li>• Analista de sistemas</li> <li>• No conozco ninguno</li> <li>• Otros</li> </ul>
		5	¿Cuánto conoce acerca de las tecnologías para el desarrollo de software más solicitadas por las empresas en Ecuador actualmente?	<ul style="list-style-type: none"> <li>• Mucho</li> <li>• Medianamente</li> <li>• Poco</li> <li>• Nada</li> </ul>
	accesibilidad	6	¿Ha interactuado alguna vez con un chatbot?	<ul style="list-style-type: none"> <li>• Si / No</li> </ul>
		7	¿Cómo le resulta más cómodo acceder a un chatbot?	<ul style="list-style-type: none"> <li>• Descargando una aplicación dedicada</li> <li>• Mediante redes sociales</li> <li>• Mediante página web</li> </ul>
	Duración de la sesión	8	¿Cuánto tiempo estaría dispuesto a dedicar para interactuar con un chatbot informativo?	<ul style="list-style-type: none"> <li>• De 1 a 5 minutos</li> <li>• De 6 a 10 minutos</li> <li>• De 11 minutos o más</li> </ul>
	Legibilidad de la información	9	¿Cómo preferiría que fueran las respuestas de un chatbot?	<ul style="list-style-type: none"> <li>• Solo texto</li> <li>• Texto e imágenes</li> <li>• Solo imágenes</li> </ul>
		10	En cuanto al contenido de las respuestas de un chatbot, preferiría:	<ul style="list-style-type: none"> <li>• Que conteste con información detallada</li> <li>• Que conteste solo con la información esencial</li> </ul>
	Funcionalidad	11	¿Qué acciones le gustaría que realizara el chatbot para que su conocimiento acerca del mercado laboral del desarrollo de software mejore? Seleccione una o más.	<ul style="list-style-type: none"> <li>• Que muestre la cantidad total de ofertas laborales publicadas</li> <li>• Que redireccione a la página original de la oferta</li> <li>• Que muestre las tecnologías que más demandan las empresas</li> <li>• Que busque ofertas según una tecnología específica</li> <li>• Que me diga el significado de las tecnologías que desconozca</li> <li>• Otros</li> </ul>

## **Anexo N° 2: Código del Web Scraping**

```

from selenium import webdriver
import time
import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore

# opciones de firebase
cred = credentials.Certificate("gcp-projects-6d8cd-firebase-adminsdk-ou4dq-8d8d2d544.json")
firebase_admin.initialize_app(cred)
db = firestore.client()

# opciones de navegador
options = webdriver.ChromeOptions()
options.add_argument('--start-maximized')
options.add_argument('--disable-extensions')

driver_path = '/Users/robertogarcia/Downloads/chromedriver'
driver = webdriver.Chrome(driver_path, options=options)

# inicializamos el navegador
driver.get('https://ec.computarepato.com/trabajo-de-desarrollador')

time.sleep(3)

# identificadores de elementos
ofert_xpath = "//ul[@class='fals pag']/a"
next_btn_xpath1 = "/html/body/main/div[4]/div[2]/div[2]/nav/upon[6]"
next_btn_xpath = "/html/body/main/div[4]/div[2]/div[2]/nav/upon[7]"
description_xpath = "/html/body/main/div[2]/div/div[2]/div[2]/p[1]"
titulo_xpath = "/html/body/main/div[3]/h1"
lugar_xpath = "/html/body/main/div[1]/p"
imagen_xpath = "/html/body/main/div[2]/div/div[3]/div/div[1]/div[1]/img"
array = [ 'JAVASCRIPT', 'JAVA', 'SQL', '.NET', 'C#', 'API REST',
          'ANGULAR', 'VUE.JS', 'PHP', 'NODE.JS', 'LINUX', 'POSTGRES',
          'MYSQL', 'SPRING BOOT', 'GRAPHQL', 'SCRUM', 'DUMS', 'WEB SERVICES', 'REST',
          'SOAP', 'JASPERSREPORT', 'JAVA SERVER PAGES', 'HTML', 'CSS', 'BOOTSTRAP',
          '.NET CORE', 'BAZON', 'SQL SERVER', 'ASP.NET', 'POWERCARDER', 'KAFKA',
          'PYSPARK', 'NEXT.JS', 'T-SQL', 'C++', 'QA', 'CMS', 'RESTFUL API', 'EMBER',
          'REACT', 'IOS', 'FLUTTER', 'BART', 'SIT', 'SITHUN', 'SITLAR', 'SQLITE',
          'PYTHON', 'KOTLIN', 'ANGRIDS', 'ASPX', 'TSQL', 'MVC', 'LARAVEL', 'VISUAL BASIC',
        ]
descripcion = []
counter = {}

i = 2

all_oferts = driver.find_elements_by_xpath(ofert_xpath)

for ofert in all_oferts:
    ofert.click()
    print("nuevo")
    descripcion1 = driver.find_element_by_xpath(description_xpath).text
    lugar1 = driver.find_element_by_xpath(lugar_xpath).text
    titulo1 = driver.find_element_by_xpath(titulo_xpath).text
    url_ofert1 = driver.current_url

    mayus = descripcion1.upper()

    # guardamos array descripcion
    for item in array:
        q = mayus.find(item)
        if q != -1:
            descripcion.append(item)
    print(descripcion)

try:
    imagen1 = driver.find_element_by_xpath(imagen_xpath).get_attribute('src')
    # guardamos en firebase
    db.collection('ofertas1').add({
        'titulo': titulo1,

```

*Figura 50 Código en Python del web scraping Computrabajo. Elaborado por Dayanna Vera (2022)*

```

from selenium import webdriver
import time
import pandas as pd

# opciones de navegador
options = webdriver.ChromeOptions()
options.add_argument('--start-maximized')
options.add_argument('--disable-extensions')
options.add_argument('--incognito')

driver_path = '/Users/dayanna/Downloads/chromedriver'
driver = webdriver.Chrome(driver_path, options=options)

# Inicializamos el navegador
driver.get('https://es.linkedin.com/jobs/openrecruitment-software-engineer?position=1&pageNum=0')

time.sleep(1)

# Variables auxiliares
counter = 0
item = 1
cant = 0

# Identificadores de elementos
xpathAllOfertas = '//*[@id="main-content"]/section[2]/ul/li'
xpath_desc1 = '/html/body/div[1]/div/section/div[2]/div/section[1]/div/div/section/div' #>div
xpath_desc2 = '/html/body/div[1]/div/section/div[2]/div/section/div/div/section/div' #>g

# Recorrer todas las ofertas
while counter < 5:
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(1)
    counter += 1

allOfertas = driver.find_elements_by_xpath(xpathAllOfertas)

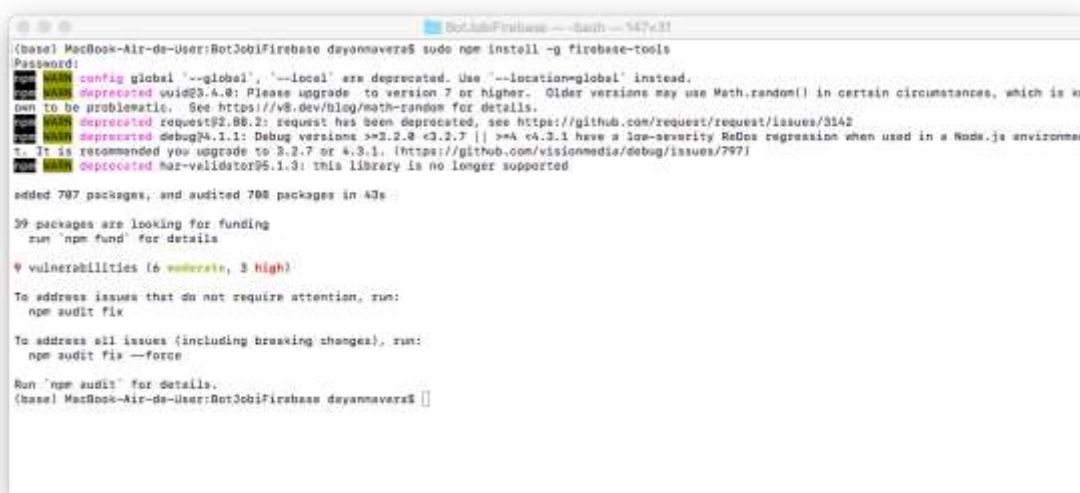
while item < len(allOfertas):
    try:
        # oferta dentro de un <div>
        xpathOp1 = '//*[@id="main-content"]/section[2]/ul/li[' + str(item) + ']/div/a'
        time.sleep(2)
        op1 = driver.find_element_by_xpath(xpathOp1)
        driver.execute_script('arguments[0].click();', op1)
        time.sleep(1)
        desc = driver.find_element_by_xpath(xpath_desc1).text
        cant += 1
        item += 1
    except:
        try:
            # oferta dentro de etiquetas <a>
            xpathOp2 = '//*[@id="main-content"]/section[2]/ul/li[' + str(item) + ']/a/div[2]/a'
            time.sleep(2)
            op2 = driver.find_element_by_xpath(xpathOp2)
            op2.click()
            driver.get(driver.current_url)
            time.sleep(1)
            desc = driver.find_element_by_xpath(xpath_desc2).text
            cant += 1
            item += 1
        except:
            if str(driver.current_url) != 'data:', '':
                driver.back()
                time.sleep(1)
                driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
                item += 1
            else:
                print("no se pueden extraer mas datos, cerrando navegador...")
                driver.close()

```

Figura 51 Código en Python del web scraping LinkedIn. Elaborado por Dayanna Vera (2022)

### Anexo N° 3: Programación del código JavaScript para el chatbot

Para iniciar con el desarrollo del chatbot existen dos opciones, utilizar el editor de código online disponible en el apartado fulfillment de Dialogflow o utilizar el editor de código de preferencia en el entorno local. En el desarrollo de este proyecto se utilizó el editor de código de preferencia y para poder ejecutar el código de forma local se realizó la conexión a firebase mediante el uso de su consola (CLI), para ello se descargaron los paquetes y dependencias necesarias la la implementación dl código, como se muestra a continuación.



```
(base) MacBook-Air-de-User:BotJobiFirebase dayannavera$ sudo npm install -g firebase-tools
Password:
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated har-validator@5.1.3: this library is no longer supported

added 707 packages, and audited 708 packages in 43s

39 packages are looking for funding
  run `npm fund` for details

4 vulnerabilities (1 moderate, 3 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
(base) MacBook-Air-de-User:BotJobiFirebase dayannavera$
```

Figura 52 Instalacion firebase-tools y CLI de Firebase. Elaborado por Dayanna Vera (2022).



```
added 707 packages, and audited 708 packages in 43s

39 packages are looking for funding
  run `npm fund` for details

4 vulnerabilities (1 moderate, 3 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
(base) MacBook-Air-de-User:BotJobiFirebase dayannavera$ firebase login
1. Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

2. Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? Y
3. To change your data collection preference at any time, run "firebase logout" and log in again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=643863238849-farham457hneki16i8d6sr92hg819s8.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformadmin.projects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform-response_type=code&state=656873847&redirect_uri=http%3A%2F%2Flocalhost%3A4080%2F

Waiting for authentication...

✔ Success! Logged in as dayannavera1@gmail.com
(base) MacBook-Air-de-User:BotJobiFirebase dayannavera$
```

Figura 53 Inicio de sesion en la CLI de firebase. Elaborado por Dayanna Vera (2022).





## Anexo 4: Integraciones entre plataformas

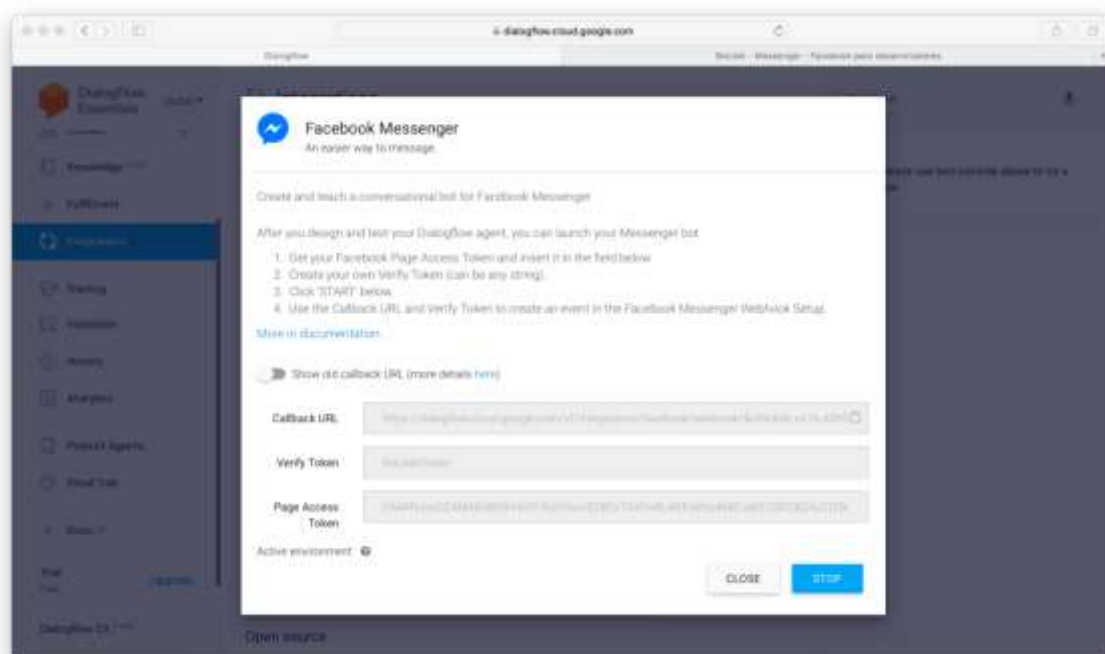


Figura 56 Apartado integraciones, en Dialogflow. Elaborado por Dayanna Vera (2022)

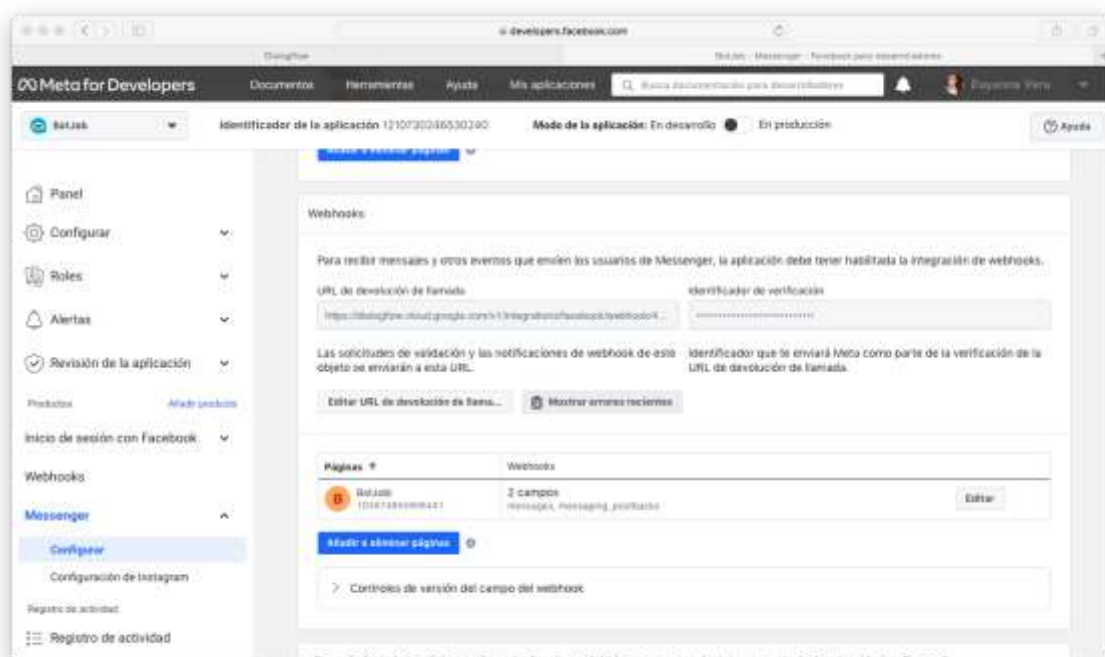


Figura 57 Apartado Webhooks, en Meta for Developers. Elaborado por Dayanna Vera (2022).

## Anexo 5: Entrenamiento del Lenguaje Natural en Dialogflow

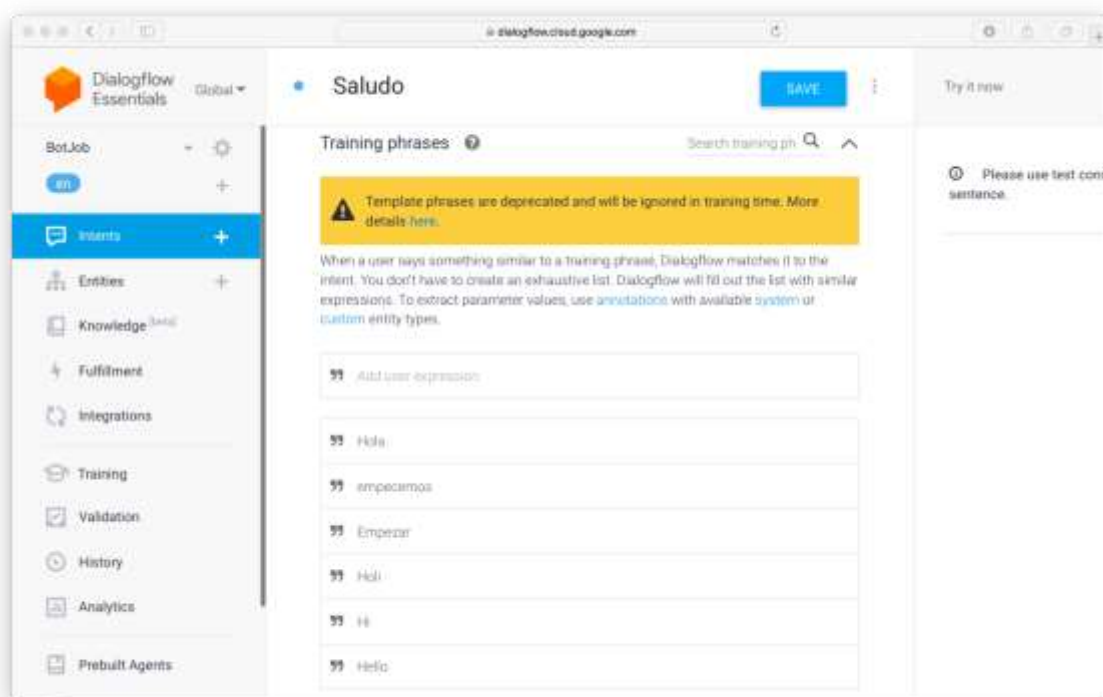


Figura 58 Frases de entrenamiento Intent Saludo. Elaborado por Dayanna Vera (2022)

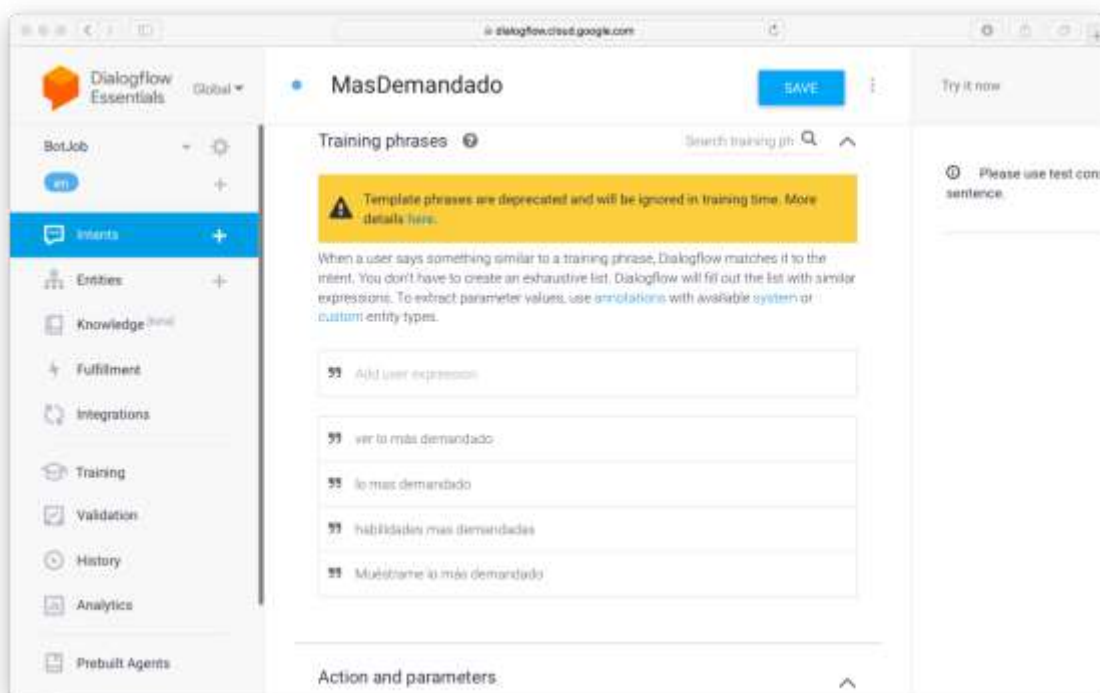


Figura 59 Frases de entrenamiento intent Más Demandado. Elaborado por Dayanna Vera (2022)

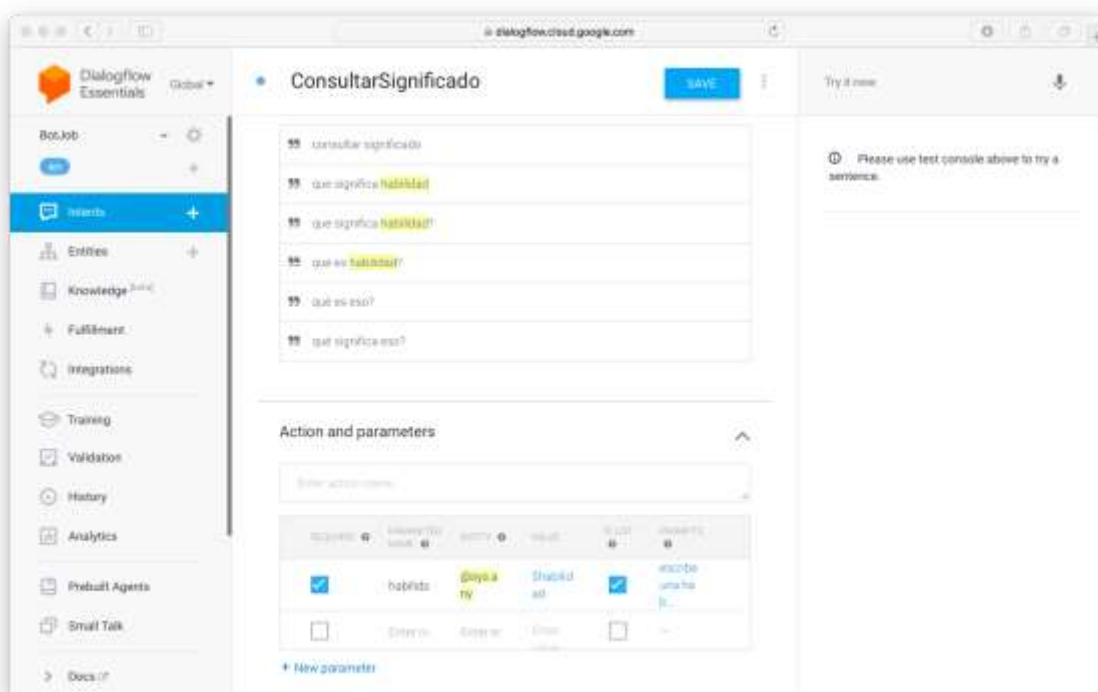


Figura 60 Frases de entrenamiento intent Consultar significado. Elaborado por Dayanna Vera (2022)

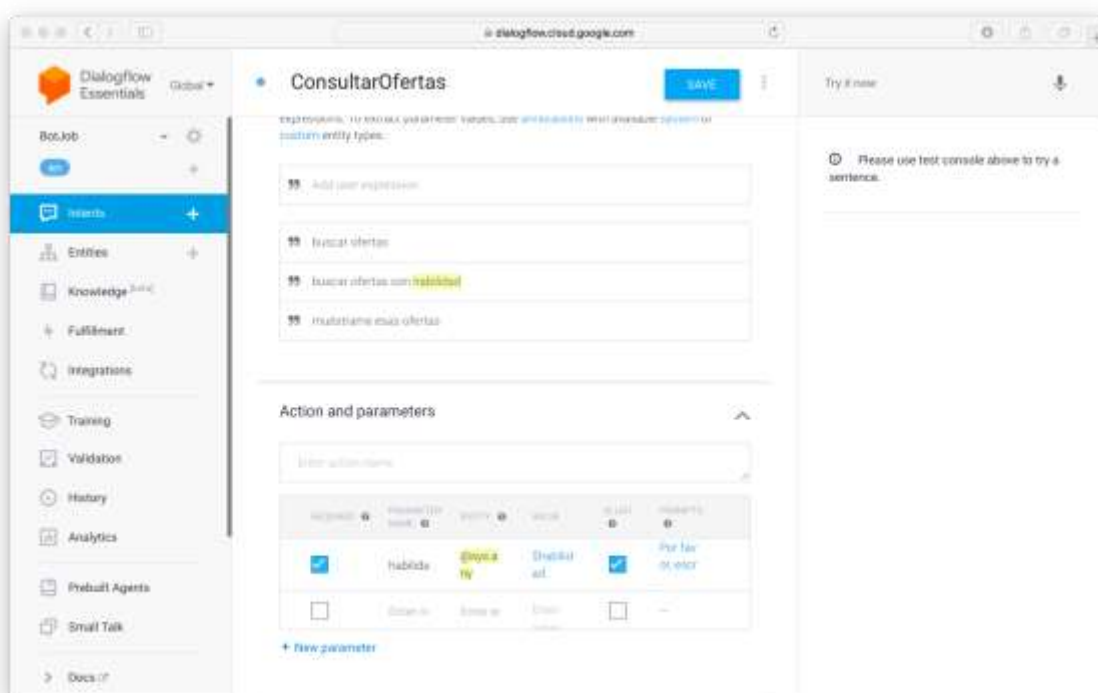


Figura 61 Figura 60 Frases de entrenamiento intent Consultar ofertas. Elaborado por Dayanna Vera (2022)

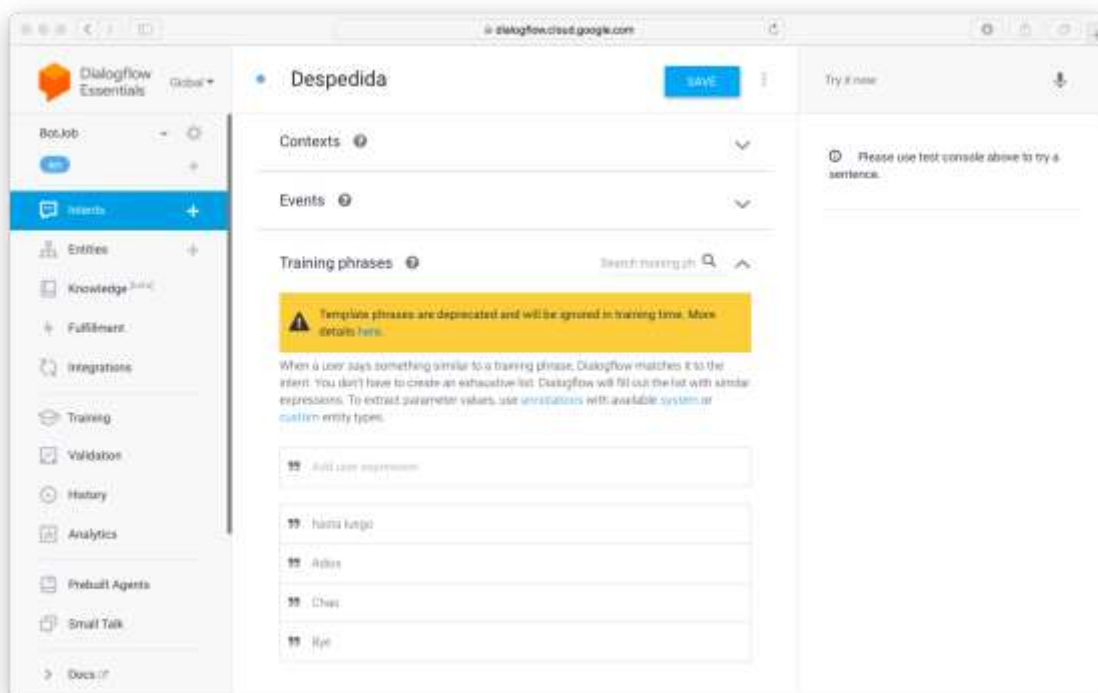


Figura 62 Figura 60 Frases de entrenamiento intent Despedida. Elaborado por Dayanna Vera (2022).

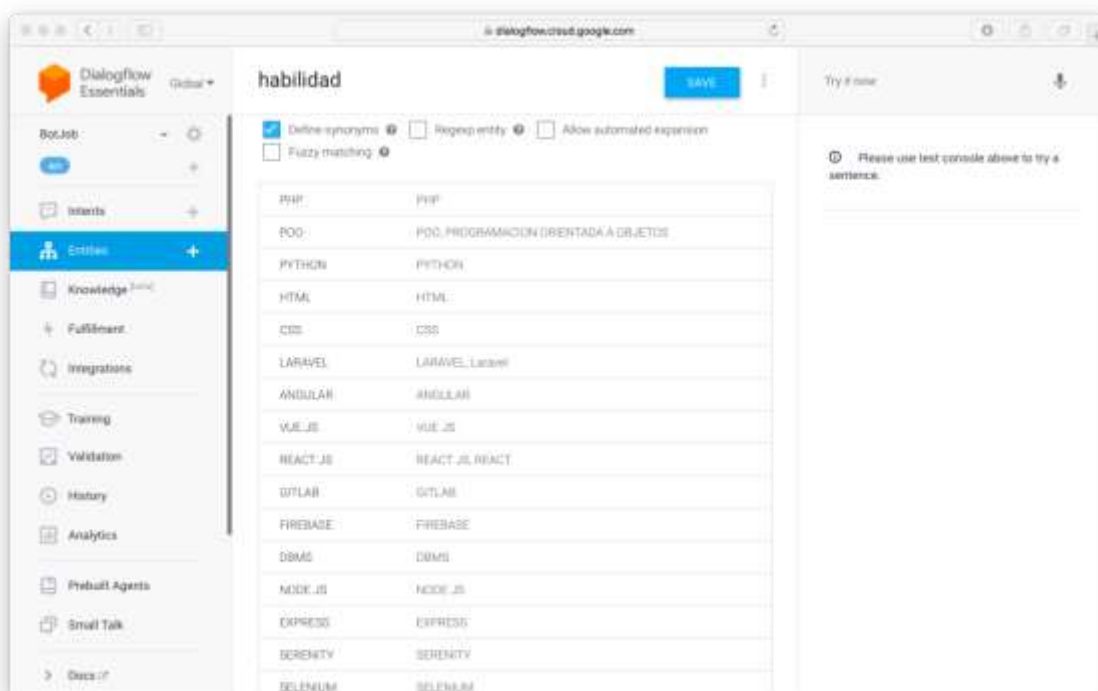


Figura 63 Entity habilidad. Elaborado Dayanna Vera (2022)..

## Anexo N° 6: Manual de usuario

### Introducción

El presente chatbot se desarrolló por la necesidad de dar a conocer cuáles son las habilidades digitales que actualmente más solicitan las empresas en el área del desarrollo de software en Ecuador. Durante la lectura de este documento debe entenderse por habilidades digitales al conjunto de lenguajes de programación, bases de datos, frameworks y demás tecnologías utilizadas en el desarrollo de software, por ejemplo PHP, Django, SQL, etc. La mayor cualidad del presente chatbot es su capacidad de responder con datos reales obtenidos directamente de los portales de empleo Computrabajo y LinkedIn, además le permitirá buscar ofertas por habilidades de forma práctica y eficiente.

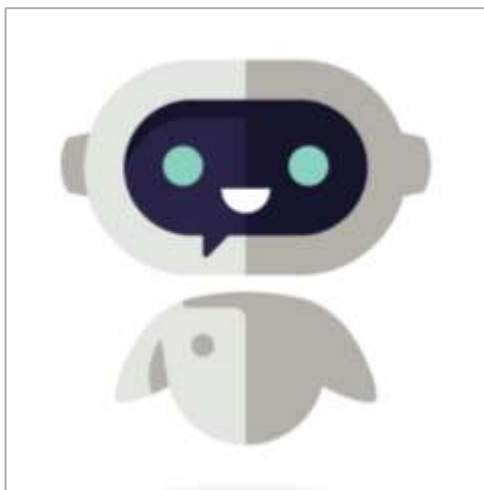
A lo largo de este documento encontrará el apartado *Acceso al Chatbot*, en donde se especifica la manera en que debe acceder al chatbot, enseguida se encuentra el apartado *Diagrama general del sistema*, en donde se presenta la estructura del sistema de forma general y finalmente se tiene la *Interacción con el chatbot*, en donde se detallan los pasos a seguir para hacer un correcto uso al chatbot.

### Acceso al chatbot

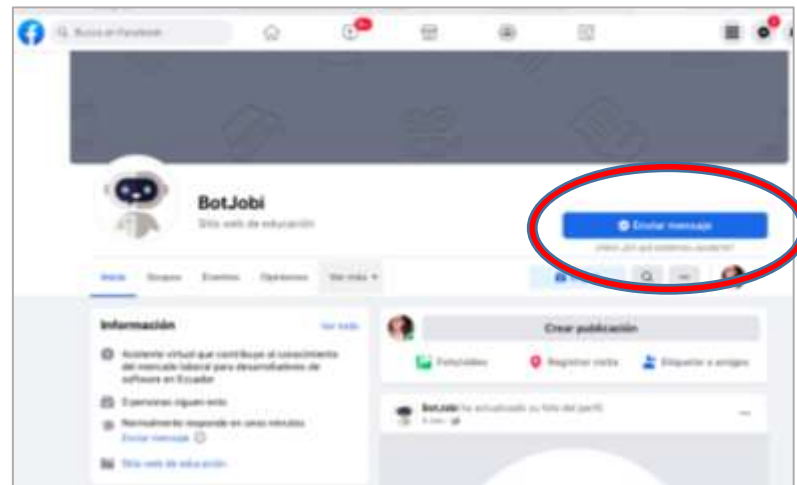
- **Requisitos:** poseer una cuenta de usuario en Facebook, acceso a internet, dispositivo móvil u ordenador.

Para acceder al chatbot realice los siguientes pasos:

1. Iniciar sesión en Facebook con tu usuario y contraseña
2. Buscar la página “BotJobi”, elija la cuenta que tenga la siguiente foto de perfil

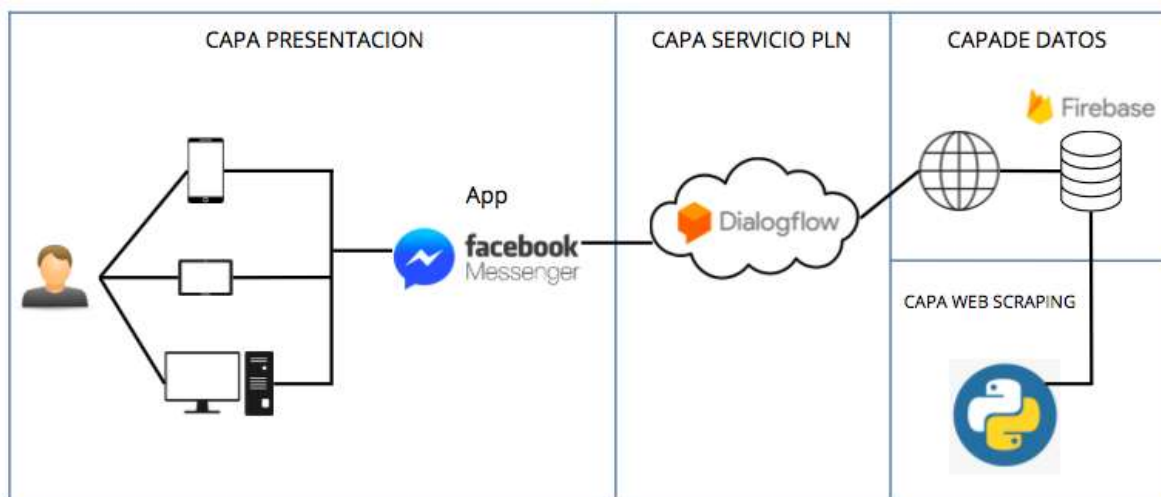


3. Tocar el boton de enviar un mensaje, lo encontrará al inicio de la página ya sea que acceda desde un móvil o un computador



### Diagrama general del sistema

Este chatbot funciona con una base de datos alojada en la nube, y a su vez esta base de datos fue previamente llenada con datos que fueron resultado de un proceso llamado web scraping o raspado web, que consiste en una búsqueda automática de datos sobre una determinada pagina en internet. Así mismo el chatbot utiliza un servicio de Google para realizar el procesamiento del lenguaje natural (PLN) llamado Dialogflow. Para un mejor entendimiento de la estructura detrás de este chatbot, vea la imagen a continuación.



## Interacción con el chatbot

El chatbot cuenta con tres funcionalidades principales que son:


- Consultar lo más demandado
- Consultar significados de las habilidades
- Consultar ofertas por habilidad

Sin embargo, para iniciar con la interacción se recomienda seguir el siguiente flujo:

### 1) Saludar

Al ingresar en el chat de BotJobi aparecerá automáticamente el boton “Hola”, el cual debe tocar para iniciar.

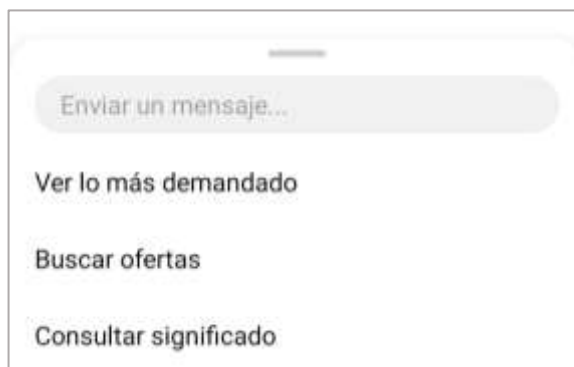


Seguidamente BotJobi responderá con un mensaje introductorio, de ahí en adelante se deberá utilizar las opciones del menú persistente, puede acceder a este desde el icono  que se encuentra a la derecha del campo donde se escribe el mensaje.

**Nota:** si al enviar el primer “Hola” BotJobi responde con un mensaje incorrecto, por favor ingrese nuevamente la palabra “Hola” de forma manual.

## 2) Consultar lo más demandado

Al desplegar el menú persistente, observará opciones como las siguientes:



En este paso deberá elegir “Ver lo más demandado”, y recibirá como respuesta un mensaje de texto con datos obtenidos de la base, podrá identificar estos datos extraídos ya que se encuentran escritos en mayúsculas, los registros numéricos corresponden también a los datos obtenidos del raspado web.

## 3) Consultar significados

Luego de ver lo más demandado probablemente se encuentre con habilidades que no conozca, por esto podrá consultar sus significados. Elija la opción “Consultar significado”, BotJobi le pedirá que ingrese una habilidad **para buscar su significado**, es importante que al **escribir** la habilidad lo haga **todo en letras mayúsculas**, ya que es así como se encuentran almacenadas en la base de datos y por ende si las escribe en minúsculas BotJobi no encontrará ninguna coincidencia.

## 4) Consultar ofertas

Luego de conocer el significado, podrá buscar ofertas laborales, seleccione la opción “Consultar ofertas” e ingrese el nombre de la habilidad en letras mayúsculas. Al obtener los resultados, se mostrarán mensajes de tipo tarjeta y colocados en carrusel según la cantidad



de ofertas que se hayan encontrado hasta un máximo de 10 ofertas, esto debido a una restricción por parte de Messenger.

BotJobi le invitará a interactuar con cualquiera de sus funciones de forma aleatoria, es decir, puede volver a consultar significados, ofertas o ver lo mas demandado de forma aleatoria siempre y cuando vuelva a elegir la opción que desee desde el menu persistente.

### **Perdida de contexto**

Es probable que al consultar un significado o buscar ofertas BotJobi no encuentre resultados, en esos casos para volver a consultar significados u ofertas deberá siempre volver a elegir la opción desde el menú persistente, si por el contrario ingresa una habilidad sin haber seleccionado previamente una opción entonces el chatbot no reconocerá lo que intenta buscar debido a que no tiene un contexto de la habilidad ingresada y mostrará alguno de sus mensajes por defecto.

### **5) Despedida**

Cuando desee finalizar la interacción puede escribir alguna frase de despedida como “Chao”, “Bye”, “Adios”, entre otras. Esto hará que el agente virtual le muestre un mensaje de despedida motivándolo a sacarle provecho a la información adquirida.

## Anexo N°7: proceso de revisión de Meta Developers

Inicialmente el chatbot solo se encontraba disponible para el desarrollador administrador, por ello para disponerlo al público fue necesaria la revisión del mismo por parte del equipo de Meta Developers

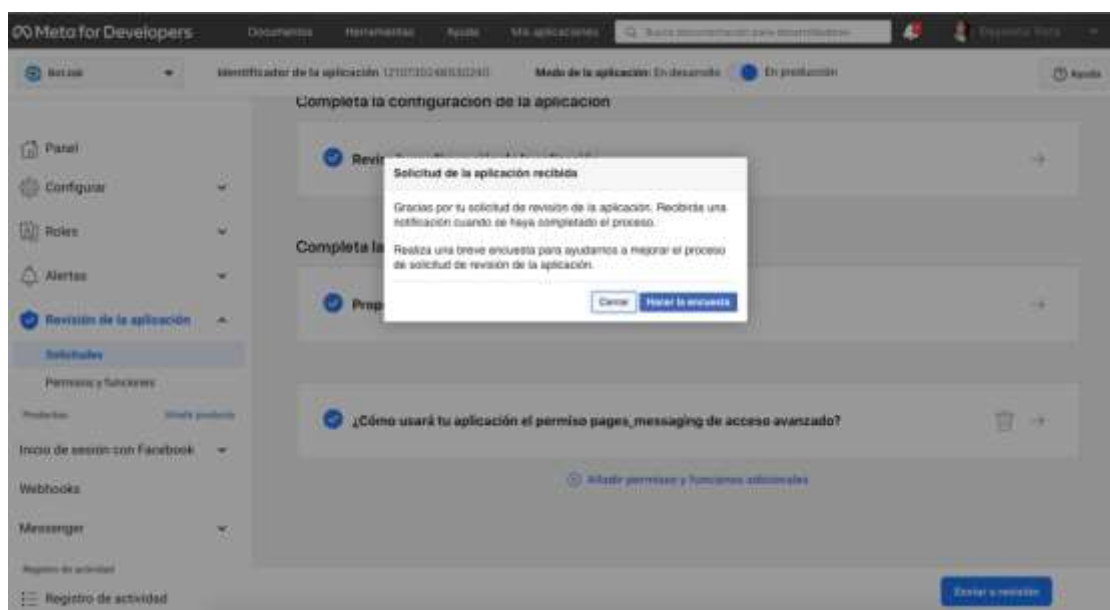


Figura 64 Envío de solicitud de revisión de aplicación, Meta Developers. Elaborado por Dayanna Vera(2022)

## **Bibliografía**

- Gomez, H.(2019). Agente conversacional para búsqueda y recomendación de ofertas laborales. Sevilla. Obtenido de <https://reunir.unir.net/handle/123456789/9564>
- Tigreros, D.(2022). Chatbot para atender las necesidades y mejorar las condiciones del clima laboral del trabajador en el hospital básico Vicente Pino morán ubicado en daule. Guayaquil. Obtenido de: <http://repositorio.ug.edu.ec/handle/redug/60089>
- Múnera, M., Salazar, L., & Osorio, A. (2022). Estudio inicial de un chatbot para estudiantes de la modalidad virtual de la Escuela Interamericana de Bibliotecología. Investigación Bibliotecológica: archivonomía, bibliote cología e información 36 (90): 13-30. Obtenido de: <https://doi.org/10.22201/iibi.24488321xe.2022.90.58452>
- Castro, L. (2021). Chatbot conversacional de apoyo al uso del aula extendida de la Universidad Simón Bolívar. Antioquia. Obtenido de: <https://hdl.handle.net/20.500.12442/8103>
- Murgu, A.(2021). Desarrollo de un ChatBot para la asistencia eficiente y automática a los estudiantes desde la Delegación de Alumnos de la ETSISI. Madrid. Obtenido de: <https://oa.upm.es/68319/>
- Llorens, J. (2021). Aplicación web para obtener de manera automática un estado del arte de un tema de investigación. Alicante. <http://hdl.handle.net/10045/115976>
- Sánchez, D. (2020). Plataforma de recomendación de habilidades tecnológicas según puesto de trabajo para profesionales de TI, en función de la demanda en las bolsas de trabajo digitales. Lima. <https://hdl.handle.net/20.500.12724/12351>
- Hernández, A. T., Vázquez, E. G., Rincón, C. A. B., García, J. M., Maldonado, A. C., & Ibarra-Orozco, R. (2015). Metodologías para análisis político utilizando Web Scraping. Res. Comput. Sci., 95, 113-121.
- Richardson, L. (2007). Beautiful soup documentation. Dosegljivo: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [Dostopano: 7. 7. 2018].
- Vásquez, A. C., Quispe, J. P., & Huayna, A. M. (2009). Procesamiento de lenguaje natural. Revista de investigación de Sistemas e Informática, 6(2), 45-54.

- Fernández, D. (2019). Integración en ROS de herramientas de Web Scraping para análisis de video. <https://riull.ull.es/xmlui/bitstream/handle/915/13471/Integracion%20en%20ROS%20de%20herramientas%20de%20Web%20Scraping%20para%20 analisis%20de%20video.pdf?sequence=1>
- International Data Corporation (2021, diciembre 14). Se espera un crecimiento de inversión de TI del 9,4% para 2022 en America Latina: IDC. <https://www.idc.com/getdoc.jsp?containerId=prLA49040422>
- Krotov, V., & Silva, L. (2018, septiembre). Legality and Ethics of Web Scraping. Retrieved August 5, 2022, from [https://www.researchgate.net/profile/Vlad-Krotov/publication/324907302\\_Legality\\_and\\_Ethics\\_of\\_Web\\_Scraping/links/5aea622345851588dd8287dc/Legality-and-Ethics-of-Web-Scraping.pdf](https://www.researchgate.net/profile/Vlad-Krotov/publication/324907302_Legality_and_Ethics_of_Web_Scraping/links/5aea622345851588dd8287dc/Legality-and-Ethics-of-Web-Scraping.pdf)
- Jia, J. (2003). The Study of the Application of a Keywords-based Chatbot System on the Teaching of Foreign Languages, ArXiv preprint cs/0310018, <https://arxiv.org/abs/cs/0310018>
- Arias, Á. (2014). Bases de Datos con MySQL: 2ª Edición. IT Campus Academy. <https://books.google.es/books?hl=es&lr=&id=EojJCgAAQBAJ&oi=fnd&pg=PA13&dq=que+son+bases+de+datos&ots=FFd9ZNiU8c&sig=zJOcxPFU4OJKiWvD5zRG-v9Jn2U#v=onepage&q=que%20son%20bases%20de%20datos&f=false>
- Sabharwal, N., & Agrawal, A. (2020). Introduction to Google Dialogflow. In: Cognitive Virtual Assistants Using Google Dialogflow. Apress, Berkeley, CA. [https://doi.org/10.1007/978-1-4842-5741-8\\_2](https://doi.org/10.1007/978-1-4842-5741-8_2)
- Dryer, A.J., and Stockton, J. 2013. "Internet 'Data Scraping': A Primer for Counseling Clients," New York Law Journal. Retrieved from <https://www.law.com/newyorklawjournal/almID/1202610687621>
- Código Orgánico Integral Penal (2014). [https://tbinternet.ohchr.org/Treaties/CEDAW/Shared%20Documents/ECU/INT\\_CEDAW\\_ARL\\_ECU\\_18950\\_S.pdf](https://tbinternet.ohchr.org/Treaties/CEDAW/Shared%20Documents/ECU/INT_CEDAW_ARL_ECU_18950_S.pdf)
- Sommerville, "Ingeniería del Software", ed. S. Edición. 2009, Madrid, España. [https://books.google.com.ec/books/about/Ingenier%C3%ADa\\_del\\_software.html?i](https://books.google.com.ec/books/about/Ingenier%C3%ADa_del_software.html?i)

d=gQWd49zSut4C&printsec=frontcover&source=kp\_read\_button&hl=es-419&redir\_esc=y#v=onepage&q&f=false

Brookshear J.(1993). Teoría de la computación Addison Wesley iberoamericana  
Wilmington Delaware. <https://dialnet.unirioja.es/servlet/libro?codigo=373480>