

UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO

CARRERA:

INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:

Ingeniera de Sistemas

TEMA:

CREACIÓN DE UN PROTOTIPO DE CHATBOT PARA EL APOYO EN EL PROCESO DE ENSEÑANZA-APRENDIZAJE VIRTUAL DE LA CÁTEDRA DE INGENIERÍA DE SOFTWARE DE LA CARRERA DE CIENCIAS DE LA COMPUTACIÓN.

AUTORA:

DEISY TATIANA AUZ ROSERO

TUTOR:

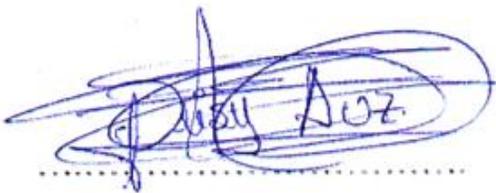
FRANKLIN EDMUNDO HURTADO LARREA

Quito, marzo del 2021

CESIÓN DE DERECHOS DE AUTOR

Yo Auz Rosero Deisy Tatiana, con documento de identificación N° 0401014865, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del trabajo de titulación intitulado: **CREACIÓN DE UN PROTOTIPO DE CHATBOT PARA EL APOYO EN EL PROCESO DE ENSEÑANZA-APRENDIZAJE VIRTUAL DE LA CÁTEDRA DE INGENIERÍA DE SOFTWARE DE LA CARRERA DE CIENCIAS DE LA COMPUTACIÓN.**, mismo que ha sido desarrollada para optar por título de Ingeniera de Sistemas en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En la aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autora me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



Deisy Tatiana Auz Rosero

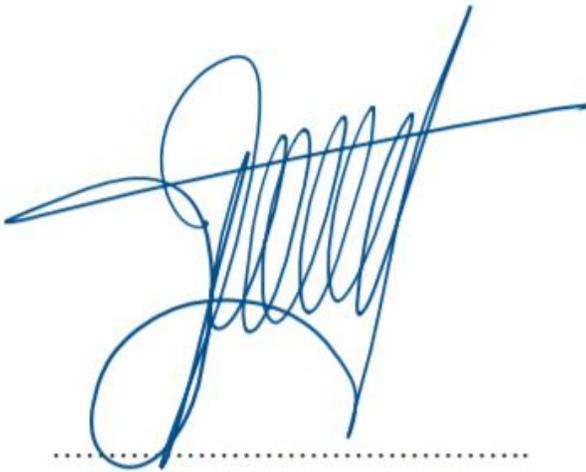
C.I: 0401014865

Quito, marzo del 2021

DECLARATORIA DE COAUTORIA DEL DOCENTE TUTOR/A

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto técnico, CREACIÓN DE UN PROTOTIPO DE CHATBOT PARA EL APOYO EN EL PROCESO DE ENSEÑANZA-APRENDIZAJE VIRTUAL DE LA CÁTEDRA DE INGENIERÍA DE SOFTWARE DE LA CARRERA DE CIENCIAS DE LA COMPUTACIÓN., realizado por Auz Rosero Deisy Tatiana, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, marzo del 2021

A handwritten signature in blue ink, consisting of a series of loops and a long horizontal stroke, positioned above a dotted line.

Franklin Edmundo Hurtado Larrea

C.I:1713382016

DEDICATORIA

El presente trabajo va dedicado con todo mi amor y cariño a mis padres Ligia y Polibio, pilares fundamentales quienes me han impulsado a seguir adelante y afrontar las adversidades siendo una guía con su esfuerzo y sacrificio, a mis hermanos Verónica, Dennis y Yamile quienes me han apoyado en cada etapa de mi vida, a mis sobrinos por compartir su alegría y entusiasmo.

A mi amado esposo Jhonny, mi compañero inseparable por su gran dedicación y su infinita paciencia quien me brindó palabras de aliento y su ayuda en el transcurso de mi etapa universitaria.

A mis profesores, quienes me impartieron los conocimientos necesarios para enfrentarme en el mundo laboral y a mis compañeros con quienes compartí muchas horas de estudio y me brindaron su compañía y amistad.

AGRADECIMIENTO

Agradezco en primer lugar a la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana, por brindarme el apoyo educativo y académico que me permitió crecer a nivel profesional y humano.

A Dios por otorgarme la salud y la energía a través de la carrera para concluir el presente proyecto.

A mis padres quien me han brindado su apoyo incondicional de diferentes maneras, y me han motivado para que concluya mi carrera universitaria a pesar de las adversidades.

A mi tutor el Ing. Franklin Hurtado por su guía y asesoramiento en cada etapa del desarrollo del presente proyecto.

Y a todas las personas que de alguna u otra manera hicieron posible pudiera culminar con esta etapa de mi vida.

Índice

Resumen	1
Abstract	2
Introducción	3
Antecedentes	3
Problema de estudio	3
Justificación.....	5
Objetivos	6
Metodología	6
Marco Referencial.....	9
Capítulo 1	11
Marco teórico	11
1.1. Los chatbots en el ámbito educativo.....	11
1.1.1. Chatbot.....	11
1.1.1.1. Inteligencia Artificial	11
1.1.1.2. Machine Learning	11
1.1.1.3. Inteligencia Artificial en el ámbito educativo.....	12
1.1.1.4. Chatbots en la educación universitaria.	13
1.1.2. Tipos de Chatbots	15
1.2. Base de conocimiento.....	16

1.2.1 Marco de Conocimiento	16
1.3. Procesamiento del Lenguaje Natural (NLP).....	17
1.3.1. Librerías para NLP	17
1.4. Metodología para la construcción de Chatbots.....	18
1.4.1. Tecnologías para la construcción de chatbots	20
1.5. Base de datos MongoDB	23
1.5.1. Diseño de bases de datos documentales	24
1.6. Python.....	25
1.6.1. PEP 8	25
Capítulo 2	27
Análisis y diseño	27
2.1. Requerimientos	27
2.1.1. Requerimientos funcionales	27
2.1.2. Requerimientos no funcionales	28
2.2. Análisis del contenido del marco de conocimiento	28
2.3. Análisis técnico del tipo de chatbot	29
2.4 Modelamiento.....	30
2.4.1. Casos de Uso	30
2.4.2. Análisis del funcionamiento del prototipo.....	33
2.4.3. Diagrama de componentes.....	34

2.4.4.	Diagrama del metamodelo conceptual de la base de datos	35
2.4.5.	Diagrama de clases	36
2.4.6.	Diagrama de Secuencia	37
Capítulo 3	40
Construcción	40
3.1.	Buenas prácticas de codificación	40
3.2.	Lineamientos para la construcción	40
3.2.1.	Selección de librerías para el Procesamiento del Lenguaje Natural	41
3.2.2.	Selección de librerías para el procesamiento de datos	43
3.2.3.	Selección de submódulos de la librería sklearn para la consulta ingresada	44
3.2.4.	Selección del algoritmo para la emisión de la respuesta	45
3.3.	Arquitectura implementada del prototipo.....	47
3.3.1.	Construcción de la base de conocimiento	47
3.3.2.	Construcción de la capa lógica.....	49
3.3.2.1.	Construcción de la clase chatbot	50
3.3.2.2.	Construcción de la Interfaz de Procesos Automatizada (API).....	51
3.3.3.	Construcción de la aplicación web.....	52
3.3.3.1.	Creación de la interfaz web del usuario.....	52
3.3.3.2.	Creación de la interfaz web del usuario docente	53
3.3.3.3.	Notificación vía correo electrónico.....	53

3.4.	Perspectiva Navegacional.....	55
3.4.1.	Código Relevante	56
	Capítulo 4	59
	Pruebas	59
4.1.	Pruebas funcionales	61
4.2.	Pruebas de confiabilidad.....	66
4.3.	Pruebas de eficiencia	71
4.4.	Usabilidad.....	74
4.5.	Pruebas de portabilidad	74
	Conclusiones y recomendaciones.....	76
	Referencias Bibliográficas	79
	ANEXOS.....	83

Índice de figuras

Figura 1: Tablero Scrum de la herramienta kanbantool con las tareas realizadas.	7
Figura 2: Plan analítico de la cátedra de Ingeniería del Software	10
Figura 3: Tabla comparativa de las bases de datos SQL y MongoDB.....	24
Figura 4: Marco de Conocimiento del Chatbot.....	29
Figura 5: Caso de Uso de la Asistencia Virtual generada por el prototipo	31
Figura 6: Caso de Uso administración de usuarios	32
Figura 7: Caso de Uso del inicio de sesión del docente	32
Figura 8: Caso de Uso de administración del chatbot.....	33
Figura 9: Diagrama de despliegue que representa el funcionamiento del prototipo.....	34
Figura 10: Diagrama de Componentes del prototipo	35
Figura 11: Metamodelo conceptual de la colección textos	36
Figura 12: Diagrama de clases del prototipo.....	37
Figura 13: Diagrama de secuencia del patrón de Arquitectura MVC	38
Figura 14: Descripción del capítulo 3 del Plan Analítico	49
Figura 15: Código para encontrar la similitud de las entradas y la base del conocimiento	50
Figura 16: Código de implementación de la API.....	51
Figura 17: Interfaz web del usuario.....	52
Figura 18: Interfaz Web del usuario docente	53
Figura 19: Código de implementación de la clase correo.py	54
Figura 20: Notificación de correo electrónico.....	55
Figura 21: Segmento de código para insertar y modificar información de la base de conocimiento desde la interfaz web	57

Figura 22: Segmento de código para la eliminación de información de la base de conocimiento desde la interfaz web	57
Figura 23: Segmento de código de los métodos para conseguir, insertar y actualizar el texto de la base de conocimiento	58
Figura 24: Resultados de las pruebas de funcionales	63
Figura 25: Imagen de la base de conocimiento del Modelo Cascada	63
Figura 26: Resultados de las pruebas de confiabilidad	68
Figura 27: Resultados de las pruebas de confiabilidad	69
Figura 28: Resultados de la herramienta Jmeter en pruebas de estrés	73
Figura 29: Resultado de la herramienta Jmeter	73
Figura 30: Resultado de las pruebas de usabilidad	74

Índice de tablas

Tabla 1: Tabla Comparativa de Librerías NLP	41
Tabla 2. Tabla comparativa de Librerías.....	43
Tabla 3: Características de los submódulos de sklearn	45
Tabla 4: Comparación de algoritmos de similitud	46
Tabla 5: Campos del documento textos	47
Tabla 6: Campos del documento saludos.....	48
Tabla 7: Plan de Pruebas	59
Tabla 8: Resultados Pruebas Funcionales	62
Tabla 9: Resultados de las pruebas funcionales planteadas en el escenario 2	64
Tabla 10: Matriz de resultados pruebas funcionales caso 2	65
Tabla 11: Matriz de resultados pruebas funcionales caso 3	65
Tabla 12: Matriz de Resultados de las pruebas funcionales caso 4	66
Tabla 13: Escenario de pruebas de confiabilidad.....	66
Tabla 14: Casos prueba	67
Tabla 15: Resultados de las pruebas de confiabilidad escenario 2	70
Tabla 16: Resultados pruebas de eficiencia	71
Tabla 17:Tabla Comparativa de la portabilidad.....	75

Resumen

En el presente documento se describe la construcción del prototipo de chatbot informativo de preguntas frecuentes (Frequently Asked Questions), FAQ por sus siglas en inglés; con diferentes herramientas y librerías de Python, que permiten el Procesamiento de Lenguaje Natural (NLP), análisis de datos y emisión de respuesta con la información almacenada en una base de conocimiento, la cual está constituida por la información relevante obtenida de la bibliografía principal de la cátedra y almacenada en una base de datos del tipo NoSQL (no relacional); obteniendo como resultado una aplicación de fácil uso para los estudiantes de Ingeniería del Software de la carrera en Ciencias de la Computación, sede Quito; que nació con la necesidad de facilitar la comprensión de conceptos básicos impartidos en la cátedra respecto a determinados temas desarrollados en el proceso educativo, el prototipo de chatbot es capaz de analizar las preguntas ingresadas por el usuario y ofrecer una respuesta acertada. En el caso de que la información sobre alguna pregunta ingresada no estuviera contemplada, se emite un correo electrónico al docente encargado de administrar la base de conocimiento, que indica la necesidad de alimentación sobre la información no encontrada con el fin de mantenerla actualizada.

Abstract

This document describes the construction of an informative chatbot prototype for frequently asked questions (FAQ), with different Python tools and libraries. This prototype allows Natural Language Processing (NLP), data analysis and answering with the information stored in a knowledge base, which is made up of relevant information obtained from the main bibliography of the subject and stored in a NoSQL database (non-relational); resulting in an easy-to-use application for the students of Software Engineering in the career of Computer Science, campus Quito, which born with the need to facilitate the understanding of basic concepts taught in the subject about certain topics developed in the educational process, the chatbot prototype is capable of analyzing the questions entered by the user and offering a correct answer. If the information of any question entered is not contemplated, an email is sent to the teacher who manage the knowledge base, indicating the need to feed on the information not founded, in order to keep it updated.

Introducción

Antecedentes

En el Siglo XX, Alan Turing se planteó la interrogación ¿las máquinas pueden pensar?, lo cual dio paso a que surgiera la búsqueda de respuestas desde el ámbito informático. A través de la Inteligencia Artificial se abrieron muchos campos de investigación que permitieron dar solución a esta interrogante, cuya respuesta más cercana empezó con el desarrollo de los chatbots, programas informáticos que incorporan sistemas de Inteligencia Artificial con la capacidad de aprender a base de un conjunto de datos manteniendo el flujo de una conversación con el usuario y brindar información. El primer chatbot se conoce como Eliza, quien emula a un psicoanalista y procesa el lenguaje natural tratando de mantener una conversación textual con el paciente. (Garcia, Fuertes, & Molas, 2018).

Con el paso del tiempo, los chatbots han evolucionado significativamente, siendo implementados en diferentes ámbitos, entre ellos, el ámbito educativo en el cual representan un valioso recurso para el apoyo del aprendizaje, permitiendo tanto a los docentes como a los estudiantes mantener una comunicación fluida y desarrollar tareas como: acompañamiento, resolución de inquietudes y retroalimentación mediante respuesta de preguntas frecuentes.

Problema de estudio

Durante el proceso educativo los seres humanos obtienen la información necesaria para solventar problemas de diversa índole, el más evidente, que el docente no dispone de mucho tiempo para atender de manera personalizada a sus estudiantes durante el proceso de aprendizaje; sin embargo, son muchas las interrogantes que se plantean durante este proceso, dudas que en ocasiones no pueden ser resueltas por los docentes ya que en la época actual, el profesor deja de ser el actor encargado de la transmisión del conocimiento y se convierte en un mediador, quien

debe guiar la actividad del estudiante con el fin de que éste construya su propio intelecto, dejando atrás las técnicas tradicionales basadas en la memorización como lo sustentan los exponentes del modelo constructivista Ausubel (1983) con la teoría del aprendizaje verbal significativo, Piaget (1969) inspirado en la psicogenética; y Vigotsky (1934) con la teoría socio cultural del desarrollo del aprendizaje. A esto se suman las tendencias actuales que obligan a buscar nuevas alternativas para mejorar el proceso de enseñanza-aprendizaje y el desarrollo de competencias y habilidades para desenvolverse en una sociedad tecnificada. (Coloma & Tafur, 1999).

El alumno acude al docente para resolver dudas o inquietudes que nacen de sus propias expectativas, de su propio análisis a los temas impartidos en clases; sin embargo, éstas muchas veces no alcanzan a ser absueltas por el docente ya que sus recursos casi siempre son limitados, siendo el tiempo uno de sus mayores limitantes; este es un argumento de la enseñanza teórica donde se cree que “el descuido de la consideración personalizada de la enseñanza, puede favorecer el decrecimiento de la atención del alumno” (Atkins, 2002).

Sin embargo, en la actualidad el estudiante no solo puede acudir al docente, sino también hace uso de otras fuentes, siendo una de las más recurrentes el uso de fuentes digitales, donde el alumno se encuentra rodeado de un sinnúmero de medios cada vez más sofisticados que sirven de gran ayuda en el proceso educativo, los cuales no siempre brindan la información para solventar las dudas específicas que un estudiante busca; además la información obtenida no siempre está alineada al enfoque que el docente tiene con la materia impartida. Sin embargo, en la actualidad el estudiante no solo puede acudir al docente, sino también hace uso de otras fuentes, siendo una de las más recurrentes el uso de fuentes digitales, donde el alumno se encuentra rodeado de medios cada vez más sofisticados que sirven de gran ayuda en el proceso educativo, los cuales no siempre brindan la información necesaria para solventar las dudas específicas que un estudiante busca; además la

información obtenida no siempre está alineada al enfoque que el docente tiene con la materia impartida.

Justificación

La creación de un chatbot para el apoyo en el proceso de enseñanza – aprendizaje virtual en la cátedra de Ingeniería del Software de la carrera en Ciencias de la Computación tiene como objetivo solventar dudas o inquietudes de quienes forman parte de este proceso, mediante la ayuda de un asistente virtual que permitirá una comunicación constante con alta disponibilidad, cuyas respuestas son automáticas y las realiza en función de las instrucciones dadas, permitiendo así que el usuario pueda capacitarse en el momento deseado de manera personalizada, con respuestas específicas a preguntas frecuentes; siendo el chatbot un asistente automático del docente que imparte la cátedra. Los sistemas educativos en la actualidad deben atender nuevas demandas que satisfagan la necesidad de los estudiantes, creando ambientes de aprendizaje enriquecidos tecnológicamente, de fácil adaptabilidad y con excelentes resultados.

Según Bill Gates, hasta las asignaturas más complejas y subjetivas pueden enseñarse con la ayuda de un chatbot como tutor, porque entiende que la clave del aprendizaje de cualquier destreza yace en la posibilidad de obtener apoyo y *feedback* de alguien con conocimientos relevantes en el campo del que se trate, a lo que no tienen acceso muchos estudiantes. En el caso del chatbot, no se trata de un apoyo cualquiera, sino de conversaciones reales y completas que ayudan a los estudiantes a atravesar cada asignatura con un enfoque personalizado, como si tuvieran un tutor personal, pero con la ventaja de que están disponibles en todo momento, en cualquier lugar, sin la necesidad de descargar un programa. (Pérez, 2018)

Objetivos

Objetivo general:

Crear un prototipo chatbot como ayudante virtual en el proceso de enseñanza - aprendizaje virtual de la cátedra de Ingeniería del Software de la carrera Ciencias de la Computación de la Universidad Politécnica Salesiana.

Objetivos específicos:

- Investigar sobre la naturaleza de los chatbots y su utilización como asistentes virtuales para la resolución de preguntas frecuentes.
- Implementar una base de datos con la información que el chatbot estará en la capacidad de responder.
- Desarrollar un prototipo de chatbot con una interfaz que permita la iteración usuario-máquina
- Entrenar el chatbot con la información necesaria para que este puede emitir respuestas acertadas.
- Realizar las pruebas necesarias para conocer el tiempo de respuesta y la carga necesaria.

Metodología

Metodología para el desarrollo

Para el desarrollo del presente proyecto se utilizó el marco de trabajo Scrum ya que, a lo largo del proyecto, se realizaron entregas de productos incrementales tras adaptar los cambios con base en la retroalimentación obtenida por parte del tutor del proyecto. Para ayudar en el flujo de tareas se utilizó la herramienta en línea Kanbantool, la cual es gratuita y permite registrar el movimiento de las tareas a través del tablero Scrum, como se puede ver en la figura 1.

Tablero Scrum en la herramienta Kabantool

Que hacer	Haciendo	Hecho	Revisado
+	+	+	+
		<div style="background-color: yellow; padding: 5px; margin-bottom: 5px;"> Información de la materia almacenada en una base de datos no relacional re </div> <div style="background-color: yellow; padding: 5px;"> Diferenciación de ambientes para los tres tipos de usuarios establecidos. re </div>	<div style="background-color: yellow; padding: 5px; margin-bottom: 5px;"> Análisis de herramientas en Python para la creación de chatbots con Inteligencia Artificial re </div> <div style="background-color: yellow; padding: 5px; margin-bottom: 5px;"> Análisis del contenido de la materia para la construcción del marco de conocimiento re </div> <div style="background-color: yellow; padding: 5px; margin-bottom: 5px;"> Análisis técnico del tipo de chatbot re </div> <div style="background-color: yellow; padding: 5px; margin-bottom: 5px;"> Motor del chatbot implementado en Python con las herramientas analizadas. re </div> <div style="background-color: yellow; padding: 5px; margin-bottom: 5px;"> Interfaz web conectada al motor del chatbot por medio de la API desarrollada. re </div> <div style="background-color: yellow; padding: 5px;"> Proceso de emisión de correo electrónico para la alimentación de la base de conocimiento. re </div>

Figura 1: Tablero Scrum de la herramienta Kanbantool con las tareas realizadas.

Elaborada por: Deisy Auz

Planificación

En la planificación se obtuvieron todos los requerimientos para el desarrollo del chatbot. Se establecieron las tareas a realizar y los Sprints a partir de las funcionalidades sobre las que se trabajó.

Se establecieron los roles de Scrum de los involucrados en el proyecto, definiendo de esta manera:

- **Product owner:** Se definieron como dueños del producto, a aquellos docentes que imparten la cátedra de Ingeniería del Software en la sede Quito.

- **Scrum master:** Se designó como Scrum master, al tutor del presente proyecto de titulación, puesto que él se encargó de la gestión de reuniones y presentación de avances.
- **Equipo Scrum:** El equipo Scrum, encargado de realizar los cambios y desarrollar los incrementos del proyecto, está conformado únicamente por Deisy Auz, autora del proyecto de titulación.
- **Usuarios:** El rol de usuarios fue asignado tomando en cuenta que el chatbot será utilizado por tres tipos de usuarios: usuario administrador, usuario docente y usuario visitante.

Una vez definidos los roles, se procedió con el establecimiento de los backlogs.

Desarrollo Incremental

Scrum es un marco de trabajo cuyo objetivo principal es el desarrollo de productos terminados y utilizables que aporten incrementos significativos al producto final. Estos son presentados como entregables al final de cada Sprint.

Para el desarrollo del presente proyecto se utilizó la técnica de prototipado, enfoque que permitió ordenar las etapas del proceso para el desarrollo, creando un conjunto de especificaciones de diseño, enfocado en la arquitectura del chatbot. Este prototipado fue dividido por incrementos, los mismos que se desarrollaron dentro de los Sprints, para conseguir incrementos significativos del producto.

Al final de cada Sprint se entregó el avance incremental desarrollado de cada componente del prototipo, siendo evaluado de acuerdo con la disposición del Scrum master:

1. Análisis de herramientas en Python para la creación de chatbots con Inteligencia Artificial.
2. Análisis del contenido de la materia para la construcción del marco de conocimiento.
3. Análisis técnico del tipo de chatbot.

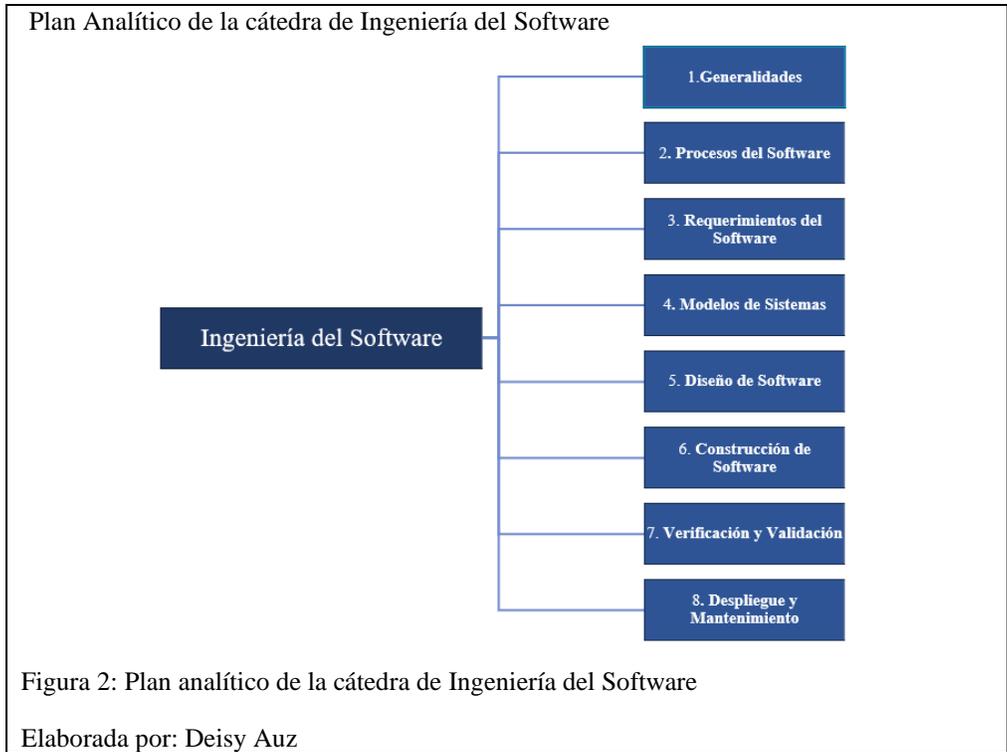
4. Motor del chatbot implementado en Python con las herramientas analizadas.
5. Información de la materia almacenada en una base de datos no relacional.
6. Interfaz web conectada al motor del chatbot por medio de la API desarrollada.
7. Proceso de notificación vía correo electrónico, indicando que la información no se encuentra en la base del conocimiento.
8. Diferenciación de ambientes para los tres tipos de usuarios establecidos.

Pruebas

A través del plan de pruebas ubicado en el anexo 2, se realizaron las pruebas necesarias para evaluar el funcionamiento del chatbot y analizar los aspectos que requieren de cambios.

Marco Referencial

Ingeniería del Software es una cátedra que se imparte en séptimo nivel de la carrera en Ciencias de la Computación de la Universidad Politécnica Salesiana sede Quito, la misma que depende de la materia de análisis y diseño de sistemas de sexto nivel. La cátedra es parte del campo de formación de fundamentos teóricos de la unidad profesional, es de modalidad presencial y cubre un total de 200 horas, distribuidas en 80 horas para el componente de docencia, 80 horas para el componente prácticas de aplicación y experimentación de aprendizajes y 40 horas para el trabajo autónomo. En esta asignatura se estudian los métodos para la aplicación de un enfoque sistemático, del desarrollo, operación y mantenimiento del software, así como también toda la documentación, procedimientos y reglas. De acuerdo con lo establecido en el plan analítico del período 55 proporcionado por los docentes de la cátedra de Ingeniería del Software, en la figura 2 se visualizan de manera general los temas que se imparten en la cátedra. En el anexo 1 se detalla de manera completa el plan analítico citado. (Universidad Politécnica Salesiana 2019 de la carrera en Ciencias de Computación).



Capítulo 1

Marco teórico

1.1. Los chatbots en el ámbito educativo

1.1.1. Chatbot

Los chatbots son programas informáticos que imitan la conversación humana, utilizando para ello distintas técnicas de Machine Learning para la adquisición, organización, procesamiento y emisión de información. Estas técnicas hacen uso de reglas para el Procesamiento del Lenguaje Natural (Natural Language Processing) NLP, reglas con las cuales el chatbot responde acorde a la conversación iniciada por el usuario. (Neves, Barros, & Hodges, 2006)

El uso de los chatbots como asistentes virtuales surge con base en la necesidad de ayudar a los seres humanos para completar tareas al entregar información relevante sobre un tema o una pregunta específica. La información manejada por un chatbot está determinada por la necesidad que este atiende, pudiendo ir desde información comercial hasta información científica y específica por cada usuario. (Boris, 2019)

1.1.1.1. Inteligencia Artificial

Se define como Inteligencia Artificial a la rama de las Ciencias de la Computación que, a través de la historia, los seres humanos han desarrollado para simular el comportamiento y capacidades propias del cerebro humano; la Inteligencia Artificial procesada desde una máquina u ordenador permite en la actualidad realizar un sinnúmero de tareas en diferentes ámbitos. (Banda, 2014)

1.1.1.2. Machine Learning

Machine Learning, o Aprendizaje de Máquina, es una rama de la Inteligencia Artificial fundamentada en hacer que una computadora logre aprender sobre una determinada tarea con base

en la experiencia. Este aprendizaje involucra el proceso de búsqueda a la respuesta que mejor se adapte a la tarea en función de los ejemplos, conjuntos de entrenamiento o marcos de conocimiento previamente cargados en el sistema. (Banda, 2014)

1.1.1.3. Inteligencia Artificial en el ámbito educativo

En el ámbito educativo la informática ha alcanzado una marcada importancia por su amplia aplicación en diferentes aspectos, lo que en la actualidad se denomina informática educativa, siendo el más destacado el de la Inteligencia Artificial en los sistemas de enseñanza y aprendizaje adaptativos basados en la interacción.

La Inteligencia Artificial trae consigo innovación para las instituciones educativas y ejerce una transformación profunda tanto en la gestión educativa, cuanto, en los procesos educativos de docentes y estudiantes, puesto que contribuye a reducir las dificultades de falta de tiempo, comunicación, atención y comprensión que pueden darse en una clase presencial, transformándolas en información de refuerzo que puede obtenerse a través de la tecnología. (Jara & Ochoa, 2020).

Los usos de la Inteligencia Artificial en el aula son diversos y sirven tanto para los estudiantes como para los docentes; van desde el aprendizaje de las materias a través de un bot que corrija las respuestas, donde el propio estudiante puede obtener aprendizaje desde sus errores; o también permite diseñar tutoriales o programas didácticos que sirven de apoyo en el proceso de enseñanza-aprendizaje, donde pueden interactuar estudiantes y profesores.

Diferentes organizaciones mundiales se encuentran desarrollando proyectos para la implementación de nuevas funciones de la Inteligencia Artificial en la educación, pudiendo citar a la UNESCO, organismo que ha diseñado materiales didácticos en línea para su agenda 2030. China, a través de Squirrel AI, posee un alto interés en la aplicación de la Inteligencia Artificial en la educación para desarrollar tareas extraescolares que brindan servicios a más de 2 millones de

estudiantes. Eduband, a través de la Universidad de Pensilvania ha creado un proyecto para la aplicación de la Inteligencia Artificial en educación secundaria para potenciar las aptitudes de los estudiantes y mejorar los resultados con un aprendizaje basado en sus propias reacciones. En España, la Universidad CDU Cardenal Herrera ha desarrollado un proyecto para el uso de la Inteligencia Artificial para que tanto profesores como estudiantes puedan resolver consultas de tipo administrativo desde un bot de asistencia. (Soria, y otros, 2016)

1.1.1.4. Chatbots en la educación universitaria.

Los chatbots, al ser programas que utilizan Inteligencia Artificial para su funcionamiento, son parte integral en el ámbito educativo, sin embargo, se debe recalcar que el ámbito educativo involucra un amplio aspecto que va desde la educación inicial hasta la educación profesional por lo que se debe limitar el uso de los chatbots al ámbito de la educación universitaria. Partiendo de la anterior premisa se tienen diversos autores quienes señalan la importancia de los chatbots en el proceso de enseñanza-aprendizaje universitario, siendo los más destacados:

- Michael Nietzel señala que los chatbots son un agente importante para la retención estudiantil, es decir, que son un gran apoyo para que los estudiantes no abandonen sus carreras al ofrecer respuestas teóricas a interrogantes básicas y comunes que los estudiantes no preguntan en su momento. (Nietzel, 2020)
- Para los autores Nesreen Mejbri, Fathi Essalmi y Vasile Rus, en su investigación sobre sistemas educativos basados en simulación y conversación inteligente, detallan sobre la importancia de responder a dudas concretas que los estudiantes pudieran tener, sin entrar en el campo práctico o necesariamente supervisado. (Mejbri, Essalmi, & Rus, 2017).
- Rainer Winkler y Matthias Söllner, por otro lado, realizan un amplio estudio sobre el potencial de los chatbots en el ámbito educativo, llegando a la conclusión de que estos

sistemas son de gran apoyo sobre todo cuando se trata de evitar el contacto con una figura humana autoritaria al momento de realizar preguntas sobre conceptos básicos, puesto que estos agentes se muestran amistosos ante los errores cometidos. Estos autores también recalcan que los chatbots pueden ser utilizados en cualquier materia impartida a nivel profesional, sobre todo en las carreras referentes a psicología, medicina y bienestar, idiomas y sistemas de información. (Winkler & Söllner, 2018)

- Para Ammar Adil, los chatbots educativos deben realizar dos tareas claves: analizar una pregunta y formular una respuesta, esto debe acompañar al objetivo de que los chatbots benefician a la educación superior al proporcionar repuestas rápidas a preguntas sobre conceptos básicos, sin importar el tipo de materia y sin necesidad de demorar el proceso al investigar en textos extensos o emitir la respuesta a los estudiantes vía correo electrónico, liberando al docente de este tipo de tareas para centrarse en tareas más relevantes. (Adil, 2020)
- Guillem Garcia, Marc Fuertes y Núria Molas, señalan que los chatbot, si bien pueden ayudar en la educación superior, estos deben manejar un tema (o materia) específica y deben responder las preguntas frecuentes sobre conceptos básicos, sin abordar temas prácticos como operaciones o procedimientos complejos. (Garcia, Fuertes, & Molas, 2018)

Tomando en cuenta lo anterior se puede decir que los chatbots son la innovación en el proceso enseñanza-aprendizaje, pueden ser aplicables en cualquier ámbito de la educación universitaria, sin embargo, hay que tomar en cuenta que estos deben estar centrados en una sola materia y que, por el momento, deben responder conceptos básicos y no entrar a temas prácticos o de operaciones complejas.

1.1.2. Tipos de Chatbots

Se debe recalcar que la tipología de los chatbots ha sido definida por diversos autores, sin embargo, la tipología más comúnmente usada es aquella establecida por los autores Kiran Ramesh, Surya Ravishankaran, Abhishek Joshi y Kallpesh Chandrasekaran en su artículo *A Survey of Design Techniques for Conversational Agents*, donde clasifican los chatbots según su alcance:

- Alcance con base en los objetivos: dentro de esta clasificación se encuentran:
 - a. Chatbots basados en tareas: orientados al usuario para ayudarle a cumplir una tarea de manera satisfactoria.
 - b. Chatbots conversacionales: aquellos que utilizan el aprendizaje automático para mantener el flujo de una conversación de la manera más natural y personalizada.
 - c. Chatbots informativos: también conocidos como chatbots de soporte, funcionan como un sistema de preguntas frecuentes (Frequently Asked Questions), FAQ por sus siglas en inglés. Son utilizados para responder de manera rápida las preguntas realizadas por los usuarios.
- Alcance con base en conocimiento
 - a. Chatbots de dominio abierto: estos chatbots emplean algoritmos de Inteligencia Artificial para aprender a medida que interactúan con los usuarios.
 - b. Chatbots de dominio cerrado: su función es mantener una conversación fluida sobre un tema específico sin aprender de las interacciones con los usuarios sino utilizando una Inteligencia Artificial de manera controlada.
- Alcance con base en servicios
 - a. Chatbots basados en servicios interpersonales: son aquellos chatbots desarrollados para interactuar con varios individuos.

- b. Chatbots basados en servicios intrapersonales: estos chatbots son desarrollados para ser el asistente virtual de una sola persona y adaptarse a sus necesidades. (Ramesh, Ravishankaran, Joshi, & Chandrasekaran, 2017)

1.2. Base de conocimiento

La base de conocimiento contiene toda la información que el sistema necesita para dar solución a diferentes problemas. Ésta por lo general se comprende de la base de hechos y el motor de inferencia establecido así en los sistemas expertos. (Bardaró, Ibañez, & Agüero, 2013).

La base de conocimiento es importante ya que determina qué tan eficaz es un sistema, puesto a que éste depende de la información vertida en ella, el contenido se puede adquirir de un experto humano ya que este tiene el conocimiento para resolver de manera apropiada los problemas de su ámbito profesional, se debe tomar en cuenta que este conocimiento es teórico.

Para el tratamiento del conocimiento existen diferentes formas de representarlo de manera adecuada, habitualmente se utilizan estas tres:

- Redes semánticas: El conocimiento se representa en una red con nodos y arcos.
- Objeto-atributo-valor (OAV): Se representa de manera gráfica el conocimiento.
- Marcos o frames: Se representa el conocimiento a través de un esquema.

1.2.1 Marco de Conocimiento

Un marco se trata de un esquema en el cual se almacena la información de manera jerárquica, se crea a través de una red de nodos y relaciones entre nodos, lo cual facilita su manejo, cada marco está constituido por casillas denominadas *slots* en las cuales se encuentran los conceptos y sus características. Los marcos organizados de manera jerárquica en un base de conocimiento pueden estar regidos por criterios o principios. (Davará, 1994)

1.3. Procesamiento del Lenguaje Natural (NLP)

El Procesamiento del Lenguaje Natural (Natural Language Processing), NLP por sus siglas en inglés, es parte de la Inteligencia Artificial la misma que permite a los ordenadores interpretar la información y obtener un significado a partir de datos textuales. Dentro del NLP se encuentran diversas tareas, que permiten el análisis textual y la eventual respuesta del sistema, estas son:

- **Tokenización:** es una tarea del NLP que segmenta las palabras dentro de un texto, utilizando los espacios en blanco o los signos de puntuación para delimitarlas.
- **Analizador sintáctico:** esta tarea es usada para eliminar la ambigüedad en una cadena de texto.
- **Extracción de información:** esta tarea es la encargada de obtener información específica del texto. (Verspoor & Cohen, 2013)

1.3.1. Librerías para NLP

Existen diversas librerías para la implementación de NLP, estas permiten el uso de sus herramientas y algoritmos, facilitando su implementación para el desarrollo de aplicaciones que implementan esta tecnología.

- **NLTK**

NLTK (Natural Language ToolKit) es una librería que consta con una gran cantidad de módulos y herramientas para realizar NLP en el lenguaje de programación Python. Fue desarrollado, primeramente, para uso pedagógico en la universidad de Pennsylvania por Steven Bird, Edward Loper y Ewan Klein en el año 2001. (Bird & Loper, 2006)

- **TextBlob**

TextBlob es una librería desarrollada en lenguaje Python para el procesamiento de datos textuales y permite realizar tareas de NLP. Fue desarrollada independientemente por Steven Loria en 2013 tomando como base la librería NLTK de la cual se la considera una extensión. (Loria, 2020)

- **Gensim**

La librería Gensim esta implementada en lenguaje Python y fue desarrollada para el NLP de grandes cantidades de texto mediante algoritmos en línea, al contrario de otras librerías con algoritmos locales. Fue desarrollada por Radim Rehurek en el año 2009 (Rehurek, 2020)

- **SpaCy**

SpaCy es una librería de NLP desarrollada en Cython y Python para NLP enfocada en ser un software para el uso industrial y empresarial. Su desarrollo se llevó a cabo en el MIT por Matthew Honnibal en el año 2015. (SpaCy, 2020)

1.4. Metodología para la construcción de Chatbots

Se debe tomar en cuenta que no existe una metodología formalizada y aceptada mundialmente para la construcción de chatbots, por lo que la construcción de este tipo de proyectos se realiza a criterio del desarrollador, sin embargo, varios autores señalan que se debe seguir pasos específicos para su correcta construcción. Los pasos más comúnmente utilizados para el desarrollo de chatbots informativos (FAQ), son los establecidos por los autores Jeevan Thukrul, Aditya Srivastava y Gaurav Thakkar:

- NLP: en este primer paso se realiza el procesamiento de la información ingresada por el usuario en forma de pregunta, con el fin de transformarla en datos que la máquina pueda entender y procesar, siendo capaz de interpretar ciertos errores o buscar sinónimos a las palabras ingresadas.
- Procesamiento de Datos: este paso se realizan las siguientes tareas:
 - Lettercase: tarea encargada de llevar las palabras a un estándar único, ya sea todas las letras en mayúscula o, de preferencia, a minúscula.
 - Remoción de ruido: esta tarea es la encargada de eliminar y reemplazar, todos los caracteres especiales, como signos de pregunta, de exclamación, números, signos matemáticos y letras tildadas, dentro de una oración dejando únicamente letras y espacios.
 - Remoción de palabras comunes: esta tarea es de suma importancia puesto que es la encargada de eliminar las palabras vacías, también llamadas stopwords, que son palabras que no aportan un valor semántico a la oración o pregunta, tales como artículos, pronombres personales, pronombres posesivos y conectores; dejando como resultado, únicamente las palabras clave.
- Análisis de la consulta ingresada del usuario: una vez que la consulta ingresada por el usuario ha sido estandarizada, se evalúa las palabras claves y se las compara con todo el conjunto de documentos llamado corpus y cada palabra (*token*) se convierte parte de una matriz de funciones TF-IDF, donde se mide la frecuencia que la frase aparece en los documentos, y luego lo compara con el número de documentos que mencionan este término dentro del corpus.

- Emisión de respuesta: tras el análisis realizado en el anterior paso, la aplicación responde al usuario acorde a lo solicitado y tomando en cuenta que se realiza un proceso de similitud entre la información ingresada y la base de conocimiento. (Thukrul, Srivastava, & Thakkar, 2020)

1.4.1. Tecnologías para la construcción de chatbots

1.4.1.1. Tensorflow

Tensorflow es una interfaz que facilita la creación de modelos de aprendizaje automático desarrollada en Python. Fue desarrollado por Google como parte del proyecto DistBelief de código cerrado en el año 2011. En el año 2015 fue liberado como software de código abierto y como la segunda generación de Google Brain, siendo el sucesor de DistBelief. (Dean, 2015)

1.4.1.2. Scikit-learn

Scikit-learn es un módulo de aprendizaje automático desarrollado para Python con el fin de apoyar proyectos de aprendizaje de máquina supervisado y no supervisado. Fue desarrollado en 2007 por David Cournapeu, sin embargo, fue lanzado públicamente, por varios autores que se involucraron en su desarrollo, en 2010. (Pedregosa, 2011)

La librería scikit-learn proporciona varios submódulos para realizar la tokenización y posterior análisis del texto ingresado:

- CountVectorizer: es un submódulo de scikit-learn el cual permite realizar la tokenización de textos creando una matriz de recuento de tokens basado en un diccionario.
- HashingVectorizer: es un submódulo de scikit-learn que permite realizar la tokenización de textos creando una matriz de ocurrencia de tokens basado en el texto ingresado.

- TfidfTransformer: es un submódulo de scikit-learn que toma una matriz de recuento de tokens y la transforma en una representación de termino-frecuencia TF donde se toma en cuenta la cantidad de veces que cierta palabra aparece en el documento, o en una representación de termino-frecuencia-inverso del texto TF-IDF donde se compara el termino-frecuencia con todos los documentos donde se mencione al termino.
- TfidfVectorizer: con este submódulo de scikit-learn, se toma el texto y se lo transforma en una matriz de funciones TF-IDF, realizando las operaciones del submódulo CountVectorizer seguido por la operación del submodulo TfidfTransformer. (scikit-learn Organization, 2020)

1.4.1.3. Emisión de respuesta

Existen varios algoritmos que permiten la selección y emisión de respuesta, que es el paso más importante para un asistente virtual, entre los más destacados se tiene:

- Ratcliff Obershelp similarity: también conocido como patrón de coincidencias Gestalt, este algoritmo permite calcular la similitud entre cadenas de texto basándose en la similitud de caracteres y encontrando una subsecuencia de longitud máxima para las cadenas de texto.

Esto es expresado por la fórmula siguiente:

$$D_{ro} = \frac{2K_m}{|S_1| + |S_2|}$$

Donde:

- K_m es el número de coincidencias de caracteres
- $|S_1|$ y $|S_2|$ son las longitudes de las cadenas de texto respectivas. (Ilyankou, 2014)

- Postfix similarity: es el algoritmo de similitud más sencillo de implementar, pues se basa en la estructura de datos de pila y expresiones regulares básicas para encontrar la similitud. (León, 2012)
- Monge-Elkan: este algoritmo mide el promedio de los valores de similitud de dos cadenas de texto que contienen varios tokens, el promedio de similitud se mide a través de la distancia que estos tienen, con la fórmula:

$$MonElkan(A, B) = \frac{1}{|A|} \sum_{i=1}^{|A|} \max\{sim(A_i, B_j)\}_{j=1}^{|B|}$$

Donde:

- A y B son los textos
- $|A|$ y $|B|$ son los números de tokens respectivamente. (Angeles & Gamez, 2015)
- Coseno Similarity: conocido como la similitud de coseno, este algoritmo mide la similitud entre los vectores, donde se analizan a través del coseno del ángulo si estos apuntan aproximadamente a la misma dirección. La fórmula utilizada es:

$$sim(x, y) = \frac{x \cdot y}{\|x\| + \|y\|}$$

Donde

- x,y son vectores (comprenden de la longitud del vector)
- $\|x\|$, $\|y\|$ son la norma euclidiana del vector

Si la medida calculada es 0 significa que los vectores no coinciden, y si la medida se acerca a 1 significa que los vectores son similares. La similitud de coseno es muy utilizada en el análisis de texto que involucran similitud de documentos. (Thukrul, Srivastava, & Thakkar, 2020)

1.5. Base de datos MongoDB

La base de datos MongoDB es una base orientada a documentos de código abierto, se caracteriza por su fácil desarrollo, la estructura de los datos es similar a JSON. Un documento está conformado por campos los cuales pueden ser arrays o arrays de documento lo que brinda ciertas ventajas:

- No depende de claves, ya que permite hacer consultas de manera rápida.
- Trae funciones predefinidas que permiten el manejo de las colecciones creadas.
- Los documentos se pueden utilizar en diferentes lenguajes de programación.

Se puede comparar a una colección o conjunto de una base de datos MongoDB, como una tabla de una base de datos relacional SQL ya que estas presentan similitud al momento de almacenar la información, la base de datos no relacional guarda la información en documentos, mientras que en la base de datos relacional la guarda en registros o filas, para una mejor comprensión se puede visualizar la figura 3.

Tabla comparativa de base de datos relacional y no relacional

SQL	MongoDB
Tabla	Conjunto
Fila	Documento
Columna	Campo
Clave principal	ObjectId
Índice	Índice
Ver	Ver
Tabla u objeto anidado	Documento incrustado
Matriz	Matriz

Figura 3: Tabla comparativa de las bases de datos SQL y MongoDB
 Tomada de: <https://aws.amazon.com/es/nosql/>

Las colecciones de MongoDB no necesitan llevar un esquema, ya que se puede diseñar el esquema en base a la necesidad tomando en cuenta la aplicación a utilizar.

1.5.1. Diseño de bases de datos documentales

Las bases de datos NoSQL, o no relacionales, no son diseñadas bajo el modelo entidad-relación puesto que no manejan entidades y las relaciones que manejan no tienen el mismo enfoque que las relaciones en las bases de datos SQL, por lo cual el autor Diego Sevilla en su artículo “Descubrimiento y visualización de esquemas de bases de datos NoSQL” propone estos diseños:

- Modelo y meta-modelo conceptual: en este modelo se visualiza las diferentes relaciones con los posibles conceptos de alto nivel, describiendo el dominio.

- Metamodelo lógico: presenta una visualización de los campos al ser utilizados asociados a un diagrama de clases.
- Modelo y meta-modelo físico: permite integrar el meta-modelo y el modelo lógico para visualizar su funcionamiento. (Sevilla, 2018)

1.6. Python

Python fue lanzado a finales de la década de 1980, por Guido van Rossum, como un nuevo lenguaje de programación interpretado y de alto nivel cuyo enfoque es programación orientado a objetos (POO), aunque también es compatible con programación estructurada. (Venners, 2003)

1.6.1. PEP 8

Python, al ser diseñado como un lenguaje de programación de alto nivel, propone una codificación mucho más ordenada, la cual sea sencilla de leer y comprender para cualquier desarrollador. Con base en esto se han realizado varias convenciones sobre las mejoras del lenguaje de programación, construyendo con el tiempo el documento de propuestas de mejora de Python (Python Enhancement Proposals), PEP por sus siglas en inglés, documento en el cual se describe el diseño, características o nuevos procesos de Python, a su vez está dividido en PEPs numeradas, de las cuales, cada una contiene información única que no se repite en las demás PEPs. De este modo el documento PEP 8 es el único que establece las mejores prácticas para la codificación en Python. (Python , 2001)

1.7. Framework Flask

El framework Flask es un marco de aplicación desarrollado en Python, que permite la integración del entorno web con aplicaciones desarrolladas en el mismo lenguaje, de manera sencilla y rápida permitiendo integrar librerías de terceros. Flask permite utilizar módulos y

librerías según la necesidad del desarrollador, ya que a diferencia de otros frameworks de Python no posee módulos y librerías ya predeterminados, lo que permite integrar solo lo necesario y tener control sobre el código utilizado. (Flask, 2010)

Capítulo 2

Análisis y diseño

2.1. Requerimientos

Para el diseño y construcción del prototipo se establecieron los siguientes requerimientos:

2.1.1. Requerimientos funcionales

El prototipo de chatbot debe:

- Responder correctamente a preguntas relacionadas con la cátedra de Ingeniería del Software.
- Presentar imágenes que aporten un valor explicativo para aquellas preguntas en las cuales no baste solo con el concepto.
- Responder que no entiende la pregunta, si no existe información relacionada a una pregunta realizada, simultáneamente se emite un correo electrónico al docente encargado del ambiente de administración indicando que no se encontró información sobre dicha pregunta.
- Ser informativo, ya que éste recibe preguntas y emite respuestas.
- Permitir almacenar la información relevante de la cátedra de Ingeniería del Software, la misma que debe estar almacenada en una base de datos no relacional.
- Registrar a los docentes que van a utilizar el ambiente de administración.
- Permitir realizar las funciones de leer, crear, editar y eliminar (funciones CRUD) información, a través del ambiente de administración del sistema.

2.1.2. Requerimientos no funcionales

- El prototipo está implementado sobre una página web y no dependerá de tecnologías de comunicación existentes (whatsapp, telegram, fb messenger).

2.2. Análisis del contenido del marco de conocimiento

En esta etapa se analizó el contenido que conforma el plan analítico de la cátedra de Ingeniería del Software, el cual consta de ocho capítulos conformados por varios subtemas los mismos que abordan el ciclo de vida del software.

Estos temas contienen la información relevante para la creación del marco de conocimiento el cual consta de dos ramas principales:

- La primera consta toda de la información analizada del plan analítico donde se obtuvieron los conceptos e imágenes que el chatbot utiliza para emitir sus respuestas.
- La segunda está conformada por todos los saludos y frases que responde el chatbot para dar sentido a la conversación.

La cátedra tiene una amplia bibliografía; sin embargo, su libro base es Ingeniería del Software, del autor Ian Sommerville, del que se han tomado los conceptos alineados al plan analítico, para alimentar a la base del conocimiento.

En la figura 4 se visualiza los contenidos del marco de conocimiento del chatbot de manera concisa, el cual está conformado en su mayoría por los temas concernientes al plan analítico descritos en el anexo 1 y por otra parte por los saludos y frases que responde el chatbot para simular una conversación.

Marco de Conocimiento del Chatbot

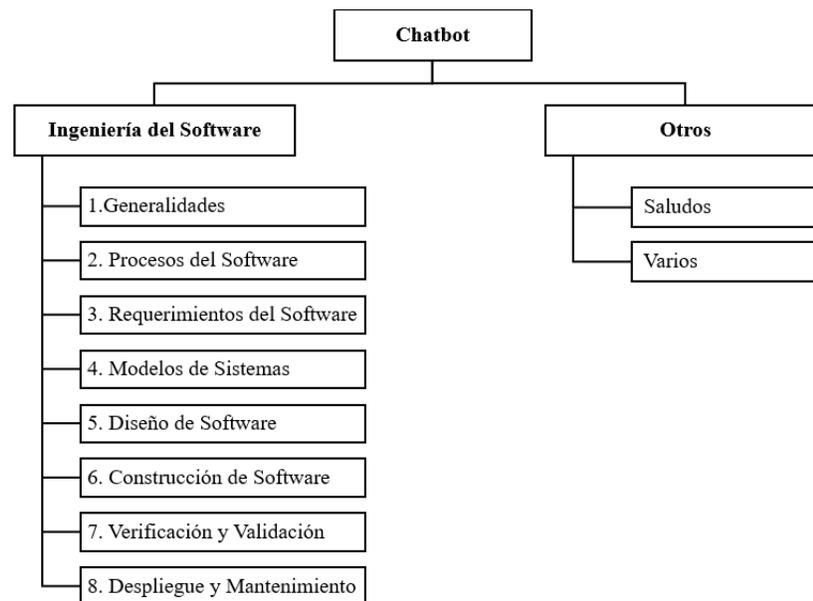


Figura 4: Marco de Conocimiento del Chatbot

Elaborada por: Deisy Auz

Se debe tomar en cuenta que el marco del conocimiento de los chatbots puede ser aplicable para cualquier materia, sin embargo, se debe destacar que los chatbots en el ámbito educativo deben estar orientados a un tema específico que responda preguntas concretas con conceptos básicos y sin entrar en temas prácticos o de operaciones.

2.3. Análisis técnico del tipo de chatbot

Tomando en cuenta los diferentes requerimientos se realizó un análisis en el que se establecieron los siguientes parámetros:

- Debe estar orientado a la cátedra de Ingeniería del Software.
- Debe responder a las preguntas más frecuentes relacionadas a Ingeniería del Software.

- El chatbot es explicativo pues proporciona la información referente a la pregunta ingresada y de ser necesario presenta imágenes relacionadas con la respuesta entregada.
- La información contenida en la base de conocimiento del chatbot proviene de los libros de la cátedra utilizados por los docentes.

Una vez finalizado el análisis se estableció que el tipo de chatbot informativo (FAQ) se adapta a las necesidades del proyecto, e involucra la utilización de librerías para el procesamiento del lenguaje natural y el procesamiento de datos, las mismas que permiten dar una respuesta acertada a la pregunta recibida.

2.4 Modelamiento

Con el fin de entender de mejor manera el análisis realizado se procedió a crear diagramas, los cuales describen de manera gráfica la arquitectura, el comportamiento y la transmisión de datos del sistema.

2.4.1. Casos de Uso

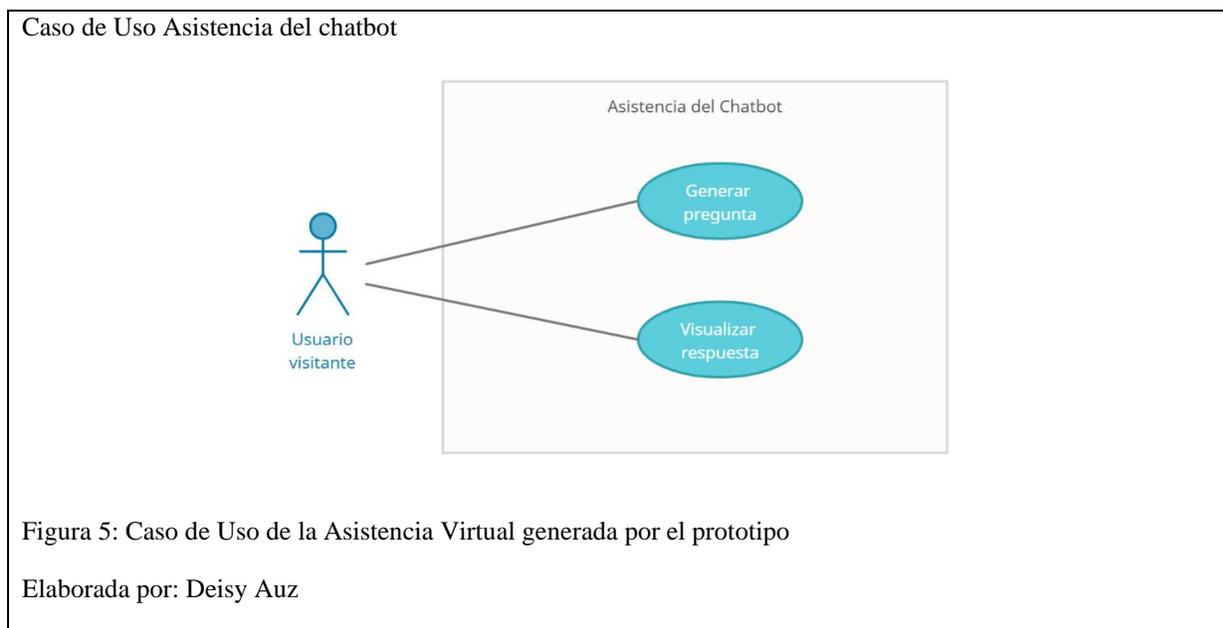
Para poder establecer los diferentes casos de uso con base en los requerimientos se realizó la delimitación de los usuarios que acceden al chatbot, tomando en cuenta que cada uno, puede realizar distintas acciones en el sistema, siendo estos:

- Usuario Visitante
- Usuario Administrador
- Usuario Docente

Se realizó una diferenciación entre el usuario administrador y el usuario docente, puesto que el usuario administrador puede asignar al usuario docente para que realice cambios en la base del conocimiento, mientras que el usuario administrador solo puede asignar los roles y no tiene permisos para ingresar información a la gestión de la base del conocimiento.

Usuario visitante

Es el usuario en general que utilizará el prototipo desde la página web. En la figura 5 se muestra la asistencia virtual que realiza el chatbot al momento que el usuario visitante realice una pregunta a través de la interfaz. El prototipo procesa la pregunta realizada y emite una respuesta la cual es visualizada por el usuario.



Usuario Administrador

El usuario administrador asigna los privilegios a las personas encargadas de administrar la base del conocimiento, pudiendo ser los docentes encargados de impartir la materia u otros. En la figura 6 se puede visualizar el caso de uso de administración de usuarios.

Caso de Uso Administración de Usuarios

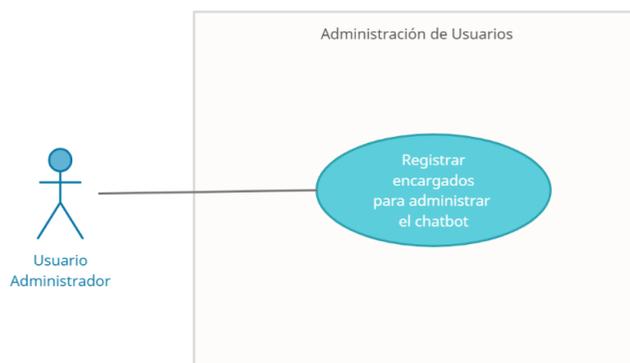


Figura 6: Caso de Uso administración de usuarios

Elaborada por: Deisy Auz

Usuario Docente

Este usuario es el encargado de administrar el ambiente del chatbot, para lo cual una precondición es iniciar sesión, desde la página web.

Caso de Uso del inicio de sesión del docente

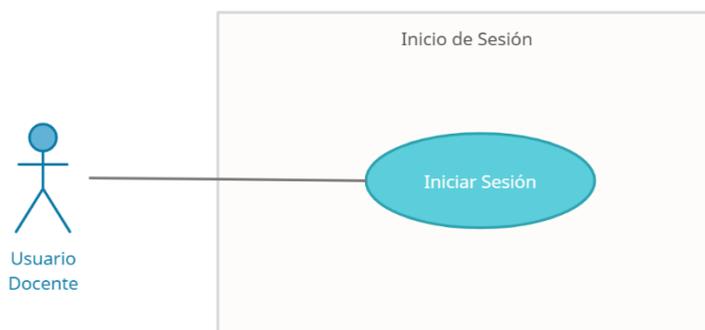
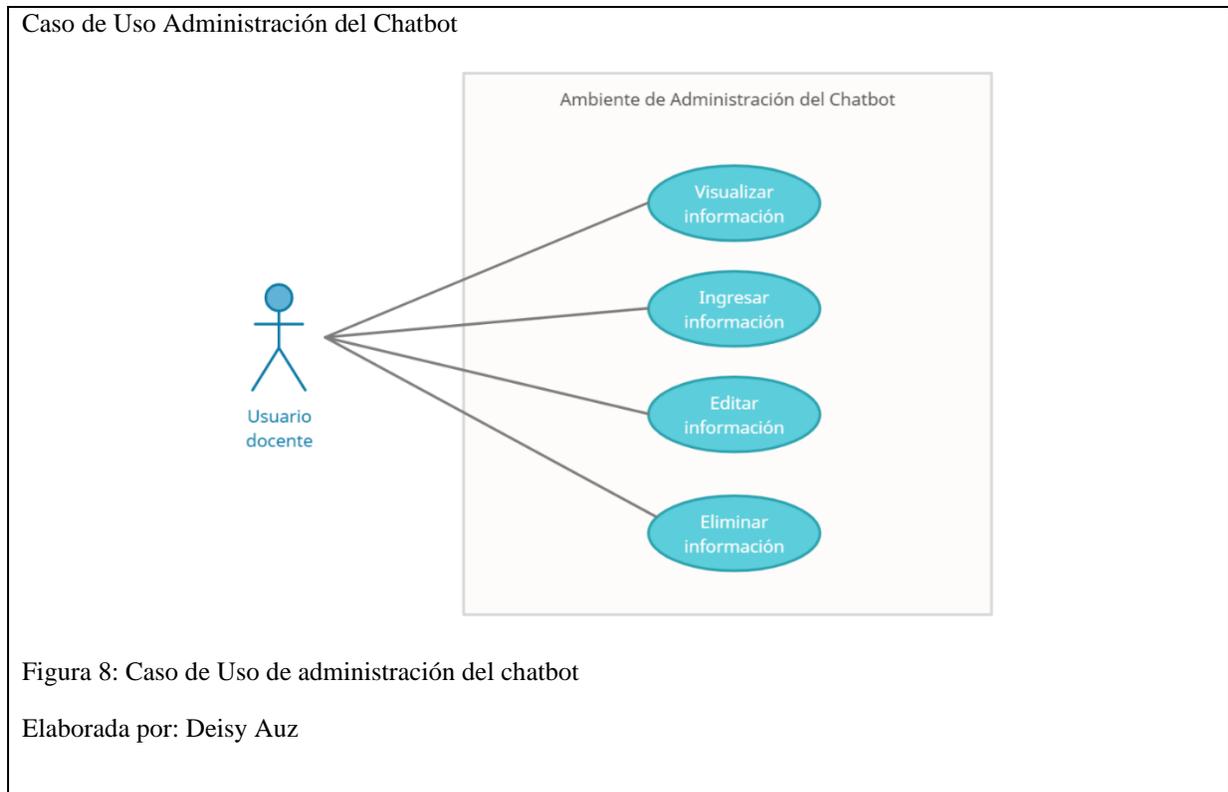


Figura 7: Caso de Uso del inicio de sesión del docente

Elaborada por: Deisy Auz

En la figura 7 se observa el caso de uso del inicio de sesión por parte del usuario docente.



En la figura 8 se muestra el caso de uso de la administración del chatbot, mediante el cual el docente es el encargado de administrar el prototipo de chatbot, el mismo que puede ingresar, leer, editar y eliminar información de la base del conocimiento.

2.4.2. Diagrama de despliegue del funcionamiento del prototipo

El usuario interactúa con el prototipo mediante el navegador web, quien realiza las peticiones mediante el protocolo TCP/IP hacia el servidor, en el cual se aloja el prototipo, e interpreta la información a través del Procesamiento del Lenguaje Natural y busca la información en la base de conocimiento; si la respuesta se encuentra, esta será emitida en la interfaz; caso contrario, si la respuesta no se encuentra, se emite un correo electrónico, mediante protocolo de correo simple (Simple Mail Transfer Protocol), SMTP por sus siglas en inglés, al docente encargado de la administración de la base del conocimiento. Esto se puede observar en la figura 9.

Diagrama de despliegue del funcionamiento del prototipo

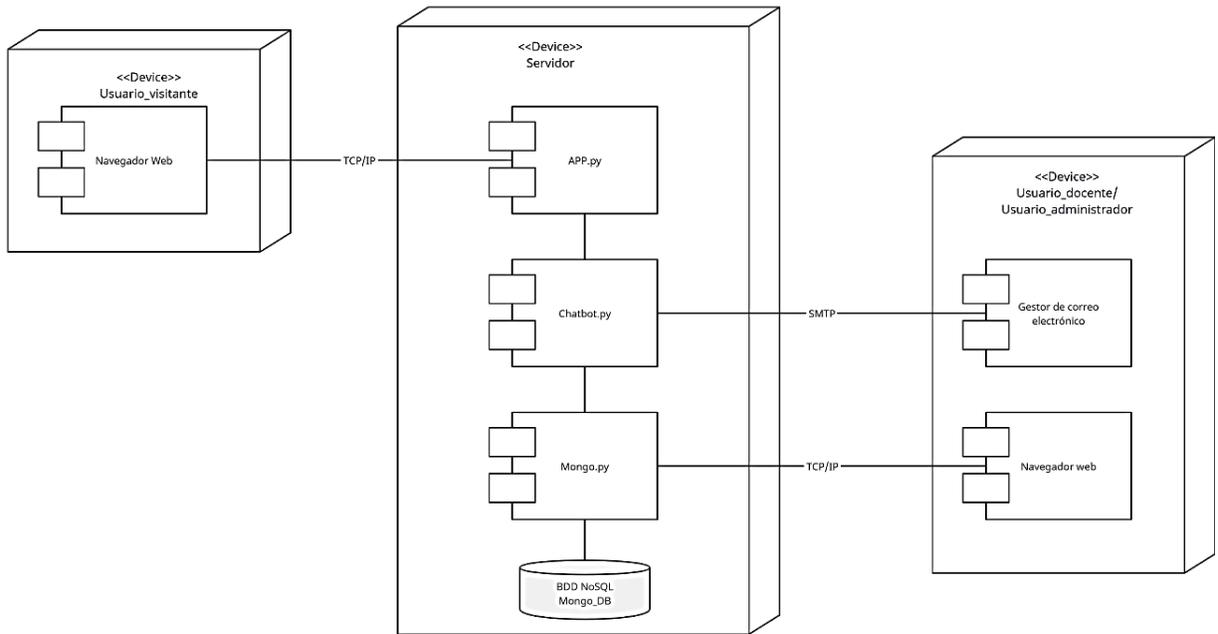


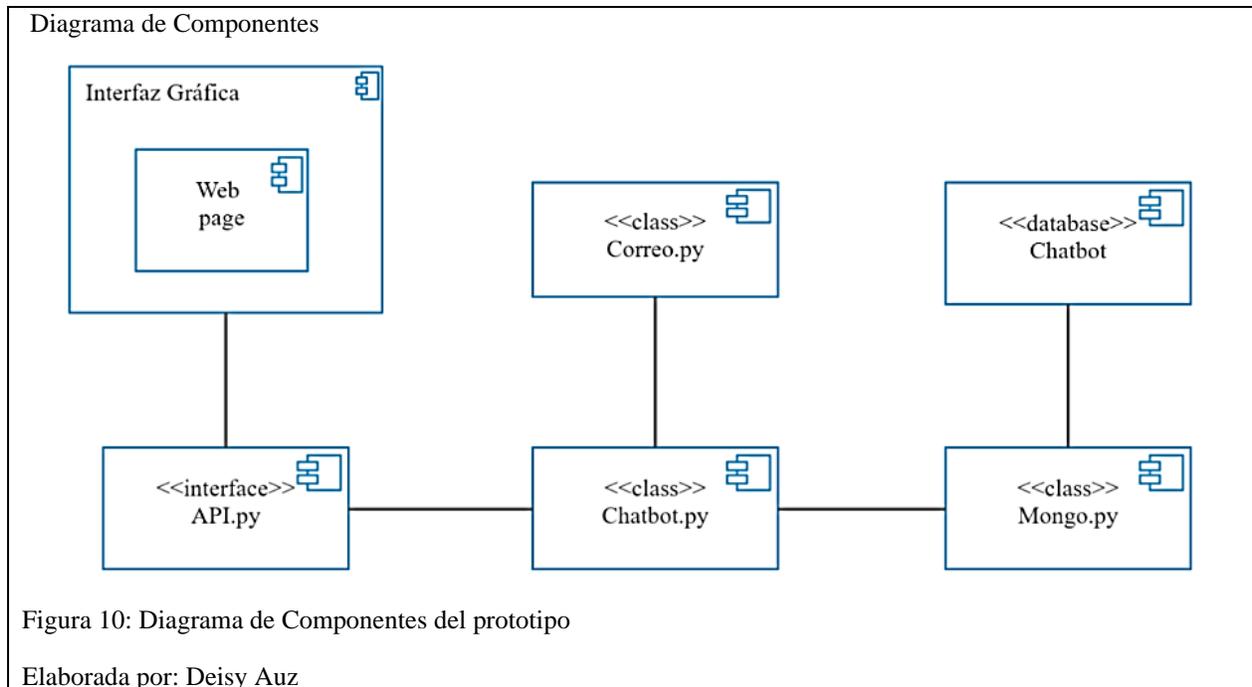
Figura 9: Diagrama de despliegue que representa el funcionamiento del prototipo

Elaborada por: Deisy Auz

2.4.3. Diagrama de componentes

Con base en el análisis del funcionamiento del chatbot se establecieron los componentes a utilizar, generando el diagrama que se visualiza en la figura 10, el mismo que está integrado por:

- La interfaz gráfica
- La API
- Las clases chatbot, correo y mongo,
- La base de datos chatbot



2.4.4. Diagrama del metamodelo conceptual de la base de datos

La base del conocimiento está alojada en una base de datos no relacional (NoSQL), la misma que fue implementada en MongoDB. Consta de la base de datos denominada chatbot, con una colección llamada textos, la cual incluyen los diferentes documentos en los que se aloja la información necesaria para que el prototipo pueda emitir una respuesta apropiada, además también se creó la colección saludos, donde se encuentran diferentes frases y saludos que responde el prototipo.

Diagrama del metamodelo conceptual de la colección textos

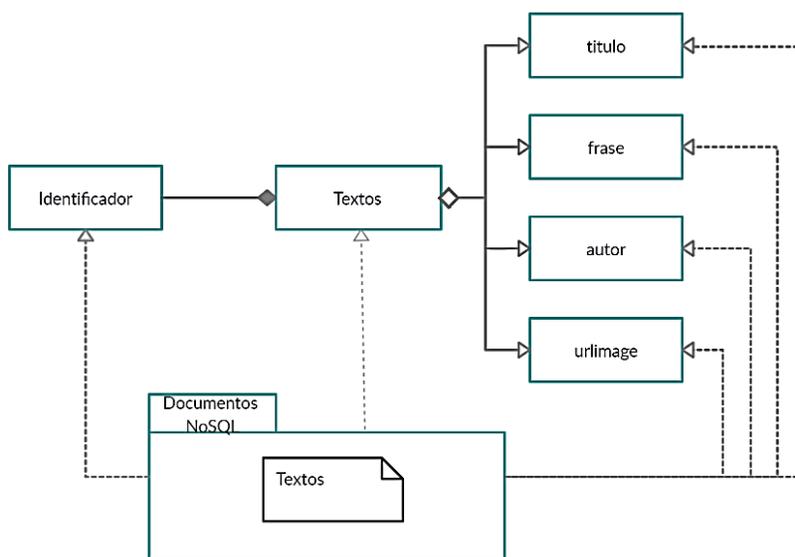


Figura 11: Metamodelo conceptual de la colección textos

Elaborada por: Deisy Auz.

En la figura 11 se visualiza el diagrama del metamodelo conceptual de la colección de textos perteneciente a la base de datos chatbot la misma que está conformada por los campos título donde se agregan las palabras clave, la frase que consta de los conceptos o descripción del tema referentes a la palabra clave, el autor campo que indica la fuente de donde se obtuvo la imagen y la urlimage que contiene la url de la imagen a mostrar.

2.4.5. Diagrama de clases

Con el fin de comunicar el diseño del prototipo se utilizó el diagrama de clases, el cual permite modelar las relaciones entre las diferentes entidades que involucran a la aplicación como se visualiza en la figura 12. El diagrama está formado por cuatro clases, de las cuales la clase principal es la clase Chatbot.

Diagrama de clases

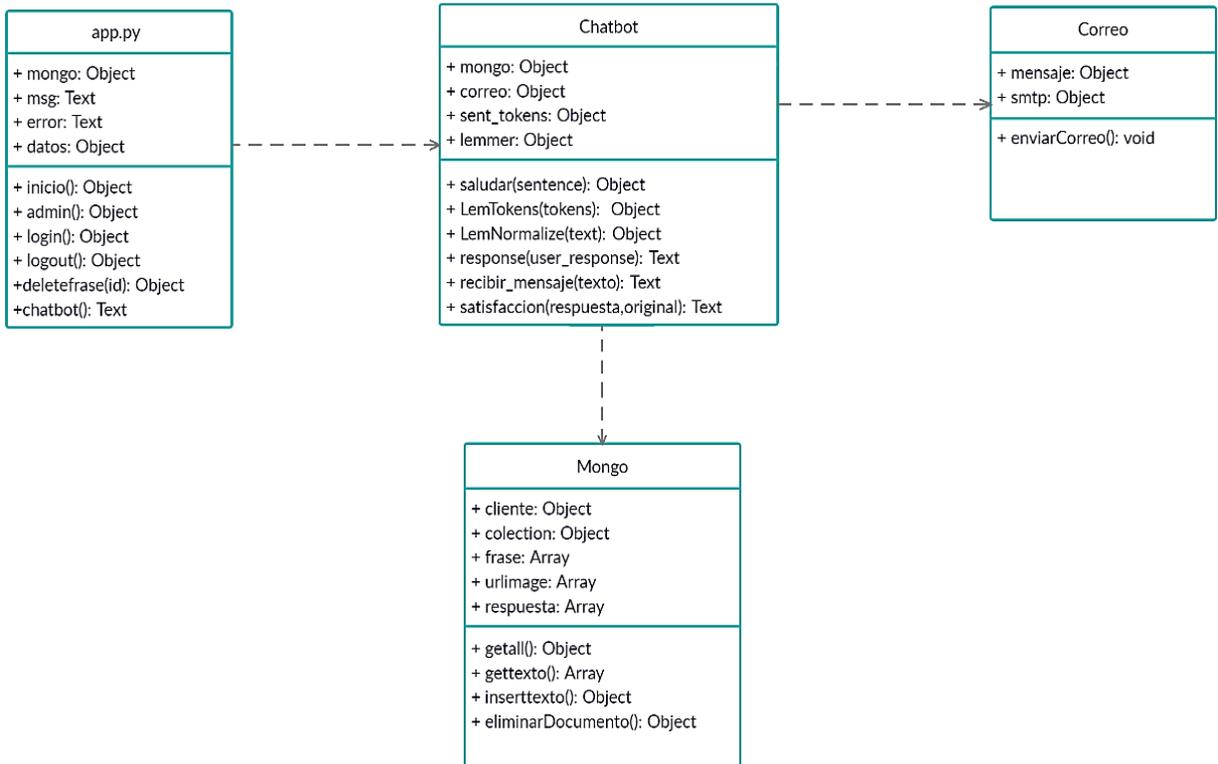


Figura 12: Diagrama de clases del prototipo

Elaborada por: Deisy Auz

2.4.6. Diagrama de Secuencia

La aplicación se adapta al patrón de arquitectura MVC, ya que el prototipo está separado en tres componentes donde la interfaz de usuario es la vista, la base de conocimiento es el modelo, y la capa lógica que incluye la API y la clase chatbot realizan las diferentes funciones del controlador.

Diagrama de secuencia de la arquitectura Modelo Vista Controlador

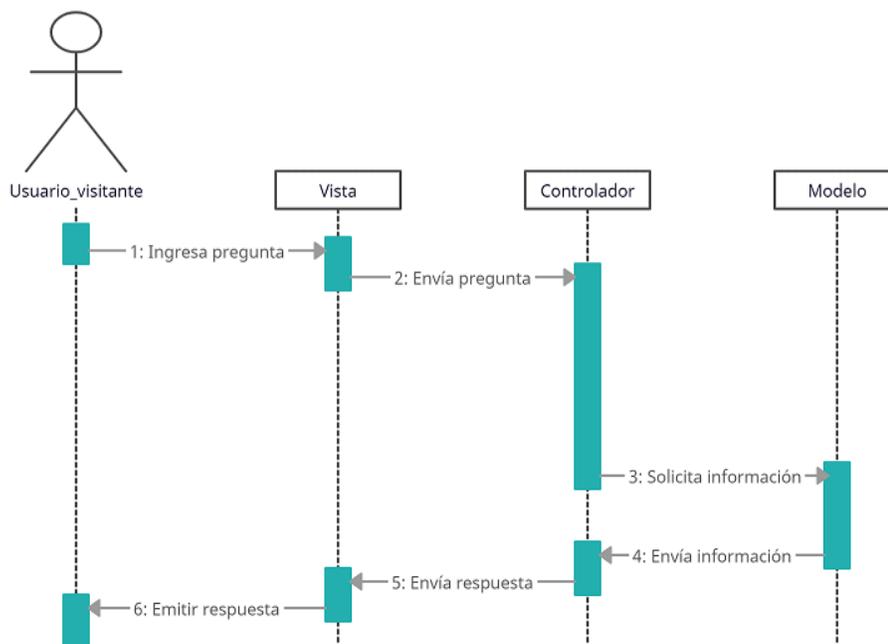


Figura 13: Diagrama de secuencia del patrón de Arquitectura MVC

Elaborada por: Deisy Auz

En la figura 13 se presenta la arquitectura MVC a través de un diagrama de secuencia el cual indica la interacción entre los diferentes componentes:

- En primera instancia se tiene al usuario visitante, en cuya línea de tiempo se inicia la secuencia al ingresar la pregunta por medio de la vista (interfaz web).
- El componente vista es activado al recibir la pregunta del usuario, luego la envía hacia el controlador (capa lógica).
- La capa lógica, o controlador, recibe la pregunta enviada por la vista y realiza el procesamiento necesario para luego solicitar información a la base de datos (modelo)
- El modelo (base de datos NoSQL) se activa al recibir la petición de información.

- El controlador recibe la información emitida por el modelo, la procesa y la envía como respuesta hacia la vista, la cual se encarga de presentarla al usuario.

Capítulo 3

Construcción

3.1. Buenas prácticas de codificación

Con base en la guía de estilo para el código Python PEP-0008 se siguieron los siguientes lineamientos en cuanto a la nomenclatura de funciones, clases, métodos, variables, etc.

- Las importaciones se colocaron al inicio del archivo, antes de la declaración de variables.
- Para el diseño del código se usó *hanging indent* que consiste en aplicar tabulaciones en todas las líneas, excepto la primera que lleva una indentación de cuatro espacios.
- Los comentarios son oraciones completas y su primera palabra comienza con una letra mayúscula.
- Los nombres de paquetes tienen un nombre corto y en minúsculas.
- Los nombres de clases utilizan la convención *CapWords* donde las palabras comienzan con mayúsculas.
- Los nombres de las variables globales, al igual que las funciones, son escritas con palabras separadas por un guión para una mejor legibilidad.
- Se utiliza la palabra reservada *self* en el primer argumento de los métodos de instancia.

3.2. Lineamientos para la construcción

Con base en los pasos propuestos por los autores Thukrul, Srivastava y Thakkar se establecieron los siguientes pasos a efectuar para la construcción del prototipo:

1. Seleccionar herramientas para Procesamiento de Lenguaje Natural (NLP)
2. Elegir librerías para el procesamiento de datos

3. Analizar librerías para la consulta ingresada
4. Devolver una respuesta

Para satisfacer cada paso propuesto se realizó una selección de tecnologías, para determinar cuál de estas se adapta mejor al requerimiento.

3.2.1. Selección de librerías para el Procesamiento del Lenguaje Natural

Existen diferentes maneras de crear un chatbot ya que estos son utilizados en diversos ámbitos, en la actualidad existen gran variedad de plataformas, herramientas y librerías, siendo estas gratuitas o pagas, también pueden encontrarse como servicios web en la nube.

Una limitante de las plataformas y herramientas para el desarrollo de chatbots es el código, ya que este no es compartido, por ende, no puede ser adaptado al trabajo a desarrollar, en la tabla 1 se analizan diferentes librerías de código abierto utilizadas para el Procesamiento del Lenguaje Natural, donde se definieron los siguientes campos:

- **Librerías:** Hace referencia a las librerías de código abierto que se puedan utilizar el lenguaje Python.
- **Tareas NPL:** Son las tareas que se utilizan en el Procesamiento del Lenguaje Natural.
- **Ventajas:** Características favorables que presenta la librería analizar.
- **Desventajas:** Características desfavorables que presenta la librería analizar.

Tabla 1: Tabla Comparativa de Librerías NLP

Librerías	Tareas de NLP	Ventajas	Desventajas
NLTK	Análisis Lingüístico Análisis de Sentimientos	Esta librería es la más completa y robusta ya que cuenta con varios recursos	No analiza de manera semántica los enunciados.

	<p>Análisis y Razonamiento</p> <p>Semántico</p> <p>Tokenización</p> <p>Clasificación</p> <p>Derivación</p> <p>Categorización</p> <p>Etiquetado</p>	<p>léxicos y una amplia documentación sobre su API.</p> <p>Tokenización rápida</p> <p>Admite mayor cantidad de idiomas</p>	<p>No cuenta con una vectorización de palabras integrado</p>
TextBlob	<p>Etiquetado</p> <p>Extracción de frases</p> <p>Análisis de sentimientos</p> <p>Clasificación</p> <p>Tokenización</p> <p>Frecuencia de palabras</p>	<p>Es menos compleja comparada con la librería NLTK.</p> <p>Trabaja mejor con textos cortos</p>	<p>No es una librería de NLTK pues se basa en esta para su construcción.</p> <p>No soporta el idioma español</p>
Gensim	<p>Busca similitud de documentos</p> <p>Transmisión de datos</p> <p>Representación de espacios vectoriales</p>	<p>Trabaja con extensos datasets y extensos flujos de datos</p> <p>Trabaja con base en la frecuencia de repetición de palabras</p>	<p>Diseñado para modelado de textos no supervisados</p> <p>No tiene suficientes herramientas para NLP</p>
SpaCy	<p>Tokenización</p> <p>Vectorización de palabras</p> <p>Segmentación de oraciones</p> <p>Reconocimiento de caracteres</p>	<p>Enfocado a modelos estadísticos</p> <p>Esta diseñado con base en POO</p> <p>Utiliza redes neuronales para entrenar a los modelos</p>	<p>Es menos flexible que NLTK</p> <p>La tokenización es más lenta que NLTK</p> <p>El soporte de lenguaje en español está en desarrollo.</p>

Nota: Tabla de Comparación de las librerías más conocidas para NLP

Elaborada por: Deisy Auz

Una vez analizadas las librerías más comunes para NLP y con base en las necesidades del prototipo se optó por utilizar la biblioteca NLTK ya que ésta es una librería de código abierto muy robusta la cual consta de librerías y paquetes de Python que permiten procesar el lenguaje natural y realizar procesamiento de textos por lo que es más flexible al momento de desarrollar un chatbot FAQ, además posee soporte en lenguaje español y cuenta con una API Rest.

3.2.2. Selección de librerías para el procesamiento de datos

Como todo chatbot este prototipo realiza procesamiento de datos, no se debe confundir con lo que realizan los gestores de base de datos, por lo que se procedió hacer una comparativa de las librerías de Python más utilizadas para este fin, como se muestra en la tabla 2 donde se tomó en cuenta los siguientes criterios:

- Librerías: Hace referencia a las librerías que realizan procesamiento de datos.
- Tareas: Son las tareas que se pueden realizar con dicha librería.
- Ventajas: Características favorables que presenta la librería.
- Desventajas: Características desfavorables que presenta la librería.

Tabla 2. Tabla comparativa de Librerías

Librerías	Tareas Involucradas	Ventajas	Desventajas
scikit-learn	Algoritmos de aprendizaje supervisado. Algoritmos de aprendizaje no supervisado Extracción de características Regresión Preprocesamiento	Fácil de usar Admite la mayoría de las tareas NLP Muy efectivo en el aprendizaje supervisado La preparación de datos no es exigente	No está enfocado al aprendizaje profundo. No tiene soporte para caracteres especiales.

	Clasificación Selección de modelo Reducción de dimensiones	Excelente opción para análisis de datos.	
TensorFlow	Técnicas de aprendizaje automático Análisis de Sentimientos Resumen de textos Extracción Clasificación Regresión Lineal Correlaciones	Fácil de usar, con pocos parámetros. Excelente en el aprendizaje profundo. Programación para redes neuronales profundas. Apta para una amplia gama de problemas neuronales.	Requiere de un elevado tiempo de procesamiento. No se puede determinar con exactitud cuantas neuronas y capas son necesarias para el aprendizaje. El aprendizaje profundo es lento y conlleva mucho tiempo.

Nota: Comparación de las librerías de Python para procesamiento de datos

Elaborada por: Deisy Auz

La librería sklearn cuenta con una API que permite la utilización de varias herramientas destinadas para el procesamiento de datos, además es compatible con otras librerías de Python, por lo que, se escogió esta librería tras la realización del análisis tomando en cuenta sus ventajas ya que permite realizar análisis de datos y se pueden realizar el mayor número de tareas NLP.

3.2.3. Selección de submódulos de la librería sklearn para la consulta ingresada

En esta etapa se debe extraer el contenido ingresado en el chatbot por el usuario, para lo cual se analizaron los submódulos de la librería sklearn que permitan realizar la extracción de características de texto. Como se visualiza en la tabla 3.

Tabla 3: Características de los submódulos de sklearn

Submódulos de la librería scikit-learn	Descripción
<code>feature_extraction.text.CountVectorizer</code>	Convierte una colección de documentos de texto en una matriz de recuentos de tokens
<code>feature_extraction.text.HashingVectorizer</code>	Convierte una colección de documentos de texto en una matriz de ocurrencias de tokens.
<code>feature_extraction.text.TfidfTransformer</code>	Transforma una matriz de recuento en una representación TF o TF-IDF normalizada
<code>feature_extraction.text.TfidfVectorizer</code>	Convierte una colección de documentos sin procesar en una matriz de funciones TF-IDF.

Nota: Descripción de los submódulos de la librería sklearn para la extracción de características de texto

Tomada de: <https://scikit-learn.org/stable/modules/classes.html>

Después de analizar los diferentes submódulos de la librería `scikit.learn` se seleccionó el submódulo `feature_extraction.text.TfidfVectorizer` para la extracción de características de texto ya que nos permite convertir el texto en números y convertirlo en una matriz de vectores a través de las funciones TF-IDF.

3.2.4. Selección del algoritmo para la emisión de la respuesta

Para que la respuesta sea acorde a la pregunta ingresada por el usuario se deben utilizar algoritmos que permiten dar una respuesta acertada, por lo que se analizó los siguientes algoritmos de similitud como se muestra en la tabla 4

Tabla 4: Comparación de algoritmos de similitud

Algoritmo	Categoría	Ventajas	Desventajas
Ratcliff Obershelp Similarity	Similitud basada en secuencia	Simplicidad Trabaja bien con textos cortos	No trabaja bien con textos largos. No toma en cuenta el significado semántico
Cosine Similarity	Similitud basada en tokens	Eficiencia Computacional Aplicable a textos largos Toma en cuenta el significado semántico	No es eficiente al comparar frases cortas.
Posffix Similarity	Simple	Eficiencia Computacional Trabaja bien en textos cortos.	Es un algoritmo muy primitivo Ineficiente para palabras largas. Necesita un preprocesamiento especial.
Monge Elkan	Híbrido	Toma en cuenta el significado semántico. Se puede utilizar para cualquier tipo de texto. Trabaja mejor en textos con errores ortográficos	Puede ser computacionalmente ineficiente para texto largo.

Nota: Tabla comparativa de los diferentes algoritmos de similitud usables para la emisión de respuesta.

Elaborada por: Deisy Auz.

Tomando en cuenta que el prototipo incluye una interacción con el usuario que involucra textos extensos, se concluyó que el mejor algoritmo para emitir una respuesta es Cosine Similarity ya que compara las consultas ingresadas con las respuestas existentes como puntos en el espacio (vectores), luego determina el coseno del ángulo para establecer el valor de similitud.

3.3. Arquitectura implementada del prototipo

El prototipo fue implementado tomando en cuenta el patrón arquitectónico modelo vista controlador el cual consta de:

- La base del conocimiento alojada en la base de datos no relacional MongoDB
- La capa lógica formada de la clase chatbot y API
- La interfaz de usuario y el ambiente de administración.

3.3.1. Construcción de la base de conocimiento

En primera instancia se procedió a crear la base de datos chatbot, que almacena la información de la cátedra, para ello con base en el análisis del contenido del marco de conocimiento se crean las colecciones saludos y textos.

Tabla 5: Campos del documento textos

Campo	Descripción
Id	Identificador principal del fichero
Titulo	En este campo se almacenan las palabras claves referentes a la pregunta de la cátedra de Ingeniería del Software.
Frase	Campo que guarda las respuestas correspondientes a las preguntas más frecuentes.
Autor	En este campo se ingresada la referencia de donde fue tomada una imagen, debido a derechos de autor.
urlimage	Campo que almacena las direcciones web de las imágenes referentes a las respuestas. Se muestra la url de la imagen debido a restricciones por derechos de autor.

Elaborada por: Deisy Auz

En la tabla 5 se describen los campos contenidos en los documentos pertenecientes a la colección textos de la base de datos chatbot.

Tabla 6: Campos del documento saludos

Campo	Descripción
Id	Identificador principal del fichero
Tipo	En este campo se indica si la información es de entrada o de salida
Frase	Campo que guarda las respuestas aleatorias a emitir el chatbot

Nota: Descripción de los campos contenidos en los documentos pertenecientes a la colección saludos.

Elaborada por: Deisy Auz

En la tabla 6 se describen los campos contenidos en los documentos pertenecientes a la colección saludos de la base de datos chatbot.

Marco de Conocimiento

El marco de conocimiento se generó por niveles tomando en cuenta el plan analítico de la materia. El nivel uno consta de las definiciones de los temas de cada capítulo, el nivel dos comprende de las definiciones y conceptos de los subtemas, el nivel tres está constituido por toda la información impartida en el contenido del capítulo, del que se han tomado las respuestas a las preguntas más frecuentes.

De acuerdo con el análisis del contenido de la cátedra, cada capítulo proporciona la información necesaria para responder a las preguntas frecuentes, por lo que se procedió analizar cada uno de manera detallada, comparando si la información de temas, subtemas, y el contenido que estos sujetan, luego se realizó la descripción de cada capítulo formando el marco del conocimiento como se visualiza en la figura 14.

De este marco se procedió a sacar las palabras clave y las respuestas a partir del nivel 1, las mismas que fueron ingresadas en la base de datos.

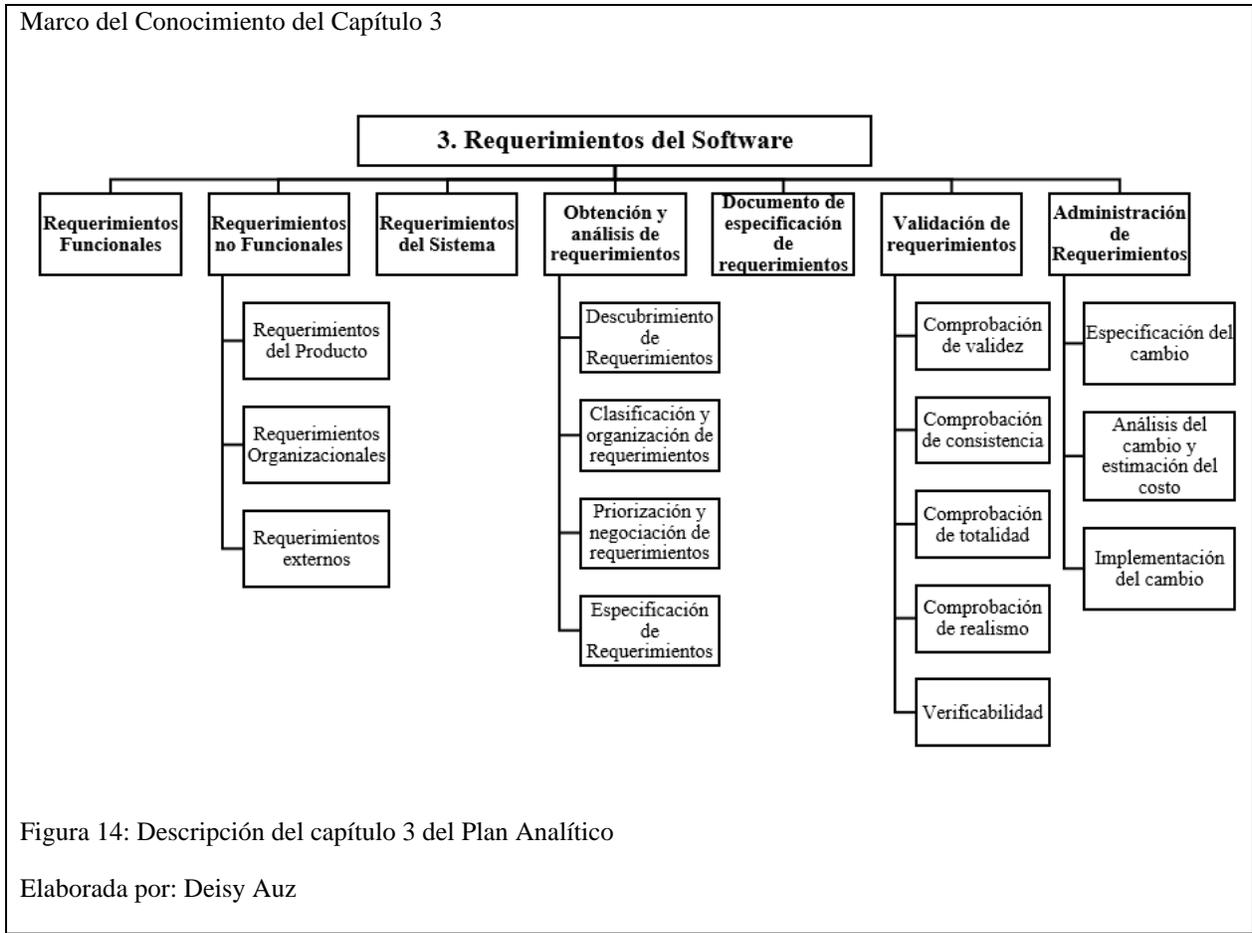


Figura 14: Descripción del capítulo 3 del Plan Analítico

Elaborada por: Deisy Auz

Luego se procedió a ingresar en la base de datos la información obtenida en el marco de conocimiento. Tras la realización de los pasos anteriores se obtiene la base de conocimiento ya que esta almacenada toda la información concerniente a la cátedra y las frases que el chatbot va a responder.

3.3.2. Construcción de la capa lógica

La capa lógica está constituida por dos partes principales que interactúan para la transferencia de datos entre la vista (capa de presentación) y el modelo (base del conocimiento), la primera parte conformada por la clase chatbot donde se integran las librerías y algoritmos analizados y la segunda constituida por la API.

3.3.2.1. Construcción de la clase chatbot

La clase chatbot, es la clase principal del prototipo. Esta construida para realizar las tareas de NLP, procesamiento de datos, análisis de la consulta ingresada del usuario y emisión de una respuesta acorde a la pregunta realizada. Estas tareas están establecidas en métodos, siendo el método más importante el response, en el cual se encuentran los algoritmos que permiten analizar la pregunta ingresada.

Espacio Vectorial y Similitud

En la figura 15 se observa el código implementado, donde el algoritmo *Vectorizer* extrae características del texto y las agrupa en una matriz, también llamada bolsa de palabras, se le llama así, ya que la información pierde la estructura de oración, de esta manera se eliminan las palabras duplicadas. Luego se importa el diccionario *stopwords* en idioma español de la librería de Python NLTK. El algoritmo realiza un índice de búsqueda en el corpus (conjunto de documentos) y devuelve una matriz de vectores. Luego el algoritmo de similitud *Cosine Similarity* encontrará la similitud a través de la comparación de los vectores entre la pregunta ingresada por el usuario y la base del conocimiento alojada en el base de datos.

Espacio Vectorial y Similitud

```
def response(self, user_response):
    self.sent_tokens.append(user_response)
    TfIdfVec = TfIdfVectorizer(tokenizer=self.LemNormalize, stop_words=stopwords.words('spanish'))
    tfid = TfIdfVec.fit_transform(self.sent_tokens)
    vals = cosine_similarity(tfid[-1], tfid) # Comparamos la pregunta del usuario, con el texto
    idx = vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfid = flat[-2]
```

Figura 15: Código para encontrar la similitud de las entradas y la base del conocimiento

Elaborada por: Deisy Auz

3.3.2.2. Construcción de la Interfaz de Procesos Automatizada (API)

Para la conexión entre la página web (vista) y el motor del chatbot (capa lógica) se desarrolló una API que permite la transferencia de datos entre estos dos componentes aplicando métodos propios del framework Flask, mismo que permite la construcción de aplicaciones web con lenguaje de programación Python.

Se creó la clase API.py donde se importa el paquete Flask y se inicializa la aplicación que gestiona el servicio a través de la ruta principal `@app.route('/')`, seguido de la función `inicio()` que permite gestionar solicitudes HTTP.

```
Construcción de la API

from flask import Flask, render_template, url_for, request, session
from werkzeug.utils import redirect
from chatbot import Chatbot
from mongo import Mongo

app = Flask(__name__)
app.secret_key = b'_Ch4Tb0t2o2o.##$'

@app.route('/')
def inicio():
    return render_template('index.html')
```

Figura 16: Código de implementación de la API

Elaborada por: Deisy Auz

En la figura 16 se puede visualizar el método `inicio()` que establece la conexión entre la página web `index.html` y la clase chatbot.

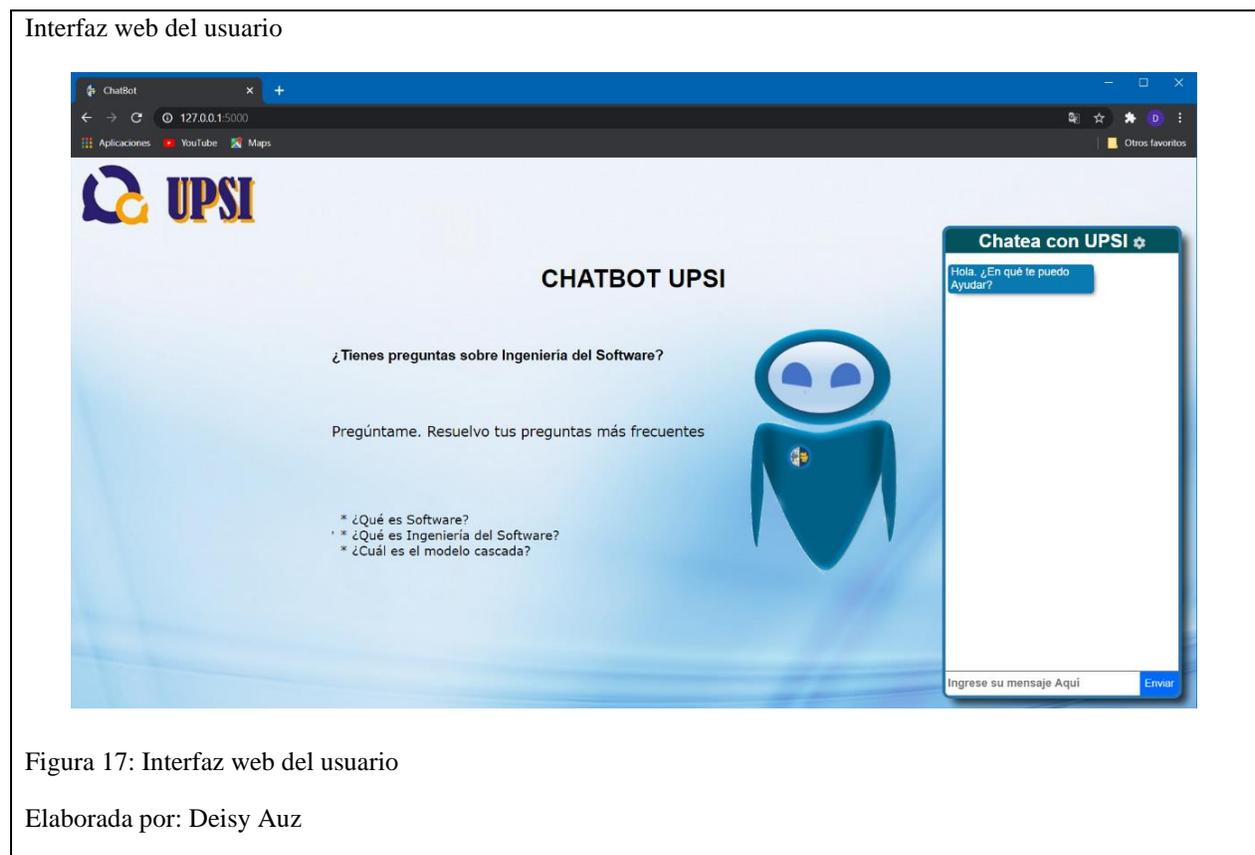
3.3.3. Construcción de la aplicación web

3.3.3.1. Creación de la interfaz web del usuario

Para la construcción de la interfaz web de usuario visitante, se utilizaron diferentes herramientas de desarrollo de páginas web. Estas herramientas son descritas a continuación:

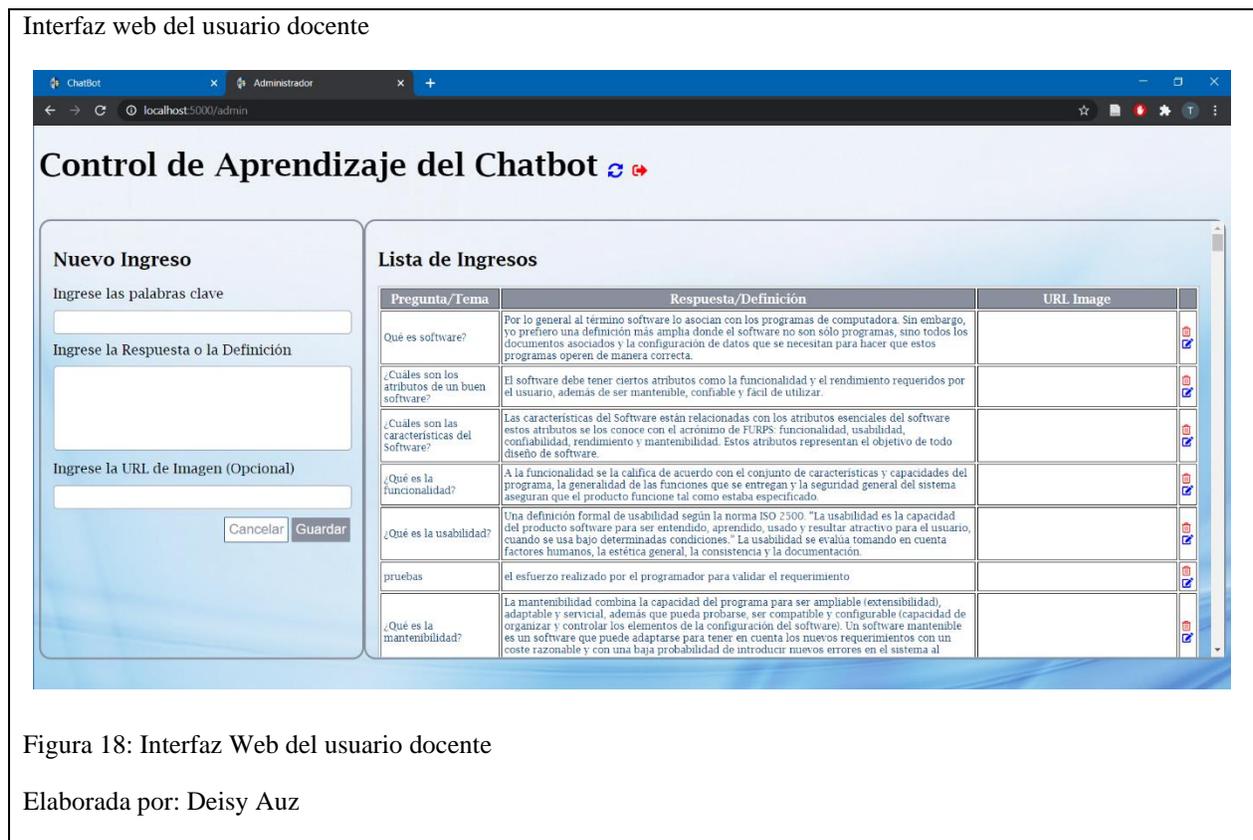
- HTML para la creación de la página web.
- Javascript para la transferencia de datos entre la vista y la capa lógica.
- CSS para el diseño y estilo de las páginas web.

Se realizó un desarrollo minimalista como se visualiza en la figura 17, el mismo que permite al usuario visitante una fácil utilización del prototipo, la ventana del chatbot se encuentra ubicado en la parte derecha de la página web, el cual recibe al usuario con un saludo.



3.3.3.2. Creación de la interfaz web del usuario docente

Para que el encargado de la administración de la base del conocimiento pueda acceder a la información, se procedió a crear una página que permita el inicio de sesión, al validar los datos correspondientes al usuario docente, se ingresa al ambiente de administración que se visualiza en la figura 18, donde el docente encargado puede realizar las funciones CRUD (crear, leer, actualizar y borrar) en la base del conocimiento.



3.3.3.3. Notificación vía correo electrónico

Para la construcción de la clase correo.py que tiene como función emitir un correo al docente encargado del ambiente de administración de la base del conocimiento, se procedió de la siguiente manera:

- Se importaron las librerías email y smtplib que permiten el envío de información bajo el protocolo de mensajería simple SMTP.
- Luego se hace la inicialización de constantes que permiten ingresar al correo.
- La emisión del correo se realiza a través del gestor de correo electrónico Outlook, ya que este permite la integración a este tipo de herramientas.
- A través de un texto plano se completan los campos para emitir el correo, donde se toma el mensaje emitido por la clase chatbot. Como se visualiza en la figura 19.

Clase correo.py

```

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from smtplib import SMTP

class Correo:
    def __init__(self):
        self.mensaje = MIMEMultipart("alternative")
        self.smtp = SMTP("smtp.live.com") # Outlook
        self.smtp.starttls()

    def enviarCorreo(self, mensaje):
        self.mensaje["From"] = "chatbot.2020@hotmail.com"
        self.mensaje["to"] = "chatbot.2020@hotmail.com"
        self.mensaje["subject"] = "RESPUESTA CHATBOT"

        # Mensaje en Texto Plano.
        msg = "Chatbot dice\n\n" + mensaje
        texto = MIMEText(msg, "plain")
        self.mensaje.attach(texto)

```

Figura 19: Código de implementación de la clase correo.py

Elaborada por: Deisy Auz

El prototipo al no presentar información dentro de la base del conocimiento emite un correo al docente encargado del ambiente de administración, indicando si una respuesta fue insatisfecha por parte del usuario, o si alguna pregunta realizada por el usuario no se encontró en la base del conocimiento.



En la figura 20 se puede observar el correo electrónico que se emite desde el chatbot, en el cual se indica que la respuesta fue insatisfecha por parte del usuario, indicando cual fue la pregunta realizada y la respuesta emitida.

3.4. Perspectiva Navegacional

La página web que contiene al prototipo presenta una interfaz gráfica simple y amigable para el usuario, en una estructura clara e intuitiva permitiendo ubicar los distintos componen de la página de manera rápida. El componente de interacción entre chatbot y usuario se encuentra ubicado en la parte derecha de la pantalla, y presenta un mensaje de saludo al usuario. El componente de texto del usuario es visible, presenta la indicación de que se debe escribir en él, además se visualiza el botón *Enviar* indicando que se debe presionar para entablar la conversación con el chatbot, sin embargo, para ingresar el mensaje también se puede presionar el botón *Enter*.

Para diferenciar entre los mensajes enviados por el usuario y los mensajes emitidos por el chatbot se han utilizado colores distintivos; para los mensajes del usuario se utilizó el color gris claro con letras negras y para los mensajes del chatbot en una gama del color azul con letras blancas.

Los mensajes emitidos por el chatbot son concretos y dan la información necesaria solicitada, en caso de que la respuesta amerite responder con una imagen para dar una explicación más clara, esta se visualizará de manera pequeña, al clickear sobre la misma, está se abrirá en una nueva página para poder observar de manera ampliada dicha información.

En la parte superior de la ventana del chatbot se encuentra una imagen de engranaje la cual direcciona al inicio de sesión del administrador de la base del conocimiento, al realizar la verificación de datos se ingresa al ambiente de administración donde se puede realizar las funciones CRUD, donde se puede ingresar las palabras claves, información relacionada a dichas palabras; de ser necesario, el autor dueño de la imagen y la url de la misma.

3.4.1. Código Relevante

Para poder realizar los diferentes cambios en la base del conocimiento se procedió a:

- Desplegar la información de la base del conocimiento en una tabla.
- Crear enlaces con iconos referentes a las actividades a realizar (editar o eliminar) en una columna de la tabla de información.
- Colocar *textbox* del lado izquierdo de la página web los mismos que permiten un fácil ingreso o edición de la información.

En la figura 21 se visualiza el código implementado en la interfaz `app.py` para la realización de funciones CRUD, por parte del docente encargado de la administración de la base del conocimiento. Para lo cual se toma la información de los *textbox* de la página web de la

administración de la base de conocimiento y se inserta a la base de datos MongoDB o a su vez se edita.

```
Segmento de código para la inserción y modificación de información en la base de conocimiento desde la interfaz web
```

```
if request.method == 'POST':
    documento = {'titulo': '', 'frase': '', 'urlimage': ''}
    if 'pregunta' in request.form and 'respuesta' in request.form:
        documento['titulo'] = request.form['pregunta']
        documento['frase'] = request.form['respuesta']
    if 'url_imagen' in request.form and len(request.form['url_imagen']) > 0:
        documento['urlimage'] = request.form['url_imagen']
    if('idpregunta' in request.form and request.form['idpregunta']):
        if mongo.updatetexto(documento,request.form['idpregunta']):
            msg = "Modificado correctamente"
        else:
            error = "No se pudo actualizar el valor correctamente"
    else:
        if mongo.inserttexto(documento):
            msg = "Ingresado correctamente"
        else:
            error = "No se pudo ingresar ningún valor"
```

Figura 21: Segmento de código para insertar y modificar información de la base de conocimiento desde la interfaz web

Elaborada por: Deisy Auz

Si se desea eliminar la información el docente encargado debe dar click en icono designado en el ambiente de administración y el método *deletefrase(id)* borrara la información de la base de datos, este método se puede observar en la sección de código de la figura 22.

```
Segmento de código para la eliminación de información de la base de conocimiento desde la interfaz web
```

```
@app.route('/api/delete-frase/<id>', methods=['GET', 'POST'])
def deletefrase(id):
    mongo = Mongo()
    mongo.eliminarDocumento(id)
    return redirect(url_for('admin'))
```

Figura 22: Segmento de código para la eliminación de información de la base de conocimiento desde la interfaz web

Elaborada por: Deisy Auz

Una vez que se han realizado las acciones funciones CRUD a nivel de la interfaz web del portal de administración, estas se deben replicar sobre la base de datos, para lo cual se han implementado distintos métodos que permiten realizar estas funciones.

En primera instancia se definieron los métodos para obtener el texto de la base de datos, para insertar un nuevo texto a la colección y para actualizar los textos ya ingresados, como se muestra en la sección de código de la figura 23.

Segmento de código de los métodos de conseguir texto, insertar texto y actualizar texto de la base de conocimiento

```
def gettexto(self):
    respuesta = []
    colection = self.dbname['textos']
    for frase in colection.find({}):
        try:
            respuesta.append(frase['titulo'])
        except KeyError:
            print("No está bien definido el documento de la colección.")
    return respuesta

def inserttexto(self, documento):
    colection = self.dbname['textos']
    return colection.insert_one(documento)

def updatetexto(self, documento, id):
    colection = self.dbname['textos']
    myquery = {'_id': ObjectId(id)}
    newvalues = {'$set': documento}
    return colection.update_one(myquery, newvalues)
```

Figura 23: Segmento de código de los métodos para conseguir, insertar y actualizar el texto de la base de conocimiento

Creada por: Deisy Auz

Capítulo 4 Pruebas

Al tratarse de un software que debe interactuar con el usuario, el prototipo debe ser evaluado en los aspectos de calidad del software, basados en la ISO 9126 y en la investigación “Evaluating Quality of Chatbots and Intelligent Conversational Agent” realizada por los autores Nicole Radziwill y Morgan Benton quienes establecen que al no existir un referente para realizar las pruebas a chatbots y agentes conversacionales se debe basar en la ISO 9126 y se adaptan a las necesidades a evaluar.(Radziwill & Benton, s. f.).

Tomando en cuenta lo antes mencionado se planteó el plan de pruebas (anexo 2) el mismo que se visualiza en la tabla 7 y consta de:

- Característica: hace referencia a las características descritas en la ISO 9126
- Métrica: medida que establece el estándar al que se desea llegar.
- Tipo de prueba: describe el nombre de la prueba a realizarse.
- Descripción de la prueba: describe los procesos a realizar para evaluar la prueba.
- Instrumento: el instrumento a utilizar vinculado al tipo de prueba.
- Casos/Escenarios: describe el número de casos en los que se ejecutará la prueba.

Tabla 7: Plan de Pruebas

Característica	Métrica	Tipo de Prueba	Descripción de la Prueba	Instrumento	Casos / Escenarios
Funcionalidad	Porcentaje de cubrimiento de funcionalidades, con un mínimo aceptable del 90%	Prueba Funcional	Se realizó una encuesta a los usuarios finales, sobre la funcionalidad del prototipo, donde se le indica al estudiante la variante a probar. Además, se realizaron diferentes casos prueba.	Matriz de resultados	5

Confiabilidad	Porcentaje de respuestas acertadas, obtenido al comparar la respuesta recibida con la respuesta esperada con un mínimo aceptable del 90%	Prueba Funcional	Se preguntó a los usuarios finales el número de respuestas correctas y se verificará que la respuesta que emite el prototipo concuerde con la respuesta esperada.	Matriz de resultados	5
Eficiencia	Promedio de los tiempos de respuesta de los diferentes escenarios con un tiempo máximo de 5s	Prueba de Rendimiento y Prueba de Estrés	Se midió el tiempo de respuesta del prototipo.	Matriz de resultados Herramienta JMeter	1
Usabilidad	Nivel de comprensión del usuario sobre el uso del prototipo, se espera que el nivel mínimo de usabilidad sea medio.	Prueba de Usabilidad	Se preguntó a los usuarios finales (estudiantes y docentes) el nivel de facilidad de uso del prototipo.	Cuestionario con preguntas cerradas	1
Portabilidad	Número de navegadores en los que el prototipo funciona correctamente, con un mínimo aceptable de tres navegadores	Prueba Comparativa	Se probó el prototipo en cuatro navegadores para conocer si en ellos están disponibles todas sus funcionalidades.	Matriz Comparativa	1

Elaborada por: Deisy Auz

Cálculo del tamaño de la Muestra

Es necesario establecer una muestra ya que las pruebas no se pueden realizar con todos los usuarios pertenecientes a la carrera, en este caso con 600 estudiantes, por lo que la muestra es representativa y permite realizar las pruebas con cierto número de estudiantes.

Para el cálculo del tamaño de la muestra se lo realizó a partir de la siguiente fórmula estadística:

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

Donde:

n=Tamaño de la muestra a buscar

N= Tamaño de la población (Estudiantes de la carrera en Ciencias de la Computación) con un total de 600 estudiantes

Z=Parámetro estadístico que depende del Nivel de Confianza (NC = 90%)

e=La cantidad de error de muestreo aleatorio resultado de la elaboración de la investigación establecido en 12%

p=Probabilidad de que ocurra el evento(éxito)

q=Probabilidad de que no ocurra el evento

De la cual se obtuvo un total de muestra de 44 estudiantes pertenecientes a la carrera en Ciencias de la Computación.

4.1. Pruebas funcionales

Para la realizar las pruebas funcionales se tomó en cuenta los requerimientos y el plan de pruebas establecidos, fijando dos escenarios:

- Con el perfil usuario visitante, se ingresa a la interfaz web del prototipo y se realizan preguntas.
- Con el perfil usuario docente, se verificó el inició de sesión y el ingreso al ambiente de administración de la base de conocimiento del sistema.

Resultados

En la Tabla 8 se muestra un resumen de los resultados obtenidos tras realizar las pruebas funcionales, en los cuatro casos planteados, las mismas que después de ser realizadas se compararon con las métricas establecidas en el plan de pruebas llegando a la conclusión de que cumplen con lo esperado por lo cual es aprobado cada caso planteado.

Tabla 8: Resultados Pruebas Funcionales

Pruebas Funcionales	Repeticiones	Estado
Caso1: Verificar el correcto funcionamiento del prototipo	1120	Aprobado
Caso2: Inicio de sesión del docente	5	Aprobado
Caso 3: Verificar el correcto funcionamiento para funciones CRUD solo con texto	10	Aprobado
Caso 4: Verificar el correcto funcionamiento para funciones CRUD con imágenes	10	Aprobado

Elaborada por: Deisy Auz

Para el caso 1 se plantearon dos escenarios:

- Escenario 1: Realizar pruebas de funcionalidad con los 44 estudiantes seleccionados de la muestra
- Escenario 2: Realizar pruebas de funcionalidad tomando en cuenta un concepto ingresado en la base de datos y compararlo con la respuesta emitida

En el Escenario 1, los estudiantes de la muestra calificaron al chatbot según su experiencia en cuanto a la funcionalidad del prototipo, para lo cual llenaron una encuesta en línea (anexo 3), dando como resultado que el prototipo emite respuestas a las preguntas planteadas en un 99% de las ocasiones, como se muestra en la figura 24, esto basado en la matriz de resultados del anexo 4.

Resultados de las pruebas funcionales caso 1



Figura 24: Resultados de las pruebas de funcionales

Elaborada por: Deisy Auz

En el escenario 2, se procedió a preguntar de diferentes maneras esperando a recibir la misma respuesta. Se seleccionó la pregunta referente al Modelo Cascada ingresado en la base del conocimiento como se visualiza en la figura 25.

Imagen de la Base de Conocimiento

Palabras Clave	Definición / Respuesta	Derechos de Autor
modelo cascada	El modelo cascada, sugiere un enfoque sistemático y secuencial, disciplinado y basado en análisis, diseño, pruebas y mantenimiento. Al final de cada etapa se reúnen y revisan los documentos para garantizar que se cumplen los requerimientos antes de avanzar a la fase siguiente. Pionero en guiar el proceso de desarrollo de software dirigido por un plan, introduciendo una planificación de casa fase antes de empezar a trabajar en ella.	Imagen obtenida de: http://imgtz.com/i/QV3Xq76.jpeg

Figura 25: Imagen de la base de conocimiento del Modelo Cascada

Elaborada por: Deisy Auz

En la tabla 9 se observan las diferentes preguntas realizadas al chatbot respecto a una misma pregunta en este caso sobre el Modelo Cascada.

Tabla 9: Resultados de las pruebas funcionales planteadas en el escenario 2

Preguntas ingresadas por el usuario	¿Emitió la respuesta esperada?	Observaciones
Háblame sobre la metodología cascada	Si	El prototipo con la respuesta esperada
Sabes algo sobre cascada	Si	El prototipo con la respuesta esperada
Cascada es:	Si	El prototipo con la respuesta esperada
Tienes conocimiento del modelo cascada	Si	El prototipo con la respuesta esperada
¿Qué es cascada?	Si	El prototipo con la respuesta esperada
Quiero saber sobre el modelo de desarrollo cascada	Si	El prototipo con la respuesta esperada
Me puedes decir que es la metodología ágil cascada	Si	El prototipo con la respuesta esperada
¿Cuál es el modelo cascada?	Si	El prototipo con la respuesta esperada
Que sabes de los pasos de cascada	Si	Emitió el concepto del modelo Cascada ya que en la información de la base de conocimiento no tiene definida la secuencia que sigue dicho modelo.
cascada	Si	El prototipo con la respuesta esperada

Elaborada por: Deisy Auz

Con base en la prueba realizada en el escenario 2, se determinó que el prototipo funciona de manera correcta, ya que los algoritmos de similitud comparan la pregunta ingresada, con la base del conocimiento, emitiendo la respuesta esperada.

En el caso 2 se verificó el correcto inicio de sesión del perfil docente, corroborando que no se pueda ingresar con usuarios o contraseñas diferentes a las registradas, de lo que se obtuvo la matriz de resultados indicada en la tabla 10.

Tabla 10: Matriz de resultados pruebas funcionales caso 2

Matriz de Resultados				
Objetivo: Verificar el inicio de sesión del docente				
Característica: Funcionalidad				
Caso: 2				
Variante	Ingreso de Credenciales	Respuesta esperada	Respuesta obtenida	Observación
1	Se ingresa una correcta clave y contraseña	Correcto inicio de sesión	Correcto inicio de sesión	NA
2	Se ingresa una correcta clave e incorrecta contraseña	No se autoriza el inicio de sesión	No se autoriza el inicio de sesión	NA
3	Se ingresa una incorrecta clave y correcta contraseña	No se autoriza el inicio de sesión	No se autoriza el inicio de sesión	NA
4	Se ingresa una incorrecta clave y contraseña	No se autoriza el inicio de sesión	No se autoriza el inicio de sesión	NA

Elaborada por: Deisy Auz

En la tabla 11 se muestra la matriz de resultados de las funciones CRUD realizadas en el ambiente de administración de la base del conocimiento para el caso 3.

Tabla 11: Matriz de resultados pruebas funcionales caso 3

Matriz de Resultados					
Objetivo: Verificar la realización de funciones CRUD de las palabras claves y contenido realizado por el docente en el ambiente de administración					
Característica: Funcionalidad					
Caso: 3					
Escenario	Variante	Información Ingresada	Respuesta esperada	Respuesta obtenida	Observación
1	1	Se ingresa las palabras claves y el contenido	Se guarda la información	Se guarda la información	NA
	2	Se ingresa solo una palabra clave	Emite mensaje de alerta	Emite mensaje de alerta	NA
	3	Se ingresa solo un contenido	Emite mensaje de alerta	Emite mensaje de alerta	NA
2	1	Se edita solo en las palabras clave	Se edita la información	Se edita la información	NA
	2	Se edita solo en el contenido	Se edita la información	Se edita la información	NA
3	1	Se elimina la información ingresada	Se elimina la información	Se elimina la información	NA

Elaborada por: Deisy Auz

En la tabla 12 se muestra la matriz de resultados de las funciones CRUD realizadas en el ambiente de administración de la base del conocimiento para el caso 4.

Tabla 12: Matriz de Resultados de las pruebas funcionales caso 4

Matriz de Resultados					
Objetivo: Verificar la realización de funciones CRUD de las palabras claves, contenido e imágenes realizado por el docente en el ambiente de administración					
Característica: Funcionalidad					
Caso: 4					
Escenario	Variante	Información Ingresada	Respuesta esperada	Respuesta obtenida	Observación
1	1	Se ingresa las palabras claves, el contenido e imagen	Se guarda la información	Se guarda la información	NA
	2	Se ingresa solo una imagen	Emite mensaje de alerta	Emite mensaje de alerta	NA
2	1	Se edita la imagen	Se edita la información	Se edita la información	NA
3	1	Se elimina la información ingresada	Se elimina la información	Se elimina la información	NA

4.2. Pruebas de confiabilidad

Se procedió a realizar las pruebas de confiabilidad, descritas en el plan de pruebas, para medir el nivel de confiabilidad de las respuestas emitidas por el prototipo.

Se establecieron dos escenarios para la realización de las pruebas de confiabilidad, como se puede ver en la tabla 13.

Tabla 13: Escenario de pruebas de confiabilidad

Escenario	Tipo de Prueba	Preguntas realizadas
Ambiente controlado	Prueba Funcional	1100
Ambiente no controlado	Pruebas en Caliente	20

Elaborada por: Deisy Auz

En el ambiente controlado se procedió a realizar la prueba con los estudiantes establecidos en la muestra, a los cuales se les indico que caso prueba se iba a realizar y la variante que se debía

ingresar en el chat, como se muestra en la tabla 14. Se definieron diferentes casos pruebas entre ellos errores ortográficos o de puntuación para validar la confiabilidad de las respuestas a través de los algoritmos utilizados.

Tabla 14: Casos prueba

Casos Prueba	Variante Prueba
Se ingresa una palabra clave relacionada con la pregunta	software
	Proceso
	validación
	espiral
	PUA
Se ingresan dos palabras claves relacionadas con la pregunta	modelo cascada
	Diagrama clases
	PRUEBAS ALFA
	Mantenimiento Preventivo
	sistema Heredado
Se ingresa la pregunta con signos de interrogación y tildes	¿Qué es ingeniería del software?
	¿Qué es SCRUM?
	¿Qué es vyv del software?
	Para qué se usa un prototipo?
	¿Cuáles son las características del Software?
Se ingresa una pregunta con faltas ortográficas	que son los Sprints
	cuales son los modelos ebolutibos
	¿Qué son los sistemas eredados?
	analicis del sistema
	Modelos Agiles
Se ingresa una frase informal relacionada a la pregunta	me puedes decir algo sobre el proceso de software
	que sabes sobre la ingeniería inversa
	Me dices algo de UML
	conoces sobre el diagrama de casos de uso
	que hay de las pruebas de usuario

Elaborada por: Deisy Auz

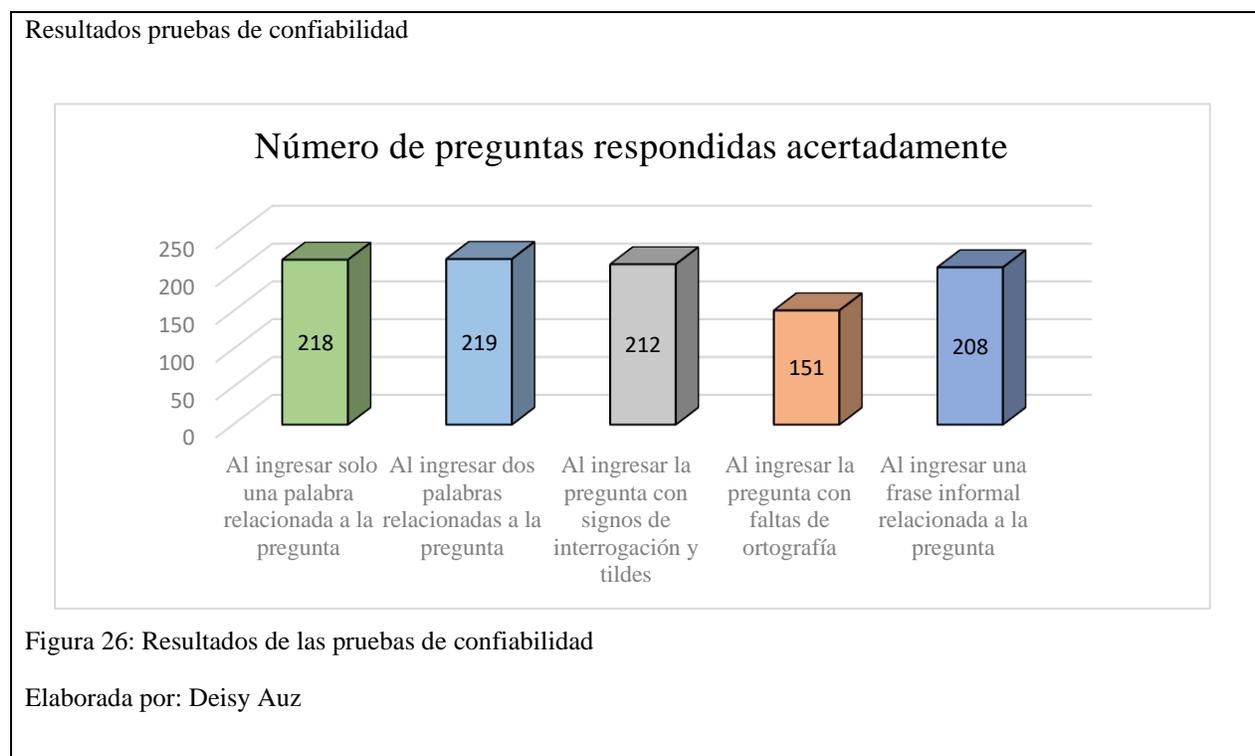
En el ambiente no controlado se procedió a realizar las pruebas con el grupo de estudiantes inscritos en la cátedra de Ingeniería del Software de la carrera en Ciencias de la Computación, quienes

ingresaron preguntas referentes a la materia tomando en cuenta su conocimiento adquirido en el transcurso del periodo académico.

Resultados

Escenario 1

En el anexo 6 se presenta la matriz de resultados obtenidos a través de la encuesta en línea (anexo 5) realizada a la muestra de estudiantes que participaron en la ejecución de las pruebas de confiabilidad, en un ambiente controlado, ya que se les indicó que preguntas ingresar tomando en cuenta los casos de usos planteados.



En la figura 26 se resume el número de preguntas respondidas de manera correcta en función de los casos de prueba.

En el caso de prueba al ingresar la pregunta con faltas de ortografía, se obtuvieron 151 respuestas acertadas de un total de 220 preguntas realizadas. Con esto se puede concluir que las palabras claves, que llevan tilde, deben estar escritas de manera correcta, para que el prototipo pueda responder acertadamente, puesto a que el algoritmo *Coseno Similarity*, encargado de la emisión de respuestas, toma en cuenta el contexto semántico.

El prototipo respondió con la información correspondiente a la pregunta 1008 preguntas de un total de 1100 preguntas realizadas por los cuarenta y cuatro estudiantes tomados en la muestra, dando como resultado un 92% de confiabilidad al momento de emitir una respuesta, como se observa en la figura 27.



Escenario 2

La matriz de resultados de las pruebas del escenario 2 se pueden ver en la tabla 15, en la cual se verifica el uso del prototipo por parte de los estudiantes, quienes ingresaron preguntas de manera formal e informal, con faltas de ortografía entre otros.

Tabla 15: Resultados de las pruebas de confiabilidad escenario 2

Frase escrita por el usuario	Respuesta Acertada (Si/No)	Observaciones
Hola	Si	El prototipo respondió al saludo ingresado.
Qué es la metodología SCRUM	Si	Se realizó una pregunta formal y la respuesta fue acertada
Que es el modelo en cascada	Si	Se ingresó una pregunta formal sin tilde y sin signos de pregunta. La respuesta fue acertada
Sabes sobre el modelo de contexto	Si	Se ingresó una pregunta informal sin tilde y sin signos de pregunta. La respuesta fue acertada
Sobre el diagrama de clases	Si	Se realizó una pregunta sin signos de puntuación y la respuesta fue acertada
Saves sobre el diagrama de dominio	Si	La respuesta fue acertada, el prototipo emitió una respuesta coherente al usuario. Ya que no hay información sobre este tema en la base del conocimiento
Sobre el diagrama de claces	No	Se realizó una pregunta con una falta de ortografía en una de las palabras claves de la búsqueda y la respuesta fue errónea
Que es ingenieria del software	Si	Se realizó una pregunta con una falta de ortografía en una de las palabras claves de la búsqueda y la respuesta fue acertada.
Modelos evolutibos	Si	Se realizó una pregunta con faltas de ortografía y la respuesta fue acertada.
mija cruzate el de diagrama de secuencia pilas	Si	Se ingresó una frase coloquial sobre una pregunta y la respuesta fue acertada
crees que la metologia Scrum es mejor que la metodologia xp	No	Se realizó una pregunta comparativa sobre dos metodologías, el prototipo emitió la respuesta de la primera metodología.
metodologa xp	Si	Se realizó una pregunta con una falta de ortografía en una de las palabras claves de la búsqueda y la respuesta fue acertada.
confiabilidad	Si	Se ingresó una pregunta con una sola palabra y la respuesta fue acertada.
modelocascada	No	Se ingresó una pregunta sin espacios y la respuesta fue errónea.

Modelo cascada	Si	Se ingresó una pregunta con varios espacios y la respuesta fue acertada
xp	Si	Se ingresó una pregunta con una palabra la respuesta fue acertada
Que es ingenieria	No	La respuesta no fue acertada respondió sobre ingeniería inversa ya que el concepto de ingeniería no se encuentra en la base del conocimiento
Sabes de diagramas	Si	Se ingresó una pregunta informal la respuesta fue acertada
buenas	Si	El prototipo respondió al saludo ingresado
chatbot sabes sobre los lenguajes de programación	Si	Se ingresó una pregunta informal la respuesta fue acertada

Elaborada por: Deisy Auz

El resultado de las pruebas en caliente las mismas que se ejecutaron con usuarios al azar y datos reales, en donde a los estudiantes no se les indicó que palabras ingresar ni cómo utilizar el chatbot, de donde se obtuvo que responde de manera acertada un 90%.

4.3. Pruebas de eficiencia

Para las pruebas de eficiencia se utilizó la herramienta JMeter la cual permite tomar el tiempo de respuesta por cada petición. Con base en el tiempo obtenido se armó la tabla 16.

Tabla 16: Resultados pruebas de eficiencia

Matriz de Análisis		
Objetivo:	Medir el tiempo de respuesta del prototipo	
Característica:	Eficiencia	
Variante	Pregunta Realizada	Tiempo de Respuesta
1	Software	1.84s
2	Proceso	1.64s
3	Validación	1.60s
4	Espiral	1.39s
5	PUA	1.60s

1	modelo cascada	1.64s
2	Diagrama Clases	1.88s
3	pruebas alfa	1.75s
4	mantenimiento preventivo	2.11s
5	sistema heredado	1.46s
1	¿Qué es ingeniería del software?	1.47s
2	¿Qué es Scrum?	1.66s
3	¿Qué es VyV del software	1.53s
4	Para qué se usa un prototipo?	1.44s
5	¿Cuáles son las características del Software?	1.52s
1	Que son los Sprints	1.66s
2	cuales son los modelos ebolutibos	1.54s
3	analicis del sistema	1.51s
4	¿Qué son los sistemas eredados?	1.86s
5	modelos agiles	1.47s
1	me puedes decir algo sobre el proceso de software	2.02s
2	que sabes sobre la ingeniería inversa	1.76s
3	me dices el significado de UML	1.39s
4	me ayudas con un diagrama de casos de uso	1.60s
5	que hay de las pruebas de usuario	1.64s

Elaborada por: Deisy Auz

El promedio de tiempo de respuesta es de 1,64s siendo este un tiempo de respuesta adecuado según lo establecido en la investigación “Chatbot Testing: Challenges and approach for testing Chatbot applications” de los autores (Singh & Jain, 2019).

Con la misma herramienta Jmeter, se realizaron las pruebas de estrés, con los siguientes parámetros:

- No de peticiones: 3
- No de usuarios: 100

- Aumento de usuarios: 100

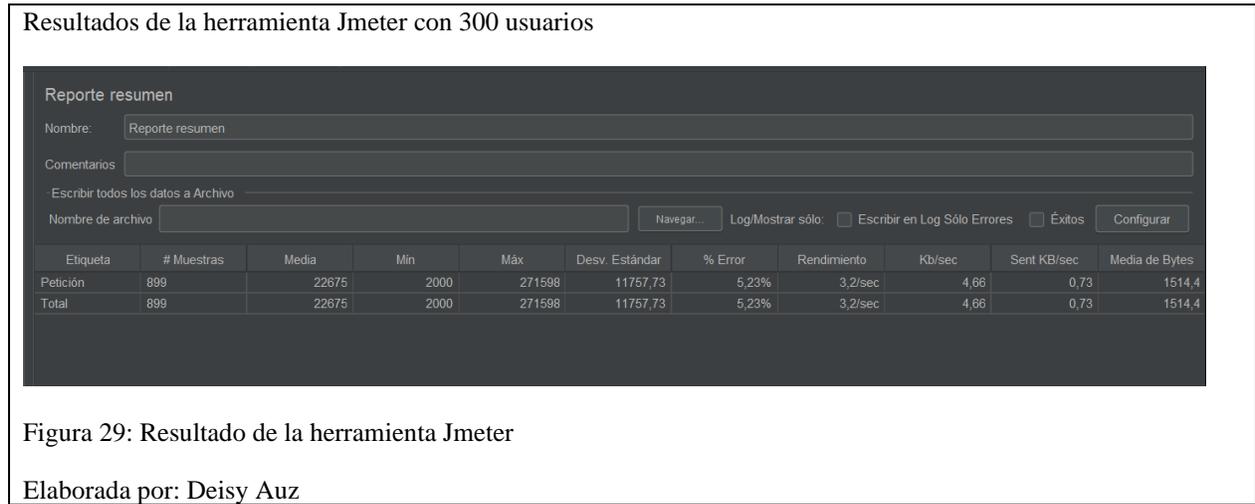
Se obtuvieron los siguientes resultados:

Con 100 usuarios:



En la figura 28 se puede observar que la cantidad de peticiones atendidas fueron 300, de 3 peticiones por cada usuario (100). En un tiempo promedio de 34s.

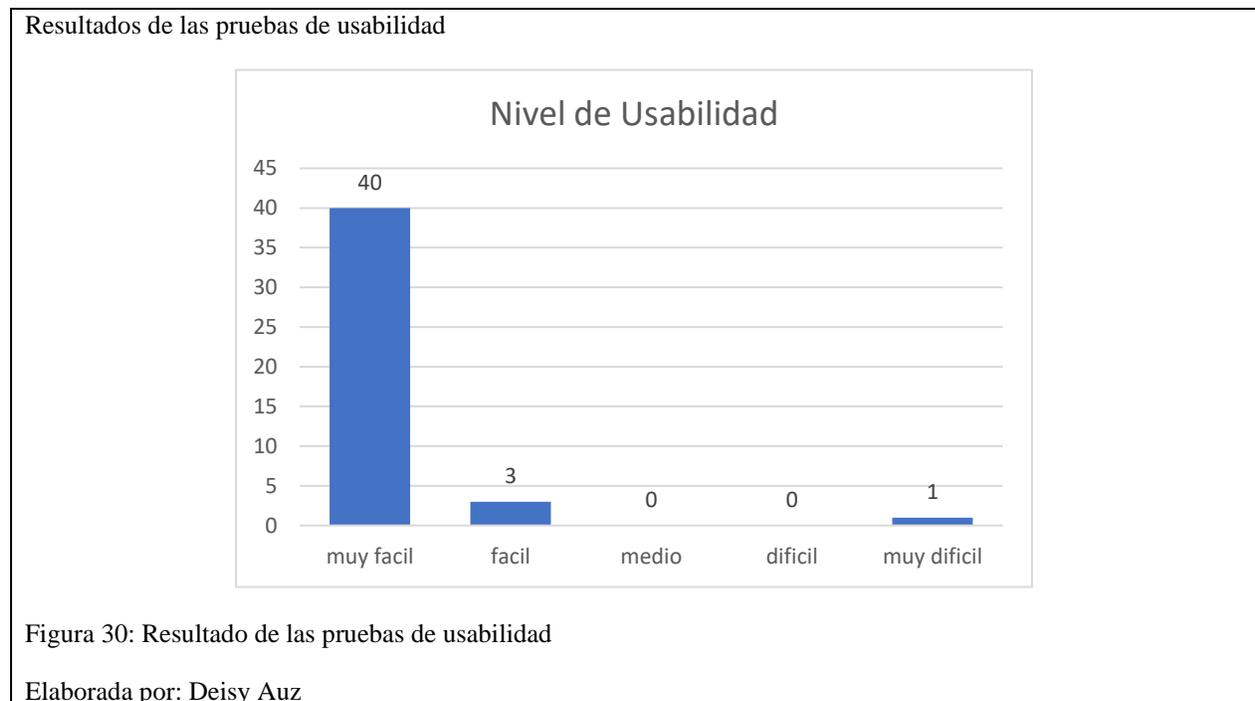
Con 300 usuarios:



En la figura 29 se puede observar que con 300 usuarios el porcentaje de error asciende a 5,12, en un tiempo promedio de respuesta de 4 minutos y 45 segundos, este tiempo no es aceptable tomando en cuenta que el tiempo máximo de respuesta establecido es de 5 segundos.

4.4. Usabilidad

El nivel de usabilidad se lo estableció por medio de la encuesta realizada de manera online ubicada en el anexo 7, donde los estudiantes calificaron entre 5 niveles de usabilidad siendo el primero de muy fácil uso y el quinto de uso difícil.



Como se puede visualizar en la figura 30 los estudiantes calificaron con un nivel de usabilidad de muy fácil, ya que el prototipo es una herramienta que no necesita de instrucciones para su uso.

4.5. Pruebas de portabilidad

Para probar la portabilidad del prototipo se realizó la ejecución del prototipo en cuatro diferentes navegadores y se verificó su funcionamiento.

Tabla 17:Tabla Comparativa de la portabilidad

Matriz de Resultados				
Objetivo:	Verificar que el prototipo funciona en diferentes navegadores			
Característica:	Portabilidad			
Funcionalidades / Navegadores	Google Chrome	Mozilla Firefox	Edge	Safari
Correcta disposición de la página	x	x	x	x
Legibilidad del Texto	x	x	x	x
Se visualizan correctamente el ingreso de preguntas	x	x	x	x
Se emiten las respuestas de manera adecuada	x	x	x	x
Se visualizan las imágenes	x	x	x	x

Como se puede observar en la tabla 17 el prototipo funciona correctamente en los diferentes navegadores probados, lo que delimita que es un prototipo portable.

Conclusiones

- Al investigar sobre los chatbots se determinó que los chatbots son utilizados como asistentes virtuales en diferentes ámbitos, ya que brindan soporte en cualquier momento, sin la necesidad de que haya una persona resolviendo las dudas, por lo que los chatbots al ser programados como asistentes, brindan las respuestas esperadas a las preguntas más frecuentes realizadas por los usuarios.
- Se implementó la base de conocimiento del chatbot sobre una base de datos no relacional, ya que este tipo de bases de datos realizan búsquedas más rápidas sobre grandes volúmenes de información, permitiendo al chatbot responder las consultas en un intervalo de tiempo más corto, al contrario de lo que sucede con las bases de datos relacionales que tardan más tiempo en responder a las consultas.
- Para la implementación de la interfaz web de comunicación entre el chatbot y el usuario, se utilizó el framework Flask, el cual permite el flujo de información entre la interfaz y la capa lógica con base en el protocolo HTTP, a su vez permite el acoplamiento de la librería encargada de la notificación de información faltante en la base de conocimiento por medio del protocolo SMTP.
- El entrenamiento del chatbot se realizó tomando en cuenta los escenarios planteados en las pruebas de confiabilidad, con el fin de alimentar al chatbot para la emisión de respuestas correctas con base en el análisis sintáctico que el chatbot realiza para comprender las preguntas realizadas y emitir una respuesta confiable.
- A través de las pruebas realizadas se verificó que el chatbot emite respuestas acertadas con base en la información almacenada en la base de conocimiento, esto permitió concluir que el chatbot emite las respuestas en un periodo corto de tiempo sin sobrecargar al sistema.

- Los chatbots son la innovación en el proceso enseñanza-aprendizaje y se pueden adaptar a cualquier materia de este ámbito ya que responden conceptos básicos y no entran a temas prácticos o que necesitan supervisión del docente.
- Los chatbots, al ser una tecnología emergente, aún no cuentan con metodologías formales preestablecidas para su construcción, por lo que los autores pueden construirlos bajo su criterio académico y profesional.
- Con base en la investigación realizada se concluye que actualmente existen servicios que permiten la construcción de chatbots con elementos ya creados, como son aplicaciones de mensajería y servicios de Inteligencia Artificial ya establecidos en la nube, los cuales pueden potenciar la capacidad de aprendizaje de este tipo de herramientas permitiendo que se utilicen en diferentes ámbitos educativos.

Recomendaciones

- Si bien los chatbots pueden ser aplicables a cualquier tipo de materia, en el ámbito educativo, se recomienda que estos estén enfocados a un solo tema específico puesto que deben estar delimitados por un marco de conocimiento, por ende, para estos fines es mejor utilizar los chatbots de FAQ ya que estos dan respuestas puntuales sobre un tema específico, ya que su Inteligencia Artificial no está enfocada a mantener una conversación.
- Para que la base de conocimiento sea más robusta se podría implementar un sistema de extracción, transformación y carga ETL por sus siglas en inglés (Extract, Transform and Load), el cual se encargará de extraer información de los principales textos de la cátedra, lo cual permite, en futuras investigaciones, acoplar este tipo de soluciones a las diversas materias impartidas en las distintas carreras de la Universidad Politécnica Salesiana.

- Para la creación de aplicaciones web basadas en Python, es recomendable utilizar el framework Flask puesto que es uno de los frameworks más sencillos y ligeros.
- Se recomienda que, para el manejo de grandes cantidades de datos no estructurados, se utilice gestores No SQL, como MongoDB, debido a que se ha demostrado en la literatura que las bases de datos no relacionales son altamente disponibles debido a su escalabilidad horizontal.

Referencias Bibliográficas

- Adil, A. (12 de 2020). *ellucian*. Obtenido de ellucian.com:
<https://www.ellucian.com/insights/how-chatbots-benefit-higher-ed>
- Angeles, M., & Gamez, A. (2015). Comparison of methods Hamming Distance, Jaro, and Monge-Elkan. *The Seventh International Conference on Advances in Databases, Knowledge, and Data Applications* (pág. 7). Mexico: DBKDA.
- Atkins, M. y. (2002). *Enseñanza efectiva en la educación superior*. London and New York: Routledge.
- Banda, H. (2014). *Inteligencia Artificial: Principios y Aplicaciones*. Quito.
- Bardaró, S., Ibañez, L., & Agüero, M. (10 de 2013). *Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones*. Obtenido de Palermo.edu:
https://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf
- Bird, S., & Loper, E. (2006). NLTK: The Natural Language Toolkit. *Researchgate*.
- Boris, G. (2019). *Developing Enterprise Chatbots: Learning Linguistic Structures*. San José de California: Springer.
- Coloma, C., & Tafur, R. (1999). EL CONSTRUCTIVISMO Y SUS IMPLICANCIAS EN EDUCACIÓN. *DIALNET*, 217-244.
- Davará, M. (1994). Introducción a los sistemas expertos. *Informática y derecho: Revista iberoamericana de derecho informático*.
- Dean, J. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems.
- Flask. (2010). *Flask web development, one drop at a time*. Obtenido de Flask web development, one drop at a time: <https://flask.palletsprojects.com/en/1.1.x/#>
- García, G., Fuertes, M., & Molas, N. (2018). Briefing Paper: Chatbots in Education.

- Garcia, G., Fuertes, M., & Molas, N. (2018). *Briefing paper: los chatbots en educación*. Catalunya: eLearn Center. Universitat Oberta de Catalunya.
- Ilyankou, I. (2014). Comparison of Jaro-Winkler and Ratcliff/Obershelp algorithms in spell check. *IB Extended Essay, Computer Science*.
- Jara, I., & Ochoa, J. (05 de 2020). *Usos y efectos de la Inteligencia Artificial en la educación*. Obtenido de publications.iadb.org:
<https://publications.iadb.org/publications/spanish/document/Usos-y-efectos-de-la-inteligencia-artificial-en-educacion.pdf>
- León, C. (2012). *Análisis Léxico y Sintáctico*. Santa Cruz de Tenerife: Universidad de La Laguna.
- Loria, S. (11 de 2020). *textblob.readthedocs.io/*. Obtenido de <https://textblob.readthedocs.io/>:
<https://textblob.readthedocs.io/en/dev/>
- Mejbri, N., Essalmi, F., & Rus, V. (2017). Educational system based on simulation and intelligent conversation. *IEEE Xplore*.
- Neves, A., Barros, F., & Hodges, A. (2006). *IAIML: A mechanism to treat. Tools with Artificial Intelligence*. ICTAI.
- Nietzel, M. (2020). How Colleges Are Using Chatbots To Improve Student Retention. *Forbes E-Megazine*.
- Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 2825-2830*.
- Pérez, G. C. (2018). *Universidades Corporativas: formación y aprendizaje en las organizaciones*. UOC. Obtenido de <http://ebookcentral.proquest.com/lib/bibliotecaupssp/detail.action?docID=5513935>.

Python . (5 de Julio de 2001). *Python developers guide: PEP 8 -- Style Guide for Python Code*.

Obtenido de PEP 8 -- Style Guide for Python Code:

<https://www.python.org/dev/peps/pep-0008/>

Ramesh, K., Ravishankaran, S., Joshi, A., & Chandrasekaran, K. (2017). A survey of Design Techniques for Conversational Agents. *Springer*.

Rehurek, R. (11 de 2020). *GENSIM Topic Modelling for humans*. Obtenido de

radimrehurek.com: <https://radimrehurek.com/gensim/wiki.html>

scikit-learn Organization. (12 de 2020). *scikit-learn.org*. Obtenido de <https://scikit-learn.org>:

[https://scikit-](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)

[learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)

Sevilla, D. (2018). Descubrimiento y visualización de esquemas de bases de datos NoSQL.

Modeling Languages.

Soria, V., Martí, R., Monterde, O., Lara, M., Capdevila, P., Herández-Leo, D., & Carrió, M.

(2016). *Impactos de la Innovación en la docencia y el aprendizaje*. Obtenido de CIDUI:

<https://www.cidui.org/revistacidui>

SpaCy. (11 de 2020). *spacy*. Obtenido de spacy.io/: <https://spacy.io/>

Thukrul, J., Srivastava, A., & Thakkar, G. (2020). DoctorBot - An Informative and Interactive Chatbot for COVID-19. *International Research Journal of Engineering and Technology*, 3033-3036.

Venners, B. (12 de 2003). The Making of Python. *artima*. Obtenido de Python:

<https://www.python.org/about/>

Verspoor, K., & Cohen, K. (2013). Natural Language Processing. *Researchgate*.

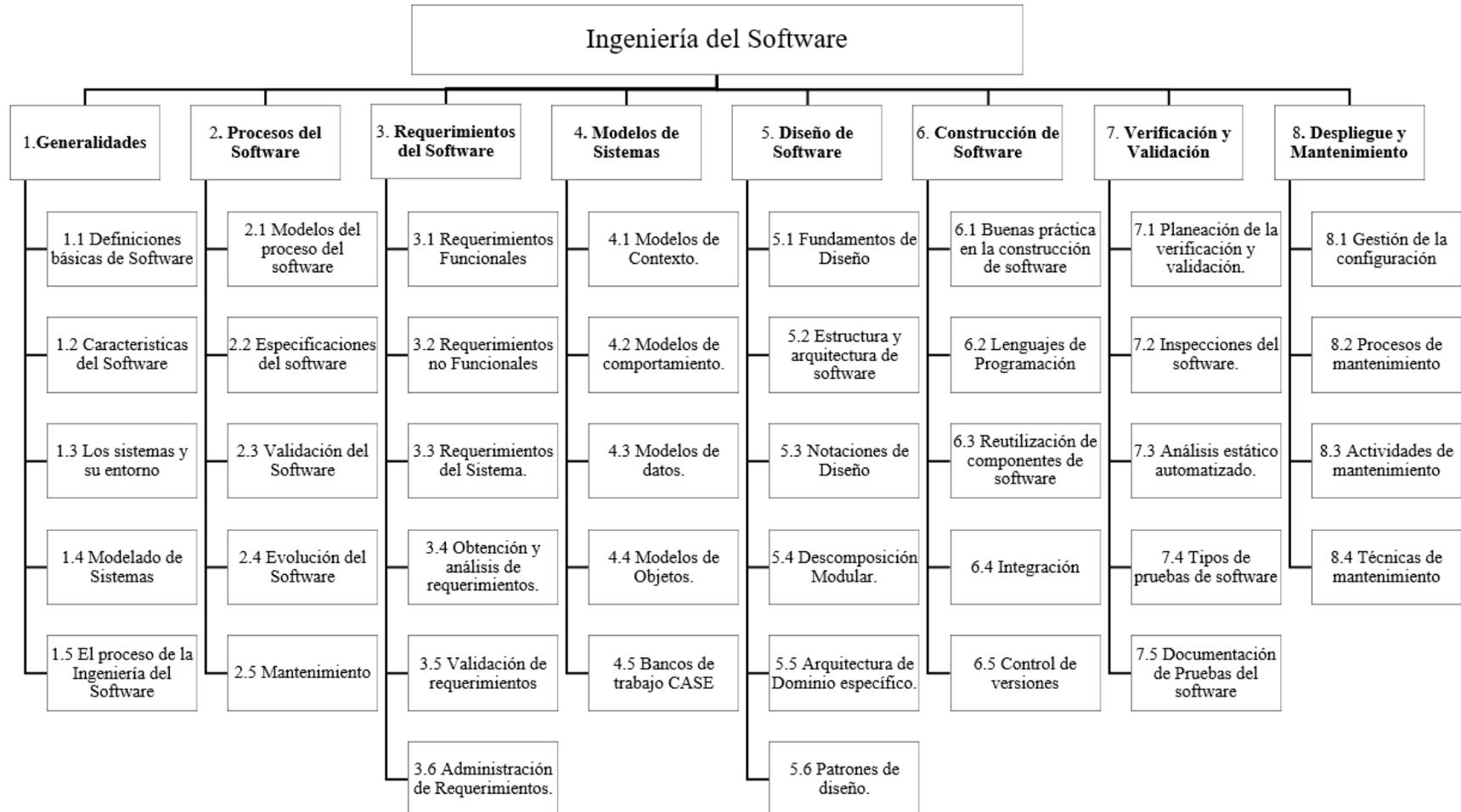
Winkler, R., & Söllner, M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. . *ResearchGate*.

Radziwill, N., & Benton, M. (s. f.). *Evaluating Quality of Chatbots and Intelligent Conversational Agents*. 21.

Singh, R., & Jain, N. (2019). *Chatbot Testing: Challenges and approach for testing Chatbot applications*. 24.

ANEXOS

Anexo1: Plan analítico de la cátedra de Ingeniería del Software



Anexo 2: Plan de Pruebas

Característica	Definición de la Característica según la ISO 9126	Métrica	Tipo de Prueba	Descripción de la Prueba	Instrumento	Caso / Escenario	Variantes o Repeticiones
Funcionalidad	Esta característica permite calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.	Porcentaje de cubrimiento de funcionalidades, con un mínimo aceptable del 90%	Prueba Funcional	Se realizó una encuesta a los usuarios finales, sobre la funcionalidad del prototipo, donde se le indica al estudiante la variante a probar. Además, se realizaron diferentes casos prueba.	Matriz de resultados	5	1100
Confiabilidad	La capacidad del producto software para mantener un nivel especificado de funcionamiento cuando se está utilizando bajo condiciones especificadas.	Porcentaje de respuestas acertadas, obtenido al comparar la respuesta recibida con la respuesta esperada con un mínimo aceptable del 90%	Prueba Funcional	Se preguntó a los usuarios finales el número de respuestas correctas y se verificará que la respuesta que emite el prototipo concuerde con la respuesta esperada.	Matriz de resultados	5	1100
Eficiencia	La capacidad del producto software para proveer un desempeño apropiado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas.	Promedio de los tiempos de respuesta de los diferentes escenarios con un tiempo máximo de 5s	Prueba de Rendimiento	Se midió el tiempo de respuesta del prototipo.	Matriz de resultados Herramienta JMeter	1	10
Usabilidad	Es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva.	Nivel de comprensión del usuario sobre el uso del prototipo, se espera que el nivel mínimo de usabilidad sea medio.	Prueba de Usabilidad	Se preguntó a los usuarios finales (estudiantes y docentes) el nivel de facilidad de uso del prototipo.	Cuestionario con preguntas cerradas	1	44
Portabilidad	La capacidad que tiene el software para ser trasladado de un entorno a otro.	Número de navegadores en los que el prototipo funciona correctamente, con un mínimo aceptable de tres navegadores	Prueba Comparativa	Se probó el prototipo en cuatro navegadores para conocer si en ellos están disponibles todas sus funcionalidades.	Matriz Comparativa	1	4

Anexo 3: Encuesta realizada a los usuarios después de la ejecución de las pruebas

Formulario del Uso del Chatbot

*Obligatorio

Funcionalidad

Cuántas veces respondió el chatbot? *

	1	2	3	4	5
Al ingresar solo una palabra	<input type="radio"/>				
Al ingresar dos palabras	<input type="radio"/>				
Al ingresar la pregunta con signos de interrogación y tildes	<input type="radio"/>				
Al ingresar la pregunta con faltas de ortografía	<input type="radio"/>				
Al ingresar una frase informal relacionada a la pregunta	<input type="radio"/>				

Anexo 4: Matriz de resultados de las pruebas funcionales

Matriz de Resultados			
Objetivo: Verificar que el prototipo responda a una pregunta realizada			
Característica: Funcionalidad			
Caso: 1			
Caso Prueba	Variante	Pregunta Realizada	Frecuencia
1	1	software	44
	2	Proceso	44
	3	validación	43
	4	espiral	43
	5	PUA	44
2	1	modelo cascada	44
	2	Diagrama clases	44
	3	PRUEBAS ALFA	44
	4	Mantenimiento Preventivo	43
	5	sistema Heredado	44
3	1	¿Qué es ingeniería del software?	44
	2	¿Qué es SCRUM?	42
	3	¿Qué es vvv del software?	44
	4	Para qué se usa un prototipo?	44
	5	¿Cuáles son las características del Software?	44
4	1	que son los sprints	44
	2	cuales son los modelos ebolutibos	44
	3	¿Qué son los sistemas eredados?	43
	4	analicis del sistema	42
	5	Modelos Agiles	43
5	1	me puedes decir algo sobre el proceso de software	44
	2	que sabes sobre la ingeniería inversa	44
	3	Me dices algo de UML	44
	4	me ayudas con un diagrama de casos de uso	43
	5	que hay de las pruebas de usuario	44

Anexo 5: Encuesta de Confiabilidad

Confiabilidad

Cuántas veces respondió el chatbot acertadamente? *

	1	2	3	4	5
Al ingresar solo una palabra	<input type="radio"/>				
Al ingresar dos palabras	<input type="radio"/>				
Al ingresar la pregunta con signos de interrogación y tildes	<input type="radio"/>				
Al ingresar la pregunta con faltas de ortografía	<input type="radio"/>				
Al ingresar una frase informal relacionada a la pregunta	<input type="radio"/>				

Anexo 6: Matriz de resultados de las pruebas de confiabilidad

Matriz de Respuestas					
Objetivo: Verificar que el prototipo responda correctamente a una pregunta realizada					
Característica: Confiabilidad					
Caso	Variante	Pregunta Realizada	Respuesta esperada	Frecuencia	Total
1	1	software	Por lo general al término software lo asocian con los programas de computadora. Sin embargo, yo prefiero una definición más amplia donde el software no son sólo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.	44	218

	2	Proceso	Un proceso de software es una serie de actividades relacionadas que conduce a la elaboración de un producto de software. Estas actividades pueden incluir el desarrollo de software desde cero en un lenguaje de programación estándar como Java o C.	44	
	3	validación	Validación de requerimientos en esta actividad verifica que los requerimientos sean realistas, coherentes y completos. Durante este proceso es inevitable descubrir errores en el documento de requerimientos. En consecuencia, deberían modificarse con la finalidad de corregir dichos problemas.	43	
	4	espiral	El modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas, el software se desarrolla en una serie de entregas evolutivas. Durante las primeras iteraciones, lo que se entrega puede ser un modelo o prototipo. En las iteraciones posteriores se producen versiones cada vez más completas.	44	
	5	PUA	El proceso unificado ágil (PUA) adopta una filosofía “serial en lo grande” e “iterativo en lo pequeño” para la elaboración de software. Describe una aproximación al desarrollo de aplicaciones que combina conceptos propios del proceso unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad.	43	
2	1	modelo cascada	La modelo cascada, sugiere un enfoque sistemático y secuencial, disciplinado y basado en análisis, diseño, pruebas y mantenimiento. Al final de cada etapa se reúnen y revisan los documentos para garantizar que se cumplen los requerimientos antes de avanzar a la fase siguiente. Pionero en guiar el proceso de desarrollo de software dirigido por un plan, introduciendo una planificación de casa fase antes de empezar a trabajar en ella.	44	219
	2	Diagrama clases	Un diagrama de clases muestra la estructura del sistema, subsistema o componente utilizando clases con sus características, restricciones y relaciones: asociaciones, generalizaciones, dependencias, etc.	44	
	3	PRUEBAS ALFA	Pruebas alfa, donde los usuarios del software trabajan con el equipo de diseño para probar el software en el sitio del desarrollador, los usuarios y desarrolladores trabajan en conjunto para probar un sistema a medida que se desarrolla.	43	
	4	Mantenimiento Preventivo	El mantenimiento preventivo consiste en la revisión constante del software para detectar posibles fuentes de problemas que puedan surgir en el futuro.	44	
	5	sistema Heredado	Un sistema heredado es un sistema antiguo que aún proporciona servicios esenciales de negocio.	44	
3	1	¿Qué es ingeniería del software?	La ingeniería del software es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento del sistema.	41	212

	2	¿Qué es SCRUM?	Scrum se refiere al trabajo en equipo, su enfoque está en la administración iterativa del desarrollo, dentro de un proceso de análisis que incorpora las siguientes actividades: la primera es la planeación del bosquejo, donde se establecen los objetivos generales del proyecto y el diseño de la arquitectura de software. A esto le sigue una serie de ciclos Sprint, donde cada ciclo desarrolla un incremento del sistema. Finalmente, la fase de cierre del proyecto, con la documentación requerida, como los marcos de ayuda del sistema y los manuales del usuario.	43	
	3	¿Qué es vyv del software?	La validación de software o, más generalmente, su verificación y validación (V&V), se crea para mostrar que un sistema cumple tanto con sus especificaciones como con las expectativas del cliente. Las pruebas del programa, donde el sistema se ejecuta a través de datos de prueba simulados, son la principal técnica de validación. Las etapas en el proceso de pruebas son: Pruebas de desarrollo, Pruebas de usuario y Pruebas de Versión	42	
	4	Para qué se usa un prototipo?	Un prototipo de software se usa: 1. En ingeniería de requerimientos, un prototipo ayuda con la selección y validación de requerimientos del sistema. 2. En el proceso de diseño de sistemas, un prototipo sirve para buscar soluciones específicas de software y apoyar el diseño de interfaces del usuario. 3. Y en la creación del prototipo del sistema, donde rápidamente se desarrolla una versión del sistema o una parte de este, para comprobar los requerimientos del cliente y la factibilidad de algunas decisiones de diseño.	44	
	5	¿Cuáles son las características del Software?	Las características del Software están relacionadas con los atributos esenciales del software estos atributos se los conoce con el acrónimo de FURPS: funcionalidad, usabilidad, confiabilidad, rendimiento y mantenibilidad. Estos atributos representan el objetivo de todo diseño de software.	42	
4	1	que son los Sprints	Los Sprints son unidades de trabajo que se necesitan para alcanzar un requerimiento definido a través de un entregable o incremento del producto. Los Sprints tienen longitud fija, por lo general de dos a cuatro semanas. El Sprint siempre comienza con una reunión de planificación, en esta se decide que se va a hacer y cómo, el número de tareas o historias de usuario que se van a realizar.	44	151
	2	cuales son los modelos ebulutibos	El objetivo de los modelos evolutivos es desarrollar software de alta calidad en forma iterativa o incremental. Sin embargo, es posible usar un proceso evolutivo para hacer énfasis en la flexibilidad, expansibilidad y velocidad del desarrollo. El reto es establecer un balance apropiado entre estos parámetros críticos del proyecto, el producto, y la satisfacción del cliente, entre estos modelos tenemos el modelo de desarrollo evolutivo, incremental, modelo espiral y el modelo de desarrollo concurrente.	44	
	3	¿Qué son los sistemas eredados?	Los sistemas heredados no son sólo sistemas de software de aplicación antiguos. Son sistemas socio-técnicos, por lo que incluyen procesos de negocio, software de aplicación, software de apoyo y sistema hardware.	44	

	4	analisis del sistema	Análisis del sistema, es una etapa temprana en el desarrollo del sistema, aclarar la arquitectura del sistema requiere cierto análisis. Las decisiones de diseño arquitectónico tienen un efecto profundo sobre si el sistema puede o no cubrir requerimientos críticos como rendimiento, fiabilidad y mantenibilidad.	9	
	5	Modelos Agiles	Los enfoques ágiles en el desarrollo de software consideran el diseño y la implementación como las actividades centrales en el proceso del software. Incorporan otras actividades en el diseño y la implementación, como la adquisición de requerimientos y pruebas. La planeación es incremental y es más fácil modificar el proceso para reflejar los requerimientos cambiantes del cliente. Entre estos modelos tenemos Proceso Unificado Ágil, Programación Extrema, Desarrollo Adaptativo de Software, Scrum	10	
5	1	me puedes decir algo sobre el proceso de software	Un proceso de software es una serie de actividades relacionadas que conduce a la elaboración de un producto de software. Estas actividades pueden incluir el desarrollo de software desde cero en un lenguaje de programación estándar como Java o C.	42	208
	2	que sabes sobre la ingeniería inversa	La ingeniería inversa es un proceso en el cual se analiza un programa para comprender la estructura de su diseño interna en un mayor grado de abstracción que el código fuente	40	
	3	Me dices algo de UML	El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común, rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas.	42	
	4	conoces sobre el diagrama de casos de uso	El diagrama de casos de uso describe un conjunto de acciones (casos de uso) que algunos sistemas o sistemas (sujetos) deben o pueden realizar en colaboración con uno o más usuarios externos del sistema (actores) para proporcionar algunos resultados observables y valiosos a los actores u otros interesados del sistema(s)	42	
	5	que hay de las pruebas de usuario	Pruebas de usuario, donde los usuarios reales o potenciales de un sistema prueban el sistema en su propio entorno. hay tres diferentes tipos de pruebas de usuario: Pruebas alfa, pruebas beta y pruebas de aceptación	42	

Anexo 7: Encuesta realizada al usuario sobre la usabilidad

Usabilidad

En cuanto a la facilidad de uso, ud considera que el chatbot fue *

1 2 3 4 5

Muy Fácil de Usar Muy Difícil de Usar

Enviar