



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TEMA:**

Desarrollo de un prototipo de Chatbot en WhatsApp para acceder a información en tiempo real sobre la producción de una empresa dedicada al procesamiento y empaque de camarones de la ciudad de Guayaquil.

**AUTOR:**

**Bermúdez Mora, Jonathan David**

**Trabajo de titulación previo a la obtención del título de  
INGENIERO EN SISTEMAS COMPUTACIONALES**

**TUTOR:**

**Ing. Yong Yong, Byron Severo**

**Guayaquil, Ecuador  
02 de septiembre de 2024**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

## CERTIFICACIÓN

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por **Bermúdez Mora, Jonathan David**, como requerimiento para la obtención del título de **Ingeniero en Sistemas Computacionales**.

## TUTOR

1202728380  
BYRON SEVERO  
YONG YONG

Nombre de reconocimiento (DN):  
1.3.6.1.4.1.37442.10.4=1202728380, o=YONG  
YONG BYRON SEVERO, ou=Certificado Persona  
Jurídica EC (FIRMA), givenName=BYRON SEVERO,  
2.5.4.97=1202728380001, c=EC,  
serialNumber=1202728380, sn=YONG YONG,  
cn=1202728380 BYRON SEVERO YONG YONG,  
email=byron.yong@cu.ucsg.edu.ec,  
2.5.4.13=Certificado para Persona Natural con RUC,  
st=GUAYAS, I=MARISCAL SUCRE  
Fecha: 2024.09.18 08:22:47 -05'00'

f. \_\_\_\_\_  
**Ing. Yong Yong, Byron Severo**

Guayaquil, a los 02 días del mes de septiembre del año 2024



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

## DECLARACIÓN DE RESPONSABILIDAD

Yo, **Bermúdez Mora, Jonathan David**

### DECLARO QUE:

El Trabajo de Titulación, **Desarrollo de un prototipo de Chatbot en WhatsApp para acceder a información en tiempo real sobre la producción de una empresa dedicada al procesamiento y empaque de camarones de la ciudad de Guayaquil**, previo a la obtención del título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

**Guayaquil, a los 02 días del mes de septiembre del año 2024**

### EL AUTOR



firmado digitalmente por  
JONATHAN DAVID  
BERMUDEZ MORA

f. \_\_\_\_\_  
**Bermúdez Mora, Jonathan David**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

## AUTORIZACIÓN

Yo, **Bermúdez Mora, Jonathan David**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Desarrollo de un prototipo de Chatbot en WhatsApp para acceder a información en tiempo real sobre la producción de una empresa dedicada al procesamiento y empaque de camarones de la ciudad de Guayaquil**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

**Guayaquil, a los 02 días del mes de septiembre del año 2024**

### EL AUTOR



Escaneado digitalmente por  
**JONATHAN DAVID  
BERMUDEZ MORA**

f. \_\_\_\_\_  
**Bermúdez Mora, Jonathan David**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

**Ing. Ana Isabel Camacho Coronel  
DECANO O DIRECTOR DE CARRERA**

f. \_\_\_\_\_

**Ing. Edison Toala  
COORDINADOR DEL ÁREA O DOCENTE DE LA CARRERA**

f. \_\_\_\_\_

**Ing. Roberto García.  
OPONENTE**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

## REPORTE COMPILATIO

 CERTIFICADO DE ANÁLISIS  
magister

Jonathan Bermudez \_TIC\_ISC  
24-08-20

**< 1%**  
Textos  
sospechosos



- 0% Similitudes  
0% similitudes entre comillas  
0% entre las fuentes mencionadas
- 2% Idiomas no reconocidos (ignorado)
- 11% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: Jonathan Bermudez \_TIC\_ISC 24-08-20.pdf  
ID del documento: 2511c01ac11834d2e10427acc1dfc66b3bd653e7  
Tamaño del documento original: 5,25 MB  
Autores: []

Depositante: Byron Severo Yong Yong  
Fecha de depósito: 20/8/2024  
Tipo de carga: interface  
fecha de fin de análisis: 20/8/2024

Número de palabras: 19.274  
Número de caracteres: 140.142

TUTOR

鄺勇

Firmado digitalmente por  
1202728380 BYRON SEVERO  
YONG YONG  
Fecha: 2024.08.20 16:49:00 -05'00'  
Versión de Adobe Acrobat  
Reader: 2024.002.20991

f. \_\_\_\_\_  
Yong Yong, Byron Severo

## **AGRADECIMIENTO**

Quisiera iniciar expresando mi profundo agradecimiento a Dios por iluminar mi camino hasta este momento crucial en el cual debo defender mi trabajo de titulación. También quiero extender mi gratitud a Empacadora Grupo Granmar Empagran S.A., una empresa privada que generosamente me ha brindado la oportunidad de realizar mi investigación de titulación. Además, no puedo dejar de mencionar el invaluable apoyo de María del Carmen Mora Pérez, mi tía querida, quien ha estado a mi lado desde los inicios de mis estudios superiores, ofreciéndome su guía y aliento incondicional. A todos ellos, mi más sincero agradecimiento por creer en mí y por contribuir al éxito de este importante paso en mi carrera académica.

Y finalmente, quiero manifestar mi gratitud a mis docentes y al personal que labora dentro de la universidad quienes con su apoyo ayudaron a formarme como profesional. A lo largo de la carrera fueron constantes en su colaboración y en cada clase me ayudaron a forjar y permitieron demostrar mi experiencia como profesional. Además, quiero agradecer a mi tutor, Byron Yong, por su paciencia y su orientación que fueron de gran ayuda para desarrollar este proyecto.

## **DEDICATORIA**

Quisiera dedicar este trabajo de titulación con profundo amor y gratitud a mis padres, Margarita y Vicente. Desde el primer día hasta hoy, su apoyo incondicional ha sido la fuerza impulsora detrás de cada paso que he dado en la vida. Este logro es tanto de ustedes como mío.

A mi hermano Paul, quien siempre ha sido un ejemplo vivo de disciplina y perseverancia, a mis amigos, Anthony, Stephanie, Mario y Liss, por su inquebrantable apoyo y aliento durante esta travesía. Vuestras palabras de ánimo y compañía fueron un impulso invaluable para culminar esta etapa de mi vida. Así mismo dedico este trabajo a Mari, por haber sido pilar fundamental en la decisión de volver a retomar mis estudios.



# ÍNDICE

AGRADECIMIENTO.....	VII
DEDICATORIA.....	VIII
INDICE DE IMÁGENES .....	XII
ÍNDICE DE TABLAS .....	XIV
Resumen.....	XV
Abstract .....	XVI
Introducción.....	2
Capítulo I: El Problema .....	5
ANÁLISIS DEL PROBLEMA. ....	5
Ubicación del Problema en un Contexto .....	5
Causas y Consecuencias del Problema.....	5
Delimitación del Problema .....	7
Formulación del Problema .....	7
Evaluación del Problema .....	7
OBJETIVOS .....	8
Objetivo General .....	8
Objetivos Específicos.....	8
ALCANCES DEL PROBLEMA .....	9
JUSTIFICACION E IMPORTANCIA .....	11
PREGUNTA DE INVESTIGACIÓN.....	12
VARIABLES DE LA INVESTIGACIÓN .....	12
Capítulo II: Marco Teórico .....	14
2.1 Chatbot.....	14
2.2 Plataforma Whatsapp.....	15
2.3 Software. ....	17
2.3.1 Lenguajes de programación.....	17
2.3.2 Lenguajes de programación: Python. ....	17
2.3.3 PIP.....	19
2.3.4 Entorno virtual virtualenv.....	20
2.3.5 Framework Flask. ....	20
2.3.6 WebHook .....	20
2.3.7 Biblioteca Request.....	21

2.3.8 API.....	21
2.3.8.1 Tipos de APIS.....	22
2.3.8.2 API Web.....	23
2.3.9 Navegador Web.....	23
2.3.10 C#.....	24
2.3.11 Bases de datos.....	24
2.3.11.1 Tipos de base de datos.....	26
2.3.11.2 Sistema de gestión de base de datos.....	26
2.3.12 Entorno de desarrollo.....	27
2.3.12.1 Visual Studio Code.....	28
2.3.12.2 Microsoft Visual Studio.....	29
2.3.12.3 Framework .NET.....	30
2.3.13 Servidor Web.....	30
2.3.13.1 Internet Information Services.....	32
2.3.14 Windows Server.....	33
¿Por qué utilizar Visual Studio y Visual Studio Code para el desarrollo del prototipo?.....	34
¿Por qué utilizar Python y Flask para el desarrollo del prototipo?.....	35
Capítulo III: Metodología De La Investigación.....	36
Investigación Descriptiva y Explicativa.....	36
POBLACIÓN Y MUESTRA.....	36
Población:.....	36
Muestra:.....	37
Instrumentos de recolección de datos.....	37
Tabulación de encuestas.....	39
Metodología de desarrollo de software.....	46
Capítulo IV: Propuesta Tecnológica.....	48
Levantamiento de información.....	48
Diseño del prototipo.....	49
Arquitectura de la solución.....	49
Origen de Datos.....	50
Modelo entidad relación.....	51
Proceso de extracción, transformación y carga de datos.....	51
Abstracción de datos mediante procedimientos almacenados.....	51

Transformación de datos mediante procedimientos almacenados. ....	53
Depuración de los datos en los procedimientos almacenados. ....	54
Diagrama de casos de uso .....	55
Ambiente de construcción del prototipo.....	55
Instalación de VirtualEnv. ....	55
Instalación de Flask .....	55
Instalación de la librería Request.....	56
Sistema de Archivos .....	56
Desarrollo de funciones del prototipo. ....	58
Autenticación de usuarios. ....	66
Implementación del prototipo. ....	67
Nivel de aceptación.....	71
Presupuesto de implementación.....	72
CONCLUSIONES.....	74
RECOMENDACIONES .....	76
REFERENCIAS BIBLIOGRÁFICAS .....	77
GLOSARIO .....	81
ANEXOS .....	84
Anexo 1. Levantamiento de información para el desarrollo del prototipo .....	84
Anexo 2. Manual de usuario para crear una aplicación en Glitch.....	84
Anexo 3. Tabulación de Encuesta de aceptación del prototipo. ....	92
Anexo 4. Resultados de Encuesta 1.....	93
Anexo 5. Resultados de Encuesta 2.....	93
Anexo 6. Código Fuente del prototipo y de la API. ....	93

## INDICE DE IMÁGENES

Figura 1 Evolución de las exportaciones.....	3
Figura 2 Camarón: Participación por destino .....	3
Figura 3 Diagrama Ishikawa.....	6
Figura 4 Uso de aplicación.....	16
Figura 5 Estructura funcional de una API.....	22
Figura 6 Componentes de una base de datos .....	25
Figura 7 Esquema de un servidor web.....	31
Figura 8 Infraestructura del prototipo .....	34
Figura 9 Estructura organizacional.....	37
Figura 10 Resultados pregunta 1. ....	39
Figura 11 Resultados pregunta 2. ....	39
Figura 12 Resultados pregunta 3. ....	40
Figura 13 Resultado pregunta 4.....	40
Figura 14 Resultado pregunta 5.....	41
Figura 15 Resultado pregunta 6.....	41
Figura 16 Resultado pregunta 7.....	42
Figura 17 Resultado pregunta 8.....	43
Figura 18 Resultados pregunta 9. ....	43
Figura 19 Resultados pregunta 10. ....	44
Figura 20 Resultados pregunta 11. ....	44
Figura 21 Resultados pregunta 12. ....	45
Figura 22 Resultados pregunta 13. ....	45
Figura 23 Resultados pregunta 14. ....	46
Figura 24 Modelo de la metodología en cascada.....	47
Figura 25 Diagrama de Flujo de conversación.....	49
Figura 26 Arquitectura de la solución.....	50
Figura 27 Tablas de la base de datos SARH. ....	50
Figura 28 Modelo Entidad-Relación Base de datos sistemas Empaprod.....	51
Figura 29 Procedimientos almacenados .....	52
Figura 30 Estructura de un procedimiento de forma modular. ....	52
Figura 31 Ejemplo de parámetros de entrada de los procedimientos almacenados.....	53
Figura 32 Conversión de formatos y unificación de unidades. ....	53

Figura 33	Uso de Joins y manejo de subconsultas. ....	54
Figura 34	Ejemplo de trabajo automático desarrollado para el prototipo. ....	54
Figura 35	Diagrama de caso de uso. ....	55
Figura 36	Instalación VirtualEnv.....	55
Figura 37	Instalación de Flask. ....	56
Figura 38	Instalación de Request.....	56
Figura 39	Estructura básica de carpetas del proyecto. ....	56
Figura 40	Ubicación en el terminal de la estructura y carga en el entorno virtual. ...	57
Figura 41	Estructura de archivos dentro del entorno de desarrollo.....	57
Figura 42	Función verificar Token.....	58
Figura 43	Función Recibir mensaje.....	59
Figura 44	Código fuente sett.py. ....	60
Figura 45	Función obtener mensaje whatsapp. ....	60
Figura 46	Función administrar chatbot.....	61
Figura 47	Función text Mensaje.....	62
Figura 48	Ejemplo de invocación de función.....	62
Figura 49	Función buttonReply_Message(). ....	62
Figura 50	Invocación del método buttonReply_Message.....	63
Figura 51	Funcion listReply_Message(). ....	63
Figura 52	Invocación del método listReply_Message(). ....	64
Figura 53	Función enviar_Mensaje_whataspp() ....	65
Figura 54	Función ConsultaApiProduccion. ....	65
Figura 55	Invocación de la función ConsultaApiProduccion.....	66
Figura 56	Autenticación de los usuarios en el prototipo. ....	67
Figura 57	Pantalla Principal de la plataforma Glitch.....	68
Figura 58	Estructura de archivos plataforma.....	68
Figura 59	Contenido archivo requirements.txt.....	69
Figura 60	Log de carga de requisitos. ....	69
Figura 61	Ventana de compartir para obtener URL pública del proyecto.....	70
Figura 62	Pantalla principal de portal Meta for Developers.....	70
Figura 63	Configuración de la URL y el Token.....	71

## ÍNDICE DE TABLAS

Tabla 1 Criterios de Limitación de Problema.....	7
Tabla 2 Criterios de Evaluación del Problema .....	8
Tabla 3 Población Jefaturas.....	36
Tabla 4 Aceptación del prototipo.....	71
Tabla 5 Costos a considerar para una implementación completa.....	72
Tabla 6 Costos de implementación en la empresa. ....	73

## Resumen

El objetivo principal de este proyecto es desarrollar un prototipo de chatbot en WhatsApp para acceder en tiempo real a la información de la producción de una empresa dedicada al procesamiento y empaque de camarones. El marco teórico se fundamenta en la importancia de los chatbots en la mejora de la eficiencia operativa mediante la automatización de consultas y la integración de sistemas de información en tiempo real. La metodología empleada incluyó el análisis del problema con el diagrama de Ishikawa, identificando causas como la dependencia de procesos manuales, el uso de hojas de cálculo, y las limitaciones en la generación de reportes, lo que resultó en una obtención ineficiente y un acceso no oportuno a los datos. Para abordar esto, se utilizó la investigación descriptiva y explicativa, detallando las necesidades de los usuarios y la relación entre el prototipo y la mejora en el acceso a la información. Para el desarrollo del prototipo se adoptó el método en cascada, utilizando las herramientas de desarrollo: Python y Flask, integrando APIs y bases de datos relacionales. La puesta en marcha de este prototipo va a servir para que los gerentes y jefes puedan acceder a los datos críticos de la producción sin la necesidad de estar frente a un computador, de esta forma ayuda a la toma de decisiones, así como también una visión más amplia en la eficiencia operativa del modelo de negocio. El contenido del trabajo abarca el desarrollo y prueba del prototipo, integrando tanto las API de la empresa como las de WhatsApp; incluye el análisis de datos, la creación de modelos de entidad-relación y la implementación de una interfaz amigable para el usuario final. Las conclusiones destacan la efectividad del prototipo en la optimización del acceso a información y su potencial para ser implementado en otras áreas de la empresa debido a su alta aceptación por parte de los usuarios.

**Palabras Clave:** *Chatbot, producción, camarones, WhatsApp, python.*

## Abstract

The main objective of this project is to develop a WhatsApp chatbot prototype to access real-time production information for a shrimp processing and packaging company. The theoretical framework is based on the importance of chatbots in improving operational efficiency through automated queries and real-time information system integration. The methodology employed included problem analysis using the Ishikawa diagram, identifying causes such as dependency on manual processes, use of spreadsheets, and limitations in report generation, resulting in inefficient data retrieval and untimely access to information. To address this, descriptive and explanatory research was used, detailing user needs and the relationship between the prototype and improved information access. The waterfall development method was adopted, using Python and Flask, and integrating APIs and relational databases. This project is significant for the company as it allows managers and supervisors to access critical data without needing a computer, enhancing decision-making and operational efficiency. The content of the work covers the development and testing of the prototype, integrating both the company's APIs and WhatsApp's APIs; it includes data analysis, creation of entity-relationship models, and the implementation of a user-friendly interface. The conclusions highlight the effectiveness of the prototype in optimizing information access and its potential to be implemented in other areas of the company due to its high acceptance by users.

**Key words:** *Chatbot, production, shrimp, WhatsApp, python.*



## Introducción

El cultivo de camarón es una de las industrias más importantes de Ecuador, contribuyendo significativamente al PIB<sup>1</sup> y generando empleo para miles de ecuatorianos. Al mismo tiempo, los avances en tecnología: análisis de datos e inteligencia artificial han revolucionado diversas industrias, optimizando procesos y mejorando la eficiencia operativa; en este contexto, el desarrollo de un chatbot<sup>2</sup> presenta una oportunidad innovadora para mejorar la gestión y producción dentro de las empresas de este sector.

El Ecuador inicia el cultivo y explotación camaronera a partir de 1980 y desde 1982 ha experimentado un crecimiento en la producción del crustáceo logrando posicionarse como uno de los principales exportadores a nivel mundial. Con este crecimiento surge la necesidad de adoptar tecnologías avanzadas y prácticas de cultivos sostenibles; a pesar de eso la industria ha enfrentado varios desafíos como brotes de enfermedades y cambios en los precios a nivel internacional.

En el período de 1990 a 2000 la industria camaronera sufrió la aparición del virus de la mancha blanca, lo que impulsó a la práctica de medidas de bioseguridad e investigación de soluciones tecnológicas para poder disminuir los riesgos. Desde el 2010 se empezó a modernizar los cultivos camaroneros, dando las herramientas necesarias para tener un monitoreo preciso y tomas de decisiones mucho más informadas.

Las exportaciones de camarón según datos oficiales presentados por la cámara de acuicultura (“REVISTA AQUACULTURA”, 2024) presentan a partir del año 2021 un aumento significativo de las toneladas exportadas del crustáceo posicionando a China como el principal comprador del Ecuador.

---

<sup>1</sup> Producto Interno Bruto.

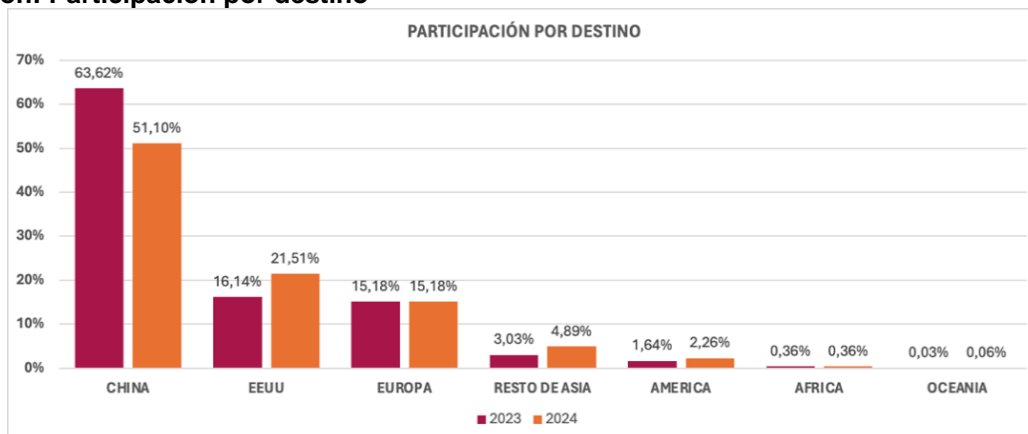
<sup>2</sup> Programa de mensajería que simula una conversación humana.

**Figura 1**  
**Evolución de las exportaciones**



**Nota:** Se evidencia la evolución significativa de las exportaciones del camarón a partir del año 2010. (“REVISTA AQUACULTURA”, 2024).

**Figura 2**  
**Camarón: Participación por destino**



**Nota:** El mercado principal de consumo de camarón ecuatoriano es el chino. (“REVISTA AQUACULTURA”, 2024).

Parte de las empresas que han aportado al crecimiento de estos indicadores es la empacadora donde se va a desarrollar el prototipo la cual tiene sede en la ciudad de Guayaquil, Ecuador, la cual inició su operación en 1982 con el objetivo de ser partícipe de la industria camaronera. Su enfoque fue la crianza y cultivo de camarón de la mejor calidad, aprovechando de las condiciones naturales de la costa ecuatoriana para el cultivo del crustáceo.

Durante las décadas de 1980 y 1990, la empacadora se expandió y consolidó, invirtiendo en tecnología avanzada y desarrollando una infraestructura sólida con laboratorios de larvas, centros de engorde y plantas de procesamiento. Mejoró sus prácticas de cultivo, lo que incrementó su capacidad de producción y la calidad de sus productos.

En las últimas dos décadas, la empacadora ha enfatizado la innovación y la sostenibilidad como pilares de su estrategia empresarial; ha implementado métodos acuícolas sostenibles, como el uso de sistemas de recirculación de agua, alimentación balanceada y control biológico de enfermedades; estas iniciativas también mejoran la eficiencia y la calidad del producto final; logrando obtener varias certificaciones internacionales, entre las que se destacan: Aquaculture Stewardship Council (ASC), Best Aquaculture Practices (BAP) y Global G.A.P., mismas que garantizan que los productos cumplen con rigurosos estándares en términos de prácticas ambientales, sociales y de seguridad alimentaria; convirtiéndola una de las principales exportadoras de camarón de Ecuador, con una fuerte presencia en mercados internacionales: Estados Unidos, Europa y Asia.

Considerando la creciente relevancia de la interacción automatizada en la plataforma de mensajería WhatsApp y respaldada por la accesibilidad generalizada de su interfaz entre la mayoría de los usuarios, surge la posibilidad de crear una nueva herramienta destinada a facilitar la consulta de datos en tiempo real. La propuesta permite transferir el soporte de la aplicación WhatsApp, permitiendo focalizar los esfuerzos en la depuración, refinamiento y presentación de datos para el óptimo aprovechamiento por parte de los usuarios de la aplicación.

Este proyecto busca gestionar mejor los datos al permitir el acceso directo a la información en tiempo real a través de WhatsApp, eliminando la necesidad de recurrir a otros departamentos o esperar la instalación de un sistema más complejo; esta estrategia aceleraría la obtención de información crucial para la gerencia de producción y mejoraría significativamente la experiencia del usuario al brindar respuestas rápidas y accesibles en cualquier momento del día, ofreciendo un enfoque ágil y eficiente que permite obtener información instantánea sin depender de procedimientos burocráticos, contribuyendo a una toma de decisiones más informada y oportuna, y fortaleciendo así la eficacia operativa del proceso.

# Capítulo I: El Problema

En este Capítulo se describe la situación problema de la obtención de información sobre la producción. Los sistemas existentes, aunque acceden a información de la producción camaronera no proporciona una herramienta que permita el acceso a datos en tiempo real lo cual limita la capacidad de tomar decisiones informadas y oportunas.

## **ANÁLISIS DEL PROBLEMA.**

### **Ubicación del Problema en un Contexto**

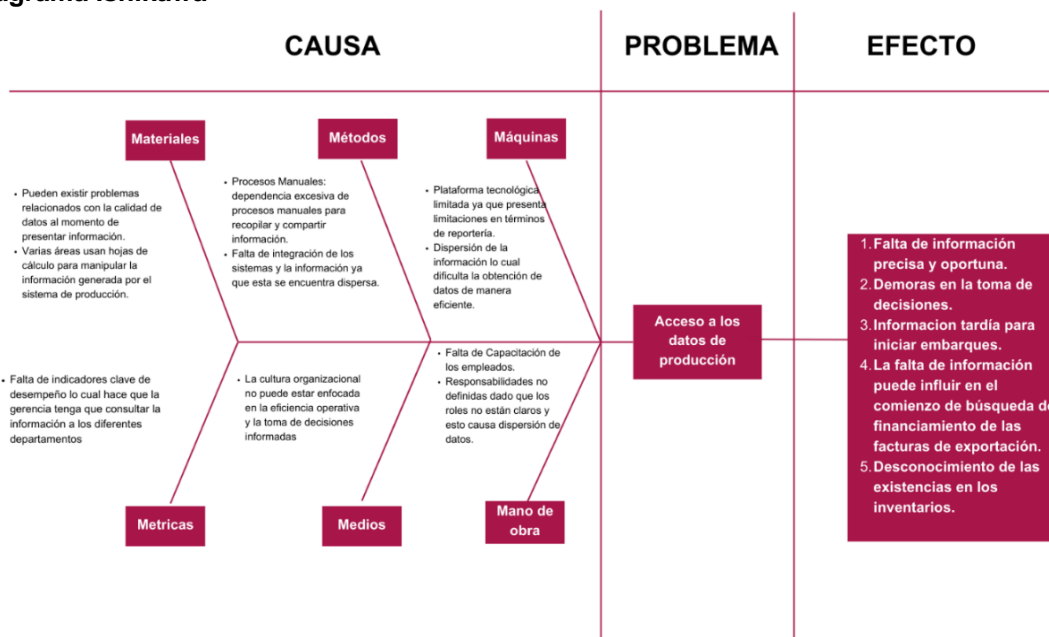
Las consultas y solicitudes de información por parte de los jefes y gerentes, desde datos de inventario hasta cifras de producción, a menudo resultan en demoras y dependen de procesos manuales, lo que impacta negativamente en la eficiencia de toma de decisiones.

La empresa se encuentra en una situación en la que la información sobre la producción de camarones está fragmentada y manejada por diversas personas y departamento; la dispersión de datos genera procesos ineficientes, ya que los laborales deben buscar y recopilar información de múltiples fuentes.

### **Causas y Consecuencias del Problema**

Para identificar y analizar la problemática y su origen, se empleó el método de Ishikawa, también conocido como diagrama de espina de pescado o diagrama de causa y efecto; este método, que se detalla en la Figura 3, permite descomponer y visualizar las diversas causas que contribuyen a un problema específico, la representación gráfica facilita la identificación de las relaciones entre las distintas variables y factores, permitiendo una comprensión más profunda y estructurada de la situación analizada.

**Figura 3**  
**Diagrama Ishikawa**



Luego del análisis realizado con el diagrama se pueden identificar las principales causas del problema:

1. Corresponde a la presentación de información que puede verse afectada por problemas de calidad de datos debido a la manipulación en hojas de cálculo por diversas áreas, la excesiva dependencia de procesos manuales para recopilar y compartir datos, y la falta de integración de los sistemas, lo que resulta en información dispersa.
2. La plataforma tecnológica presenta limitaciones en términos de reportería y, junto con la dispersión de la información, dificulta la obtención de datos de manera eficiente; además, la falta de indicadores clave de desempeño obliga a la gerencia a consultar información dispersa en diferentes departamentos.

La cultura organizacional, al no estar enfocada en la eficiencia operativa y la toma de decisiones informadas, se ve afectada por la falta de capacitación de los empleados y la indefinición de responsabilidades, lo que resulta en roles poco claros y en la dispersión de datos.

Como consecuencia de lo expuesto anteriormente, la gerencia y las jefaturas no disponen de información precisa y oportuna cuando la necesitan. Esto se debe a una serie de factores, incluyendo la falta de una plataforma tecnológica adecuada, la dispersión de la información, y la ausencia de indicadores clave de desempeño; la obtención de datos se complica aún más porque las solicitudes de información se realizan a través de WhatsApp o llamadas telefónicas a los laborales del área de producción, lo que incrementa el riesgo de errores y retrasos en la comunicación.

### **Delimitación del Problema**

A continuación, se presenta la delimitación del problema:

**Tabla 1**  
**Criterios de Limitación de Problema**

<b>#</b>	<b>Criterios</b>	<b>Detalles</b>
1	Campo	Tecnología
2	Área	Producción
3	Aspecto	Chatbot, WhatsApp, comunicación por mensajería, preguntas y respuestas
4	Propuesta	Desarrollar un prototipo en la plataforma de WhatsApp para consultar información de la producción del proceso de empaque de camarón

### **Formulación del Problema**

Tras realizar una observación detallada de los problemas, se ha determinado la necesidad de implementar un chatbot que permitirá una recopilación y distribución más eficiente de la información; este automatizará la respuesta a consultas frecuentes y el acceso a datos importantes, reduciendo así la dependencia de la comunicación informal y manual. De este modo, se minimizarán los errores y retrasos asociados con el uso de WhatsApp y llamadas telefónicas para solicitar información; el chatbot se integrará con el sistema de producción de la empacadora, consolidando la información dispersa y proporcionando a la gerencia y jefaturas datos precisos y en tiempo real, facilitando así la toma de decisiones basadas en información confiable y actualizada.

### **Evaluación del Problema**

De los diez criterios definidos para la evaluación del problema, se escogerán siete, los cuales se indican en la tabla siguiente:

**Tabla 2**  
**Criterios de Evaluación del Problema**

#	Criterios	Contexto
1	Delimitado	El problema se circunscribe al proceso de consulta de la información de la empresa empacadora de camarón de la ciudad de guayaquil la cual no accede a la información de manera oportuna ya que es solicitada a intermediarios que consultan información en los sistemas y la manipulan para entregarla a gerencia.
2	Claro	La definición del problema permite comprender el impacto de no poseer una consulta directa utilizando el sistema de mensajería WhatsApp para consultar la información.
3	Relevante	Para resolver el problema se requiere del desarrollo de un prototipo de chatbot en WhatsApp el cual responderá las consultas sobre la producción de la empacadora.
4	Evidente	Como se indica en la <b>Figura 3</b> las causas que se indican son claras, no requieren mayor detalle para poder definir el problema.
5	Original	La empresa no cuenta con otra herramienta tipo chatbot implementada dentro de sus soluciones.
6	Factible	La empresa dedicada al procesamiento y empaque de camarones cuenta con los recursos necesarios y se tiene acceso a la información y los recursos para el desarrollo del chatbot.
7	Identifica productos esperados	El desarrollo del prototipo de chatbot en WhatsApp brindará la información de la producción en tiempo real en base a la información más común que solicita gerencia a los distintos departamentos de la empacadora.

## OBJETIVOS

### Objetivo General

Crear un prototipo de Chatbot en WhatsApp para agilizar la consulta de información sobre la producción en una empresa de procesamiento de camarones donde la herramienta proporcione respuestas instantáneas.

### Objetivos Específicos

1. Aplicar las herramientas de desarrollo de software definidas y utilizadas por la empresa en la creación del prototipo, asegurando su coherencia con los estándares de desarrollo de la compañía.
2. Desarrollar e implementar una interfaz de usuario que integre un banco de preguntas donde este recurso facilite la comprensión y respuestas precisas a las consultas vinculadas con la producción de la empacadora.
3. Optimizar la precisión y rapidez de respuestas del chatbot mediante la abstracción, transformación y depuración de datos en tiempo real<sup>3</sup>. Se enfocará en crear un proceso automatizado para recolectar, limpiar y organizar datos del procesamiento de camarones, asegurando que el chatbot disponga de información actualizada y precisa para las consultas en que se realicen en WhatsApp.
4. Integrar el chatbot con los sistemas de información internos de la empresa, asegurando la obtención y presentación precisa de datos en tiempo real relacionados con el procesamiento de camarones.
5. Medir el nivel de aceptación del chatbot por parte de los usuarios mediante encuestas y análisis de comentarios con la finalidad de obtener una evaluación sobre la aceptación del mismo.

## **ALCANCES DEL PROBLEMA**

El desarrollo del prototipo será exclusivo para la aplicación de mensajería instantánea WhatsApp con el fin de aprovechar las características y funcionalidades particulares de esta plataforma lo cual nos ayudará a garantizar una experiencia de usuario fluida y familiar.

Durante las pruebas del prototipo, se utilizarán datos reales por tanto los resultados de esas pruebas dependerán de los datos. Esta medida garantizará que los datos de producción y empaque de camarones, provenientes directamente de la operación empresarial, sean un recurso valioso para la alta gerencia; la incorporación de datos reales no solo respaldará la evaluación y mejora del chatbot, sino que

---

<sup>3</sup> ETL por sus siglas en inglés significa "Extract, Transform, Load"



también facilitará una representación precisa de las condiciones operativas. Este enfoque asegurará una evaluación más exacta y pertinente, manteniendo al mismo tiempo la integridad de la información empresarial.

Limitar el chatbot al banco de preguntas obtenido del levantamiento de información implicará enfocar la funcionalidad del sistema exclusivamente en las consultas y respuestas identificadas durante el proceso de obtención de información. De este modo nos ayudará a tener precisión y relevancia en cada una de las respuestas que nos proporciona el prototipo, ya que se adapta de forma específica a las necesidades de las consultas que realizan los usuarios con frecuencia.

Esta limitante nos facilita la gestión y actualización de la información ya que el prototipo se enfocará únicamente en todas las interrogantes previamente definidas; no obstante, es importante considerar la posibilidad de expansiones y actualizaciones de su funcionalidad, evaluando la inclusión de nueva información en base a las interacciones reales con los usuarios con el objetivo de mantener relevante la utilidad del prototipo ante una implementación completa.

El enfoque del desarrollo estará específicamente orientado a la creación de un prototipo funcional del chatbot para WhatsApp. En lugar de una implementación completa, se priorizará la identificación y solución de desafíos iniciales, la validación de conceptos y la obtención de retroalimentación de los usuarios. Este enfoque permitirá ajustes iterativos basados en la experiencia del usuario antes de considerar una implementación a mayor escala.

La implementación del prototipo utilizando la aplicación WhatsApp impone ciertos desafíos relacionados con la conectividad y el servicio de esta plataforma las cuales se mencionan a continuación:

1. Dependencia de una conexión a Internet estable:
  - 1.1. Necesidad de una conexión estable entre el cliente y los servidores de la empresa.
2. En regiones con cobertura limitada o baja conectividad, los usuarios pueden experimentar:

- 2.1. Demoras en las respuestas del chatbot.
  - 2.2. Dificultades para acceder al servicio.
  - 2.3. Limitaciones debido a la falta de acceso al servicio de Internet de la empresa.
3. Variabilidad en la calidad del servicio de WhatsApp:
- 3.1. Influencia de factores externos como:
    - 3.1.1. Picos de demanda.
    - 3.1.2. Actualizaciones de la aplicación.
    - 3.1.3. Problemas técnicos en la plataforma.
  - 3.2. Impacto negativo en la disponibilidad y velocidad de respuesta del chatbot.
  - 3.3. Generación de una experiencia del usuario menos consistente.
4. Restricciones en las políticas de uso de WhatsApp:
- 4.1. Limitaciones en ciertas funcionalidades del chatbot debido a las políticas de WhatsApp.

La fase de pruebas se realizará de forma escalada, comenzando con un grupo de usuarios internos, antes de mostrar la funcionalidad a los usuarios externos; esto nos permitirá realizar una evaluación preliminar por los usuarios internos, dando la oportunidad de hacer ajustes en función de los comentarios antes de exponer el prototipo a la totalidad de los usuarios. La retroalimentación será importante para mejorar la experiencia del prototipo antes de una implementación amplia.

## **JUSTIFICACION E IMPORTANCIA**

La importancia de este trabajo radica en que la automatización de respuestas a consultas frecuentes y el acceso inmediato a datos críticos a través de un chatbot reduce significativamente el tiempo necesario para obtener información. WhatsApp, siendo una plataforma ampliamente utilizada y familiar para la mayoría de los

empleados, facilita su adopción y uso efectivo, mejorando la eficiencia en la recopilación y distribución de información. El chatbot puede integrarse con el sistema de producción de la empaedora, consolidando la información dispersa y proporcionando datos precisos y en tiempo real a la gerencia y las jefaturas, lo cual es crucial para la toma de decisiones informadas y oportunas.

La implementación de este prototipo es relevante ya que marca un hito en el desarrollo de herramientas de este tipo para esta industria; es de suma importancia demostrar que los sistemas internos implementados brindan acceso a datos de forma fácil y confiable; con la implementación de este prototipo dará pauta para comprender mejor las necesidades reales de información por parte de la gerencia. De forma general, el desarrollo de este prototipo proporcionará una herramienta adicional importante para que los gerentes tomen decisiones con información actualizada.

## **PREGUNTA DE INVESTIGACIÓN**

¿El prototipo ChatBot de WhatsApp tendrá la capacidad de responder las preguntas de la producción de la empaedora de camarones de manera oportuna?

## **VARIABLES DE LA INVESTIGACIÓN**

**Uso del prototipo:** Se medirá mediante la cantidad de preguntas generadas diariamente; para realizar este seguimiento, se implementará un registro detallado en la base de datos, que incluirá la fecha, la hora y el usuario que genera cada solicitud de información, el cual permitirá analizar patrones de uso, identificar picos en la demanda y evaluar la eficacia del prototipo en la atención de consultas. Estos datos ayudarán a mejorar continuamente el servicio, ajustando el producto según las necesidades y comportamientos de los usuarios.

**Eficacia de respuesta:** El prototipo debe ser capaz de responder cada pregunta relacionada con la producción en un tiempo de no más de 15 segundos bajo condiciones normales; con esta medición se asegura que los gerentes y jefes obtengan la información sin demoras significativas. En caso de detectar tiempos superiores al ya mencionado, se debe analizar las causas y aplicar medidas correctivas para optimizar el rendimiento del prototipo.

**Coherencia:** El prototipo debe proporcionar respuestas que estén alineadas con la información de producción; las preguntas formuladas al prototipo deben mantenerse dentro del ámbito de los temas que el chatbot está diseñado para resolver, evitando cualquier desviación. Para garantizar esta coherencia, se implementarán filtros y reglas específicas que guíen las interacciones del prototipo, asegurando que solo se manejen preguntas pertinentes.

**Satisfacción del prototipo:** Se llevará a cabo una encuesta de satisfacción para evaluar el prototipo y determinar si cumple con las necesidades de información de los usuarios; esta recogerá opiniones detalladas sobre la precisión, relevancia y utilidad de las respuestas proporcionadas por el prototipo.

La implementación de este prototipo tiene como objetivo posicionarse como una solución integral de consulta de información. Al identificar las aristas del problema, queda claro qué soluciones debemos desarrollar. Será crucial realizar un exhaustivo levantamiento de información y formular preguntas específicas relacionadas con la producción, para cubrir la mayoría de las dudas que surgen en las jefaturas y gerencias respecto al estado de la producción.

## Capítulo II: Marco Teórico

Se investigaron conceptos relacionados a: chatbots, sus aspectos fundamentales, características y ventajas, junto con los principios de facilidad de uso y experiencia del usuario, utilidad, así también se validó bibliografía de desarrollo de software, implementación y seguridad en sistemas de información, Este estudio sentó las bases para el diseño y creación del prototipo de chatbot destinado a una empresa dedicada al procesamiento de camarones.

### 2.1 Chatbot.

IBM (2024) define a un chatbot como un software creado para simular la conversación humana con un usuario; estos programas pueden ayudar a los usuarios a encontrar información al responder instantáneamente a preguntas y solicitudes, ya sea mediante entrada de texto, audio, o ambas, sin necesidad de intervención humana o de realizar una búsqueda manual.

Dentro de los lenguajes de programación con los cuales se puede desarrollar los chatbots tenemos varios, pero según (Sánchez-Ticona et al., 2023) la principal herramienta para el desarrollo de los chatbots es python.

La popularidad de los chatbots en las empresas se debe a la reducción de costos de servicio al cliente y su capacidad para atender a múltiples usuarios al mismo tiempo; su interacción ha evolucionado, acercándolos a los usuarios como compañeros amigables. (Adamopoulou & Moussiades, 2020) revela que las solicitudes a chatbots en redes sociales son tanto emocionales como informativas, con más del 40% enfocadas en lo emocional, sin buscar información específica. El aprendizaje automático permite a los chatbots detectar sentimientos y relacionarse emocionalmente con los clientes, similar a los operadores humanos.

Según Solutions, 2020 los chatbots son valorados por las empresas por las siguientes ventajas que se enlistan a continuación:

**Respuestas inmediatas:** Los chatbots empresariales permiten a las empresas satisfacer esta demanda proporcionando respuestas inmediatas.

**Impulsar los ingresos y reducir costos:** Los chatbots inteligentes guían a los clientes en sus compras, aumentando las ventas, pueden recordar las preferencias de los clientes y promover ventas adicionales de manera sutil, permite atender a más clientes sin aumentar sus costos de operación, disminuyendo las consultas entrantes y logrando altas tasas de resolución en la primera llamada.

**Maximizar las habilidades de los laborales:** Al automatizar consultas que normalmente requerirían intervención humana directa, los chatbots liberan tiempo para que el personal existente se enfoque en interacciones de mayor valor con los clientes, en tareas de control y mejora de la información, lo que permite optimizar el recurso humano y optimizar la eficiencia operativa.

**Alcanzar nuevos canales:** Los chatbots facilitan conversaciones de consultas automatizadas que involucran a los clientes o laborales y ofrecen asesoramiento y soporte personalizado, sin necesidad de nuevos equipos de oficina para gestionar cada canal o red.

**Disponibilidad 24/7:** Los funcionarios desean soporte inmediato, disponible las 24 horas del día, los 365 días del año; requieren enviar un mensaje con una consulta mientras están en una reunión o utilizar la voz para actualizar informes mientras se desplazan al trabajo, utilizando todos los dispositivos y servicios que ya emplean diariamente.

**Comprender mejor a los empleados y aumentar su compromiso:** Los datos conversacionales en primera persona son una fuente valiosa para comprender tendencias y sentimientos de usuarios, informando el desarrollo de productos y servicios. Desde niveles granulares para marketing individualizado hasta niveles macro para identificar tendencias generales, estos datos revelan necesidades de los usuarios que, al ser atendidas, aumentan el compromiso y se traducen en mejores resultados económicos.

## **2.2 Plataforma Whatsapp.**

WhatsApp es una aplicación de mensajería gratuita y multiplataforma perteneciente a Meta Platforms, Inc, cuyas funcionalidades principales permite realizar: llamadas de voz, video llamadas, transferencia de archivos (texto, imagen,

sonido, etc.), GIS<sup>4</sup>, mensajes de texto, y más, con una conexión Wi-Fi o sistema de acceso a internet vía GPRS<sup>5</sup>, 3G<sup>6</sup>, LTE<sup>7</sup>, 4G<sup>8</sup>, gestionar comunidades (grupos).

Goodwin (2023) menciona que al 2024 cuenta con más de dos mil millones de usuarios activos, es especialmente popular entre amigos y familiares debido a su accesibilidad y características directas.

Para una mayor apreciación de la plataforma Whatsapp en la Figura 4 se muestra la interfaz gráfica de usuario.

**Figura 4**  
**Uso de aplicación**



**Nota:** Descripción de la pantalla de chats de la aplicación WhatsApp.

Según WhatsApp Business, 2024 , WhatsApp ofrece una aplicación llamada WhatsApp Business que, a diferencia de la versión estándar, incluye varias

<sup>4</sup> Sistema de información geográfica.

<sup>5</sup> General Packet Radio Service.

<sup>6</sup> Tecnología de tercera generación.

<sup>7</sup> Long Term Evolution.

<sup>8</sup> Tecnología de cuarta generación.

funcionalidades adicionales diseñadas para facilitar la gestión de la comunicación empresarial. Se mencionan características clave de esta aplicación para negocios:

1. **Perfil de Empresa:** Permite crear un perfil de tipo empresa donde se puede incluir información de esta como: dirección, descripción del negocio, sitio web, correo electrónico y horario de atención.
2. **Herramientas de Mensajería:** Permite crear respuestas rápidas para preguntas que se realizan de forma frecuente, así como también parametrizar mensajes de bienvenida y de ausencia dependiendo del caso.
3. **Catálogo de Productos:** Permite compartir el catálogo de productos o servicios directamente en la aplicación dando la facilidad de promocionar o vender estos.
4. **Integración con API:** Ofrece una API que permite integrar WhatsApp Business con sistemas empresariales y otros softwares lo que facilita la escalabilidad y automatización de las comunicaciones.

## **2.3 Software.**

### **2.3.1 Lenguajes de programación.**

Según Hemmendinger (2024) un lenguaje de programación de computadoras es cualquier de los lenguajes utilizado para escribir instrucciones detalladas para una computadora digital. Estas instrucciones pueden ejecutarse directamente en el lenguaje de máquina específico del fabricante, mediante un lenguaje ensamblador, o tras ser traducidas desde un lenguaje de alto nivel. Aunque hay muchos lenguajes de programación como, por ejemplo: C++, Java, PHP, Visual Basic, C#, Python Visual Basic.Net solo unos pocos son de uso común (C#, Java, Python, Visual Basic.Net).

### **2.3.2 Lenguajes de programación: Python.**

Para la construcción del prototipo se utilizó Python en su versión 3.11.2, el cual es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus avanzadas estructuras de datos lo hacen ideal para el



desarrollo rápido de aplicaciones y como lenguaje de scripting para integrar componentes, su sintaxis simple y legible reduce los costos de mantenimiento.

Guido Van Rossum & Fred L. Drake (2011) menciona que python fue creado a principios de la década de 1990 por Guido van Rossum en el Stichting Mathematisch Centrum en los Países Bajos como sucesor del lenguaje ABC. Aunque Guido sigue siendo el autor principal, Python incluye muchas contribuciones de otros desarrolladores. En 1995, Guido continuó su trabajo en la Corporation for National Research Initiatives en Virginia, lanzando varias versiones del software. En mayo de 2000, él y el equipo central de desarrollo se trasladaron a BeOpen.com para formar el equipo BeOpen PythonLabs, y en octubre de ese año, se movieron a Digital Creations (ahora Zope Corporation). En 2001, se formó la Python Software Foundation (PSF), una organización sin fines de lucro creada para gestionar la propiedad intelectual de Python, con Zope Corporation como miembro patrocinador. Todas las versiones de Python son de código abierto y, en su mayoría, han sido compatibles con la GPL<sup>9</sup>.

Guido Van Rossum & Fred L. Drake (2011) define a Python como un lenguaje de código abierto, lo que hace que su código fuente esté disponible al público y pueda ser modificado y distribuido; lo cual fomenta el crecimiento de este consolidando una comunidad grande y un ecosistema de desarrolladores que contribuyen con el lenguaje, así como también con la creación de bibliotecas las cuales, mediante foros, grupos de discusión y recursos educativos han facilitado enormemente la resolución de problemas y consultas. Python posee una fundación llamada Python Software Foundation la cual promueve el desarrollo continuo del lenguaje lo que garantiza la constante evolución y adaptabilidad a las nuevas tecnologías.

Guido Van Rossum & Fred L. Drake (2011) también menciona que Python es un lenguaje de programación versátil y poderoso que se utiliza en una amplia variedad de aplicaciones, incluyendo el análisis de datos, la inteligencia artificial (IA), el desarrollo web, la automatización de tareas, y la creación de scripts. En el ámbito del análisis de datos, Python es muy popular debido a sus bibliotecas como Pandas y NumPy, que facilitan la manipulación y el análisis de grandes conjuntos de datos

---

<sup>9</sup> GPL General Public Licence por sus siglas en inglés

(McKinney, 2017). En inteligencia artificial y aprendizaje automático, bibliotecas como TensorFlow, Keras y PyTorch permiten a los desarrolladores construir y entrenar modelos complejos de manera eficiente (Martelli & Ascher, 2002). En el desarrollo web cuenta con frameworks como Django y Flask son ampliamente utilizados para el desarrollo de aplicaciones robustas y escalables.

Según la Python Software Foundation (2024b), python soporta módulos y paquetes, promoviendo la modularidad y reutilización del código; el intérprete y su amplia biblioteca estándar están disponibles gratuitamente en formato fuente o binario para todas las principales plataformas y pueden distribuirse libremente.

### **2.3.3 PIP.**

Según la Python Software Foundation (2023), PIP es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python; es una herramienta esencial para los desarrolladores de Python, ya que facilita la instalación y actualización de paquetes y sus dependencias. Su nombre "PIP" es un acrónimo recursivo que significa "Pip Installs Packages".

Reitz & Schlusser (2016) menciona que las funciones claves del gestor de paquetes son las siguientes:

- **Instalación de Paquetes:** Permite instalar paquetes desde el Python Package Index (PyPI).
- **Gestión de Dependencias:** Resuelve e instala las dependencias de los paquetes automáticamente.
- **Actualización de Paquetes:** Facilita la actualización de paquetes a versiones más recientes.
- **Desinstalación de Paquetes:** Permite desinstalar paquetes y sus dependencias no utilizadas.
- **Entornos Virtuales:** Se integra bien con venv y virtualenv para la gestión de entornos virtuales.

### **2.3.4 Entorno virtual virtualenv.**

La Python Software Foundation (2024a) menciona que el módulo venv es una herramienta que permite crear "entornos virtuales" independientes en Python, con sus propios paquetes instalados, estos entornos se construyen sobre una instalación existente de Python, conocida como "base", y pueden estar aislados opcionalmente de otros entornos, lo que limita los paquetes disponibles. Al trabajar dentro de un entorno virtual, las herramientas de instalación como PIP instalan paquetes automáticamente en el entorno sin necesidad de indicarlo explícitamente. Fue creado por Ian Bicking (Ian Bicking, 2018) en 2007 como respuesta a la necesidad de gestionar dependencias de manera más eficiente en proyectos Python. Su objetivo era proporcionar un medio para trabajar en proyectos con diferentes requerimientos sin afectar al sistema global de Python. El proyecto ha sido mantenido y desarrollado activamente por la comunidad desde entonces.

### **2.3.5 Framework Flask.**

Miguel Grinberg (2014) define a Flask como un ligero y minimalista framework web para Python, reconocido por su simplicidad y fácil aprendizaje, lo que lo convierte en una opción ideal para desarrollar rápidamente aplicaciones web; proporciona herramientas simples y flexibles que permiten a los desarrolladores enfocarse en la lógica de la aplicación sin preocuparse por la complejidad del framework.

El prototipo utilizará Flask para poder manejar las solicitudes HTTP, lo que nos permite la conectividad con las APIs de WhatsApp Business por donde se enviarán y recibirán los mensajes del usuario por medio de la plataforma, así mismo, nos permitirá realizar la validación del token de autenticación con la plataforma de Meta.

### **2.3.6 WebHook**

Según Nemesis (2023) los webhooks son un mecanismo para que un sistema informe a otro sobre eventos en tiempo real, facilitando la comunicación entre aplicaciones a través de HTTP. Se utilizan para automatizar tareas, como enviar correos electrónicos al registrarse un usuario, sincronizar datos entre sistemas, enviar notificaciones como alertas por SMS, y permitir integraciones personalizadas entre servicios o APIs para adaptarse a las necesidades de los desarrolladores.

Parte de las integraciones que requiere el prototipo como lo es la API de WhatsApp se realiza por medio de la utilización de Flask. Esta herramienta permite la integración ya que posee la capacidad de manejar las peticiones http de forma eficiente. Se considera Flask ya que es un requerimiento explícito de la empresa patrocinante por ser parte de los estándares de desarrollo de esta.

Desarrollar los WebHooks utilizando Flask nos permite una rápida integración y un desarrollo mucho más ágil obedeciendo los estándares de la empresa.

### **2.3.7 Biblioteca Request.**

Según Al Sweigart (2020) request es una biblioteca desarrollada en Python que permite realizar solicitudes http de forma clara y manejable lo cual la hace altamente reconocida; ofrece funciones de fácil comprensión ayudando en el manejo de solicitudes HTTP GET, POST, y otros métodos. Request posibilita la interacción con servicios web lo cual ayuda a los desarrolladores a escribir código de forma reducida para poder trabajar con APIs y otros recursos webs.

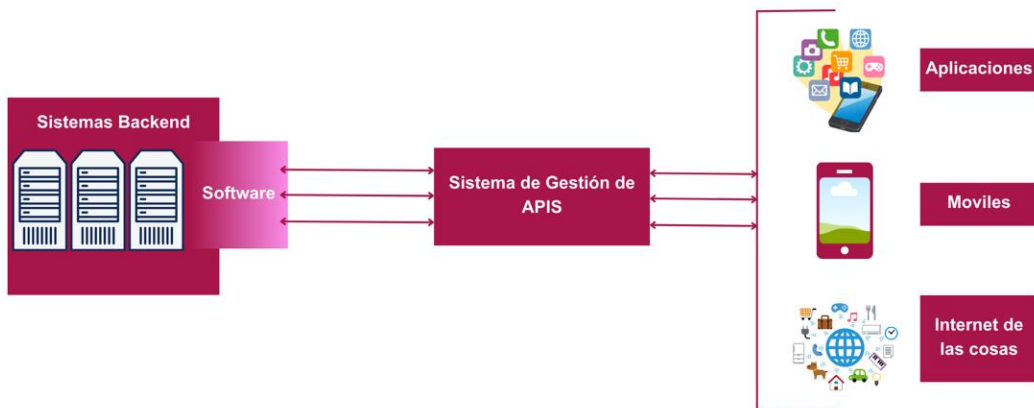
### **2.3.8 API.**

Según Amazon Web Services, Inc., 2023 una API es una herramienta que permite a los sistemas, aplicaciones y plataformas comunicarse entre sí obedeciendo ciertas reglas, protocolos y estándares los cuales garantizan la interacción. API es el acrónimo de “Interfaz de Programación de Aplicaciones” donde el termino aplicación hace referencia a cualquier software o sistema con alguna función en especifica; la interfaz se puede definir como un contrato entre dos aplicaciones donde se define la forma de comunicarse y estableciendo solicitudes y respuestas.

Las APIs poseen documentación donde se detalla como los desarrolladores deben estructurar estas solicitudes y las respuestas para de esta forma lograr la integración entre sistemas.

A continuación, se especifica en la Figura 5 la estructura de una API donde se grafica todos los componentes con los cuales puede interactuar.

**Figura 5**  
**Estructura funcional de una API**



**Nota:** Las API proporcionan una manera más simple de conectar su infraestructura, tanto para desarrollar aplicaciones nativas de la nube como para compartir datos con clientes y otros usuarios externos. Tomado de (Red Hat, Inc., 2023)

La estructura de las API se describe comúnmente en relación con el concepto de cliente y servidor. El cliente es la aplicación que realiza la solicitud, mientras que el servidor es la aplicación que envía la respuesta. Por ejemplo, en el caso de una aplicación móvil que consulta datos meteorológicos a través de una API, la base de datos meteorológicos del instituto actúa como servidor y la aplicación móvil como cliente.

### 2.3.8.1 Tipos de APIS.

Amazon Web Services, Inc., 2023 clasifica las APIs según su funcionamiento los cuales se describen a continuación:

**API de SOAP**<sup>10</sup>: Esta forma de interacción utiliza el protocolo simple de acceso a objetos y el intercambio de información se realiza mediante XML. Fueron populares en el pasado, pero su uso ha venido disminuyendo por su flexibilidad en comparación a otras opciones.

**API de RPC**: Conocidas como llamadas a procedimientos remotos, estas interfaces permiten que el cliente llame a funciones en el servidor, recibiendo luego el resultado correspondiente.

---

<sup>10</sup> Simple Object Access Protocol por sus siglas en inglés.

**API de WebSocket:** esta forma de comunicación utiliza objetos tipo JSON<sup>11</sup> para la transmisión de datos. JSON es un formato ligero y fácil de interpretar lo que facilita la interacción entre sistemas y aplicaciones. Esta tecnología destaca por su ligereza superando en algunos aspectos a las API REST.

**API de REST:** Consideradas las más populares y flexibles en la actualidad, estas interfaces permiten que el cliente envíe solicitudes al servidor como datos, activando funciones internas que luego devuelven los resultados al cliente.

Para el prototipo se utilizará API de Rest ya que de esta forma se asegura la interoperabilidad entre el sistema de producción de la empacadora y el chatbot desarrollado en Python, así como también API Rest siguen estándares bien definidos en métodos HTTP<sup>12</sup> (GET, POST, PUT, DELETE) la cual nos permitirá una comunicación más sencilla y efectiva.

Es importante indicar que la implementación de API Rest no es solo una recomendación técnica, sino que también responde a un requerimiento explícito por parte de la empresa donde se implementará el prototipo; este requerimiento se fundamenta principalmente en la integración a futuro de sistemas tanto internos como externos a fin de garantizar una comunicación fluida entre diferentes servicios y plataformas.

#### **2.3.8.2 API Web.**

Una API web o API de servicio web, es un servicio del lado del servidor lo que facilita la interacción entre aplicaciones. Se debe considerar que mientras todos los servicios web son APIs, no todas las APIs son servicios web; como por ejemplo la API de REST el cual es un tipo especial de API web que posee una arquitectura estándar siendo entre las APIs las más comunes.

#### **2.3.9 Navegador Web.**

Berners-Lee et al., 2000 define a un navegador web es un software diseñado para permitir a los usuarios acceder y navegar por información en la World Wide Web.

---

<sup>11</sup> JavaScript Object Notation

<sup>12</sup> HyperText Transfer Protocol por sus siglas en inglés.

Funciona como una interfaz entre el usuario y la web, interpretando el código HTML, CSS y otros lenguajes de marcado para mostrar páginas web y permitir la interacción con ellas. Los navegadores web también pueden incluir características como la gestión de marcadores, historial de navegación, gestión de descargas y extensiones que amplían su funcionalidad.

Algunos ejemplos de navegadores web populares incluyen Google Chrome, Mozilla Firefox, Microsoft Edge, Safari y Opera.

### **2.3.10 C#.**

C# es un lenguaje de programación orientado a objetos y multi propósito desarrollado por Microsoft. Este lenguaje es altamente utilizado en el desarrollo de aplicaciones de escritorio, aplicaciones web y aplicaciones móviles teniendo enfoque principalmente en entornos basados en el framework .NET. (Microsoft Corporation, 2024a).

Anders Hejlsberg, quien es conocido por participar en el desarrollo de los lenguajes TypeScript y Turbo Delphi fue quien creó C# en el año 2000 el cual fue bautizado originalmente como SMC (Simple Managed C) pero por problemas de registro de marca fue cambiado su nombre al que actualmente conocemos. El nombre C# proviene de la nota música Do Sostenido el cual está representado por el símbolo #, indicando de esta forma que C# es un avance con respecto al lenguaje de programación C++. (De León, 2023).

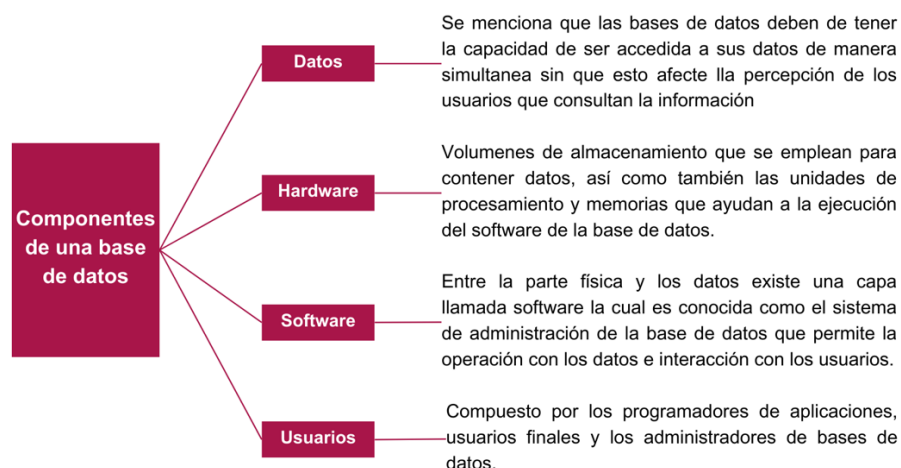
C# ha sido seleccionado para realizar la programación backend de las APIs, conforme a los requerimientos de la empresa empacadora de camarones. Este lenguaje de programación que se utiliza ampliamente dentro de la empresa, lo que facilita su integración con los sistemas existentes por su compatibilidad con Microsoft SQL Server, que se empleará para gestionar las consultas relacionadas con la producción de camarones y para registrar los logs de uso y acceso, asegura un entorno de desarrollo eficiente y cohesivo

### **2.3.11 Bases de datos.**

Coronel et al., 2011 define a una base de datos como un sistema organizado de almacenamiento de información que permite gestionar y acceder a datos de manera eficiente, consistiendo en conjuntos de datos relacionados entre sí, estructurados de acuerdo con un modelo de datos específico. Estos datos pueden ser manipulados mediante consultas y actualizaciones, lo que facilita la recuperación de información relevante por lo cual se pueden utilizar en una amplia gama de aplicaciones, desde sistemas de gestión empresarial hasta sistemas de información científica, siendo fundamentales para la toma de decisiones y la automatización de procesos.

Date (2001) define a un sistema de bases de datos como un sistema informático diseñado para gestionar registros de manera automatizada; se puede visualizar como un armario electrónico para almacenar información, siendo un depósito o contenedor de datos, permitiendo a los usuarios realizar diversas operaciones, como agregar nuevos registros, insertar datos en registros existentes, recuperar información, modificar datos, eliminar registros individuales o eliminar conjuntos de registros completos. El autor enlista 4 componentes que posee una base de datos, a continuación, el detalle gráfico.

**Figura 6**  
**Componentes de una base de datos**



**Nota:** Se introduce el concepto de Sistema de administración de base de datos (DBMS)<sup>13</sup> al incluir los 4 componentes.

<sup>13</sup> DBMS Data Base Management Systems por sus siglas en inglés



### **2.3.11.1 Tipos de base de datos.**

Teniendo presente el interés del desarrollo del prototipo y conociendo la infraestructura con la que cuenta la empresa empacadora de camarón vamos a conceptualizar las bases de datos OLAP<sup>14</sup> y OLPT<sup>15</sup>.

Según Kimball & Ross (2002) OLAP y el modelado dimensional son importantes para el análisis de datos en entornos empresariales. OLAP proporciona un marco para el soporte a la toma de decisiones, mientras que el modelado dimensional ofrece una metodología eficaz para estructurar datos, mejorar el rendimiento de consultas y facilitar el uso de los datos por parte de los usuarios. Estos conceptos son ayudados al desarrollo y funcionamiento de sistemas de análisis de datos avanzados.

Connolly & Begg (2015) define a OLTP como un gestor de bases de datos creado para manejar grandes cantidades de transacciones cortas y rápidas en tiempo real. Este tipo de bases de datos están optimizadas para llevar a cabo operaciones de inserción, actualización, eliminación y consulta de manera eficiente sin dejar atrás la integridad de los datos y soportando altos niveles de concurrencia; su uso es relevante cuando se requiere niveles altos de transaccionalidad, como por ejemplo sistemas bancarios, de comercio electrónico y sistemas de gestión empresarial.

### **2.3.11.2 Sistema de gestión de base de datos.**

Kimball & Ross (2002) define el sistema de gestión de bases de datos como una aplicación que permite definir, crear y mantener bases de datos, garantizando un acceso controlado la cual consiste en la base de datos en sí misma, el sistema propiamente dicho y los programas de aplicación asociados por lo que se asemeja al modelo de programación orientada a objetos, donde se distingue entre la implementación interna y la especificación externa de un objeto, visible para los usuarios; esta abstracción de datos facilita cambios en la implementación interna sin afectar a los usuarios. De manera análoga, los sistemas de bases de datos separan la estructura física y lógica de los datos, almacenando esta información en la base de

---

<sup>14</sup> OLAP Online Analytical Processing por sus siglas en inglés.

<sup>15</sup> OLPT Online Transaction Processing por sus siglas en inglés.

datos mediante el sistema, que sirve de intermediario entre la base de datos y los programas de aplicación.

Según Marín (2024) los principales sistemas gestores de bases de datos relacionales son:

**MySQL:** es un sistema gestor de bases de datos relacional ampliamente utilizado, especialmente en páginas web y aplicaciones de software libre.

**Oracle:** ha sido históricamente el sistema gestor de bases de datos preferido en el ámbito empresarial, conocido por su robustez y funcionalidad completa.

**PostgreSQL:** es un sistema gestor de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia BSD<sup>16</sup>.

**Microsoft SQL Server:** es un sistema de gestión de bases de datos relacional basado en Transact-SQL, diseñado para manejar grandes volúmenes de datos de manera simultánea para múltiples usuarios. Propiedad de Microsoft ofrece soporte exclusivo, alta escalabilidad, estabilidad y seguridad, capacidad para cancelar consultas, y un potente entorno gráfico de administración con comandos DDL<sup>17</sup> y DML<sup>18</sup>.

Microsoft SQL Server será el motor de base de datos que utilizaremos para gestionar las consultas relacionadas con la producción de camarones. Nos basaremos en la base de datos existente para registrar los logs de uso y acceso. Esta elección se debe a los requerimientos de la empresa, que ya tiene su información almacenada en este sistema de gestión de bases de datos.

### **2.3.12 Entorno de desarrollo**

AWS, Inc (2023) define a un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) como una aplicación que proporciona un conjunto de herramientas y facilidades para los desarrolladores de software con el objetivo de maximizar su productividad. Un IDE típico incluye un editor de código fuente, un depurador, un

---

<sup>16</sup> BSD Berkeley Software Distribution por sus siglas en inglés.

<sup>17</sup> DDL Data Definition Language por sus siglas en inglés.

<sup>18</sup> DML Data Manipulation Language por sus siglas en inglés.

compilador o intérprete, y a menudo herramientas de construcción y control de versiones. Los IDEs están diseñados para simplificar el proceso de desarrollo, permitiendo a los programadores escribir, probar y depurar su código de manera eficiente dentro de un único entorno.

Para el entorno de desarrollo del prototipo, utilizaremos Visual Studio Code, que nos permitirá programar en Python, así como también, para el desarrollo de la API emplearemos Visual Studio 2022 en su versión Community. Ambos IDEs <sup>19</sup> están autorizados y son utilizados por la empresa de procesamiento de camarones, por lo que debemos adaptarnos a las herramientas de desarrollo que tienen implementadas. A continuación, se define cada una de las herramientas mencionadas.

### **2.3.12.1 Visual Studio Code**

Gamarra (2024) menciona que Visual Studio Code (VS code) es un editor de código fuente desarrollado y anunciado por Microsoft en el año 2015. Su característica principal es ser gratuito y de código abierto lo que permite una experiencia de desarrollo flexible al ser compatible con muchos lenguajes de programación y tecnologías en el desarrollo de software.

Posee depuración integrada, finalización de escritura de código inteligente (IntelliSense), control de versiones integrando Git y un terminal de líneas de comandos. Este editor tiene dentro de sus opciones un MarketPlace con el cual permite extender su usabilidad mediante librerías y extensiones lo cual brinda a los desarrolladores herramientas útiles y con esto mejorar su experiencia en el desarrollo de código.

Se utilizará Visual Studio Code para desarrollar en Python el prototipo, implementando en este entorno el flujo de preguntas y respuestas que manejará el chatbot ya que es un entorno de desarrollo integrado (IDE) altamente eficiente, conocido por su versatilidad, extensibilidad y compatibilidad con diversas herramientas de desarrollo, incluyendo aquellas específicas para Python; cuenta con extensiones que facilitan la depuración y la integración con APIs externas, esenciales para el desarrollo del prototipo; su uso responde a un requerimiento explícito de la

---

<sup>19</sup> IDE Integrated Development Environment por sus siglas en inglés.

empresa, garantizando que las herramientas utilizadas estén alineadas con sus estándares y preferencias la cual asegura una mayor coherencia en el proceso de desarrollo y una futura implementación exitosa del prototipo en el entorno productivo completo.

### **2.3.12.2 Microsoft Visual Studio.**

Según anandmeg (2023), Microsoft Visual Studio es un entorno de desarrollo integrado que permite escribir, editar, depurar código, así como también implementar aplicaciones. Complementario a esto, Visual Studio ofrece herramientas de completado inteligente de código, control de versiones, extensiones y ambientes colaborativos que mejoran la fase de desarrollo de software. Este entorno de desarrollo integrado posee un instalador basado en cargas de trabajo lo que permite realizar una instalación personalizada. Posee también soporte a múltiples lenguajes de programación como por ejemplo C++, C#, JavaScript, TypeScript y Python.

Este entorno de desarrollo ha evolucionado constantemente desde su lanzamiento en los años 90, lo que lo convierte en una herramienta muy conocida por los desarrolladores. La primera versión, Visual Studio 97 empezó integrando lenguajes como Visual Basic y Visual C++.

En 1998, Visual Studio en su Versión 6.0 integró soporte para el desarrollo web y el diseño de formularios para Microsoft Windows. La versión de Visual Studio 2005 mejoró el rendimiento y la compatibilidad con ASP.NET 2.0 y SQL Server 2005.

Para la versión del 2010 de Visual Studio se realizó una renovación total de su interface, mejora la integración con las herramientas de colaboración. En las versiones del 2012, 2013 y 2015 fue integrando cambios en los logos y mejoras para el desarrollo móvil y aplicaciones universales, así como también integrando funcionalidad y conectividad con Azure.

En la versión 2019 se introduce una nueva interfaz de usuario, mejora en el rendimiento del IDE y nuevas características como la depuración de contenedores y la integración con el nuevo ambiente colaborativo de Microsoft llamado Azure DevOps. (Historia de Visual Studio, 2024).

### **2.3.12.3 Framework .NET**

Powers & Snell, 2015 menciona que el Framework .NET abarca no solo el entorno de ejecución administrado, sino también una amplia gama de clases base, bibliotecas y funciones esenciales que potencian la eficiencia en la programación dentro del entorno .NET. Este conjunto de clases y funciones proporciona la mayoría de las características estándar que un desarrollador necesita para sus proyectos.

Cada uno de los lenguajes compatibles con .Net Framework puede hacer uso de las funcionalidades que posee este marco de trabajo gracias al Common Type System. Entre las funciones que ofrece el framework menciona que permite una serie de operaciones con archivos y comunicaciones, manejo de listas de datos y colecciones y un amplio soporte para aplicaciones basadas en Windows.

Para asegurar la compatibilidad de las aplicaciones con las versiones del .Net Framework, cada iteración de este se maneja como una entidad independiente por lo cual podemos encontrar versiones desde la 1.0 pasando por las versiones 1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.7 y 4.8.

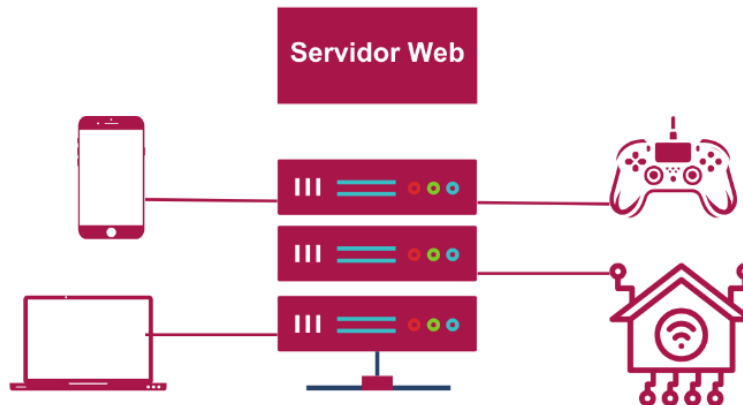
### **2.3.13 Servidor Web.**

Para la implementación de las APIs que brindaran la información de la producción de camarones se utilizará un servidor web por lo cual mencionaremos a continuación sus diferentes definiciones.

Corao & Vanega (2023) define a un servidor web como un software o hardware que almacena contenido web y accede a páginas web a través del protocolo HTTP. Este servidor procesa solicitudes de los clientes y devuelve respuestas entregando el contenido web solicitado a través de Internet.

A continuación, la Figura 7 presenta un esquema básico que ilustra cómo varios clientes pueden acceder a un servidor web el cual muestra la interacción entre múltiples dispositivos cliente, como computadoras, tabletas y teléfonos inteligentes, que se conectan a un servidor centralizado a través de Internet.

**Figura 7**  
**Esquema de un servidor web**



**Nota:** Un servidor web soporta a distintos tipos de clientes los cuales pueden ser, pc, dispositivos móviles, y consolas de video juegos.

Existe una variedad de software para servidores web que pueden ser utilizados, los cuales pueden ejecutarse sobre Linux y Windows. A continuación, se describen los más utilizados:

- **Apache HTTP Server:** Es uno de los servidores web más antiguos y ampliamente utilizados. Es de código abierto y es conocido por su estabilidad y flexibilidad. Apache es altamente configurable y es compatible con una amplia gama de sistemas operativos.
- **Nginx:** Es un servidor web de alto rendimiento y código abierto, diseñado para manejar cargas pesadas de tráfico web. Nginx es conocido por su eficiencia en el manejo de conexiones simultáneas y su capacidad para servir contenido estático de manera rápida.
- **LiteSpeed Web Server:** Es un servidor web comercial que se promociona por su alta velocidad y eficiencia en el uso de recursos. LiteSpeed es compatible con Apache y puede funcionar como un reemplazo directo de éste.
- **Caddy:** Es un servidor web de código abierto con un énfasis en la simplicidad y la automatización. Caddy se destaca por su facilidad de configuración y su capacidad para generar certificados SSL/TLS

automáticamente.

- **Microsoft Internet Information Services (IIS):** Es el servidor web desarrollado por Microsoft para sistemas Windows. IIS es altamente integrado con otras tecnologías de Microsoft y es una opción popular para aquellos que desarrollan aplicaciones web en el ecosistema de Microsoft

Teniendo en cuenta que las APIs a desarrollar se utilizará el lenguaje de programación C#, se recomienda emplear Microsoft Internet Information Services (IIS) la cual nos permite tener una integración completa entre estas dos tecnologías. Internet Information Services está optimizado y orientado a trabajar con aplicaciones .NET Framework lo que permite aprovechar las características de este servidor web y asegurar un rendimiento óptimo lo que permite una implementación fluida.

La decisión también se basa en que IIS es parte de la infraestructura tecnológica de la empresa auspiciante y es requerimiento exclusivo que las APIs sean publicadas utilizando esta infraestructura lo cual garantiza futuras expansiones, modificaciones o nuevos requerimientos.

### **2.3.13.1 Internet Information Services.**

Según John Hart (2024) Internet Information Services (IIS) es una aplicación de tipo servicio para servidores web creada por Microsoft para los sistemas operativos basados en Windows, aplicación permite a los desarrolladores publicar y administrar sitios web así como también aplicaciones y servicios todo esto en un servidor Windows.

(Microsoft Corporation, 2024b) enlista una breve descripción de su historia y evolución:

Internet Information Services (IIS) ha evolucionado significativamente desde su introducción como un componente opcional en Windows NT 3.51 en 1995. Su versión 2.0, lanzada con Windows NT 4.0 en 1996, mejoró la seguridad y el rendimiento. En 2000, IIS 3.0, incluido en Windows 2000 Server, ofreció mejoras en escalabilidad y gestión. Entre 2000 y 2010, IIS continuó mejorando con versiones importantes como IIS 6.0 en Windows Server 2003, que introdujo el Modo de Aislamiento Web, y IIS 7.0

en Windows Server 2008, que adoptó una arquitectura modular y un nuevo modelo de configuración basado en XML. Desde 2010, IIS ha seguido modernizándose con IIS 8.0 en Windows Server 2012, que mejoró la escalabilidad y el soporte para WebSocket, e IIS 8.5 en Windows Server 2012 R2, que optimizó el rendimiento y la administración. Las versiones recientes, como IIS 10 en Windows Server 2016 y 2019, continúan mejorando la seguridad, el rendimiento y la compatibilidad con los estándares web modernos.

Para este prototipo en particular, se requerirá Internet Information Services (IIS) a fin de lograr la exposición de las APIs en la internet, ya que las tecnologías y arquitectura del servidor de la empresa Empacadora de Camarones así lo requiere, lo cual hace que éste requerimiento (la utilización de Internet Information Services) sea determinante para ello, y a su vez asegura la debida alineación del prototipo con la existente Infraestructura tecnológica; no sólo simplificando el despliegue integrado de las APIs, sino que ofrece la debida afinidad y mejora con el marco tecnológico corporativo, permitiendo una marcha eficiente y simplificada.

#### **2.3.14 Windows Server**

Krause (2021) define a Windows Server como un Sistema Operativo de característica Servidor – Cliente que contiene un conjunto de programas y aplicaciones que se complementan entre sí para el manejo administrativo de recursos compartidos en una red y con distintos clientes u otros sistemas Operativos o servicios que acceden a sus recursos a nivel corporativo. Esta amalgama de Sistemas Operativos fue creada para “servir” un amplio abanico de servicios propios de un servidor, tales como web hosting, Bases de Datos, Directorio Activos con Controladores de Dominio, almacenamiento sincronizado, seguridad en la red y otros. Su robusta arquitectura, extensa escalabilidad y administración amigable lo ubican entre los sistemas operativos de servidores de mayor acogida por empresas desde las más pequeñas hasta las mayores en tamaño y estructura.

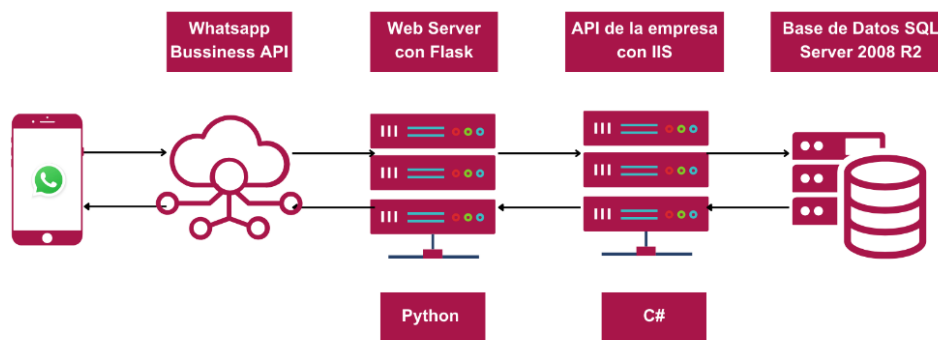
La última versión de Windows Server lanzada hasta 2024 por Microsoft es Windows Server 2022. Sin embargo, dentro de los servidores de la empresa, se encuentra la versión Windows Server 2012 R2, por tal motivo el despliegue de la API se hará en este ambiente.



Para comprender el trabajo realizado se grafica en la Figura 8 lo siguiente:

- Resumen de las herramientas de software a utilizar donde se detalla las herramientas específicas que se emplearan en el desarrollo del prototipo, todas ellas ya en uso por la empresa auspiciante.
- Representación gráfica de la infraestructura informática del prototipo, que muestra visualmente la infraestructura utilizada por la empresa auspiciante y sobre la cual se desarrollará el prototipo.

**Figura 8**  
**Infraestructura del prototipo**



**Nota:** Esquema gráfico de la implementación del prototipo.

### ¿Por qué utilizar Visual Studio y Visual Studio Code para el desarrollo del prototipo?

Visual Studio ha sido seleccionado para el desarrollo del prototipo porque es el entorno de desarrollo integrado (IDE) estándar utilizado por la empresa, formando parte integral del kit de herramientas de desarrollo aprobado. Este IDE es fundamental, ya que el sistema de producción principal de la empresa, denominado EMPAPROD, ha sido desarrollado en Visual Studio utilizando C#, lo que asegura coherencia y compatibilidad con las soluciones tecnológicas existentes.

Por otro lado, Visual Studio Code también es parte del kit de herramientas de la empresa y se utiliza en el desarrollo de ciertos sensores biométricos. Como miembro del equipo de desarrollo, se dispuso por parte de la empresa auspiciante

utilizar esta herramienta debido a su flexibilidad y ligereza, que facilitan el desarrollo en proyectos específicos que requieren un entorno de edición más ágil.

La utilización de ambas herramientas garantiza que el prototipo se desarrolle de manera consistente con los estándares tecnológicos de la empresa y se integre sin problemas con los sistemas actuales.

### **¿Por qué utilizar Python y Flask para el desarrollo del prototipo?**

Python y Flask han sido seleccionados para el desarrollo del prototipo porque son herramientas estándar aprobadas y utilizadas por la empresa en diversos proyectos, siendo Python un lenguaje de programación ampliamente adoptado dentro de la empresa que facilita la integración con los sistemas propios, y además, el desarrollador de este trabajo posee un sólido dominio de Python y Flask, lo que garantiza un desarrollo eficiente utilizando estas herramientas ya conocidas tanto por la empresa como por el desarrollador, lo que asegura no solo la coherencia con los estándares internos sino también una curva de aprendizaje mínima y una implementación más fluida.

Flask, también forma parte del kit de herramientas de desarrollo de la empresa y ha sido solicitado específicamente para este proyecto debido a facilidad en el desarrollo; permite construir aplicaciones y servicios web de manera rápida, lo que es ideal para prototipos y proyectos que requieren desarrollar de manera ágil. La utilización de ambas herramientas asegura que el desarrollo del prototipo esté alineado con los estándares tecnológicos de la empresa, facilitando su integración con los sistemas internos y mantenimientos futuros.

En consecuencia, el uso del entorno de desarrollo y lenguaje de programación fue decisión de la empresa auspiciante de este proyecto, para asegurar la integración con los sistemas internos, bases de datos y demás componentes. Esta decisión se basa en la compatibilidad técnica y, a su vez, en facilitar el mantenimiento y la escalabilidad a futuro.

## Capítulo III: Metodología De La Investigación

En este capítulo se describe el proceso investigativo aplicado en este proyecto detallando los siguientes aspectos:

- Investigación de campo: Se utiliza la metodología de investigación descriptiva y explicativa.
- Desarrollo del prototipo: Se emplea la metodología en cascada.

### Investigación Descriptiva y Explicativa

Para la recolección de información del trabajo de campo, se emplea la investigación descriptiva, ya que se pretende detallar las características y necesidades de los usuarios en cuanto al acceso a información de producción; se utiliza la investigación explicativa para comprender la relación entre el uso del prototipo y la mejora en la eficiencia del acceso a la información.

### POBLACIÓN Y MUESTRA

#### Población:

La población de estudio estará compuesta por las jefaturas y gerencias de la empresa; la muestra será intencional y estará conformada por 11 personas que pertenecen a la jefatura, quienes por su conocimiento y experiencia en el manejo de la producción y su necesidad de acceder a información en tiempo real fueron escogidas para este estudio.

**Tabla 3**  
**Población Jefaturas**

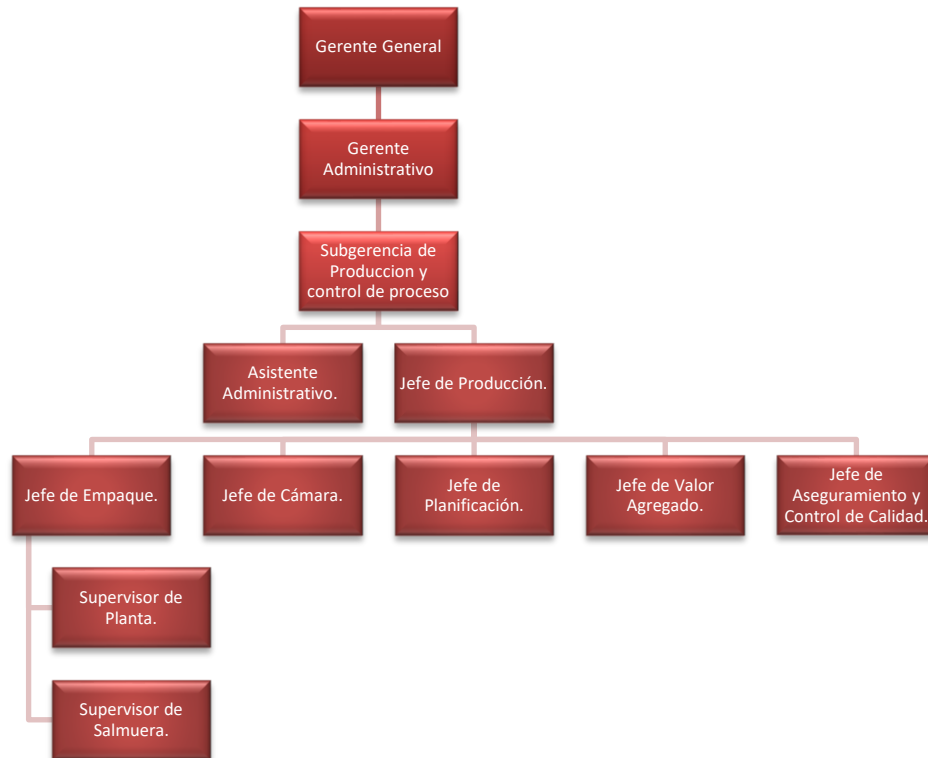
Sujetos de Estudio	# de Laborales
Gerentes	2
Jefaturas	6
Supervisores	2
Asistentes	1
<b>Total</b>	<b>11</b>

**Nota:** Se detalla la población a la cual se puede aplicar la encuesta de viabilidad del prototipo.

## Muestra:

Para el estudio, y considerando la funcionalidad del proyecto, nuestra muestra se constituye de las jefaturas y supervisores; quienes responderán una encuesta destinada al desarrollo del prototipo. En la Figura 9 se indica los funcionarios a encuestar:

**Figura 9**  
**Estructura organizacional**



**Nota:** Estructura organizacional de los laborales escogidos para la muestra

## Instrumentos de recolección de datos.

Para evaluar la viabilidad del prototipo, surge la necesidad de realizar una encuesta para obtener una opinión general sobre el uso de chatbots y recopilar información relevante para la interacción con estos. El proceso de la encuesta consiste en elaborar un formulario mediante Google Forms, lo que simplifica la distribución y tabulación de los datos.

A continuación, se observa las preguntas a responder por el personal asignado a este ejercicio.

1. ¿Para realizar actividades laborales y/o personales utiliza algún smartphone?  
 Si  No

2. El smartphone que utiliza para sus actividades tiene sistema operativo  
 Android  IOS  Otros

3. ¿Ha utilizado asistentes virtuales en aplicaciones de mensajería instantánea como WhatsApp?  
 Si  No

4. ¿Utiliza herramientas de mensajería instantánea, como WhatsApp, para comunicar o solicitar información sobre el proceso productivo del modelo de negocio?  
 Si  No

5. ¿Con qué frecuencia utiliza herramientas de mensajería instantánea como medio de comunicación para intercambiar información sobre el proceso productivo del modelo de negocio?  
 Alta  Media  Baja  Ninguna

6. ¿Le gustaría poder consultar en tiempo real mediante e uso de una aplicación móvil (chatbot) que use los servicios de Whatsapp información sobre la producción, empaque y clasificación de los camarones?  
 Si  No

7. ¿Estaría dispuesto a participar activamente en la construcción de un prototipo de chatbot aportando en la retroalimentación sobre el proceso productivo y la información a consultar?  
 Si  No

8. ¿Considera útil que un chatbot en whatsapp ayude proporcionando información sobre la producción y las exportaciones?  
 Si  No

9. ¿Considera usted que el chatbot esté disponible las 24 horas del día para consultas urgentes?  
 Si  No

10. En cuanto a la información de producción mensual, ¿considera importante poder consultar en el chatbot datos como las libras recibidas por proveedor, las libras procesadas de cola, libras procesadas de forma totalizada y las libras procesadas por Copacking?  
 Si  No

11. Respecto a la información de los reprocesos, ¿considera importante poder consultar en el chatbot datos como las libras de valor agregado por día y las libras de reproceso por día?  
 Si  No

12. Respecto a la información de inventarios, ¿considera importante poder consultar en el chatbot datos como las libras totales de inventario por tipo (por ejemplo, entero, cola, valor agregado, salmuera), las libras totales en el inventario y los inventarios por destino?  
 Si  No

13. Respecto a la producción diaria, ¿considera importante poder consultar en el chatbot la siguiente información: empaque diario por destino, libras diarias procesadas y rendimiento por piscina por día?  
 Si  No

14. ¿Considera importante poder consultar en el chatbot las siguientes informaciones: libras exportadas hasta la fecha, cantidad de contenedores exportados hasta la fecha, libras vendidas localmente hasta la fecha, ¿detalle de libras vendidas localmente por fecha específica?  
 Si  No

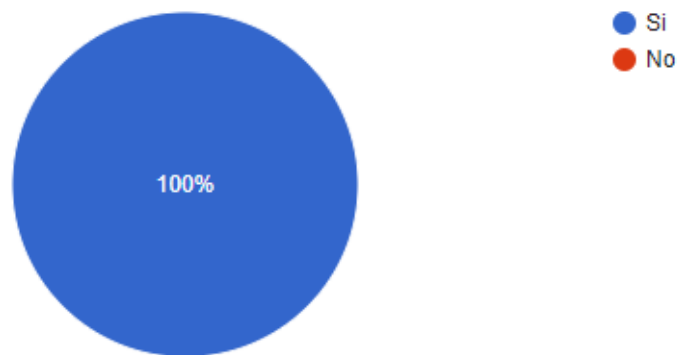
## Tabulación de encuestas

Tras la aplicación de la encuesta, se procede a la tabulación de los datos recopilados; a continuación, se presentan los resultados obtenidos en detalle:

**Pregunta 1.-** ¿Para realizar actividades laborales y/o personales utiliza algún smartphone? Todos los encuestados utilizan smartphone; a continuación, la gráfica:

**Figura 10**  
Resultados pregunta 1.

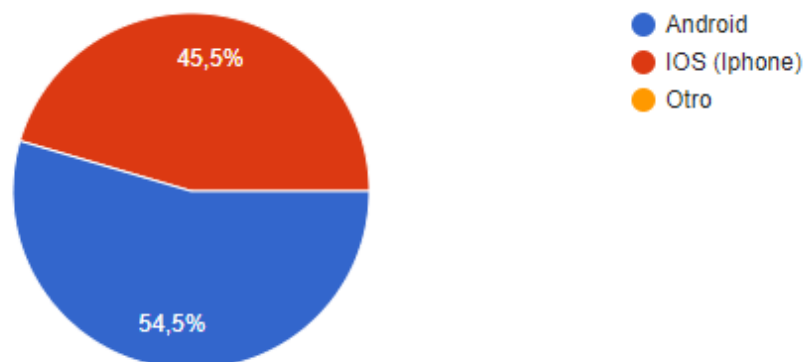
11 respuestas



**Pregunta 2.-** ¿El smartphone que utiliza para sus actividades tiene sistema operativo? Se determina que existe una utilización de sistemas operativos en smartphone de forma equitativa, como se muestra en la Figura 11

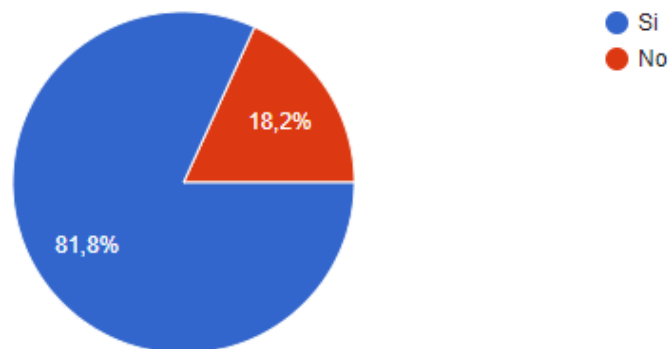
**Figura 11**  
Resultados pregunta 2.

11 respuestas



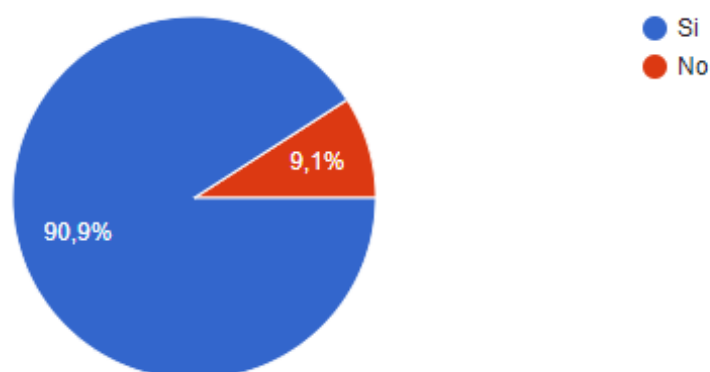
**Pregunta 3.-** ¿Ha utilizado asistentes virtuales en aplicaciones de mensajería instantánea como WhatsApp? Podemos apreciar que un pequeño grupo de laborales no ha utilizado asistentes virtuales en aplicaciones de mensajería como se muestra en la Figura 12.

**Figura 12**  
**Resultados pregunta 3.**  
11 respuestas



**Pregunta 4.-** ¿Utiliza herramientas de mensajería instantánea, como WhatsApp, para comunicar o solicitar información sobre el proceso productivo del modelo de negocio? La mayoría de encuestados utiliza las herramientas de mensajería como WhatsApp para solicitar información del proceso productivo, como se muestra en la Figura 13.

**Figura 13**  
**Resultado pregunta 4.**  
11 respuestas

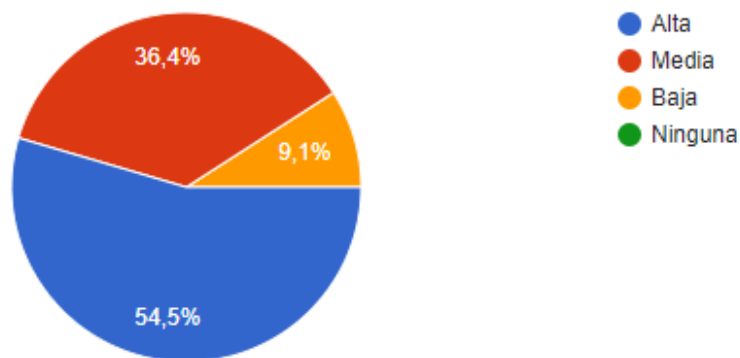


**Pregunta 5.-** ¿Con qué frecuencia utiliza herramientas de mensajería instantánea como medio de comunicación para intercambiar información sobre el

proceso productivo del modelo de negocio? El 90.9% de los participantes de la encuesta utiliza herramientas de mensajería instantánea con alta o media frecuencia para intercambiar información sobre el proceso productivo, lo que sugiere que estas herramientas son ampliamente adoptadas y consideradas importantes en su entorno de trabajo como se muestra en la Figura 14.

**Figura 14**  
**Resultado pregunta 5.**

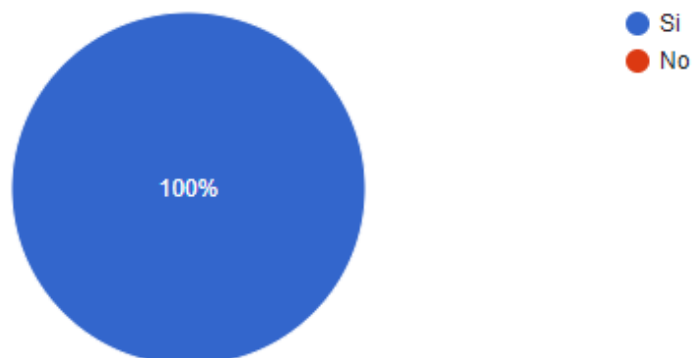
11 respuestas



**Pregunta 6.-** ¿Le gustaría poder consultar en tiempo real mediante el uso de una aplicación móvil (chatbot) que use los servicios de Whatsapp información sobre la producción, empaque y clasificación de los camarones? El 100% de los encuestados respondieron que sí les gustaría poder consultar en tiempo real esta información mediante una aplicación móvil que use WhatsApp como se muestra en la Figura 15.

**Figura 15**  
**Resultado pregunta 6.**

11 respuestas

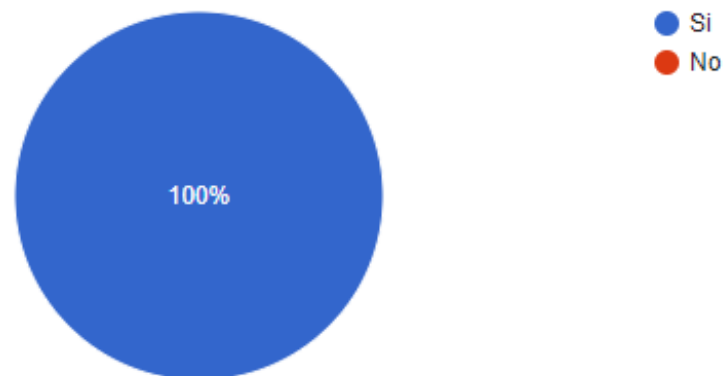




**Pregunta 7.-** ¿Estaría dispuesto a participar activamente en la construcción de un prototipo de chatbot aportando en la retroalimentación sobre el proceso productivo y la información a consultar? El 100% de los encuestados respondieron afirmativamente, indicando que estarían dispuestos a participar activamente en la construcción del prototipo de chatbot y a proporcionar retroalimentación sobre el proceso productivo y la información necesaria, a continuación, se muestra la Figura 16 con los resultados de la encuesta.

**Figura 16**  
**Resultado pregunta 7.**

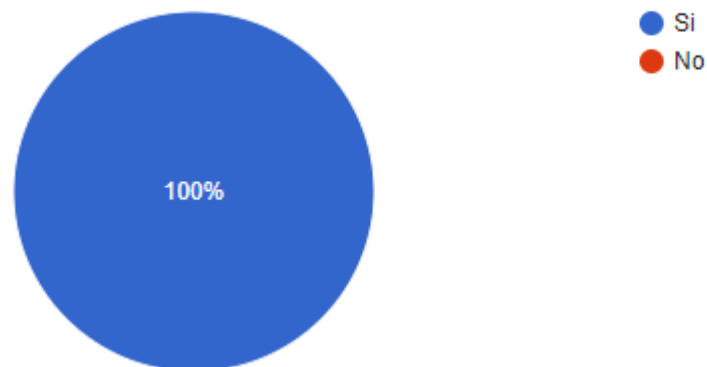
11 respuestas



**Pregunta 8.-** ¿Considera útil que un chatbot en whatsapp ayude proporcionando información sobre la producción y las exportaciones? El 100% de los encuestados respondieron de forma afirmativa donde consideran útil que un chatbot ayude proporcionando información sobre la producción y exportaciones como lo muestra en la Figura 17.

**Figura 17**  
**Resultado pregunta 8.**

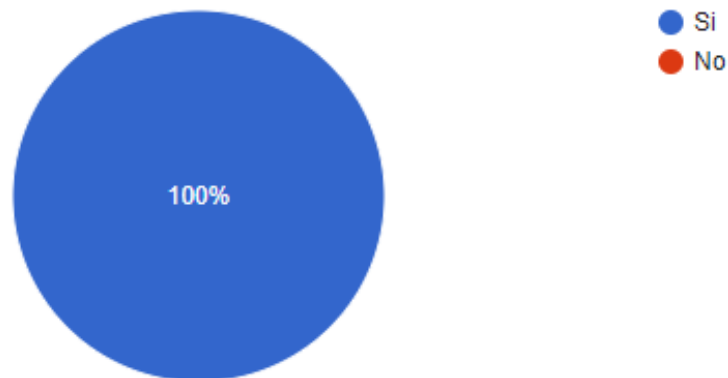
11 respuestas



**Pregunta 9.-** ¿Considera usted que el chatbot esté disponible las 24 horas del día para consultas urgentes? El 100% de los encuestados consideran importante que el chatbot esté disponible las 24 horas del día como se muestra en la Figura 18.

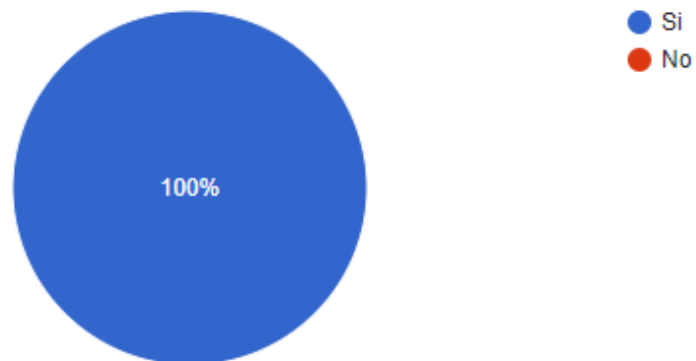
**Figura 18**  
**Resultados pregunta 9.**

11 respuestas



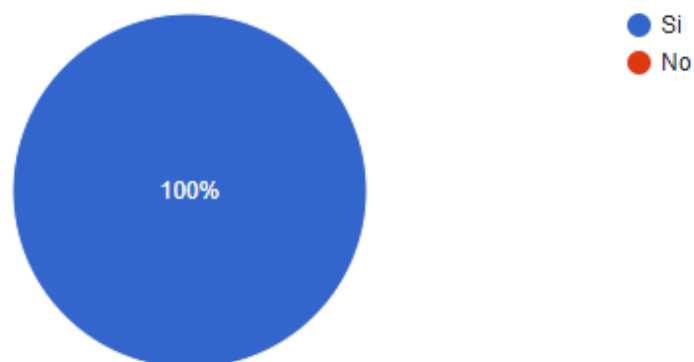
**Pregunta 10.-** En cuanto a la información de producción mensual, ¿considera importante poder consultar en el chatbot datos como las libras recibidas por proveedor, las libras procesadas de cola, libras procesadas de forma totalizada y las libras procesadas por Copacking? El 100% de los encuestados consideran importante poder consultar la información de libras recibidas y procesadas por proveedor y por proceso copacking como se muestra en la Figura 19.

**Figura 19**  
**Resultados pregunta 10.**  
11 respuestas



**Pregunta 11.-** Respecto a la información de los reprocesos, ¿considera importante poder consultar en el chatbot datos como las libras de valor agregado por día y las libras de reproceso por día? El 100% de los encuestados consideran importante consultar la información de los reprocesos en el prototipo de chatbot como muestra la gráfica de la Figura 20.

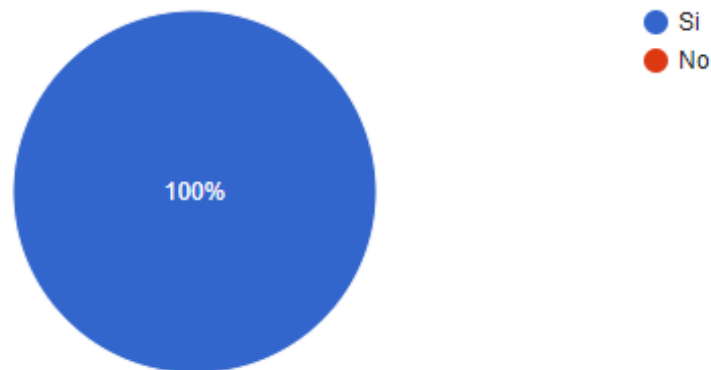
**Figura 20**  
**Resultados pregunta 11.**  
11 respuestas



**Pregunta 12.-** Respecto a la información de inventarios, ¿considera importante poder consultar en el chatbot datos como las libras totales de inventario por tipo (por ejemplo, entero, cola, valor agregado, salmuera), las libras totales en el inventario y los inventarios por destino? El 100% de los encuestados consideran importante consultar las libras totales de los inventarios por tipo tal como muestra la Figura 21.

**Figura 21**  
**Resultados pregunta 12.**

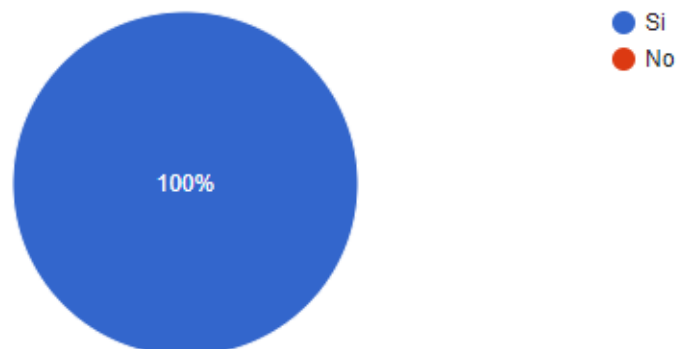
11 respuestas



**Pregunta 13.-** Respecto a la producción diaria, ¿considera importante poder consultar en el chatbot la siguiente información: empaque diario por destino, libras diarias procesadas y rendimiento por piscina por día? El 100% de los encuestados consideran importante poder consultar información de los empaques diarios por destinos y los rendimientos por piscinas como se muestra en la gráfica siguiente:

**Figura 22**  
**Resultados pregunta 13.**

11 respuestas

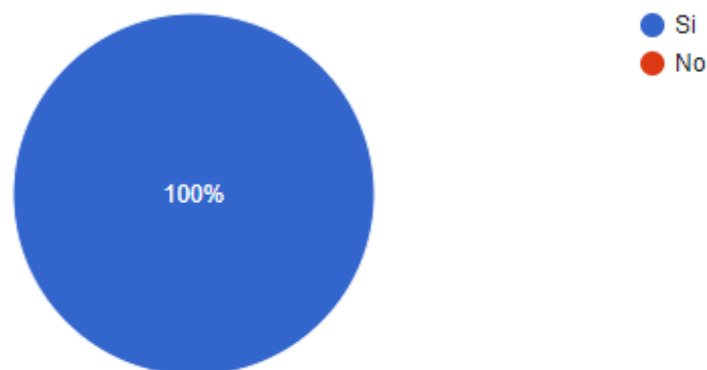


**Pregunta 14.-** ¿Considera importante poder consultar en el chatbot las siguientes informaciones: libras exportadas hasta la fecha, cantidad de contenedores exportados hasta la fecha, libras vendidas localmente hasta la fecha, ¿detalle de libras vendidas localmente por fecha específica? El 100% de los encuestados consideran importante poder consultar la información de las libras y contenedores

exportados, libras vendidas localmente por fecha tal como se muestra en la gráfica de la Figura 23.

**Figura 23**  
**Resultados pregunta 14.**

11 respuestas



Finalizado el análisis respectivo se indica que el 89% de funcionarios está de acuerdo con que la implementación de un chatbot facilitará las consultas que realizan habitualmente; este hallazgo indica una fuerte aceptación y disposición por parte del personal para adoptar esta tecnología, lo que sugiere que la implementación del chatbot no solo será bien recibida, sino que también mejorará significativamente la eficiencia y la experiencia de los usuarios en sus interacciones diarias.

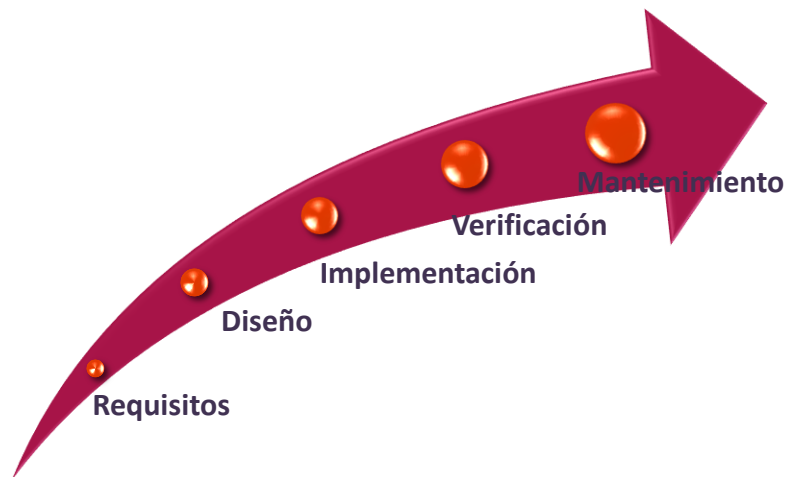
### **Metodología de desarrollo de software**

La metodología de cascada se compone de 5 fases que en su conjunto facilitan una planificación, ejecución y control detallada del proyecto; garantizando que el prototipo cumpla con los requisitos y expectativas definidas por modelo de negocio.

Lisdania de la Caridad D. & Lexys Manuel (2021) mencionan que este modelo estructura las actividades fundamentales (especificación, desarrollo, validación y evolución) en fases claramente definidas lo cual permite realizar estimaciones más precisas de tiempo y recursos, gestionando mejor las expectativas del usuario y proporcionando actualizaciones regulares sobre el progreso. Como resultado, se logra un mayor nivel de satisfacción del usuario en comparación con enfoques menos

estructurados. A continuación, se detalla la estructura gráfica de la metodología de construcción de software.

**Figura 24**  
**Modelo de la metodología en cascada**



**Nota:** El modelo en cascada es un enfoque secuencial para el desarrollo de software que comienza con la especificación de requisitos y pasa por las fases de diseño, implementación, verificación y despliegue, terminando con el soporte del software finalizado (Pressman & Maxim, 2021).

Cada uno de los elementos del modelo comprenden una fase del proyecto el cual se trabajó de forma individual para esto se explica en que consiste cada uno de ellos.

Según Lisdania de la Caridad D. & Lexys Manuel (2021), en los requisitos se define las características y funcionalidades del prototipo, así como el objetivo y alcance del desarrollo, buscando realizar una definición detallada del mismo. En la fase de diseño, se trabaja en la parte del sistema, identificando las partes de hardware y software a intervenir, realizando la arquitectura, diagramas y relaciones del prototipo.

Aguirre Barrera & Aguirre Barrera (2021) describe que, en la implementación, se realizan las tareas de programación y ejecución para obtener los entregables del prototipo. En la fase de verificación, se ejecutan las pruebas necesarias con el usuario para comparar que se cumpla lo establecido en la fase de definición de requerimientos. En caso de encontrar alguna desviación, se realizan las correcciones necesarias.

## Capítulo IV: Propuesta Tecnológica

En este capítulo se documenta el desarrollo del proyecto, manteniendo un registro detallado de cada punto de control, el cual se organiza y categoriza acorde a la metodología de construcción de software seleccionada.

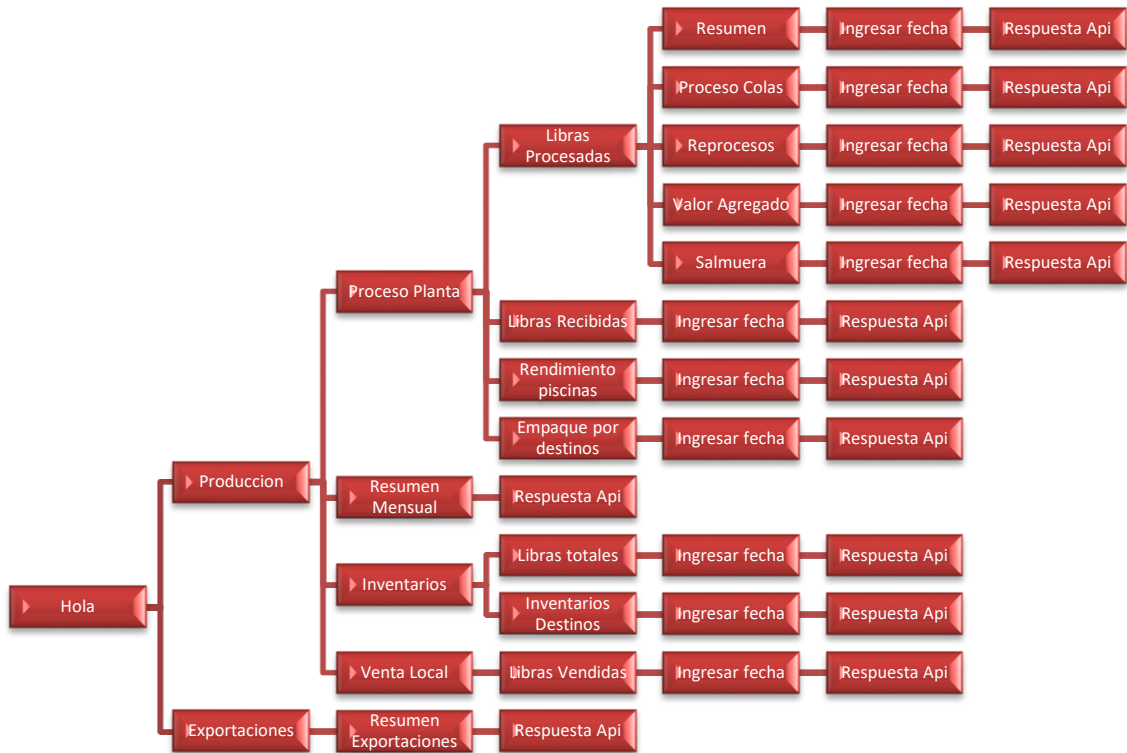
### **Levantamiento de información.**

El levantamiento de los requisitos se realizó aplicando el formato definido por la empresa auspiciante de este proyecto, el cual se detalla en el anexo 1. Durante este proceso, se concluyó que el prototipo debe ser capaz de responder a las siguientes preguntas clave relacionadas con la producción y el inventario:

- Libras diarias procesadas
- Libras recibidas por proveedor
- Empaques diarios por destino
- Rendimiento diario por piscinas
- Libras procesadas por mes
- Libras de valor agregado por día
- Libras de reprocesos por día
- Libras de cola por día
- Libras procesadas por copacking
- Libras totales de inventarios
- Libras exportadas a la fecha
- Libras de venta local vendidas por fecha.

En base a lo analizado en el levantamiento de información y para tener una mayor comprensión del flujo de conversación que se quiere tener dentro del prototipo se elaboró el siguiente diagrama:

**Figura 25**  
**Diagrama de Flujo de conversación**



**Nota:** El siguiente diagrama muestra el flujo de conversación en base a cada opción que presenta el prototipo,

**Diseño del prototipo.**

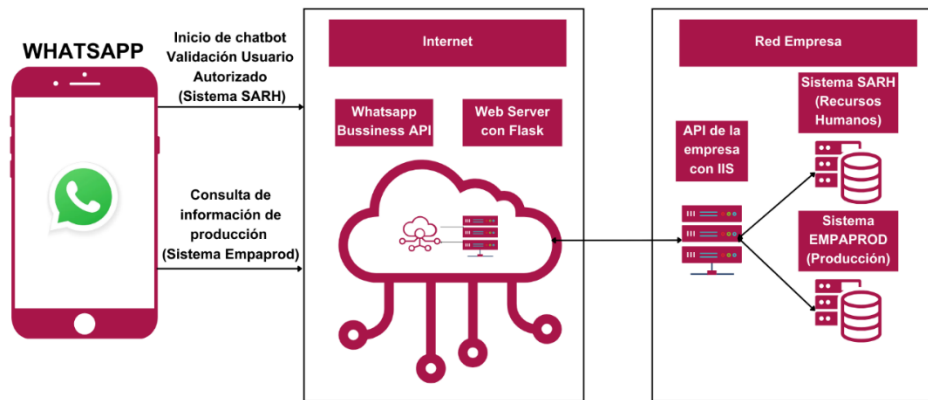
Para empezar la fase de diseño identificamos el origen de los datos y como se accede a ellos tal cual se muestra a continuación:

**Arquitectura de la solución.**

La arquitectura se presenta de la siguiente forma:



**Figura 26**  
**Arquitectura de la solución.**



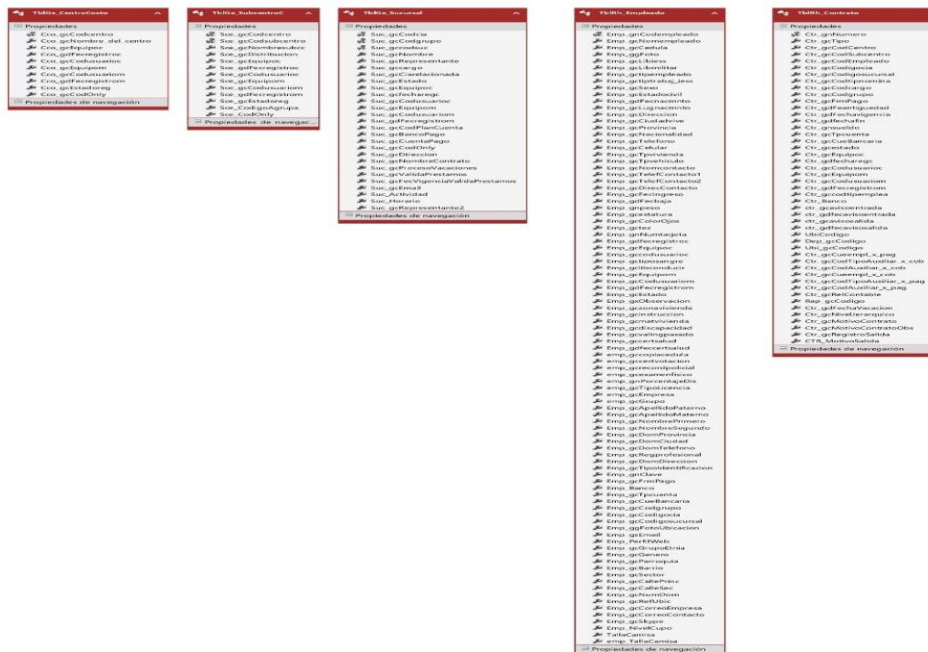
**Nota:** Se presenta la arquitectura de la solución identificando tanto la parte que será desplegada en un servidor en internet como la comunicación con los servidores de la empresa.

**Origen de Datos.**

La fuente de datos proviene de las bases de datos transaccionales de la empresa, estas son dos y almacenan la información utilizada para generar respuestas del chatbot:

**Base de datos del sistema SARH.** - Sistema de recursos humanos que valida que los usuarios sean colaboradores activos.

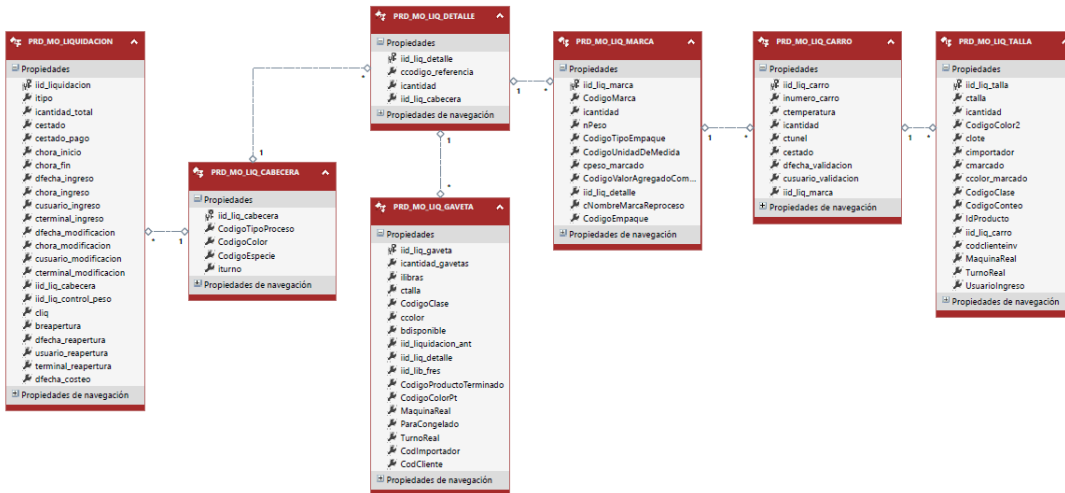
**Figura 27**  
**Tablas de la base de datos SARH.**



**Nota:** Tablas (5) de la base de datos SARH con las que el prototipo valida al usuario.

**Base de datos del sistema EMPAPROD.** - Sistema de producción que proporciona la data requerida por el prototipo su operación.

**Figura 28**  
**Modelo Entidad-Relación Base de datos sistemas Empaprod.**



**Nota:** Modelo de la estructura de la base de datos del sistema EMPAPROD consiste en 7 tablas las cuales poseen información de la producción.

## Modelo entidad relación

El prototipo interactúa con dos bases de datos de los sistemas de información de producción de la empacadora: SARH y EMPAPROD, tal como se explica en el ítem Origen de Datos, por lo que este prototipo no requiere de un modelo de entidad-relación propio.

## Proceso de extracción, transformación y carga de datos.

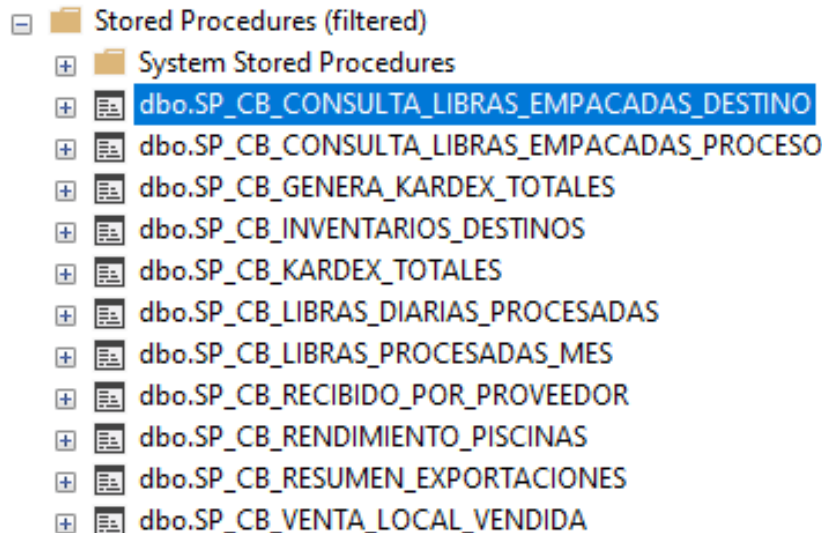
Para optimizar las respuestas del prototipo mediante la abstracción, formación y depuración de datos en tiempo real, se centra en el uso de procedimientos almacenados en SQL Server, realizando consultas en el sistema EMPAPROD.

## Abstracción de datos mediante procedimientos almacenados.

Se desarrollan procedimientos almacenados que encapsulen la lógica de negocio y realicen consultas complejas a las tablas del EMPAPROD. Estos procedimientos deben abstraer la complejidad de las consultas, permitiendo que el prototipo reciba solo la información necesaria, para se crearon 11 procedimientos los

cuales poseen las consultas a los datos de producción tal como se muestra en la Figura 29

**Figura 29**  
Procedimientos almacenados



**Nota:** Procedimientos almacenados realizados los cuales contienen las consultas del prototipo.

Se estructuran los procedimientos almacenados de manera modular, donde cada procedimiento se encargue de una tarea específica, como obtener el estado de la producción, los inventarios, o los tiempos de procesamiento. A continuación, se muestra parte del procedimiento almacenado en la Figura 30 donde su estructura se encuentra de forma modular.

**Figura 30**  
Estructura de un procedimiento de forma modular.

```

DECLARE @PROVEEDOR_DESTINO TABLE(COD_IMPORTADOR VARCHAR(20), DESTINO VARCHAR(20))
DECLARE @RESULTADO TABLE(CodigoProductoTerminado char(61), Libras decimal(18,4), Destino varchar(200))

3 INSERT INTO @PROVEEDOR_DESTINO
(
COD_IMPORTADOR,DESTINO
)
SELECT ccodigo,case when Continente is null
then 'OTROS' ELSE Continente end
FROM Empacadora.dbo.CEN_MA_IMPORTADOR IMPORTADOR WITH (NOLOCK)
union
select 'N/A','ESTADOS UNIDOS'

3 INSERT INTO @RESULTADO
SELECT cat.CodigoProductoTerminado,
Libras=SUM([dbo].[FPRD_CON_CONVERSION_PES04](talla.icantidad*marca.nPeso,marca.CodigoUnidadDeMedida,'LB'))
, DESTINO= CASE WHEN (talla.cimportador = 'N/A' AND cat.CodigoEmpaque= 'M20K010CJT2 KL') THEN 'EUROPA'
WHEN (cat.CodigoEmpaque= 'M18K010CJT1.80 KL' AND cat.CodigoMarca = 'FRR' AND talla.cimportador = 'N/A') THEN 'STOCK ASIA'
WHEN liqec.CodigoProveedor in ('081') and pd.DESTINO = 'OTROS' THEN 'ESTADOS UNIDOS SONGA' ELSE PD.DESTINO END
FROM
Empacadora.dbo.PRD_MO_LIQUIDACION liq WITH (NOLOCK)
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_CABECERA cab WITH (NOLOCK) ON liq.iid_liq_cabecera = cab.iid_liq_cabecera
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_DETALLE det WITH (NOLOCK) ON cab.iid_liq_cabecera = det.iid_liq_cabecera
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_MARCA marca WITH (NOLOCK) ON marca.iid_liq_detalle = det.iid_liq_detalle
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_CARRO carro WITH (NOLOCK) ON carro.iid_liq_marca = marca.iid_liq_marca
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_TALLA talla WITH (NOLOCK) ON talla.iid_liq_carro = carro.iid_liq_carro
INNER JOIN Empacadora.dbo.CatalogoProductoTerminado cat WITH (NOLOCK) ON cat.IdProducto=talla.IdProducto AND cat.CodigoEmpaque=marca.CodigoEm
INNER JOIN @PROVEEDOR_DESTINO PD ON PD.COD_IMPORTADOR = talla.cimportador
inner join empacadora.dbo.PRD_MO_LIQUIDACION_EC liqec WITH (NOLOCK) on liqec.iid_liquidacion = liq.iid_liquidacion
WHERE cast(liq.dfecha_ingreso as date) >= CAST(@FECHA_DESDE as date) and cast(liq.dfecha_ingreso as date) <= CAST(@FECHA_HASTA as date)
and liqec.CodigoProveedor not in('081') AND liq.cestado not in ('A')
GROUP BY cat.CodigoProductoTerminado, PD.DESTINO,talla.cimportador,cat.CodigoEmpaque,cat.CodigoMarca ,liqec.CodigoProveedor

```

**Nota:** Se muestra la información estructurada de un procedimiento almacenado.

Se utilizan parámetros de entrada claros y precisos para personalizar las consultas según las necesidades del prototipo tal como se muestra en la Figura 31.

**Figura 31**  
Ejemplo de parámetros de entrada de los procedimientos almacenados.

```
ALTER PROCEDURE [dbo].[SP_CB_CONSULTA_LIBRAS_EMPACADAS_DESTINO]
-- Add the parameters for the stored procedure here
@FECHA_DESDE DATETIME,
@FECHA_HASTA DATETIME
```

**Nota:** Se utilizan parámetros para poder filtrar la información, como parte del proceso de abstracción.

## Transformación de datos mediante procedimientos almacenados.

Se implementa lógica dentro de los procedimientos almacenados para normalizar los datos, como convertir formatos de fechas, unificar unidades de medida y manejar cadenas de texto. Como se muestra en la Figura 32, se utiliza una función para realizar transformación de unidades de medidas en el campo libras, y en la condición de la consulta se realiza la transformación a formato fecha.

**Figura 32**  
Conversión de formatos y unificación de unidades.

```
SELECT cat.CodigoProductoTerminado,
Libras=SUM([dbo].[FPRD_CONVERSION_PES04](talla.icantidad*marca.nPeso,marca.CodigoUnidadDeMedida,'LB'))
, DESTINO= CASE WHEN (talla.cimportador = 'N/A' AND cat.CodigoEmpaque= 'M20K010CJT2 KL') THEN 'EUROPA'
WHEN (cat.CodigoEmpaque= 'M18K010CJT1.80 KL' AND cat.CodigoMarca = 'FRR' AND talla.cimportador = 'N/A') THEN 'STOCK ASIA'
WHEN liqec.CodigoProveedor in ('081') and pd.DESTINO = 'OTROS' THEN 'ESTADOS UNIDOS SONGA' ELSE PD.DESTINO END
FROM
Empacadora.dbo.PRD_MO_LIQUIDACION liq WITH (NOLOCK)
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_CABECERA cab WITH (NOLOCK) ON liq.iid_liq_cabecera = cab.iid_liq_cabecera
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_DETALLE det WITH (NOLOCK) ON cab.iid_liq_cabecera = det.iid_liq_cabecera
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_MARCA marca WITH (NOLOCK) ON marca.iid_liq_detalle = det.iid_liq_detalle
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_CARRO carro WITH (NOLOCK) ON carro.iid_liq_marca = marca.iid_liq_marca
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_TALLA talla WITH (NOLOCK) ON talla.iid_liq_carro = carro.iid_liq_carro
INNER JOIN Empacadora.dbo.CatalogoProductoTerminado cat WITH (NOLOCK) ON cat.IdProducto=talla.IdProducto AND cat.CodigoEmpaque=marca.CodigoEmpaque
INNER JOIN @PROVEEDOR_DESTINO PD ON PD.COD_IMPORTADOR = talla.cimportador
inner join Empacadora.dbo.PRD_MO_LIQUIDACION_EC liqec WITH (NOLOCK) on liqec.iid_liquidacion = liq.iid_liquidacion
WHERE cast(liq.dfecha_ingreso as date) >= CAST(@FECHA_DESDE as date) and cast(liq.dfecha_ingreso as date) <= CAST(@FECHA_HASTA as date)
and liqec.CodigoProveedor not in ('081') AND liq.cestado not in ('A')
GROUP BY cat.CodigoProductoTerminado, PD.DESTINO,talla.cimportador,cat.CodigoEmpaque,cat.CodigoMarca ,liqec.CodigoProveedor
```

**Nota:** Se utilizan funciones para conversión de unidades y transformación de fechas en la condición de la consulta.

Dentro de los procedimientos almacenados se optimizan las consultas SQL mediante el uso de joins<sup>20</sup>, y el manejo adecuado de subconsultas para asegurar que las transformaciones se realicen rápidamente. A continuación, en la Figura 33 se muestra un ejemplo el uso de joins entre las tablas de la producción para obtener parte de la información de la producción y así como también una subconsulta limitada a un solo campo para filtrar estratégicamente los resultados de la consulta principal, restringiendo el conjunto de datos a un rango específico de información.

<sup>20</sup> Join es una operación que permite combinar filas de dos tablas en una base de datos.

**Figura 33**  
**Uso de Joins y manejo de subconsultas.**

```

INSERT INTO @peso_total
( iid_liquidacion, libras )
SELECT a.iid_liquidacion, SUM(Empacadora.[dbo].[FPRD_CON_CONVERSION_PES03] ((liqmarca.nPeso*liqtalla.icantidad),liqmarca.CodigoUnidadDeMedida,'LB'))
FROM Empacadora.dbo.PRD_MO_LIQUIDACION a WITH(NOLOCK)
INNER JOIN Empacadora.dbo.PRD_MO_LIQ_DETALLE det WITH(NOLOCK) ON det.iid_liq_cabecera = a.iid_liq_cabecera
inner join Empacadora.dbo.PRD_MO_LIQ_CABECERA cab WITH(NOLOCK) on a.iid_liq_cabecera=cab.iid_liq_cabecera
inner join Empacadora.dbo.PRD_MO_LIQUIDACION_EC g WITH(NOLOCK) on a.iid_liquidacion = g.iid_liquidacion
inner join Empacadora.dbo.PRD_MO_LIQ_MARCA liqmarca WITH(NOLOCK) on liqmarca.iid_liq_detalle = det.iid_liq_detalle
inner join Empacadora.dbo.PRD_MO_LIQ_CARRO liqcarro WITH(NOLOCK) on liqcarro.iid_liq_marca = liqmarca.iid_liq_marca
inner join Empacadora.dbo.PRD_MO_LIQ_TALLA liqtalla WITH(NOLOCK) on liqtalla.iid_liq_carro = liqcarro.iid_liq_carro
WHERE a.iid_liquidacion IN (SELECT liq FROM @Liquidaciones)
GROUP BY a.iid_liquidacion

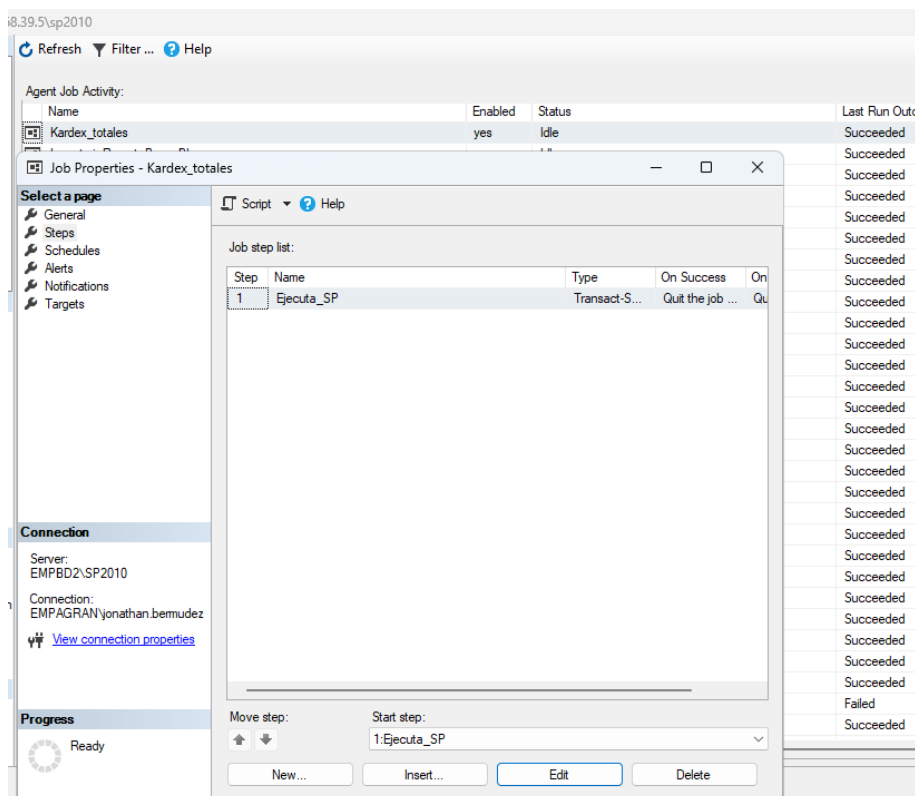
```

**Nota:** Las subconsultas se limitan a un solo campo para mejorar la búsqueda en base a la condición.

### Depuración de los datos en los procedimientos almacenados.

La depuración de datos se lleva a cabo mediante la implementación de trabajos automáticos en la base de datos los cuales realizan consultas periódicas lo cual nos permite tener recolección de datos en tiempo real. Estos trabajos automatizados extraen la información del proceso productivo de las tablas del sistema EMPAPROD, asegurando que los datos estén siempre actualizados y disponibles para su procesamiento por parte del prototipo. En la Figura 34 se muestra la programación de la tarea para la generación de los inventarios de producción.

**Figura 34**  
**Ejemplo de trabajo automático desarrollado para el prototipo.**

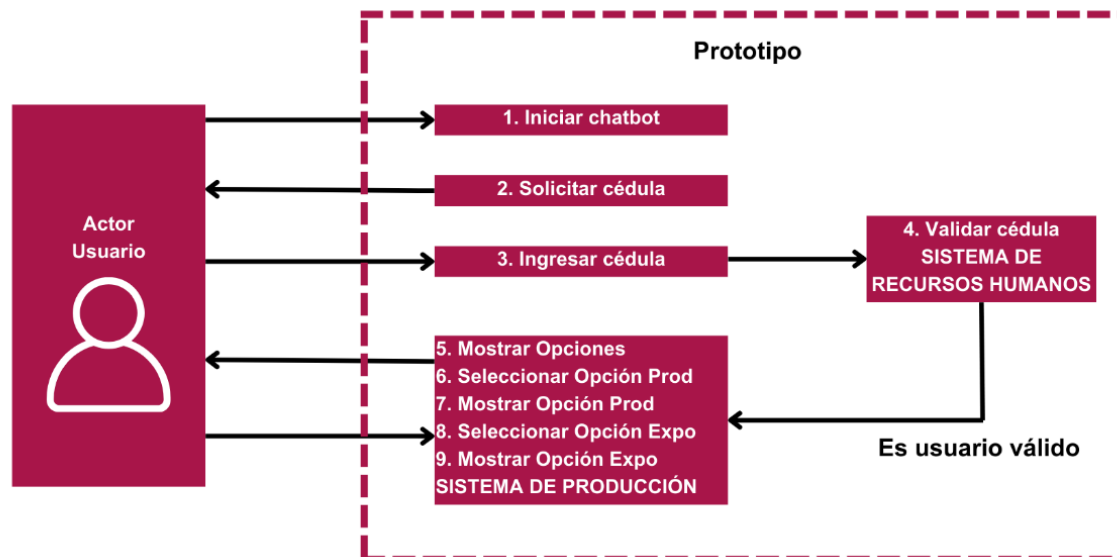


**Nota:** Estos trabajos son importantes para poder mostrar la información de forma resumida para el prototipo.

## Diagrama de casos de uso

Para identificar el flujo de trabajo del prototipo y las diferentes interacciones entre los sistemas de la empresa y el chatbot, se construyó el siguiente diagrama de caso de uso:

Figura 35  
Diagrama de caso de uso.



**Nota:** Este diagrama proporciona una visión general del funcionamiento del sistema y el flujo de información, ayudando a visualizar cómo los usuarios interactúan con el chatbot y cómo se comunican los sistemas internos para proporcionar datos en tiempo real.

## Ambiente de construcción del prototipo

### Instalación de VirtualEnv.

Para configurar el entorno virtual de Python ejecutamos el siguiente comando: “*pip3 install virtualenv*”, como se muestra en la Figura 36.

Figura 36  
Instalación VirtualEnv.

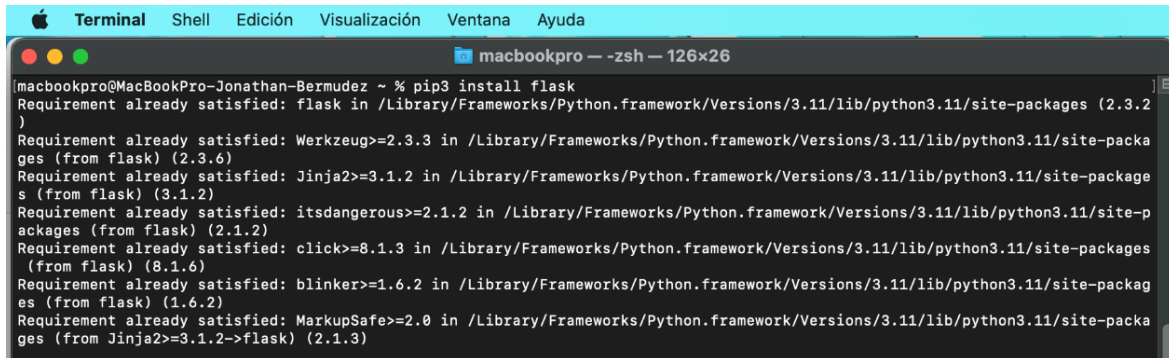
```
macbookpro -- -zsh -- 126x26
Last login: Mon Jul 8 21:08:37 on console
macbookpro@MacBookPro-Jonathan-Bermudez ~ % pip3 install virtualenv
Requirement already satisfied: virtualenv in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (20.25.1)
Requirement already satisfied: distlib<1,>=0.3.7 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from virtualenv) (0.3.8)
Requirement already satisfied: filelock<4,>=3.12.2 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from virtualenv) (3.13.3)
Requirement already satisfied: platformdirs<5,>=3.9.1 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from virtualenv) (4.2.0)
```

**Nota:** Ventana de resultado de la instalación del entorno virtual

### Instalación de Flask

Como servidor web que nos permite la comunicación entre el prototipo y las APIs de la empresa se utilizará Flask; para esto, en el terminal de desarrollo ejecutamos el comando: “*pip3 install flask*”, como se muestra en la Figura 37.

**Figura 37**  
**Instalación de Flask.**



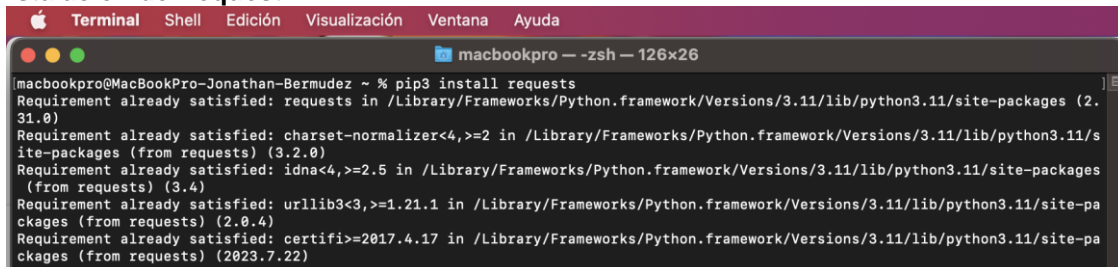
```
macbookpro@MacBookPro-Jonathan-Bermudez ~ % pip3 install flask
Requirement already satisfied: flask in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (2.3.2)
Requirement already satisfied: Werkzeug>=2.3.3 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from flask) (2.3.6)
Requirement already satisfied: Jinja2>=3.1.2 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from flask) (8.1.6)
Requirement already satisfied: blinker>=1.6.2 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from flask) (1.6.2)
Requirement already satisfied: MarkupSafe>=2.0 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from Jinja2>=3.1.2->flask) (2.1.3)
```

**Nota:** Ventana de resultado de la instalación del servidor web Flask

### Instalación de la librería Request.

Se procede a instalar la librería request, necesaria para la interacción con las APIs, mediante el comando: “*pip3 install requests*” dentro del terminal de desarrollo, como se muestra en la Figura 38.

**Figura 38**  
**Instalación de Request.**



```
macbookpro@MacBookPro-Jonathan-Bermudez ~ % pip3 install requests
Requirement already satisfied: requests in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4, >=2 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from requests) (3.2.0)
Requirement already satisfied: idna<4, >=2.5 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from requests) (3.4)
Requirement already satisfied: urllib3<3, >=1.21.1 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from requests) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages (from requests) (2023.7.22)
```

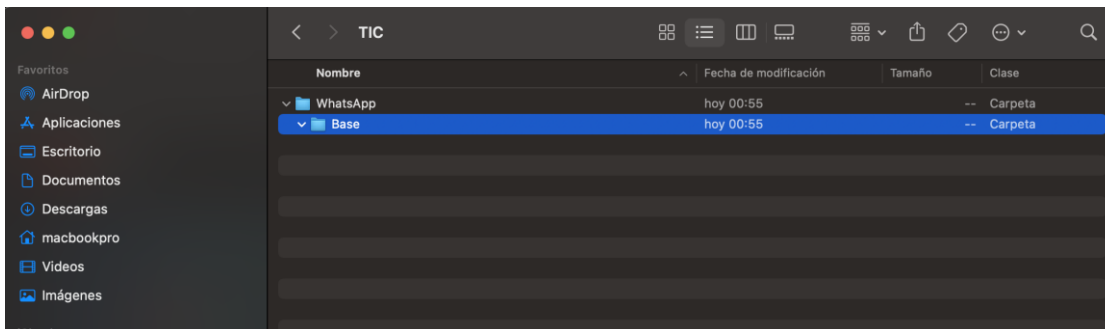
**Nota:** Ventana de resultado de la instalación de la librería Request.

### Sistema de Archivos

Para el sistema de archivos del prototipo se crea una estructura básica de carpetas como se muestra en la Figura 39.

**Figura 39**  
**Estructura básica de carpetas del proyecto.**





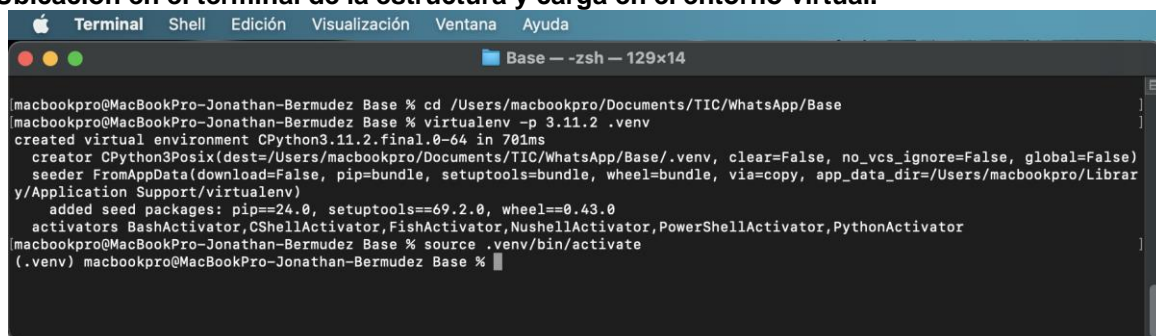
**Nota:** La estructura no obedece a un estándar en sí, queda a total criterio del desarrollador nombrar las carpetas.

Luego de crear las estructuras de carpetas se debe crear el entorno virtual dentro de este, para lo cual se abre el terminal y se navega hasta la carpeta creada con el comando:

`“cd /Users/macbookpro/Documents/TIC/WhatsApp/Base”`

para luego de esto crear el entorno virtual dentro de la misma con el siguiente comando: `“virtualenv -p 3.11.2 .venv”` tal como se muestra en la Figura 40.

**Figura 40**  
Ubicación en el terminal de la estructura y carga en el entorno virtual.

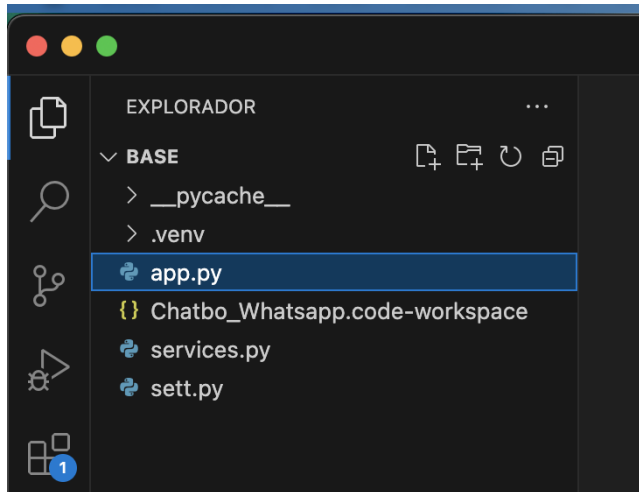


**Nota:** Se crea el entorno virtual dentro de la carpeta para poder ejecutar Flask

Una vez cargado el entorno virtual, abrir Visual Studio Code y se arrastra al explorador para comenzar a crear los elementos necesarios para el desarrollo del prototipo como se muestra en la Figura 41.

**Figura 41**  
Estructura de archivos dentro del entorno de desarrollo.





**Nota:** Se muestra estructura base de archivos del prototipo.

Dentro del mismo se muestran 3 archivos los cuales se documentan a continuación:

- **Archivo app.py.** - Este es el archivo principal de ejecución que llama a todas las funciones del prototipo.
- **Archivo sett.py.** - Este archivo es el que contiene todos los parámetros, URLs<sup>21</sup> y Tokens<sup>22</sup> de conexión al servicio de meta y a las APIs de la empresa.
- **Archivo services.py.** - Este archivo contiene la lógica de validaciones, preguntas y respuestas del prototipo.

### **Desarrollo de funciones del prototipo.**

Se desarrolla el archivo app.py que es el archivo que inicia cuando se ejecuta la aplicación en el servidor.

1. Se crea la función `verificar_token()` que sirve para verificar el token de acceso a los servicios de Meta desde el servidor web levantado.

**Figura 42**  
**Función verificar Token.**

<sup>21</sup> URL, por sus siglas en inglés Uniform Resource Locator

<sup>22</sup> Token nos sirve como llave para podernos conectar a los servicios de meta.

```

7  @app.route('/webhook', methods=['GET'])
8  def verificar_token():
9      try:
10         token = request.args.get('hub.verify_token')
11         challenge = request.args.get('hub.challenge')
12
13         if token == sett.token and challenge != None:
14             return challenge
15         else:
16             return 'token incorrecto', 403
17     except Exception as e:
18         return e,403

```

**Nota:** Código fuente función verificar\_token().

2. Se desarrolla la función `recibir_mensajes()` que es la encargada de recibir el mensaje del usuario y descomponerlo para que pueda ser interpretado por el chatbot con la petición de las diferentes opciones que tiene el prototipo.

**Figura 43**  
Función Recibir mensaje.

```

20  @app.route('/webhook', methods=['POST'])
21  def recibir_mensajes():
22      try:
23         body = request.get_json()
24         entry = body['entry'][0]
25         changes = entry['changes'][0]
26         value = changes['value']
27         message = value['messages'][0]
28         #number = services.replace_start(message['from'])
29         number = message['from']
30         messageId = message['id']
31         contacts = value['contacts'][0]
32         name = contacts['profile']['name']
33         text = services.obtener_Mensaje_whatsapp(message)
34
35         services.administrar_chatbot(text, number,messageId,name)
36         return 'enviado'
37
38     except Exception as e:
39         return 'no enviado ' + str(e)

```

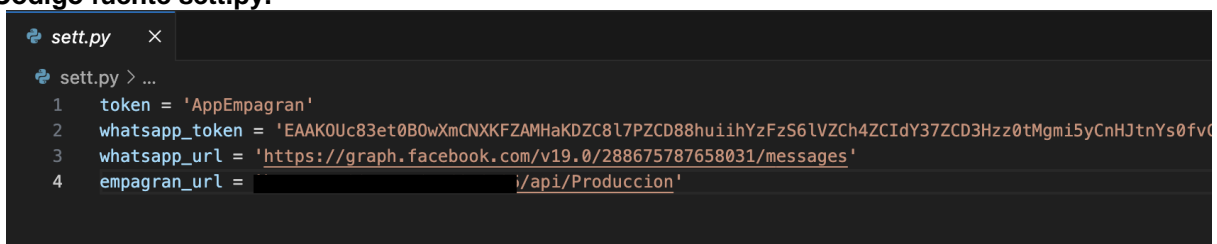
**Nota:** Código fuente función recibir\_mensaje()

El archivo `sett.py` actúa como un archivo de configuración que define varias variables necesarias para el funcionamiento del prototipo la cual permite la comunicación con los servicios externos para su operación entre las cuales tenemos:

- **Token y whatsapp\_token** son proporcionadas por la cuenta de Meta creada para la aplicación, lo que permite la autenticación y el acceso a los servicios de WhatsApp a través de la API. La variable token está configurada como 'AppEmpagran', mientras que whatsapp\_token contiene un largo token alfanumérico utilizado para la autenticación en la API de WhatsApp.
- **whatsapp\_url** almacena la URL del endpoint de la API de WhatsApp para el envío de mensajes.
- **empagran\_url** contiene la URL específica de la empresa empacadora de camarones, y está configurada con la IP pública fija.

La Figura 44 muestra como está desarrollado el archivo.

**Figura 44**  
**Código fuente sett.py.**



```

sett.py x
sett.py > ...
1 token = 'AppEmpagran'
2 whatsapp_token = 'EAAK0Uc83et0B0wXmCNXKFZAMHaKDZC8L7PZCD88huiihYzFzS6lVZCh4ZCIdY37ZCD3Hzz0tMgmi5yCnHJtnYs0fv0'
3 whatsapp_url = 'https://graph.facebook.com/v19.0/288675787658031/messages'
4 empagran_url = 'https://192.168.1.100/api/Produccion'

```

**Nota:** Código de almacenamiento de variables.

El archivo services.py contiene la comunicación con las APIs de la empresa, el flujo de información y las funciones principales de respuesta del prototipo las cuales se mencionan a continuación.

1. La función `obtener_Mensaje_whatsapp()` ayuda a descomponer el JSON obtenido de la comunicación con la API de Meta con el fin de obtener el texto que deseamos procesar, a continuación, se muestra el código en la Figura 45.

**Figura 45**  
**Función obtener mensaje whatsapp.**

```

def obtener_Mensaje_whatsapp(message):
    if 'type' not in message :
        text = 'mensaje no reconocido'
        return text

    typeMessage = message['type']
    if typeMessage == 'text':
        text = message['text']['body']
    elif typeMessage == 'button':
        text = message['button']['text']
    elif typeMessage == 'interactive' and message['interactive']['type'] == 'list_reply':
        text = message['interactive']['list_reply']['title']
    elif typeMessage == 'interactive' and message['interactive']['type'] == 'button_reply':
        text = message['interactive']['button_reply']['title']
    else:
        text = 'mensaje no procesado'

    return text

```

**Nota:** Muestra como procesa el mensaje para obtener el texto de este y retornarlo como variable.

2. Se desarrolla la función `administrar_chatbot()` la cual contiene el flujo de respuestas en base a cada palabra u opción que genere el usuario, a continuación, se muestra el código en la Figura 46.

**Figura 46**  
**Función administrar chatbot.**

```

62 def administrar_chatbot(text, number, messageId, name):
63     #mensaje que envio el usuario
64     list = []
65     global conversacion
66     markRead = markRead_Message(messageId)
67     list.append(markRead)
68     time.sleep(2)
69     print(text)
70     print(str(conversacion))
71     if "Hola" in text:
72         conversacion = 1
73         textMessage = text_Message(number, "¡Hola! 🤖 Bienvenido a ChatBot de Empagran 🇵🇷. ¿Por favor digita tu cédula para saber si estas autorizado a ve
74         footer = "Equipo Sistemas Empagran"
75         list.append(textMessage)
76     elif validarCliente(text) != "" and conversacion == 1:
77         conversacion = 2
78         body = "¡Hola! " + Nombre_Usuario + " 🤖 Bienvenido a ChatBot de Empagran 🇵🇷. ¿Cómo podemos ayudarte hoy? 🇵🇷"
79         footer = "Equipo Sistemas Empagran"
80         options = [{"🇵🇷 Produccion"}, {"🇵🇷 Exportacion"}]
81         replyButtonData = buttonReply_Message(number, options, body, footer, "sed1",messageId)
82         list.append(replyButtonData)
83     elif "Produccion" in text and conversacion == 2:
84         conversacion = 3
85         body = "La presente lista te muestra todas las consultas disponibles de la producción. ¿Que te gustaría saber?"
86         footer = "Equipo Sistemas Empagran"
87         options = [{"🇵🇷 Libras Empacadas"}, {"Proceso Salmuera"}, {"Proce (variable) footer: Literal['Equipo Sistemas Empagran']}
88         listReplyData = listReply_Message(number, options, body, footer, "sed2",messageId)
89         list.append(listReplyData)
90     elif "Exportacion" in text and conversacion == 2:
91         body = "La presente lista te muestra todas las consultas disponibles de las exportaciones. ¿Que te gustaría saber?"
92         footer = "Equipo Sistemas Empagran"
93         options = [{"Contenedores Exportados"}, {"Contenedores en planta"}, {"Exportaciones por destino"}, {"Volver menú anterior"}]
94         listReplyData = listReply_Message(number, options, body, footer, "sed3",messageId)
95         list.append(listReplyData)
96     else :
97         data = text_Message(number, "Lo siento, soy un prototipo y sigo aprendiendo, no entendí lo que me pediste")
98         list.append(data)
99     for item in list:
100         enviar_Mensaje_whatsapp(item)

```

**Nota:** Muestra el flujo básico de la conversación del chatbot.

3. Luego se tiene la función `text_Message()` que es la que le responde en forma de mensajes de texto simple al usuario.

**Figura 47**  
**Función text Mensaje.**  
**Visual Studio Code**

```

48 def text_Message(number, text):
49     data = json.dumps(
50         {
51             "messaging_product": "whatsapp",
52             "recipient_type": "individual",
53             "to": number,
54             "type": "text",
55             "text": {
56                 "body": text
57             }
58         }
59     )
60     return data

```



**Nota:** Se detalla la definición de la función y como se muestra en pantalla de usuario la respuesta del texto

**Figura 48**  
**Ejemplo de invocación de función.**

```

if "Hola" in text:
    conversacion = 1
    textMessage = text_Message(number, "¡Hola! 🙌 Bienvenido a ChatBot de Empagran
    footer = "Equipo Sistemas Empagran"
    list.append(textMessage)

```

**Nota:** Función implementada dentro del flujo de conversación del método administrar\_chatbot().

- Se implementa la función `buttonReply_Message()` la cual el prototipo utiliza para devolver respuestas tipo botones como se muestra en la Figura 49.

**Figura 49**  
**Función buttonReply\_Message().**

**Visual Studio Code**

**Chat Whatsapp**

```

101
102 def buttonReply_Message(number, options, body, footer, sedd,messageId):
103     buttons = []
104     for i, option in enumerate(options):
105         buttons.append(
106             {
107                 "type": "reply",
108                 "reply": {
109                     "id": sedd + "_btn_" + str(i+1),
110                     "title": option
111                 }
112             }
113         )
114
115     data = json.dumps(
116         {
117             "messaging_product": "whatsapp",
118             "recipient_type": "individual",
119             "to": number,
120             "type": "interactive",
121             "interactive": {
122                 "type": "button",
123                 "body": {
124                     "text": body
125                 },
126                 "footer": {
127                     "text": footer
128                 },
129                 "action": {
130                     "buttons": buttons
131                 }
132             }
133         }
134     )
135     return data

```



**Nota:** Se detalla la definición de la función y como se muestra en la pantalla del usuario la respuesta en tipo botones.

**Figura 50**  
Invocación del método `buttonReply_Message`.

```

76     elif validarCliente(text) != "" and conversacion == 1:
77         conversacion = 2
78         body = "¡Hola! " + Nombre_Usuario + " 🙌 Bienvenido a ChatBot de Empagran 🤖. ¿Cómo podemos ayudarte hoy? 🗣️"
79         footer = "Equipo Sistemas Empagran"
80         options = [{"🏭": "Produccion"}, {"🚚": "Exportacion"}]
81         replyButtonData = buttonReply_Message(number, options, body, footer, "sed1",messageId)
82         list.append(replyButtonData)

```

**Nota:** Función implementada dentro del flujo de conversación del método `administrar_chatbot()`.

- Se implementa la función `listReply_Message()` la cual el prototipo utiliza para devolver una lista predefinida de opciones como se muestra en la figura 51.

**Figura 51**  
Funcion `listReply_Message()`.

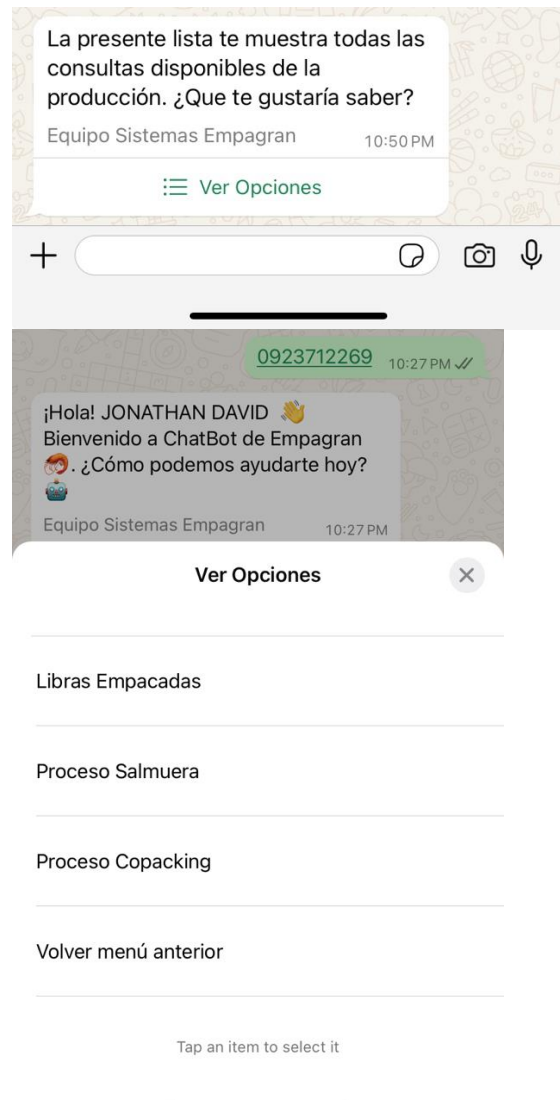
Visual Studio Code

Chat Whatsapp

```

7 def listReply_Message(number, options, body, footer, sedd,messageId):
8     rows = []
9     for i, option in enumerate(options):
10        rows.append(
11            {
12                "id": sedd + "_row_" + str(i+1),
13                "title": option,
14                "description": ""
15            }
16        )
17
18    data = json.dumps(
19        {
20            "messaging_product": "whatsapp",
21            "recipient_type": "individual",
22            "to": number,
23            "type": "interactive",
24            "interactive": {
25                "type": "list",
26                "body": {
27                    "text": body
28                },
29                "footer": {
30                    "text": footer
31                },
32                "action": {
33                    "button": "Ver Opciones",
34                    "sections": [
35                        {
36                            "title": "Secciones",
37                            "rows": rows
38                        }
39                    ]
40                }
41            }
42        }
43    )
44    return data

```



**Nota:** Se detalla la definición de la función y como se muestra en la pantalla del usuario la respuesta en tipo lista de opciones.

**Figura 52**  
**Invocación del método listReply\_Message().**

```

83 elif "Produccion" in text and conversacion == 2:
84     conversacion = 3
85     body = "La presente lista te muestra todas las consultas disponibles de la producción. ¿Que te gustaría saber?"
86     footer = "Equipo Sistemas Empagran"
87     options = ["Libras Empacadas", "Proceso Salmuera", "Proceso Copacking", "Volver menú anterior"]
88     listReplyData = listReply_Message(number, options, body, footer, "sed2",messageId)
89     list.append(listReplyData)

```

**Nota:** Función implementada dentro del flujo de conversación del método administrar chatbot().

- La interacción entre la Api de Whatsapp y el prototipo se hace por medio de la función `enviar_Mensaje_whatsapp()` la cual recibe como parámetros las estructuras JSON devueltas por las funciones `listReply_Message()`, `buttonReply_Message()`, `text_Message()` que son las estructuras definidas por la api para el

intercambio de información, el desarrollo del método se explica en la Figura 53.

**Figura 53**  
**Función enviar\_Mensaje\_whatsapp()**

```
33 def enviar_Mensaje_whatsapp(data):
34     try:
35         whatsapp_token = sett.whatsapp_token
36         whatsapp_url = sett.whatsapp_url
37         headers = {'Content-Type': 'application/json',
38                   'Authorization': 'Bearer ' + whatsapp_token}
39         print("se envia ", data)
40         response = requests.post(whatsapp_url,
41                                  headers=headers,
42                                  data=data)
43         print(response)
44         if response.status_code == 200:
45             return 'mensaje enviado', 200
46         else:
47             print (response.status_code)
48             return 'error al enviar mensaje', response.status_code
49     except Exception as e:
50         return e,403
```

**Nota:** Los parámetros whatsapp\_token y whatsapp\_url son los definidos previamente en el archivo sett.py.

7. La interacción del prototipo con las consultas de producción se realiza por medio de la API de producción implementando el método ConsultaApiProduccion() el cual recibe como parámetros la ruta de la Api y los parámetros de consulta. A continuación, se muestra el desarrollo de esta y sus resultados.

**Figura 54**  
**Función ConsultaApiProduccion.**  
**Visual Studio Code**

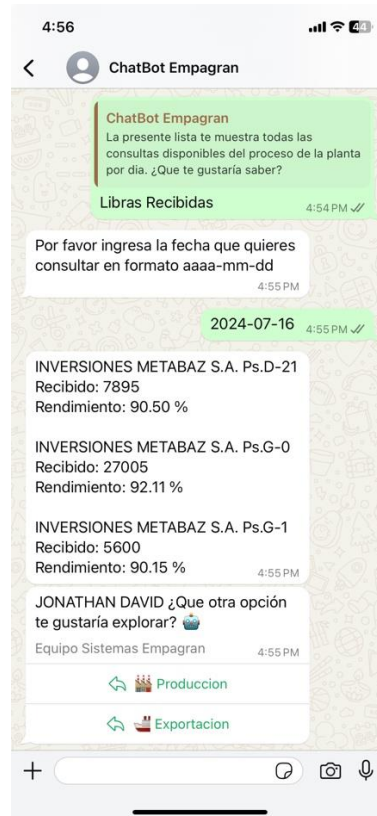
**Chat Whatsapp**



```

def ConsultaApiProduccion(api, fecha, conv):
    url = sett.empagran_url + "/" + api
    global Resultado
    print(url)
    if conv == "A":
        url = url + "?fecha=" + fecha
        response = requests.put(url)
        if response.status_code == 200:
            # Parsear la respuesta JSON
            data = response.json()
            lines = [{"item": TipoProceso, capitalize(): (item["LIBRAS"])} for item in data]
            # Calcular el total
            total_libras = sum(item["LIBRAS"] for item in data)
            lines.append(("Total Proceso del dia: (total_libras:3f)" + " Libras")
            Resultado = "\n".join(lines)
            return Resultado
        else:
            print("Error en la solicitud:", response.status_code, response.text)
    elif conv == "B":
        url = url + "?fecha=" + fecha
        response = requests.put(url)
        if response.status_code == 200:
            # Parsear la respuesta JSON
            data = response.json()
            result = ""
            for item in data:
                result += f"({item['Proveedor']})\nRecibido: {item['Recibido']} Libras\nRendimiento: {item['Rendimiento']}\n\n"
            Resultado = result
            return Resultado
        else:
            print("Error en la solicitud:", response.status_code, response.text)

```



**Nota:** Se detalla la definición de la función y como se muestra en la pantalla del usuario la respuesta en base a los parámetros ingresados.

**Figura 55**  
**Invocación de la función ConsultaApiProduccion.**

```

104 elif ("Libras Procesadas" in text or "Libras Recibidas" in text or "Empaque por Destino" in text) and conversacion == 3:
105     conversacion = 4
106     if "Libras Procesadas" in text:
107         Tipo_Variable = "A"
108     elif "Libras Recibidas" in text:
109         Tipo_Variable = "B"
110     else:
111         Tipo_Variable = "C"
112     textMessage = text_Message(number, "Por favor ingresa la fecha que quieres consultar en formato aaaa-mm-dd")
113     footer = "Equipo Sistemas Empagran"
114     list.append(textMessage)
115     elif validar_fecha(text) and conversacion == 4:
116         if Tipo_Variable == "A":
117             textMessage = text_Message(number, ConsultaApiProduccion("LibrasDiariasProcesadas", text, Tipo_Variable))
118         elif Tipo_Variable == "B":
119             textMessage = text_Message(number, ConsultaApiProduccion("LibrasDiariasRecibidas", text, Tipo_Variable))
120         else:
121             textMessage = text_Message(number, ConsultaApiProduccion("EmpaqueDestino", text, Tipo_Variable))
122     conversacion = 5
123     footer = "Equipo Sistemas Empagran"
124     list.append(textMessage)

```

**Nota:** Función implementada dentro del flujo de conversación del método administrar\_chatbot().

## Autenticación de usuarios.

La función validarCliente() implementa un mecanismo básico de autenticación de usuarios mediante la verificación de la cédula y el número de teléfono en un servicio web externo. Al recibir estos parámetros, la función construye una URL con la cual realiza una solicitud HTTP GET para validar la existencia y el permiso de acceso al chatbot en la base de datos SARH. Si la solicitud es exitosa, se obtiene el

nombre completo del usuario y se extraen los dos últimos componentes de su nombre, los cuales se retornan como resultado.

**Figura 56**  
**Autenticación de los usuarios en el prototipo.**  
**Visual Studio Code**

```
2 def validarCliente(text, telefono):
3
4     url = sett.empagran_url + "/PermisoUsuarios?cedula=" + text + "&numero=" + telefono
5     global Nombre_Usuario
6     print(url)
7     # Realizar la solicitud GET
8     response = requests.get(url)
9     if response.status_code == 200:
10        # Parsear la respuesta JSON
11        data = response.json()
12        nombres_completos = data.get("NOMBRES_COMPLETOS", "No encontrado")
13        print("nombres_completos:", nombres_completos)
14        partes = nombres_completos.split()
15        print("partes:", partes)
16        nombres = '.join(partes[-2:]) if len(partes) >= 2 else "No encontrado"
17        print("nombres:", nombres)
18        Nombre_Usuario = nombres
19        return Nombre_Usuario
20    else:
21        print('Error en la solicitud:', response.status_code, response.text)
22        return ""
```

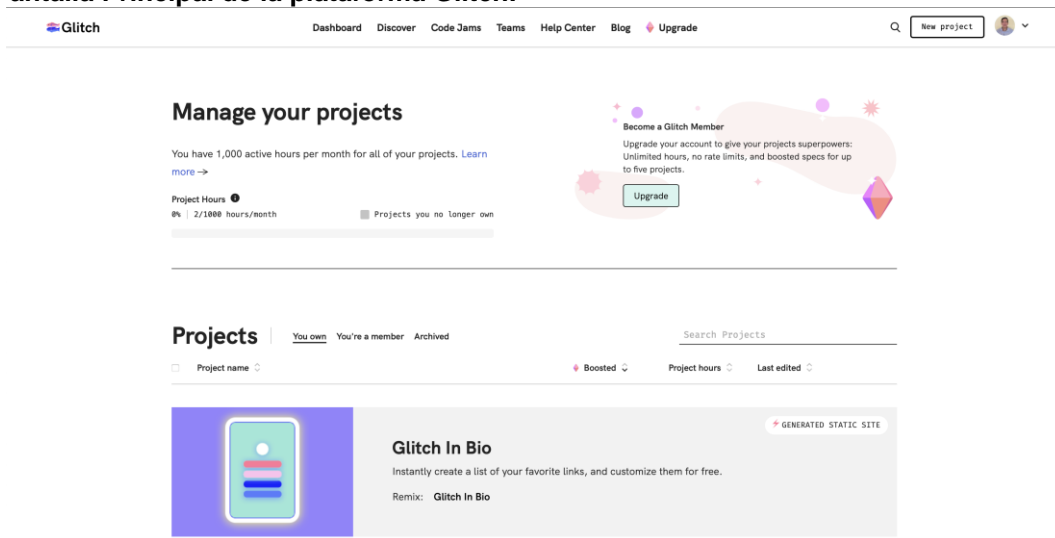


**Nota:** Definición de la función y funcionamiento dentro del chatbot

### Implementación del prototipo.

Para la implementación del prototipo se utiliza la plataforma Glitch, la cual opera en línea y permite crear aplicaciones web de manera gratuita y con plantillas predefinidas que integran el entorno de python y flask, que son las herramientas con las cuales trabaja nuestro prototipo. Para el registro se utiliza una cuenta de Google y automáticamente nos muestra la pantalla de inicio como se indica en la Figura 57.

**Figura 57**  
**Pantalla Principal de la plataforma Glitch.**

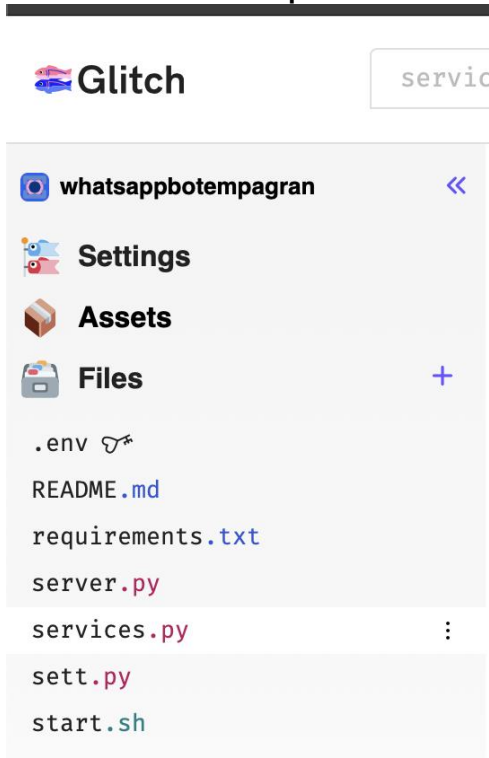


**Nota:** Nos muestra la pantalla inicial luego de iniciar sesión con Google

Para crear el proyecto se realiza los pasos detallados en el anexo 2.

Una vez creado el proyecto en la plataforma, se crean los archivos de la misma forma que tenemos en el Visual Studio Code. La estructura debe quedar como se indica en la Figura 58.

**Figura 58**  
**Estructura de archivos plataforma.**



**Nota:** El archivo server.py contiene la información del archivo app.py del proyecto local.

A la estructura de archivos de la plataforma se agrega el archivo `requirements.txt` el cual contiene los requerimientos de instalación de librerías y componentes para el funcionamiento del prototipo. A continuación, se muestra el contenido del archivo en la Figura 59.

**Figura 59**  
Contenido archivo `requirements.txt`.

```

1 blinker==1.6.2
2 certifi==2023.5.7
3 charset-normalizer==3.1.0
4 click==8.1.3
5 Flask
6 idna==3.4
7 itsdangerous==2.1.2
8 Jinja2==3.1.2
9 MarkupSafe==2.1.3
10 requests==2.31.0
11 urllib3==1.26.6
12 Werkzeug

```

**Nota:** Las versiones que muestra cada línea es la recomendada por la plataforma.

Una vez indicado los requisitos, la plataforma automáticamente empieza a cargar cada uno de ellos para luego de eso levantar el servicio de nuestra aplicación como se muestra en la Figura 60.

**Figura 60**  
Log de carga de requisitos.

```

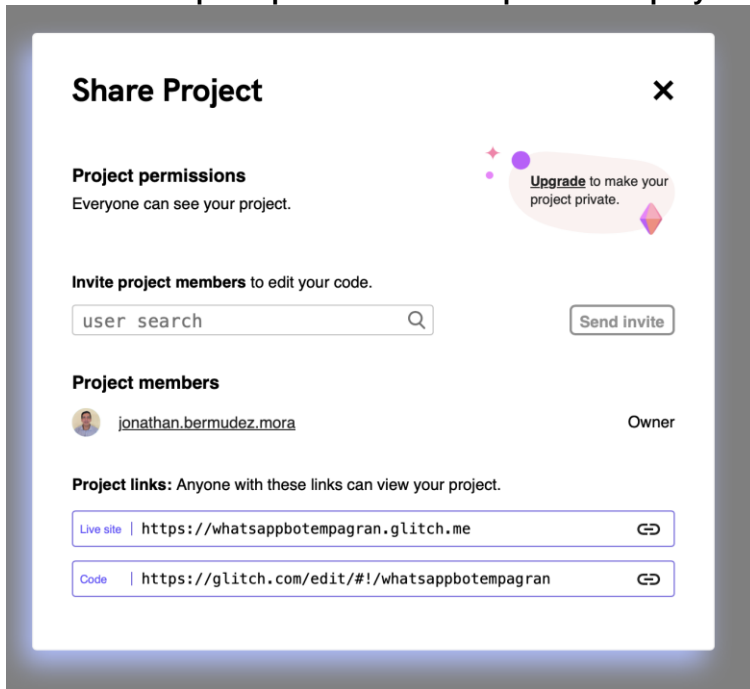
Press CTRL-C to quit
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in ./local/lib/python3.7/site-packages (24.0)
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: blinker=1.6.2 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (1.6.2)
Requirement already satisfied: certifi=2023.5.7 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 2)) (2023.5.7)
Requirement already satisfied: charset-normalizer=3.1.0 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 3)) (3.1.0)
Requirement already satisfied: click=8.1.3 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (8.1.3)
Requirement already satisfied: Flask in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 5)) (2.0.0)
Requirement already satisfied: idna=3.4 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 6)) (3.4)
Requirement already satisfied: itsdangerous=2.1.2 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 7)) (2.1.2)
Requirement already satisfied: Jinja2=3.1.2 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 8)) (3.1.2)
Requirement already satisfied: MarkupSafe=2.1.3 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 9)) (2.1.3)
Requirement already satisfied: requests=2.31.0 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 10)) (2.31.0)
Requirement already satisfied: Werkzeug in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 12)) (2.2.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from click=8.1.3->-r requirements.txt (line 4)) (4.8.1)
Requirement already satisfied: zipp>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->click=8.1.3->-r requirements.txt (line 4)) (3.7.4.3)
* Tip: There are *.env or *.flaskenv files present. Do "pip install python-dotenv" to use them.
* Serving Flask app "server" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL-C to quit

```

**Nota:** Si los requisitos cargan con éxito mostrará el mensaje de Running.

Luego de esto se obtiene la URL de nuestro proyecto mediante el botón Share que se encuentra en la parte superior derecha, la cual muestra la ventana de compartir para poder copiar la dirección del prototipo tal como se expone en la Figura 61.

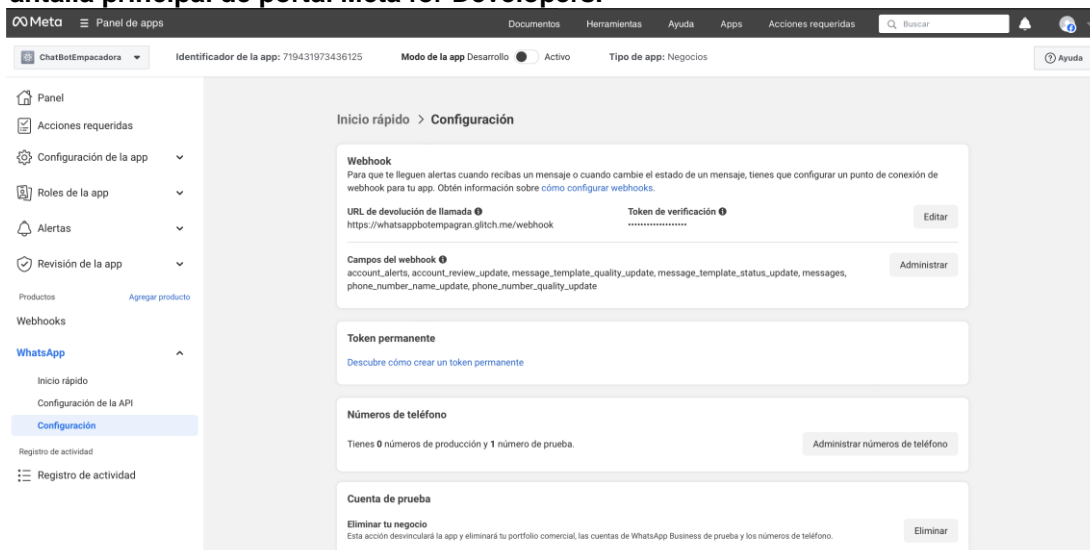
**Figura 61**  
Ventana de compartir para obtener URL pública del proyecto.



**Nota:** Importante anotar la dirección que nos muestra el campo Live site que nos servirá para comunicarnos con la Api de WhatsApp de los servicios de Meta.

Para finalizar, se inicia sesión en el portal de Meta for Developers donde está habilitado el portal de administración de las Apis de WhatsApp como se muestra en la Figura 62.

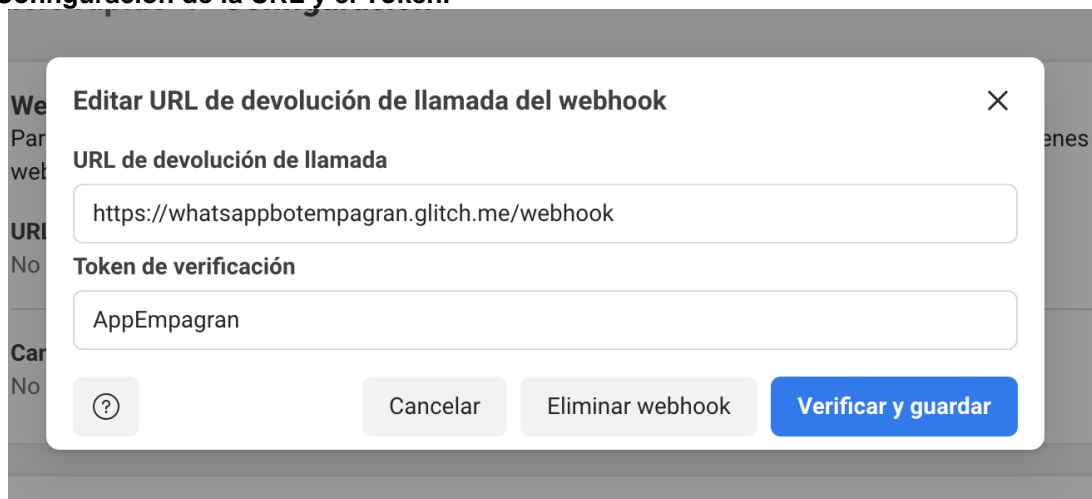
**Figura 62**  
Pantalla principal de portal Meta for Developers.



**Nota:** Esta vez se trabaja sobre la opción configuración para poder incluir la URL del prototipo.

Se ubica la opción Webhook donde se ingresa la URL previamente obtenida y el token de validación que Meta proporciona tal como se muestra en la Figura 63.

**Figura 63**  
**Configuración de la URL y el Token.**



**Nota:** El token de verificación es único y solo funciona para la aplicación previamente configurada.

Con esto se da por finalizada la puesta en marcha de la aplicación y se establece la comunicación entre las Apis de Whatsapp y el prototipo desarrollado.

### **Nivel de aceptación.**

Finalizado el prototipo y puesto en operación se procedió a aplicar a los usuarios una encuesta para determinar el nivel de aceptación, cuyos resultados se encuentran tabulados en el Anexo 3 revelando un alto nivel de aceptación del chatbot como se muestra en la Tabla 4:

**Tabla 4**  
**Aceptación del prototipo.**

#	Pregunta	Porcentaje
1	¿Cómo evalúa la facilidad de uso del chatbot de WhatsApp para resolver sus consultas sobre la producción en la empacadora?	100%
2	¿Qué tan satisfecho está con la rapidez de las respuestas proporcionadas por el chatbot?	100%
3	¿El chatbot ha sido capaz de resolver sus consultas de manera precisa y eficiente?	100%
4	¿Cómo calificaría la integración del chatbot con los sistemas existentes de la empresa? (por ejemplo, acceso a datos en tiempo real, precisión de la información, etc.)	100%
5	¿Qué tan intuitivo le parece el uso del chatbot en términos de navegación y acceso a opciones?	100%

Se destaca una alta satisfacción por parte de los usuarios; la mayoría de los encuestados evaluaron positivamente la facilidad de uso del chatbot, resaltando su capacidad para resolver consultas relacionadas con la producción de manera útil y clara también se mostraron satisfechos con la rapidez de las respuestas proporcionadas por el chatbot, lo que indica una alta efectividad en la interacción con el sistema. Se destaca también la integración del chatbot con los sistemas existentes de la empresa fue bien valorada, especialmente en términos de acceso a datos en tiempo real y precisión de la información; esto demuestra que el chatbot ha cumplido con los objetivos de proporcionar información oportuna y precisa; finalmente, el uso intuitivo del chatbot, en cuanto a navegación y acceso a opciones, fue otro aspecto destacado por los usuarios, lo que sugiere una experiencia de usuario positiva y fluida; los resultados reflejan un alto nivel de aceptación del chatbot, consolidando su valor como herramienta de apoyo en la gestión de la producción.

### **Presupuesto de implementación.**

Para llevar a cabo una implementación completa del prototipo, es necesario considerar los siguientes costos:

**Tabla 5**  
**Costos a considerar para una implementación completa.**

<b>Ítem</b>	<b>Unidad</b>	<b>Cantidad</b>	<b>Valor Unitario</b>	<b>Total</b>
Costo por horas de trabajo en el desarrollo del ChatBot	Hora / USD	120 horas	\$ 7.50	\$900.00
Costo WhatsApp Business API anual	Mensajes anuales / USD	108.000 conversaciones adicionales	\$ 0.0088	\$950.40
Suscripción Glitch Pro	Suscripción Anual	1	\$ 96.00	\$ 96.00
Servidor DELL POWEREDGE R450 INTEL XEON 8 CORE	Servidor	1	\$ 5,822.88	\$ 5,822.88
Licencia Windows Server Standar 2016	Licencia	1	\$ 398.99	\$ 398.99
Licencia Visual Studio Professional 2022	Licencia Anual	1	\$ 45.00	\$ 540.00
<b>Total</b>				<b>\$ 8,708.27</b>

**Nota:** La selección de las licencias del servidor y el equipo se realizó en base a la estructura actual de los equipos y licencias existentes en la empresa auspiciante.

Dado que la empresa cuenta con parte de la infraestructura requerida, como servidores, PC, herramientas de desarrollo y licencias, estos elementos no se

consideran dentro del costo de implementación del chatbot. A continuación, se detallan los costos incurridos por la empaadora:

**Tabla 6**  
**Costos de implementación en la empresa.**

<b>Ítem</b>	<b>Unidad</b>	<b>Cantidad</b>	<b>Valor Unitario</b>	<b>Total</b>
Costo por horas de trabajo en el desarrollo del ChatBot	Hora / USD	120 horas	\$ 7.50	\$900.00
Costo WhatsApp Business API anual	Mensajes anuales / USD	108.000 conversaciones adicionales	\$ 0.0088	\$950.40
Suscripción Glitch Pro	Suscripción Anual	1	\$ 96	\$ 96.00
<b>Total</b>				<b>\$ 1,946.40</b>



## CONCLUSIONES

La implementación del prototipo de chatbot en WhatsApp utilizando las herramientas de desarrollo definidas por la empresa resultó coherente con sus estándares de desarrollo; esto asegura que el chatbot cumple con las expectativas técnicas y con la integración adecuada de las prácticas establecidas por la empresa; la utilización de estas herramientas resultó ser efectivo para el desarrollo del prototipo y posterior operabilidad del sistema dentro del entorno del modelo de negocio.

La creación de la interfaz de usuario, que incluye listas de opciones múltiples y botones con opciones principales, facilitó la integración del banco de preguntas y una comprensión más precisa de las consultas relacionadas con la producción en la empacadora; esta interfaz es intuitiva y está completamente integrada con WhatsApp, siendo esencial para la efectividad del chatbot.

El proceso de abstracción, transformación y depuración de datos en tiempo real mediante procedimientos almacenados en bases de datos previamente optimizadas subraya la importancia de contar con datos de alta calidad y bien gestionados para la eficacia de sistemas de información automatizados.

La integración exitosa del chatbot mediante APIs con los sistemas de información de la empresa aseguró la obtención y presentación precisa de datos en tiempo real relacionados con el procesamiento de camarones, demostrando su versatilidad y aplicabilidad en el modelo de negocio.

Por lo indicado, se da cumplimiento al alcance, objetivos generales y específicos planteados en el presente proyecto, existiendo correspondencia entre los objetivos y las conclusiones, ya que el resultado del prototipo cumple más allá de las expectativas inicialmente definidas.

La pregunta de investigación formulada: "¿El prototipo ChatBot de WhatsApp tendrá la capacidad de responder las preguntas de la producción de la empacadora de camarones de manera oportuna?" ha sido respondida positivamente. El prototipo de ChatBot de WhatsApp desarrollado en este proyecto ha demostrado tener la capacidad de responder las preguntas de la producción de la empacadora de camarones de manera oportuna y eficiente mediante la integración con las APIs

internas de la empresa; esto permite que el chatbot acceda directamente a los sistemas de producción y bases de datos, obteniendo y presentando datos en tiempo real de manera eficiente.

La retroalimentación de los usuarios clave indica un alto nivel de satisfacción con la rapidez y precisión de las respuestas obtenidas a través del chatbot; la organización de datos ha sido fundamental para garantizar que el prototipo pueda responder de manera oportuna, así como la disponibilidad 24/7 permite a los gerentes y jefes acceder a información crítica en cualquier momento, mejorando significativamente la toma de decisiones y la eficiencia operativa.

Con estas premisas, se ha dado cumplimiento y respuesta a la pregunta de investigación planteada.

## RECOMENDACIONES

Expandir la funcionalidad del chatbot para incluir otros ítems de consultas y respuestas, basadas en interacciones reales con los usuarios; esto permitirá que el sistema evolucione y se adapte continuamente a las necesidades cambiantes de la empresa y sus empleados.

La aceptación del prototipo sugiere que una implementación a mayor escala podría ser beneficiosa para otros departamentos dentro de la empresa; esta expansión debe considerar la inclusión de nuevos módulos que puedan atender diferentes aspectos operativos y mejora continua en eficiencia general de la empresa.

Considerando que para el desarrollo del prototipo se utilizaron versiones gratuitas de la plataforma Glitch y el modo de pruebas de las APIs de Meta for Developers se recomienda utilizar versiones de pago de ambas plataformas. Utilizar estas versiones garantiza la ejecución ilimitada en el servidor de la plataforma Glitch y envío de mensajes desde cualquier número en la plataforma WhatsApp; lo cual con estas mejoras se asegura una experiencia fluida en la utilización del prototipo y uso ampliado por parte de los usuarios en entornos productivos de alta transaccionalidad.

## REFERENCIAS

- Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. En I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), *Artificial Intelligence Applications and Innovations* (pp. 373–383). Springer International Publishing. [https://doi.org/10.1007/978-3-030-49186-4\\_31](https://doi.org/10.1007/978-3-030-49186-4_31)
- Aguirre, J., & Aguirre, S. (2021). *Metodologías para el desarrollo de proyectos*. <http://repository.unicatolica.edu.co/handle/20.500.12237/2037>
- Al Sweigart. (2020). *Automate The Boring Stuff With Python: Practical Programming For Total Beginners* (2nd Edition). No Starch Press, Inc.
- Amazon Web Services, Inc. (2023). *¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/api/>
- anandmeg. (2023, octubre 31). *¿Qué es el IDE de Visual Studio?* <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- AWS, Inc. (2023). *¿Qué es un IDE? - Explicación de los entornos de desarrollo integrado - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/ide/>
- Berners-Lee, T., Fischetti, M., & Fernández, M. R. (2000). *Tejiendo la red: El inventor del world wide web nos descubre su origen*. Siglo XXI de España. <https://books.google.com.ec/books?id=QRe-iutQQmQC>
- Connolly, T. M., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management* (Sixth edition). Pearson Education Limited.
- Corao, F. P., & Vanegas, M. P. (2023). *Administración de servicios web. Anatomía del Internet*. Marcombo. <https://books.google.com.ec/books?id=dtavEAAAQBAJ>
- Coronel, C., Morris, S., Rob, P., & Muñoz, J. H. R. (2011). *Base de Datos: Diseño,*

*Implementación Y Administración.* CENGAGE Learning.  
[https://books.google.com.ec/books?id=KINC0Gc\\_RREC](https://books.google.com.ec/books?id=KINC0Gc_RREC)

Date, C. J. (2001). *Introducción a los sistemas de bases de datos* (SÉPTIMA EDICIÓN). Pearson Educación.

Delgado, L. & Díaz, L. (2021). Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informáticas*, 15, 37–51.  
<https://www.redalyc.org/articulo.oa?id=378366538003>

De León, H. (2023). *Aprender a programar con C#*-. <https://hdeleon.net/libro-aprender-a-programar-con-c-hector-de-leon/>

Desafíos de aptitudes de IA. (2024). *Documentación de IIS* [Blog].  
<https://learn.microsoft.com/es-es/iis/>

Gamarra, F. (2024). *Visual Studio Code*. Ediciones de la U.  
<https://books.google.com.ec/books?id=P9oJEQAAQBAJ>

Goodwin, G. E. (2023, septiembre 6). *What is WhatsApp? A guide to navigating the free Meta-owned communication platform*. Business Insider.  
<https://www.businessinsider.com/guides/tech/what-is-whatsapp-guide>

Guido Van Rossum & Fred L. Drake. (2011). *The Python Language Reference Manual: For Python Version 3.2* (revisada). Network Theory Limited.

Hemmendinger, D. (2024, marzo 24). *Computer programming language | Types & Examples | Britannica*. Computer Programming Language.  
<https://www.britannica.com/technology/computer-programming-language>

*Historia de Visual Studio: De un simple IDE a una potente plataforma de desarrollo - Administración de Sistemas.* (2024, marzo 26).  
<https://administraciondesistemas.com/visual-studio/>

Ian Bicking. (2018). *Development—Virtualenv 16.7.9 documentation* [Blog].  
<https://virtualenv.pypa.io/en/16.7.9/development.html#status-and-license>

IBM. (2024, mayo 23). *¿Qué es un chatbot? | IBM*. <https://www.ibm.com/es->

es/topics/chatbots

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (Second Edition). John Wiley & Sons, Inc.

Krause, J. (2021). *Mastering Windows Server 2019: The complete guide for system administrators to install, manage, and deploy new capabilities with Windows Server 2019*. Packt Publishing.  
<https://books.google.com.ec/books?id=et06EAAAQBAJ>

Marín, R. (2024, abril 8). Los gestores de bases de datos (SGBD) más usados. *Canal Informática y TICS*. <https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

Martelli, A., & Ascher, D. (2002). *Python Cookbook*. O'Reilly.  
<https://books.google.com.ec/books?id=yhfdQgq8JF4C>

McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (Second). O'Reilly Media, Incorporated.  
<https://books.google.com.ec/books?id=2BYfvgAACAAJ>

Microsoft Corporation. (2024a). *C# | Lenguaje de programación moderno y de código abierto para .NET*. Microsoft. <https://dotnet.microsoft.com/es-es/languages/csharp>

Microsoft Corporation. (2024b). *Internet Information Services (IIS)—Microsoft Lifecycle*. <https://learn.microsoft.com/es-es/lifecycle/products/internet-information-services-iis>

Grinberg, M. (2014). *Flask Web Development: Developing Web Applications with Python* (First Edition). O'Reilly Media, Inc.  
[https://coddyschool.com/upload/Flask\\_Web\\_Development\\_Developing.pdf](https://coddyschool.com/upload/Flask_Web_Development_Developing.pdf)

Nemesis, P. C. (2023, octubre 10). Understanding Webhooks in Python. *Technology Hits*. <https://medium.com/technology-hits/understanding-webhooks-in-python-1fcda2e87572>

Powers, L., & Snell, M. (2015). *Microsoft Visual Studio 2015 Unleashed* (Third).

- Pearson Education. <https://books.google.com.ec/books?id=EehVCgAAQBAJ>
- Pressman, R. S., & Maxim, B. R. (2021). *Ingeniería del software: Un enfoque práctico* (9a ed.). McGraw Hill Interamericana Editores. <https://books.google.com.ec/books?id=sQZ6zgEACAAJ>
- Python Software Foundation. (2023). *Pip documentation v24.0*. <https://pip.pypa.io/en/stable/>
- Python Software Foundation. (2024a). *venv—Creation of virtual environments*. Python Documentation. <https://docs.python.org/3/library/venv.html>
- Python Software Foundation. (2024b). *What is Python? Executive Summary*. Python.Org. <https://www.python.org/doc/essays/blurb/>
- Red Hat, Inc. (2023, enero 20). *¿Qué es una API y cómo funciona?* <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- Reitz, K., & Schlusser, T. (2016). *The Hitchhiker's Guide to Python: Best Practices for Development*. O'Reilly Media, Incorporated. <https://books.google.com.ec/books?id=2ZggjwEACAAJ>
- Revista Acuicultura. (2024, mayo 2). *Abril 2024, 158, 88*. [https://issuu.com/revista-cna/docs/edicion\\_158](https://issuu.com/revista-cna/docs/edicion_158)
- Sánchez-Ticona, R. J., Alcántara-Moreno, O. R., Torres-Villanueva, M., Gómez-Avila, J. A., Rodríguez-Huamán, R. E., & Santos-Fernández, J. P. (2023). *Una Revisión Sistemática del Desarrollo de Chatbots y Asistentes Virtuales en Instituciones de Educación Superior*. 113–120. <https://doi.org/10.54808/CISCI2023.01.113>
- Solutions, A. (2020, enero 22). *The Value of AI Chatbots for Business*. Medium. <https://medium.com/@artificialsolutions/the-value-of-ai-chatbots-for-business-c34ccc222884>
- WhatsApp Business. (2024). *WhatsApp Business | Transforma tu empresa*. WhatsApp Business. <https://business.whatsapp.com>

## GLOSARIO

- PIB:** Producto Interno Bruto, medida del valor total de los bienes y servicios producidos en un país durante un período específico.
- ETL:** Extract, Transform, Load por sus siglas en inglés es el proceso que implica la extracción de datos de diversas fuentes, su transformación en base a la necesidad del trabajo y su carga en un repositorio para manejo de datos.
- SIG:** Sistema de Información Geográfica, sistema que nos permite capturar, almacenar, analizar y gestionar datos geográficos y espaciales.
- GPRS:** General Packet Radio Service por sus siglas en inglés es la tecnología de comunicación de datos móviles que permite la transmisión de datos en paquetes sobre redes GSM.
- 3G:** tecnología de tercera generación, estándar de comunicación móvil que proporciona acceso a Internet y servicios multimedia.
- LTE:** Long Term Evolution por sus siglas en inglés es un estándar para la comunicación de alta velocidad para teléfonos móviles.
- 4G:** Tecnología de cuarta generación estándar de comunicación móvil que proporciona acceso a internet



y servicios multimedia con mayor velocidad y capacidad de datos que las tecnologías anteriores.

**GPL:** Licencia de software libre que permite a los usuarios copiar, modificar y distribuir software libremente.

**SOAP:** Simple Object Access Protocol por sus siglas en inglés, protocolo para el intercambio de información estructurada en la implementación por medio de servicios web en internet.

**JSON:** JavaScript Object Notation por sus siglas en inglés es un formato ligero de intercambio de datos que es fácil de leer y escribir para desarrolladores y fácil de generar e interpretar para sistemas y aplicaciones.

**HTTP:** HyperText Transfer Protocol por sus siglas en inglés es un protocolo utilizado para la transmisión de información en la World Wide Web.

**DBMS:** Data Base Management Systems por sus siglas en inglés es un sistema que permite la administración de datos en una base de datos por medio de interfaces graficas o comandos.

**OLAP:** Online Analytical Processing por sus siglas en inglés es una tecnológica que permite realizar análisis y consultas de datos multidimensionales.

**OLTP:** Online Transaction Processing por sus siglas en inglés es un sistema de gestión de bases de datos que

optimiza el procesamiento de transacciones en tiempo real.

**BSD:** Berkeley Software Distribution por sus siglas en inglés pertenece a la familia de sistemas operativos Unix derivados de software distribuido por la Universidad de California, Berkeley.

**DDL:** Data Definition Language por sus siglas en inglés es el lenguaje que utilizan las bases de datos para realizar definiciones y modificaciones de tablas y estructuras y objetos.

**DML:** Data Manipulation Language por sus siglas en inglés es el lenguaje que utilizan las bases de datos para poder manipular los datos permitiendo la inserción, actualización y eliminación de estos.

**IDE:** Integrated Development Environment por sus siglas en inglés es el ambiente de desarrollo de software donde se encuentran herramienta como el editor de código, depurador y compilador lo cuales son necesarios para construir aplicaciones.

**URL:** Uniform Resource Locator por sus siglas en inglés es una dirección la cual permite ingresar a un recurso único en la web.

**Token** Cadena de caracteres que representan datos o claves que sirven para autenticar o autorizar dentro de un proceso.

# ANEXOS

## Anexo 1. Levantamiento de información para el desarrollo del prototipo



PROCESO\_CHATBO  
T.docx

## Anexo 2. Manual de usuario para crear una aplicación en Glitch

### Prototipo de Chatbot en WhatsApp.

Jonathan David Bermúdez Mora

Julio 2024

### Índice

1. Introducción
2. Instalación
3. Creación de la Aplicación
4. Despliegue
5. Ejecución

### Introducción

#### Propósito del manual

El propósito del manual es enlistar los pasos seguidos para publicar el desarrollo del prototipo del chatbot en esta plataforma para poder tener comunicación con los clientes de la aplicación WhatsApp y también con las Apis de Meta.

#### Audiencia objetivo.

Este trabajo es parte del trabajo de integración curricular Anexo 2.

#### Requisitos previos

No requiere ningún requisito ya que no es una instalación on premise.

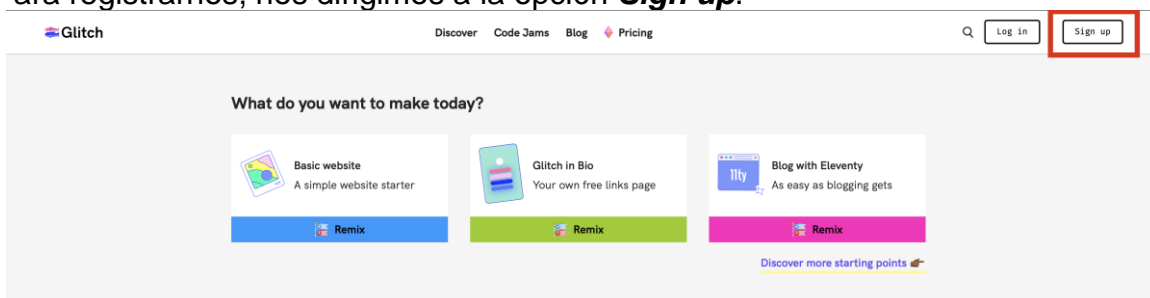
#### Instalación

Crear una cuenta en Glitch

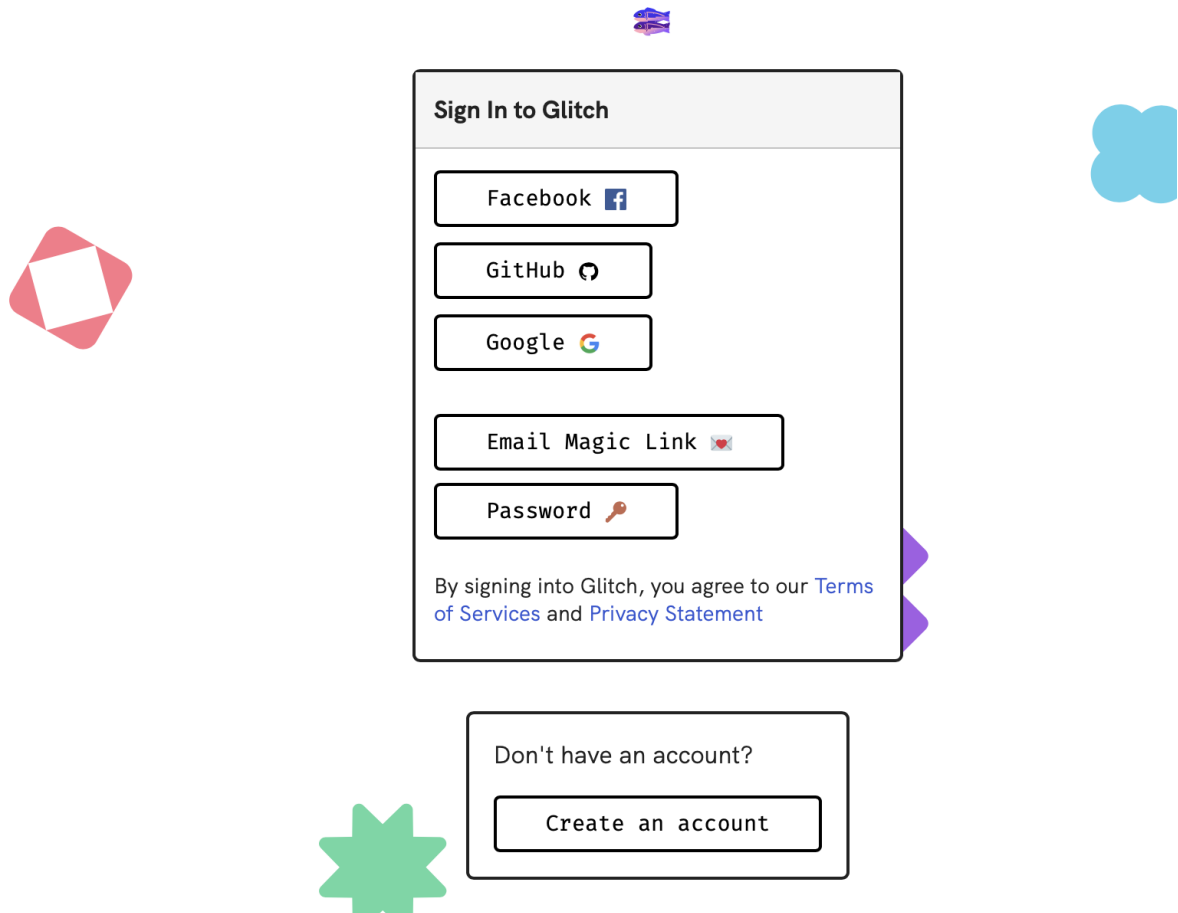
Para empezar, navegamos a la siguiente dirección:

<https://glitch.com>

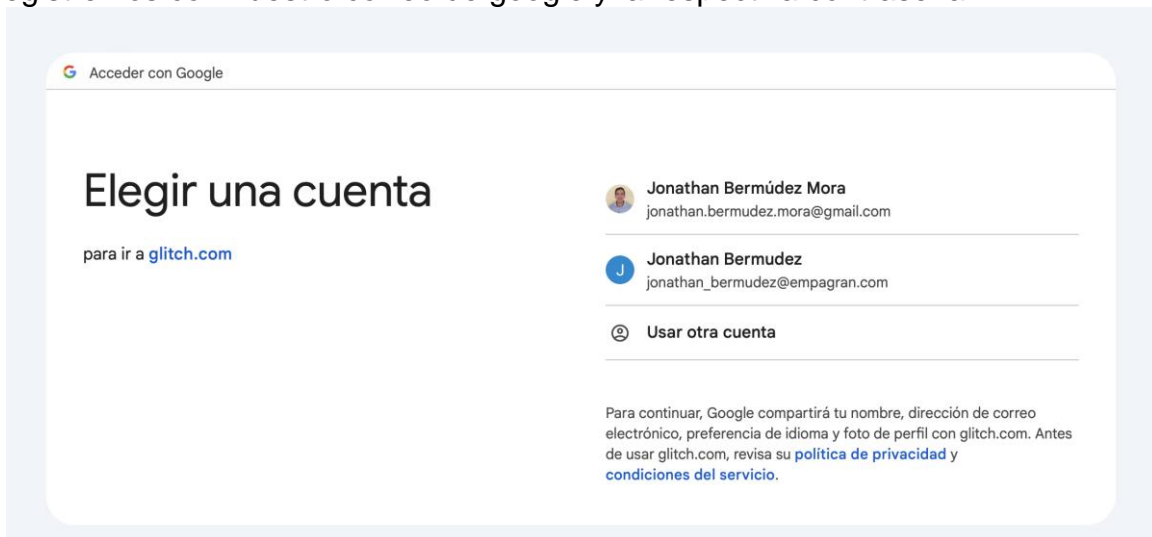
Para registrarnos, nos dirigimos a la opción **Sign up**.



Luego de esto, nos dirige a la página que nos da varias opciones de registro.



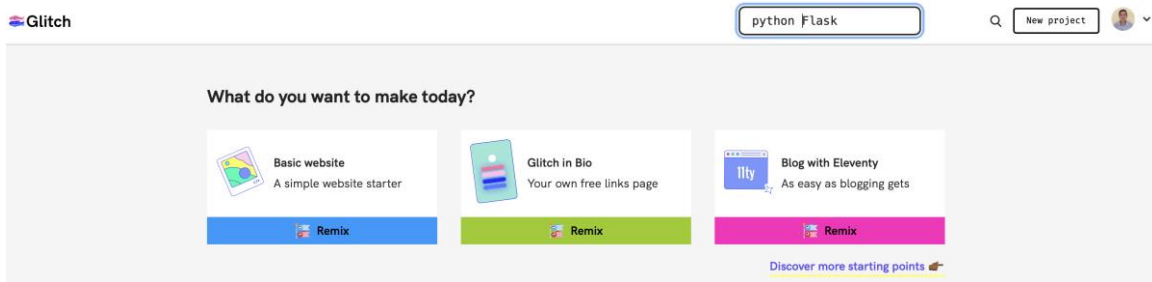
Para realizar el proceso de forma ágil utilizamos nuestra cuenta de google para poder autenticarnos dentro de la plataforma, para esto nos pedirá que nos registremos con nuestro correo de google y la respectiva contraseña.



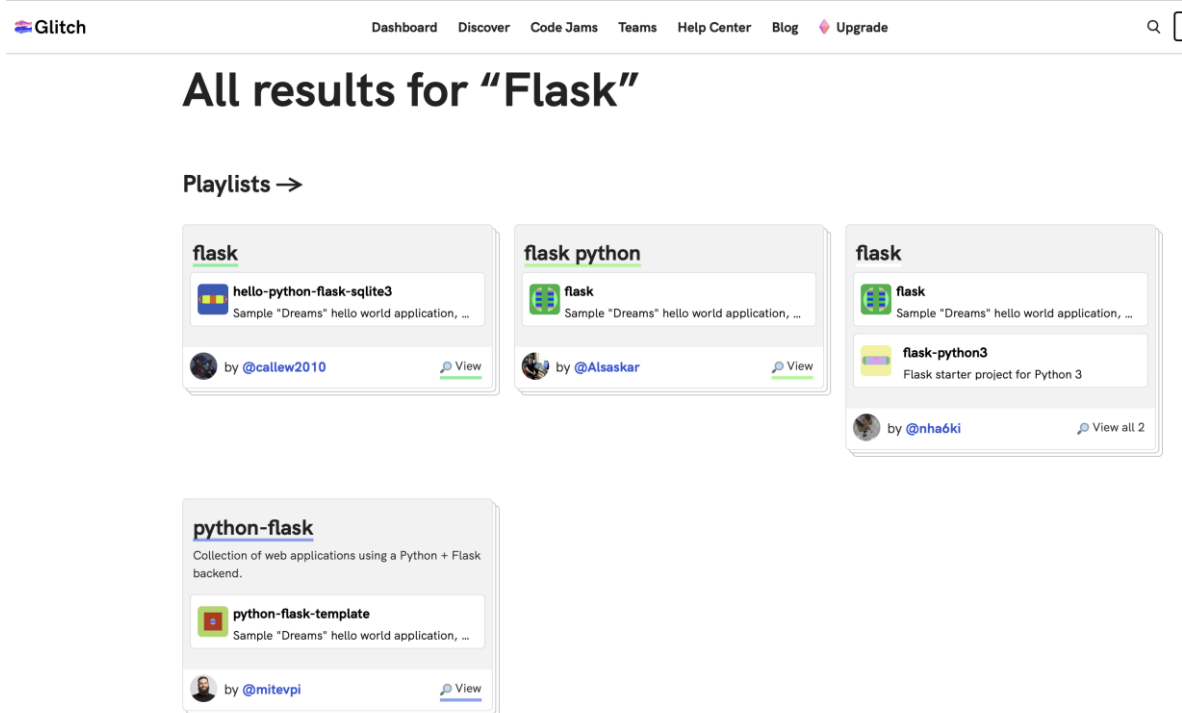
Una vez validado el usuario y contraseña y aceptando los términos y condiciones nos permitirá ingresar al portal de administración de proyectos.

Creación de la Aplicación  
Paso 1: Crear un nuevo proyecto

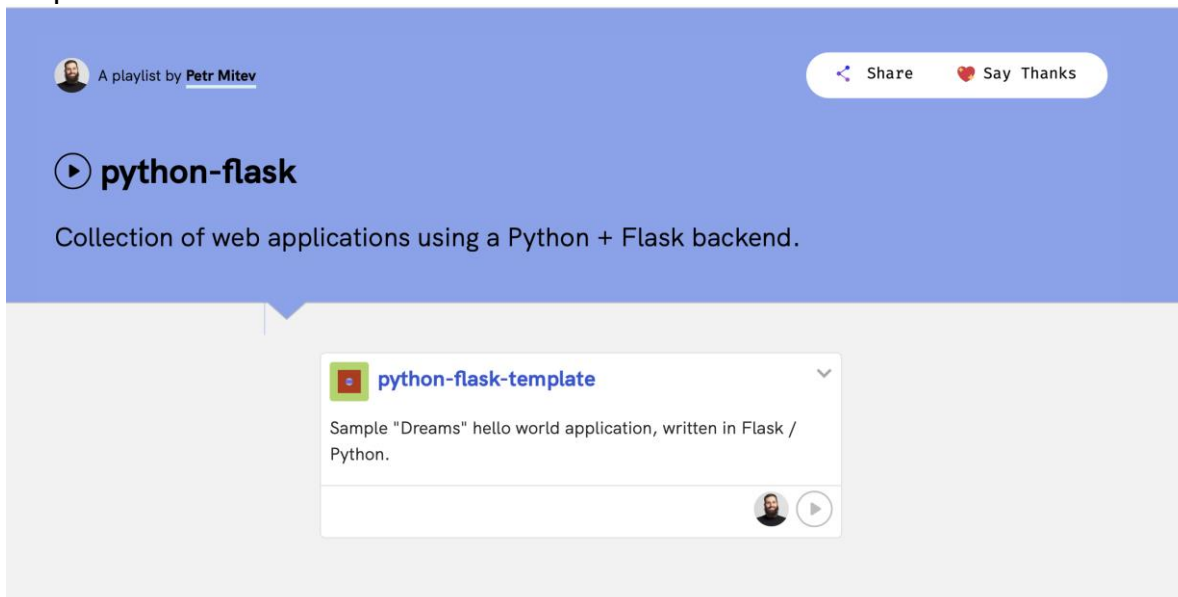
Para crear un nuevo proyecto, nos dirigimos a la parte superior derecha y sobre la imagen en forma de lupa buscamos el proyecto de tipo Flask.



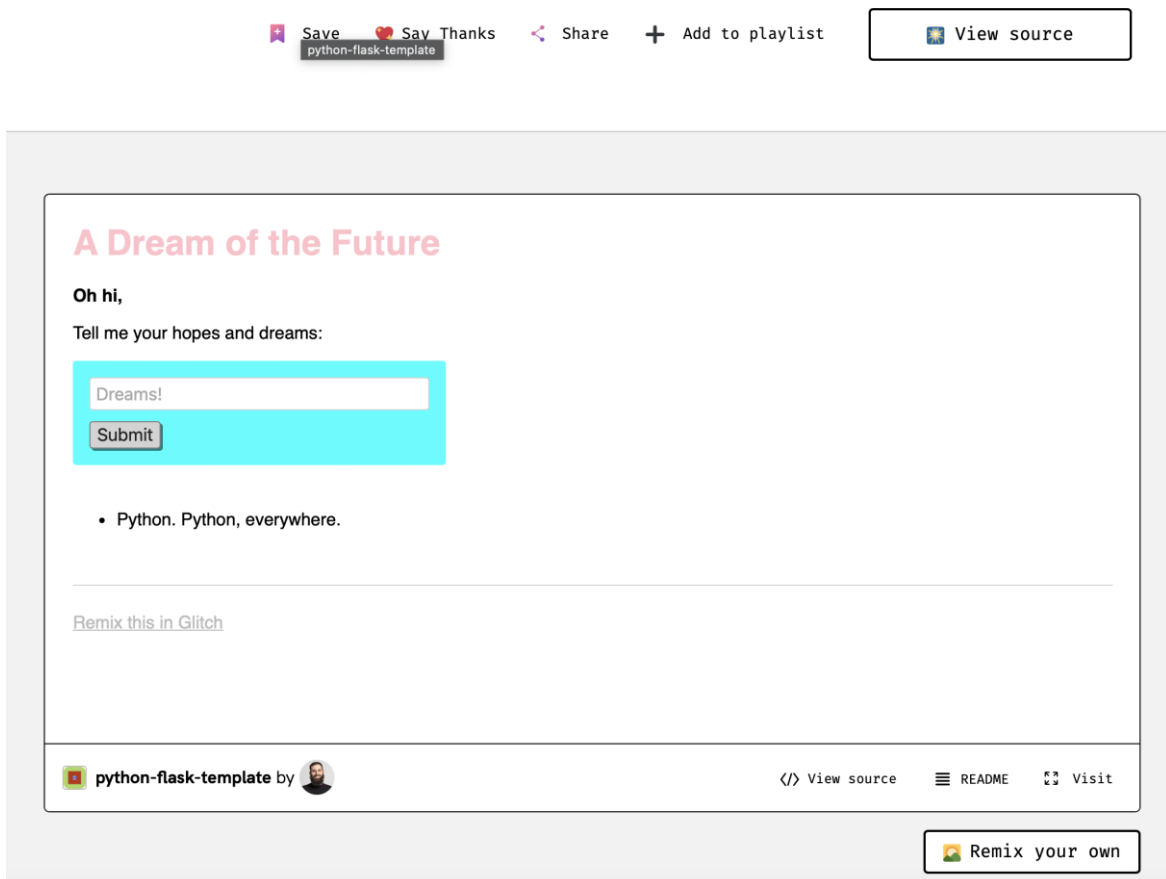
Nos cargan las plantillas que coinciden con la búsqueda que realizamos.



Debemos tener en cuenta que buscamos un proyecto plantilla que trabaje tanto con Flask y con Python por lo que escogemos a opción que nos muestra ambos componentes



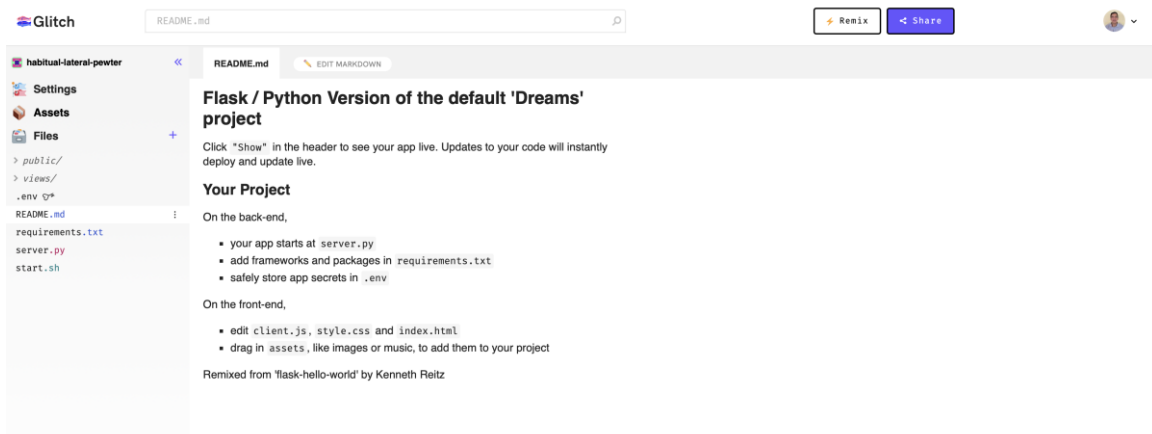
Luego seleccionamos la plantilla y nos muestra la siguiente pantalla.



Para cargar esta plantilla a nuestro proyecto escogemos la opción **Remix your own**. Esto creará la plantilla en nuestro repositorio de proyectos.

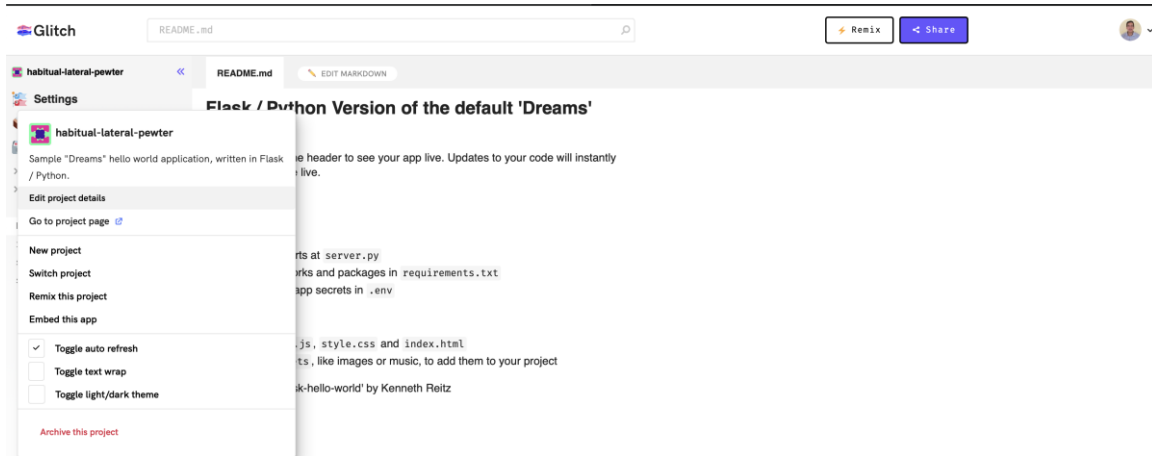
Paso 2: Configuración inicial del proyecto

Una vez cargada la plantilla nos lleva a la pantalla principal del proyecto.

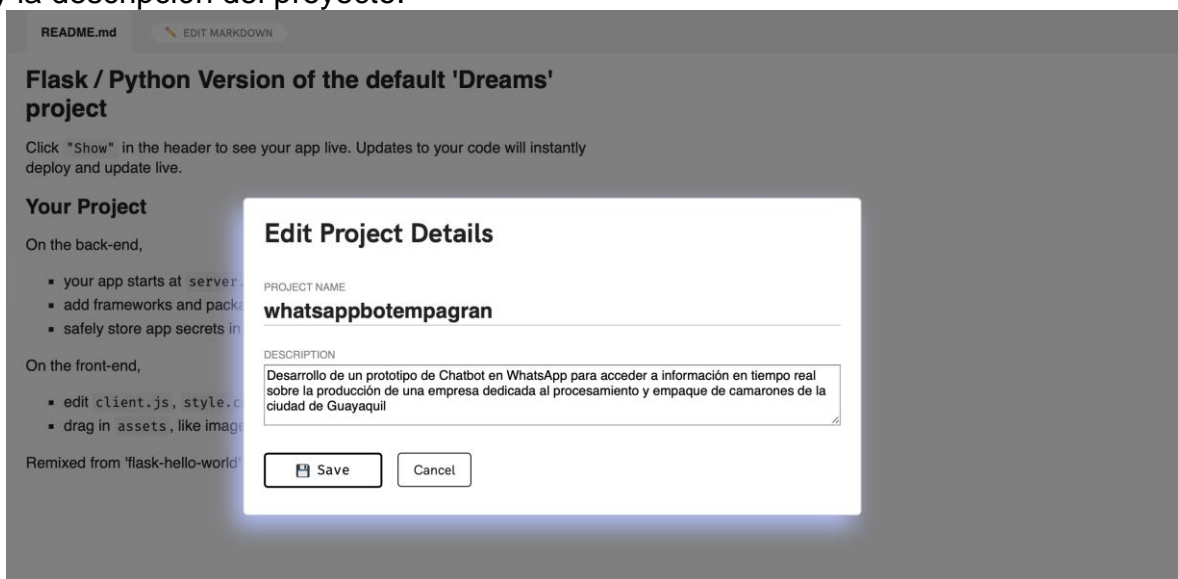


Empezamos cambiándole el nombre a uno propio que identifique dentro de **dashboard** de proyectos. En este caso llamaremos **whatsappbotempagran**.

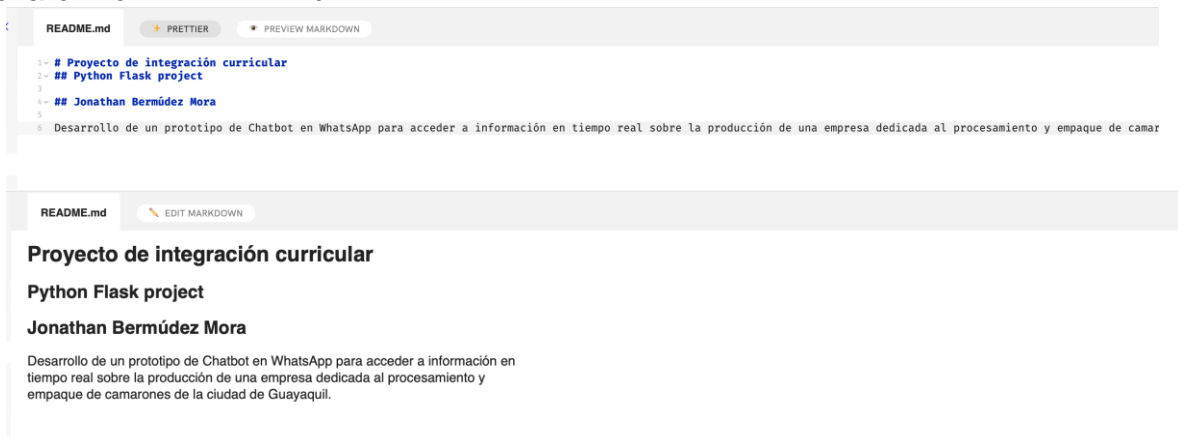
Para cambiar el nombre nos dirigimos a **Settings** para que se nos despliegue un menú flotante y seleccionamos **Edit Project Details**.



Luego, se nos apertura una ventana emergente que nos permite cambiar el nombre y la descripción del proyecto.



Presionamos el botón **Save** para guardar los cambios. Para incluir una mejor comprensión e identificación de nuestro proyecto modificamos el archivo README.md



Para finalizar podremos visualizar en el **DashBoard** principal nuestro proyecto con el nombre y la descripción previamente ingresada.

## Manage your projects

You have 1,000 active hours per month for all of your projects. [Learn more](#) →

Project Hours 0% | 4/1000 hours/month  Projects you no longer own

**Become a Glitch Member**  
 Upgrade your account to give your projects superpowers: Unlimited hours, no rate limits, and boosted specs for up to five projects.

[Upgrade](#)

## Projects

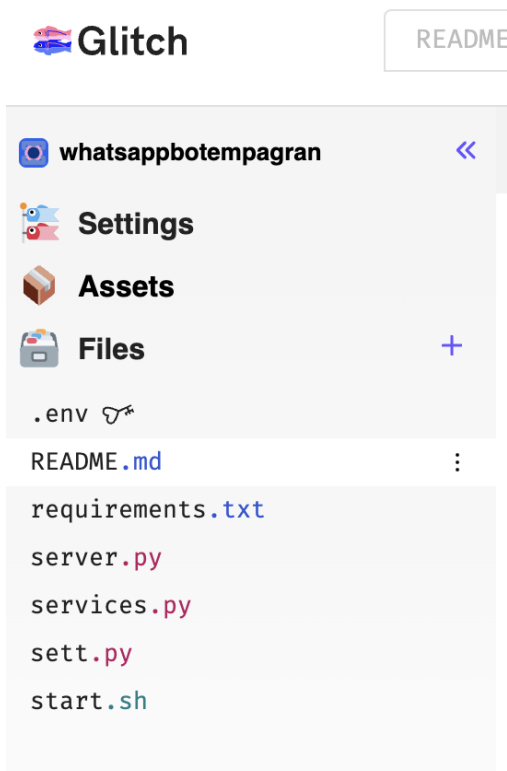
You own You're a member Archived

Search Projects

Project name	Boosted	Project hours	Last edited
<input type="checkbox"/> <b>whatsappbotempagran</b> Desarrollo de un prototipo de Chatbot en WhatsApp para acceder a información...	<input type="checkbox"/>	4	Jul 18, 2024

## Despliegue

Para el despliegue, se crea una estructura de archivos similar a la del proyecto del entorno de desarrollo integrado obedeciendo el mismo nombre de los archivos, notar que el archivo del proyecto `app.py`, será reemplazado por el archivo `server.py`, en el cual se deberá copiar el mismo código fuente que se tiene en el archivo `app.py`



Para la carga de los requerimientos de nuestra aplicación utilizamos el archivo `requirements.txt` el cual se especifica todas las librerías que necesita nuestro proyecto.



The screenshot shows the Glitch editor interface. On the left, a sidebar lists the project name 'whatsappbotempagran', 'Settings', 'Assets', and 'Files'. Under 'Files', several files are listed: '.env', 'README.md', 'requirements.txt', 'server.py', 'services.py', 'sett.py', and 'start.sh'. The 'requirements.txt' file is selected and its content is displayed in the main editor area. The content of 'requirements.txt' is as follows:

```
1 blinker==1.6.2
2 certifi==2023.5.7
3 charset-normalizer==3.1.0
4 click==8.1.3
5 Flask
6 idna==3.4
7 itsdangerous==2.1.2
8 Jinja2==3.1.2
9 MarkupSafe==2.1.3
10 requests==2.31.0
11 urllib3==1.26.6
12 Werkzeug
```

Para verificar la correcta ejecución de la aplicación verificar el archivo **start.sh** el cual posee el arranque del programa y la carga de todos los requerimientos.

The screenshot shows the Glitch editor interface with the 'start.sh' file selected. The sidebar on the left now includes 'requirements.txt' in the 'Files' list. The main editor area displays the content of 'start.sh':

```
1 #!/bin/bash
2
3 # Exit early on errors
4 set -eu
5
6 # Python buffers stdout. Without this, you won't see what you "print" in the Activity Logs
7 export PYTHONUNBUFFERED=true
8
9 python3 -m pip install -U pip
10
11 # Install the requirements
12 python3 -m pip install -r requirements.txt
13
14 # Run a glorious Python 3 server
15 python3 server.py
```

### Ejecución.

Al ser glitch una plataforma de entorno de edición en vivo, los cambios aplicados en ambos archivos mencionados en el paso de despliegue se verán reflejados en el log de ejecución el cual se encuentra en la parte inferior de la pantalla.

Glitch  Remix Share

Logs Clear Debugger

```
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->click==8.1.3->-r requirements.txt (line 4)) (3.7.4.3)

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pip in ./local/lib/python3.7/site-packages (24.0)
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: blinker==1.6.2 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (1.6.2)
Requirement already satisfied: certifi==2023.5.7 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 2)) (2023.5.7)
Requirement already satisfied: charset-normalizer==3.1.0 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 3)) (3.1.0)
Requirement already satisfied: click==8.1.3 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (8.1.3)
Requirement already satisfied: Flask in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 5)) (2.0.0)
Requirement already satisfied: idna==3.4 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 6)) (3.4)
Requirement already satisfied: itsdangerous==2.1.2 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 7)) (2.1.2)
Requirement already satisfied: Jinja2==3.1.2 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 8)) (3.1.2)
Requirement already satisfied: MarkupSafe==2.1.3 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 9)) (2.1.3)
Requirement already satisfied: requests==2.31.0 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 10)) (2.31.0)
Requirement already satisfied: urllib3==1.26.6 in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 11)) (1.26.6)
Requirement already satisfied: Werkzeug in ./local/lib/python3.7/site-packages (from -r requirements.txt (line 12)) (2.2.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from click==8.1.3->-r requirements.txt (line 4)) (4.8.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->click==8.1.3->-r requirements.txt (line 4)) (3.6.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->click==8.1.3->-r requirements.txt (line 4)) (3.7.4.3)
* Tip: There are .env or .flaskenv files present. Do "pip install python-dotenv" to use them.
* Serving Flask app 'server' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

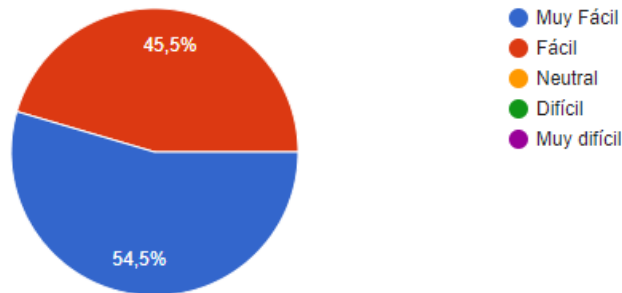
STATUS LOGS TERMINAL TOOLS PREVIEW

Con el mensaje en pantalla “Running on <http://127.0.0.1:5000>” se da por iniciada la aplicación en el ambiente de producción

### Anexo 3. Tabulación de Encuesta de aceptación del prototipo.

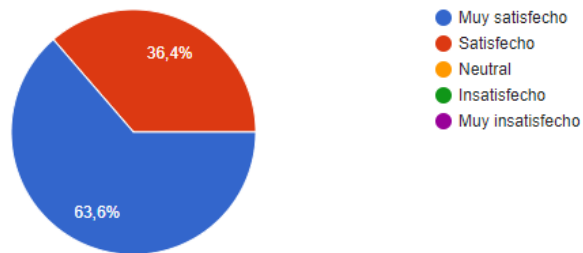
1. - ¿Cómo evalúa la facilidad de uso del chatbot de WhatsApp para resolver sus consultas sobre la producción en la empacadora?

11 respuestas



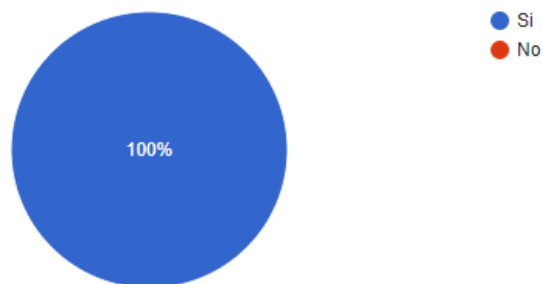
2. - ¿Qué tan satisfecho está con la rapidez de las respuestas proporcionadas por el chatbot?

11 respuestas



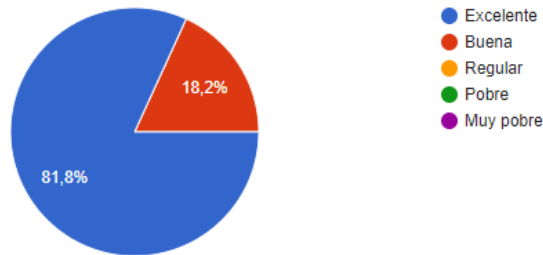
3. - ¿El chatbot ha sido capaz de resolver sus consultas de manera precisa y eficiente?

11 respuestas



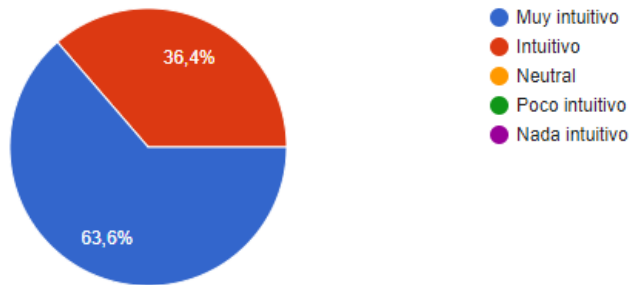
4. - ¿Cómo calificaría la integración del chatbot con los sistemas existentes de la empresa? (por ejemplo, acceso a datos en tiempo real, precisión de la información, etc.)

11 respuestas



**5. - ¿Qué tan intuitivo le parece el uso del chatbot en términos de navegación y acceso a opciones?**

11 respuestas



**Anexo 4. Resultados de Encuesta 1**

Resultados de la encuesta realizada a la muestra de los laborales para conocer su opinión sobre los chatbots.



C:\Users\  
jonathan.bermudez\

**Anexo 5. Resultados de Encuesta 2**

Resultados de la encuesta realizada a la muestra de los laborales para conocer el nivel de aceptación del prototipo.



C:\Users\  
jonathan.bermudez\

**Anexo 6. Código Fuente del prototipo y de la API.**



Codigo Fuente  
Chatbot.zip



ChatBot Api  
Produccion.zip

## DECLARACIÓN Y AUTORIZACIÓN

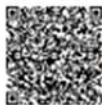
Yo, **Bermúdez Mora, Jonathan David** con C.C: # **0923712269** autor/a del trabajo de titulación: **Desarrollo de un prototipo de Chatbot en WhatsApp para acceder a información en tiempo real sobre la producción de una empresa dedicada al procesamiento y empaque de camarones de la ciudad de Guayaquil**, previo a la obtención del título de **Ingeniero en Sistemas Computacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 02 de septiembre de 2024

### EL AUTOR



Concedo autorización por:  
**JONATHAN DAVID  
BERMUDEZ MORA**

---

Nombre: **Bermúdez Mora, Jonathan David**

C.C: **0923712269**

## **REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA**

### **FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN**

<b>TEMA Y SUBTEMA:</b>	Desarrollo de un prototipo de Chatbot en WhatsApp para acceder a información en tiempo real sobre la producción de una empresa dedicada al procesamiento y empaque de camarones de la ciudad de Guayaquil		
<b>AUTOR(ES)</b>	Bermúdez Mora, Jonathan David		
<b>REVISOR(ES)/TUTOR(ES)</b>	Ing. Yong Yong, Byron Severo		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Ingeniería		
<b>CARRERA:</b>	Ingeniería en Sistemas Computacionales		
<b>TÍTULO OBTENIDO:</b>	Ingeniero en Sistemas Computacionales		
<b>FECHA DE PUBLICACIÓN:</b>	02 de septiembre de 2024	<b>No. DE PÁGINAS:</b>	92
<b>ÁREAS TEMÁTICAS:</b>	Producción de empaquera de camarón, Bots, Proceso de empaque y exportación		
<b>PALABRAS CLAVES/KEYWORDS:</b>	Chatbot, producción, camarones, WhatsApp, python		
<b>RESUMEN/ABSTRACT:</b>	<p>El objetivo principal de este proyecto es desarrollar un prototipo de chatbot en WhatsApp para acceder en tiempo real a la información de la producción de una empresa dedicada al procesamiento y empaque de camarones. El marco teórico se fundamenta en la importancia de los chatbots en la mejora de la eficiencia operativa mediante la automatización de consultas y la integración de sistemas de información en tiempo real. La metodología empleada incluyó el análisis del problema con el diagrama de Ishikawa, identificando causas como la dependencia de procesos manuales, el uso de hojas de cálculo, y las limitaciones en la generación de reportes, lo que resultó en una obtención ineficiente y un acceso no oportuno a los datos. Para abordar esto, se utilizó la investigación descriptiva y explicativa, detallando las necesidades de los usuarios y la relación entre el prototipo y la mejora en el acceso a la información. Para el desarrollo del prototipo se adoptó el método en cascada, utilizando las herramientas de desarrollo: Python y Flask, integrando APIs y bases de datos relacionales. La puesta en marcha de este prototipo va a servir para que los gerentes y jefes puedan acceder a los datos críticos de la producción sin la necesidad de estar frente a un computador, de esta forma ayuda a la toma de decisiones, así como también una visión más amplia en la eficiencia operativa del modelo de negocio. El contenido del trabajo abarca el desarrollo y prueba del prototipo, integrando tanto las API de la empresa como las de WhatsApp; incluye el análisis de datos, la creación de modelos de entidad-relación y la implementación de una interfaz amigable para el usuario final. Las conclusiones destacan la efectividad del prototipo en la optimización del acceso a información y su potencial para ser implementado en otras áreas de la empresa debido a su alta aceptación por parte de los usuarios.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	Teléfono: +593-983-714986	E-mail: <a href="mailto:jonathan.bermudez@cu.ucsg.edu.ec">jonathan.bermudez@cu.ucsg.edu.ec</a>	
<b>CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE)::</b>	Toala Quimí, Edison José		
	Teléfono: +593-990-976776		
	E-mail: edison.toala@cu.ucsg.edu.ec		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			