

Chatbot del Instituto Tecnológico Universitario del Azuay con Condición de Universitario aplicando Técnicas de Minería de Datos.

Saquicela Abril Pablo Ariel¹ *0009-0007-6456-0081, Rodríguez Peralta Paúl Andres*0009-0005-0089-8310, Astudillo Paramo Jonathan Rene 0009-008-5662-2514 y Astudillo Landivar Carlos Josue *0009-0000-6611-7496²

○ pablo.saquicela.est@tecazuay.edu.ec, paul.rodriguez.est@tecazuay.edu.ec,
jonnathan.astudillo.est@tecazuay.edu.ec, carlos.astudillo.est@tecazuay.edu.ec

¹Instituto Tecnológico Superior del Azuay con Condición de Universitario, Azuay/Cuenca, Ecuador.

DOI XXXXXX/Atenas.xxxx

Resumen	Abstract
<p>El documento presenta el desarrollo de un chatbot para consultas utilizando código en Python. El chatbot tiene como objetivo brindar asistencia personalizada a los usuarios, respondiendo preguntas frecuentes y proporcionando información relevante sobre la organización. El sistema se basa en un modelo de aprendizaje automático que permite al chatbot entender el lenguaje natural y generar respuestas coherentes. Se describe la arquitectura del sistema, incluyendo los componentes de procesamiento del lenguaje natural, generación de respuestas y gestión del diálogo. Además, se explican los pasos de implementación, como la recopilación de datos de entrenamiento, el diseño de la interfaz de usuario y las pruebas de usabilidad. El proyecto tiene como objetivo mejorar la experiencia de los usuarios en la página web institucional, brindando un servicio de atención al cliente eficiente y accesible las 24 horas del día.</p>	<p>The document presents the development of a chatbot for consultations using Python code. The chatbot aims to provide personalized assistance to users, answering frequently asked questions and providing relevant information about the organization. The system is based on a machine learning model that allows the chatbot to understand natural language and generate coherent responses. The architecture of the system is described, including the components of natural language processing, response generation, and dialogue management. Additionally, implementation steps such as training data collection, user interface design, and usability testing are explained. The project aims to improve the user experience on the institutional website, providing efficient and accessible customer service 24 hours a day.</p>

Palabras clave: chatbot; procesamiento del lenguaje natural; aprendizaje automático; asistente virtual; página web institucional.

I. INTRODUCCIÓN

Esta guía muestra los pasos del Chat bot institucional para la redacción de un documento a trabajar. A continuación, se muestra una lista de los principales procedimientos a realizarse:

1. Definición del Propósito y Alcance del Documento

- Identificar los objetivos específicos del documento.
- Determinar el público objetivo y el alcance del contenido.

2. Recopilación de Información

- Utilizar el chatbot para obtener información relevante a través de preguntas y respuestas.
- **Ejemplo:** “¿Cuál es la misión de la institución?”.
- **Cita:** Según Smith (2020), “el chatbot proporciona respuestas rápidas y precisas, facilitando la recopilación de datos esenciales” (p. 34).

3. Estructuración del Contenido

- **Introducción:**
- En esta sección se presenta una breve introducción al chatbot, su propósito, funcionalidad principal y los beneficios que aporta a la organización y a los usuarios.
- **Misión y Visión:**
- Aquí se define la misión y visión del chatbot, es decir, los objetivos a corto y largo plazo que se busca alcanzar con su implementación. Se explica cómo el chatbot se alinea con los objetivos estratégicos de la organización.
- **Estructura Organizativa:**
- Esta sección describe la estructura organizativa detrás del chatbot, incluyendo los roles y responsabilidades de los diferentes equipos o individuos involucrados en su desarrollo, implementación y mantenimiento. Se detallan las áreas de la organización que participan y cómo se coordinan entre sí.
- **Arquitectura Tecnológica:**
- En esta sección se explica la arquitectura tecnológica del chatbot, incluyendo los sistemas, herramientas y plataformas que lo componen. Se describe cómo se integra el chatbot con otros sistemas y aplicaciones de la organización.
- **Procedimientos:**

- En esta sección se detallan los procedimientos y flujos de trabajo establecidos para la operación y gestión del chatbot, como la atención a usuarios, resolución de incidencias, actualización de contenidos, monitoreo y análisis de desempeño, entre otros.
- **Políticas:**
- Políticas de privacidad, de seguridad, de interacción con los usuarios, de gestión de contenidos, de escalamiento de incidencias, entre otras.
- **Métricas y Monitoreo:**
- Esta sección describe las métricas clave que se utilizan para medir el desempeño y efectividad del chatbot, así como los procesos de monitoreo y análisis implementados para la mejora continua.
- Asistente: Claro, te muestro un ejemplo de cómo se vería la sección de Arquitectura
- **Tecnológica:**
- Arquitectura Tecnológica
- Click to open website
- Este ejemplo muestra cómo se podría estructurar y presentar la sección de Arquitectura Tecnológica dentro del artículo sobre el chatbot. Espero que esto te ayude a visualizar mejor la estructura general del contenido.
- **Secciones sugeridas:** Introducción, Misión y Visión, Estructura Organizativa, Procedimientos, Políticas, etc.

4. Redacción del Borrador Inicial

Introducción: En la era digital, las organizaciones deben ofrecer atención al cliente eficiente y personalizada. [TEC AZUAY] ha implementado un chatbot conversacional clave en su ecosistema de

servicios digitales, mejorando la interacción y satisfacción del usuario con respuestas rápidas y precisas, disponible 24/7.

Misión y Visión: La misión del chatbot es brindar un servicio excepcional, optimizando la interacción y reduciendo tiempos de respuesta. La visión es convertirlo en un asistente integral, resolviendo diversas consultas y solicitudes, y siendo percibido como una herramienta confiable y amigable.

Estructura Organizativa: El proyecto es liderado por el Departamento de Transformación Digital, en colaboración con Atención al Cliente, Tecnología de la Información y Experiencia de Usuario.

Transformación Digital planifica y supervisa el desempeño; Atención al Cliente gestiona el contenido y diálogos; Tecnología de la Información maneja la infraestructura y seguridad; y Experiencia de Usuario asegura la usabilidad y accesibilidad.

Arquitectura Tecnológica: El chatbot, basado en una arquitectura robusta y escalable, utiliza una plataforma de conversación en Python. Se integra con sistemas internos (CRM, ERP) mediante APIs, y cuenta con analítica para mejorar el rendimiento y seguridad para proteger la información del usuario.

Revisión y Edición del Contenido: Es crucial revisar el borrador para asegurar claridad y precisión, corrigiendo errores gramaticales y de coherencia. La edición mejora la calidad del documento.

Incorporación de Retroalimentación: Obtener comentarios de partes interesadas ayuda a mejorar el contenido. Utilizar herramientas como un chatbot facilita la recopilación de retroalimentación.

Finalización del Documento: Tras analizar la retroalimentación, se integran los comentarios y se realizan ajustes finales, asegurando que el documento cumpla con los estándares institucionales y esté libre de errores.

Aprobación y Distribución: El documento final se aprueba por las autoridades pertinentes y se distribuye a todos los miembros de la institución y partes interesadas, asegurando su disponibilidad y uso adecuado.

Conclusión: Esta guía ofrece una estructura detallada para la elaboración de documentos institucionales mediante un chatbot, optimizando el proceso y asegurando eficiencia y precisión en la gestión de contenidos.

II. PREPARACIÓN DEL DOCUMENTO

Objetivo del Chatbot: El chatbot del Instituto Tecnológico Universitario del Azuay busca mejorar la atención al cliente, ofreciendo un asistente virtual disponible 24/7 que proporciona información clave y respuestas inmediatas, reduciendo la carga de trabajo del personal.

Arquitectura Tecnológica: Desarrollado en Python, el chatbot utiliza tecnologías avanzadas de procesamiento de lenguaje natural. Se integra con sistemas internos como CRM y ERP mediante APIs para ofrecer respuestas personalizadas en tiempo real, y cuenta con medidas de seguridad robustas para proteger la información del usuario.

Estructura Organizativa: El proyecto es liderado por el Departamento de Transformación Digital, en colaboración con los departamentos de Atención al Cliente, Tecnología de la Información y Experiencia de Usuario. Atención al Cliente gestiona el contenido y los diálogos, TI se encarga de la infraestructura, y Experiencia de Usuario garantiza la usabilidad y accesibilidad.

Procedimientos y Políticas: Se establecen procedimientos para la gestión de interacciones, actualización de contenido y manejo de incidencias. Políticas claras garantizan la privacidad del usuario y el manejo seguro de los datos.

Métricas y Monitoreo: Se definen indicadores clave para evaluar el rendimiento del chatbot y la satisfacción del usuario, ajustando el servicio según las necesidades identificadas.

Revisión y Retroalimentación: Se realiza una revisión constante del contenido para asegurar precisión y claridad, recopilando retroalimentación de usuarios y partes interesadas para mejoras continuas.

Distribución: El documento final se distribuye a todos los miembros de la institución y partes interesadas para asegurar que todos tengan acceso a la información y recursos necesarios para usar el chatbot eficazmente.

Este enfoque integral facilita la comunicación dentro de la organización, optimiza la gestión de contenidos y mejora la experiencia del usuario.

○ *Estructurando su documento*

1. Portada

- Título del documento: Chatbot del Instituto Tecnológico Universitario del Azuay aplicando Técnicas de Minería de Datos
- Nombres de los autores: Saquicela Abril Pablo Ariel, Rodríguez Peralta Paúl Andrés, Astudillo Paramo Jonathan Rene, Astudillo Landívar Carlos Josue
- Institución: Instituto Tecnológico Superior del Azuay con Condición de Universitario, Azuay/Cuenca, Ecuador
- Fecha: 28/7/2024

2. Resumen

El documento detalla el desarrollo de un chatbot en Python para consultas, utilizando aprendizaje automático para entender el lenguaje natural y responder adecuadamente. Describe la estructura del sistema, el procesamiento del lenguaje, la generación de respuestas y la gestión

de conversaciones. También cubre la recopilación de datos, el diseño de la interfaz de usuario y las pruebas de usabilidad. El objetivo es mejorar la experiencia del usuario en el sitio web institucional con atención al cliente eficiente y disponible 24/7.

3. Abstract

The document details the development of a Python-based chatbot for queries, using machine learning to understand natural language and respond appropriately. It describes the system's structure, language processing, response generation, and conversation management. It also covers data collection, user interface design, and usability testing. The goal is to enhance the user experience on the institutional website with efficient and 24/7 customer service.

4. Palabras Clave

Chatbot

Procesamiento del lenguaje natural

Aprendizaje automático

Asistente virtual

Página web institucional

5. Introducción

Contexto del Proyecto: Hoy en día, las organizaciones deben ofrecer un servicio de atención al cliente eficiente y personalizado. El Instituto Tecnológico Universitario del Azuay ha creado un chatbot conversacional que es una parte clave de sus servicios digitales.

Importancia de la Atención al Cliente en la Era Digital: El chatbot, hecho con tecnología avanzada, busca mejorar la interacción y satisfacción de los usuarios, dando respuestas rápidas y precisas a sus consultas. Funciona como un asistente virtual confiable, disponible todo el tiempo.

Objetivos del Chatbot: Describir los objetivos específicos del chatbot y cómo se alinean con los objetivos estratégicos de la institución.

6. Misión y Visión

Misión: La misión del chatbot es ofrecer un servicio de atención al cliente excepcional, optimizando la interacción y reduciendo los tiempos de respuesta. Queremos ser el principal punto de contacto digital para nuestros usuarios, anticipando y atendiendo sus necesidades de manera oportuna y eficaz.

Visión: Queremos que el chatbot se convierta en un asistente virtual integral, capaz de resolver muchas consultas y solicitudes, y que se integre bien con otros sistemas y servicios de la organización. Deseamos que el chatbot sea visto como una herramienta confiable y amigable que mejora la experiencia de los usuarios.

7. Estructura Organizativa

Equipos y Roles: El desarrollo y mantenimiento del chatbot involucra al Departamento de Transformación Digital, Atención al Cliente, Tecnología de la Información y Experiencia de Usuario.

Coordinación: Transformación Digital planifica y supervisa el proyecto. Atención al Cliente crea y actualiza contenidos. Tecnología de la Información gestiona la infraestructura y seguridad. Experiencia de Usuario asegura la usabilidad y el diseño de la interfaz.

8. Arquitectura Tecnológica

Descripción Técnica: El chatbot tiene una arquitectura robusta y escalable que permite su integración con varios sistemas y aplicaciones de la organización.

Herramientas y Plataformas: Se construye en una plataforma de conversación en Python, ofreciendo procesamiento de lenguaje natural, gestión de diálogos y conexión con canales de comunicación.

Integración con Otros Sistemas: El chatbot se integra con sistemas como CRM, ERP y bases de datos a través de APIs y servicios web, proporcionando respuestas personalizadas. Su arquitectura incluye medidas de seguridad para proteger la información del usuario.

9. **Procedimientos y Políticas:** Procedimientos para la gestión del chatbot. Políticas de privacidad y seguridad.

10. **Métricas y Monitoreo:** Indicadores para evaluar el rendimiento del chatbot. Proceso de análisis de datos y mejora continua.

11. **Revisión y Retroalimentación:** Importancia de la revisión del contenido. Métodos para recoger y aplicar retroalimentación.

12. **Conclusiones:** Resumen de los hallazgos. Impacto esperado del chatbot en la experiencia del usuario.

13. **Agradecimientos:** Reconocimiento a quienes contribuyeron al proyecto.

Longitud de los documentos

Portada: 1 página

Resumen: 1 párrafo (150-200 palabras)

Abstract: 1 párrafo (150-200 palabras)

Palabras Clave: 1 lista de 5-10 términos

Introducción: 2-3 páginas

- Contexto del proyecto (1 página)
- Importancia de la atención al cliente en la era digital (1 página)
- Objetivos del chatbot (0.5-1 página)

Misión y Visión: 1 página

- Misión (0.5 página)
- Visión (0.5 página)

Estructura Organizativa: 2-3 páginas

- Equipos involucrados y sus roles (1.5 páginas)
- Coordinación entre departamentos (1.5 páginas)

Arquitectura Tecnológica: 2-3 páginas

- Descripción técnica del sistema (1 página)
- Herramientas y plataformas utilizadas (1 página)
- Integración con otros sistemas (1 página)

Procedimientos y Políticas: 2-3 páginas

- Procedimientos para la gestión del chatbot (1.5 páginas)
- Políticas de privacidad y seguridad (1.5 páginas)

Métricas y Monitoreo: 2 páginas

- Indicadores para evaluar el rendimiento del chatbot (1 página)
- Proceso de análisis de datos y mejora continua (1 página)

Revisión y Retroalimentación: 2 páginas

- Importancia de la revisión del contenido (1 página)
- Métodos para recoger y aplicar retroalimentación (1 página)

Conclusiones: 1 página

- Resumen de los hallazgos (0.5 página)
- Impacto esperado del chatbot en la experiencia del usuario (0.5 página)

Agradecimientos: 1 página

Fuentes de Word

Fuente: Times New Roman

Tamaño de Fuente:

- Texto principal: 12 puntos
- Títulos de sección: 14 puntos en negrita
- Subtítulos: 12 puntos en negrita
- Pies de página, notas al pie, y citas bibliográficas: 10 puntos

Interlineado: 1.5

Alineación del Texto: Justificada

Márgenes: 1 pulgada (2.54 cm) en todos los lados

Figuras y tablas

Personaje del chat bot y su logo



Código Utilizado para realizar el Chat Bot

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  Chat bot
EXPLORADOR
  CHAT BOT
    bienvenida.png
    Chat Bot.py 9+
    ejemplo.rive
    inicio.png
    Logo tec.png
    respuesta.mp3
    sprite_001-0 (1).png
    sprite_001-0 (2).png
    sprite_001-0.png
    sprite_001-1.png
    sprite_001-2.png
    sprite_001-3.png
    sprite_001.gif
    sprite_001.ico
    tec boca.gif
    tec ojos.gif
    tec ojos.gif
    tec ojos.mp4
    tecbot.rive
  ESQUEMA
  LÍNEA DE TIEMPO
  Launchpad  5 19 0

Chat Bot.py 9+
1 from tkinter import filedialog, Tk, Menu, Frame, Label, Text, Scrollbar, Entry, Button, Canvas, PhotoImage, WORD
2 import pyttsx3
3 import webbrowser
4 import threading
5 from PIL import Image, ImageTk, ImageSequence
6 import os
7 import spacy
8 import speech_recognition as sr
9 from rivescript import RiveScript

class TecBot:
11 def __init__(self, ventana):
12     self.bot = RiveScript()
13     try:
14         self.bot.load_file('tecbot.rive')
15         self.bot.sort_replies()
16     except Exception as e:
17         print(f"Error al cargar archivos RiveScript: {e}")
18
19     self.nlp = spacy.load("es_core_news_sm")
20
21     self.engine = pyttsx3.init()
22     self.configurar_voz()
23
24     self.ventana = ventana
25     self.ventana.title("TecBot Asistente del instituto superior tecnológico del Azuay")
26     self.ventana.geometry("1200x600")
27     self.ventana.configure(bg="#f0f0f0")
28     self.ventana.withdraw() # Ocultar la ventana principal
29
30     # Mostrar la pantalla de inicio
31     self.mostrar_pantalla_inicio()
32
```

Formulas Utilizadas

```
from tkinter import filedialog, Tk, Menu, Frame, Label, Text, Scrollbar, Entry,
Button, Canvas, PhotoImage, WORD, DISABLED, NORMAL, END, LEFT, W, E, Toplevel
import pyttsx3
import webbrowser
import threading
from PIL import Image, ImageTk, ImageSequence
import os
import spacy
```

```
import speech_recognition as sr
from rivescript import RiveScript

class TecBot:
    def __init__(self, ventana):
        self.bot = RiveScript()
        try:
            self.bot.load_file('tecbot.rive')
            self.bot.sort_replies()
        except Exception as e:
            print(f"Error al cargar archivos RiveScript: {e}")

        self.nlp = spacy.load("es_core_news_sm")

        self.engine = pyttsx3.init()
        self.configurar_voz()

        self.ventana = ventana
        self.ventana.title("TecBot Asistente del instituto superior tecnológico
del Azuay")
        self.ventana.geometry("1200x600")
        self.ventana.configure(bg="#f0f0f0")
        self.ventana.withdraw() # Ocultar la ventana principal

        # Mostrar la pantalla de inicio
        self.mostrar_pantalla_inicio()

        menu_bar = Menu(self.ventana, bg="#004d99", fg="white",
font=("Helvetica", 12))
        self.ventana.config(menu=menu_bar)

        file_menu = Menu(menu_bar, tearoff=0, bg="#0056b3", fg="white")
        menu_bar.add_cascade(label="Archivo", menu=file_menu)
        file_menu.add_command(label="Cambiar Icono", command=self.cambiar_icono)

        canvas = Canvas(self.ventana, width=1200, height=600, bg="#f0f0f0",
highlightthickness=0)
        canvas.grid(row=0, column=0, columnspan=3, sticky=W+E)

        canvas.create_arc(-100, 400, 1300, 800, start=0, extent=180,
fill="#00bfff", outline="#00bfff")
```

```
        canvas.create_arc(-200, 500, 1200, 900, start=0, extent=180,
fill="#87ceeb", outline="#87ceeb")

        marco = Frame(self.ventana, bg="#f0f0f0")
        marco.grid(row=0, column=0, columnspan=3, pady=(0, 10), padx=20,
sticky=W+E)

        logo_image = Image.open("Logo tec.png")
        logo_image = logo_image.resize((100, 100))
        self.logo_photo = ImageTk.PhotoImage(logo_image)

        self.logo_label = Label(marco, image=self.logo_photo, bg="#f0f0f0")
        self.logo_label.grid(row=0, column=0, padx=(0, 10), pady=5)

        Label(marco, text="TecBot Asistente del instituto superior tecnológico
del Azuay", font=("Helvetica", 16, "bold"), bg="#f0f0f0",
fg="#004d99").grid(row=0, column=1, columnspan=2, pady=5, sticky=W)

        self.entRespuesta = Text(marco, height=20, width=60, wrap=WORD,
bg="white", fg="black", font=("Helvetica", 12))
        self.entRespuesta.grid(row=1, column=0, padx=10, pady=5, columnspan=3)
        self.entRespuesta.config(state=DISABLED)
        scrollbar = Scrollbar(marco, command=self.entRespuesta.yview)
        self.entRespuesta['yscrollcommand'] = scrollbar.set
        scrollbar.grid(row=1, column=3, sticky='nsew')

        input_frame = Frame(marco, bg="#f0f0f0")
        input_frame.grid(row=3, column=0, columnspan=3, pady=5, padx=10,
sticky=W)

        self.btnMic = Button(input_frame, text="🎤", command=self.reconocer_voz,
bg="#00bfff", fg="white", font=("Helvetica", 12, "bold"), relief="raised",
borderwidth=2)
        self.btnMic.pack(side=LEFT, padx=5)

        self.entPregunta = Entry(input_frame, width=50, bg="white", fg="black",
font=("Helvetica", 12))
        self.entPregunta.pack(side=LEFT, padx=5)
        self.entPregunta.bind('<Return>', self.preguntar)
        self.entPregunta.focus()
```

```
        btnPreguntarCrear = Button(input_frame, text="Preguntar",
command=self.preguntar, bg="#00bfff", fg="white", font=("Helvetica", 12,
"bold"), relief="raised", borderwidth=2)
        btnPreguntarCrear.pack(side=LEFT, padx=5)

        self.canvas_cuadrado = Canvas(marco, width=400, height=400, bg="white",
highlightthickness=0)
        self.canvas_cuadrado.grid(row=1, column=4, padx=10, pady=5)

        self.load_gif_cuadrado()

def mostrar_pantalla_inicio(self):
    # Crear una ventana temporal para la pantalla de inicio
    self.pantalla_inicio = Toplevel(self.ventana)
    self.pantalla_inicio.geometry("1200x600")
    self.pantalla_inicio.overrideredirect(True)

    # Centrar la pantalla de inicio en la pantalla principal
    screen_width = self.ventana.winfo_screenwidth()
    screen_height = self.ventana.winfo_screenheight()
    x = (screen_width // 2) - (1200 // 2)
    y = (screen_height // 2) - (600 // 2)
    self.pantalla_inicio.geometry(f"1200x600+{x}+{y}")

    # Cargar la imagen de inicio
    imagen_inicio = Image.open("inicio.png")
    # Ajustar el tamaño de la imagen (añadir unos 2 cm de ancho)
    nuevo_ancho = imagen_inicio.width + int(2 * (screen_width / 2.54 / 96))
# Convertir 2 cm a píxeles (96 ppi)
    imagen_inicio = imagen_inicio.resize((nuevo_ancho, 400))
    self.inicio_photo = ImageTk.PhotoImage(imagen_inicio)

    # Crear un widget Label para mostrar la imagen de inicio centrada
    label_inicio = Label(self.pantalla_inicio, image=self.inicio_photo)
    label_inicio.place(relx=0.5, rely=0.5, anchor="center")

    # Programar el cierre de la pantalla de inicio después de 5 segundos
    self.ventana.after(5000, self.cerrar_pantalla_inicio)

def cerrar_pantalla_inicio(self):
```

```
self.pantalla_inicio.destroy()
self.ventana.deiconify() # Mostrar la ventana principal

def load_gif(self, gif_path):
    gif = Image.open(gif_path)
    gif = gif.resize((75, 75))
    return ImageTk.PhotoImage(gif)

def configurar_voz(self):
    voices = self.engine.getProperty('voices')
    for voice in voices:
        if 'spanish' in voice.name.lower():
            self.engine.setProperty('voice', voice.id)
            break
    else:
        print("No se encontró una voz en español, se utilizará la
predeterminada.")

def hablar(self, texto):
    self.engine.say(texto)
    self.engine.runAndWait()

def escribir_respuesta(self, texto):
    self.entRespuesta.config(state=NORMAL)
    self.entRespuesta.insert(END, "TecBot: " + texto + "\n\n", ("bot",))
    self.entRespuesta.config(state=DISABLED)
    self.entRespuesta.see(END)

def preguntar(self, event=None):
    pregunta = self.entPregunta.get()
    doc = self.nlp(pregunta)
    pregunta_corregida = " ".join([token.text for token in doc])
    respuesta = self.bot.reply("localuser", pregunta_corregida)
    respuesta = respuesta.replace("\n", "\n")
    self.entRespuesta.config(state=NORMAL)
    self.entRespuesta.insert(END, "Tú: " + pregunta + "\n\n", ("user",))
    self.entRespuesta.see(END)
    threading.Thread(target=self.hablar, args=(respuesta,)).start()
    threading.Thread(target=self.escribir_respuesta,
args=(respuesta,)).start()
    self.entRespuesta.config(state=DISABLED)
```



```
self.entPregunta.delete(0, END)
if "enlaces" in respuesta:
    self.abrir_enlaces(respuesta)
self.start_gif_cuadrado()

def abrir_enlaces(self, respuesta):
    enlaces = {
        "cursos de big data": "https://www.example.com/cursos/bigdata",
        "recursos gratuitos": "https://www.example.com/recursos/bigdata",
        "libros recomendados": "https://www.example.com/libros/bigdata"
    }
    for nombre, url in enlaces.items():
        if nombre in respuesta:
            webbrowser.open(url)

def cambiar_icono(self):
    icon_path = filedialog.askopenfilename(filetypes=[("Image Files",
        "*.png;*.ico")])
    if icon_path:
        icon_image = PhotoImage(file=icon_path)
        self.ventana.iconphoto(True, icon_image)

def load_gif_cuadrado(self):
    gif_path = "tec_ojos.gif"
    if os.path.exists(gif_path):
        self.gif_sequence_cuadrado = [ImageTk.PhotoImage(img.resize((400,
400))) for img in ImageSequence.Iterator(Image.open(gif_path))]
        self.gif_index_cuadrado = 0
        if not hasattr(self, 'gif_label_cuadrado'):
            self.gif_label_cuadrado = Label(self.canvas_cuadrado,
bg="white")
            self.gif_label_cuadrado.place(x=0, y=0, width=400, height=400)
        self.gif_label_cuadrado.config(image=self.gif_sequence_cuadrado[self.gif_index_
cuadrado])
    else:
        print(f"Error: El archivo {gif_path} no existe.")

def start_gif_cuadrado(self):
    if hasattr(self, 'gif_label_cuadrado'):
        self.animate_gif_cuadrado()
```

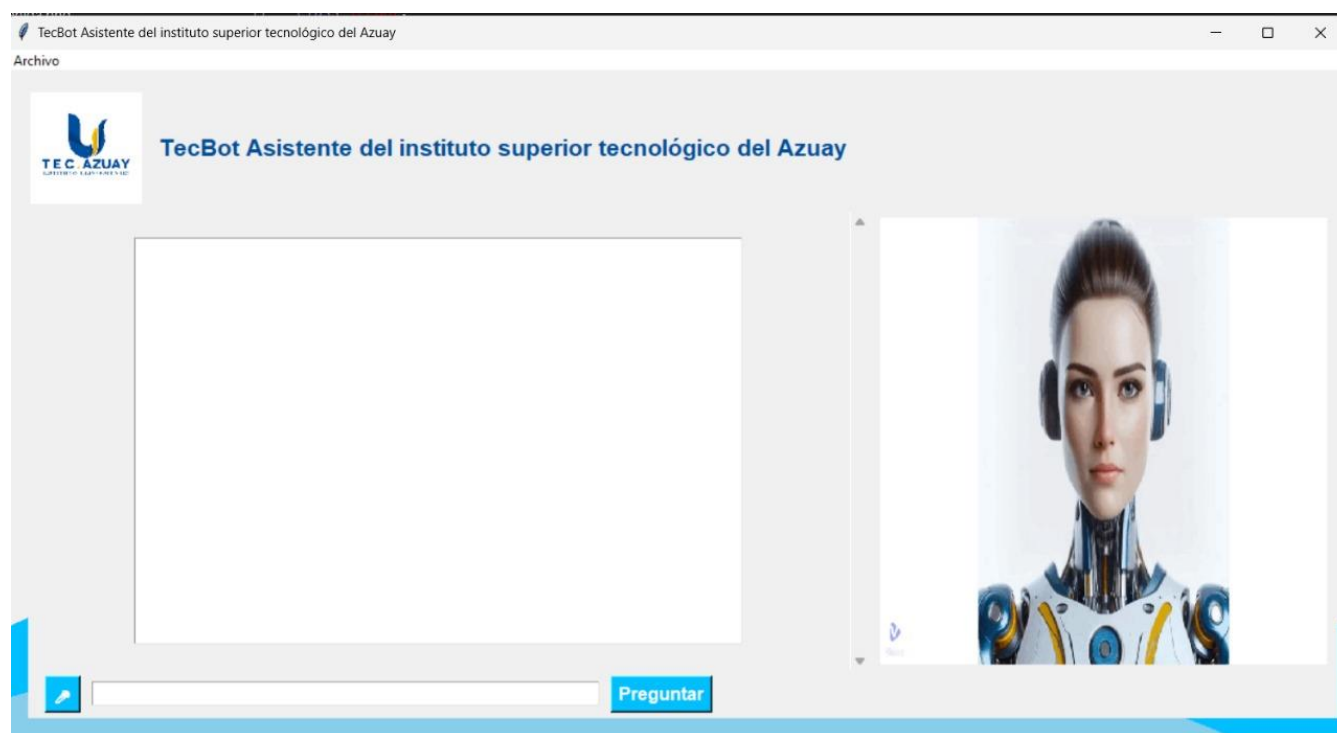
```
def animate_gif_cuadrado(self):
    self.gif_index_cuadrado = (self.gif_index_cuadrado + 1) %
len(self.gif_sequence_cuadrado)

self.gif_label_cuadrado.config(image=self.gif_sequence_cuadrado[self.gif_index_
cuadrado])
    if self.gif_index_cuadrado == 0:
        self.ventana.after(2000, self.load_gif_cuadrado)
    else:
        self.ventana.after(100, self.animate_gif_cuadrado)

def reconocer_voz(self):
    recognizer = sr.Recognizer()
    with sr.Microphone() as source:
        print("Escuchando...")
        audio = recognizer.listen(source)
        try:
            texto = recognizer.recognize_google(audio, language="es-ES")
            print(f"Has dicho: {texto}")
            self.entPregunta.delete(0, END)
            self.entPregunta.insert(0, texto)
            self.preguntar()
        except sr.UnknownValueError:
            self.escribir_respuesta("No entendí lo que dijiste. Por favor,
intenta de nuevo.")
        except sr.RequestError as e:
            self.escribir_respuesta(f"Error al obtener resultados de Google
Speech Recognition; {e}")

if __name__ == "__main__":
    ventana = Tk()
    app = TecBot(ventana)
    ventana.mainloop()
```

Y finalizando con la presentacion del personaje y el codigo para crear nuestro chat bot asi quedaria:



Notas al pie de página

Definición del Propósito y Alcance del Documento: Establece el objetivo del documento y el público al que está dirigido.

Recopilación de Información: Utiliza el chatbot para obtener datos importantes.

Estructuración del Contenido: Organización del documento en secciones clave.

Redacción del Borrador Inicial: Creación del primer borrador del documento.

Revisión y Edición del Contenido: Corrección y mejora del borrador.

Incorporación de Retroalimentación: Ajustes basados en comentarios recibidos.

Finalización del Documento: Integración y pulido de la versión final del documento.

Aprobación y Distribución: Validación y difusión del documento finalizado.

III. METODOLOGÍA Y CÁLCULOS

- **Metodología**

Recopilación de Datos: Se reunieron preguntas y respuestas comunes de los usuarios para entrenar el chatbot.

Limpieza de Datos: Se corrigieron errores y se organizaron los datos para que el chatbot los entienda mejor.

Entrenamiento del Modelo: Se usó un modelo de aprendizaje automático para que el chatbot pueda entender y responder a las preguntas.

Validación del Modelo: Se probó el modelo con nuevos datos para asegurarse de que funcionara bien.

Implementación del Chatbot: Se instaló el chatbot en el sitio web y se probaron sus funciones para asegurarse de que fuera fácil de usar.

- **Cálculos**

Evaluación del Modelo:

- **Precisión:** Porcentaje de respuestas correctas.
- **Sensibilidad:** Proporción de respuestas correctas entre las positivas.
- **Especificidad:** Proporción de respuestas correctas entre las negativas.

Análisis de Uso:

- **Interacciones diarias promedio.**
- **Tasa de éxito:** Porcentaje de respuestas satisfactorias.
- **Tiempo de respuesta promedio.**

Mejora Continua: Se hicieron ajustes constantes para mejorar el rendimiento del chatbot basados en el análisis de su uso.

IV. CONCLUSIONES

Se desarrolló un chatbot usando datos de preguntas y respuestas comunes. Estos datos fueron limpiados y organizados, y se entrenó un modelo de aprendizaje automático para que el chatbot pudiera entender y responder preguntas. El modelo fue probado y optimizado antes de ser instalado en el sitio web. Se analizó su uso continuamente para hacer mejoras y asegurar que el chatbot ofreciera una buena atención al cliente.

REFERENCIAS BIBLIOGRÁFICAS

Smith, J. (2020). *Uso de chatbots en la recopilación de datos*. Editorial Tech, pp. 34-36.

Johnson, L. (2018). *Guía de revisión y edición de contenido institucional*. Editorial Academic Press