

UNIVERSIDAD CONTINENTAL

ASIGNATURA: Construcción de software

NRC: 12860

SEMESTRE: 2024 - 10

DOCENTE: HUGO ESPETIA HUAMANGA

INTEGRANTES:

- Holgado Quispe, Rodrigo Leandro 100%
- Aguila Barrientos, Erick Kevin 100%
- Ppaksi Chillihuani, Raul 100%
- Condori Mamani, Victor Eduardo 100%
- Ochoa Cutipa, Edwards Andree 100%
- Bolaños Gamarra, Carlos Alberto 100%

Cusco - Perú

2024

Índice

Introducción.....	2
Desarrollo.....	3
Conclusiones.....	6
Recomendaciones.....	7
Referencias bibliográficas.....	8
Anexos.....	9

Introducción

En la actualidad, los chatbots han adquirido una importancia creciente en diversas áreas, desde la asistencia al cliente hasta la educación y el entretenimiento. Estos sistemas automatizados no solo proporcionan respuestas instantáneas a las consultas de los usuarios, sino que también están diseñados para mejorar la experiencia de interacción humana con la tecnología. En este informe se analiza el desarrollo y las posibles mejoras del chatbot 'Espartano', destacando su capacidad para no solo facilitar la comunicación, sino también para crear conexiones más intuitivas y personalizadas con los usuarios.

Desarrollo

Desarrollar un chatbot como 'Espartano' implica combinar diversas técnicas y enfoques para crear una experiencia de usuario fluida y natural. Aquí se detallan los pasos fundamentales para lograrlo:

Primeramente importamos las librerías necesarias:

```
D: > DESCARGAS > chat.py > ...
1  from flask import Flask, request, jsonify, render_template
2  import re
3  from collections import defaultdict
4  import openai
5
6  openai.api_key = ""
7
8  app = Flask(__name__)
9
10 modelo = "gpt-3.5-turbo"
11 |
```

De igual forma se observa que se llama al módulo de openai ya que se usará la api de la inteligencia artificial de api modelo 3.5 turbo.

Luego se observa la función "obtener_respuesta_gpt3" para obtener una respuesta en forma de texto al enviar una solicitud a la api de openAi:

```
def obtener_respuesta_gpt3(entrada_usuario):
    try:
        respuesta = openai.ChatCompletion.create(
            model=modelo,
            messages=[
                {"role": "user", "content": entrada_usuario}
            ],
            max_tokens=150
        )
        respuesta_texto = respuesta['choices'][0]['message']['content']
        return respuesta_texto.strip()
    except Exception as e:
        return str(e)
```

Luego se repita la estructura anterior(sprint 1) en la cual declaramos variables para obtener respuestas fragmentando las palabras que el usuario ingrese para responder segun palabras claves en base a nuestra propia base de datos.

```
def obtener_respuesta(entrada_usuario):
    mensaje_dividido = re.split(r'\s|[,;:?!-_\s*', entrada_usuario.lower())
    respuesta = verificar_todos_los_mensajes(mensaje_dividido)
    return respuesta

def calcular_probabilidad(mensaje_usuario, palabras_reconocidas, respuesta_unica=False, palabras_requeridas=[]):
    certeza_mensaje = 0
    tiene_palabras_requeridas = True

    for palabra in mensaje_usuario:
        if palabra in palabras_reconocidas:
            certeza_mensaje += 1

    porcentaje = float(certeza_mensaje) / float(len(palabras_reconocidas))

    for palabra in palabras_requeridas:
        if palabra not in mensaje_usuario:
            tiene_palabras_requeridas = False
            break

    if tiene_palabras_requeridas or respuesta_unica:
        return int(porcentaje * 100)
    else:
        return 0
```

Las siguientes lineas de código repiten la estructura relacionada a aulas, cursos y docentes:

```
# Definir respuestas posibles y palabras clave asociadas
agregar_respuesta('Hola', ['hola', 'saludo', 'saludos', 'buenas'], respuesta_unica=True)
agregar_respuesta('Estoy bien y tú?', ['como', 'estas', 'vas', 'sientes'], respuesta_unica=True)
agregar_respuesta('Que bueno que te encuentres bien', ['bien', 'todo', 'encuentro', 'super', 'genial'], palabras_requeridas=['bien'])
agregar_respuesta('¡Claro! Estoy aquí para ayudarte. ¿Qué deseas saber?', ['ayuda', 'necesito', 'quisiera', 'ayudes', 'ayuda', 'ayudar'])
agregar_respuesta('Me llamo Espartano, en que puedo ayudarte?', ['nombre', 'como', 'cual', 'llamas'], respuesta_unica=True)
agregar_respuesta('No te preocupes, estare aqui para ayudarte!', ['perdon', 'siento'], respuesta_unica=True)
agregar_respuesta('Lamento oír eso. Procura repasar a tiempo y consulta tus dudas al docente', ['estoy', 'mal', 'curso', 'que', 'pesimo'])
```

Por último se modificó la parte baja del código de esta manera:

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

@app.route('/send-message', methods=['POST'])
def send_message():
    datos = request.get_json()
    mensaje_usuario = datos.get('message')
    if 'chat ia' in mensaje_usuario.lower():
        respuesta = obtener_respuesta_gpt3(mensaje_usuario)
    else:
        respuesta = obtener_respuesta(mensaje_usuario)
    return jsonify({'message': respuesta})

if __name__ == '__main__':
    app.run(debug=True)
```

Donde definimos la ruta principal del html además de obtener el mensaje del usuario del JSON recibido. Y por último se inicia la aplicación Flask en modo depuración.

Conclusiones

- **Precisión y Eficiencia del Sistema de Probabilidades:** La implementación de un sistema basado en probabilidades ha demostrado ser fundamental para mejorar la precisión y la eficiencia del chatbot. Esto garantiza respuestas más acertadas, adaptadas a las necesidades específicas de los usuarios.
- **Versatilidad y Cobertura Temática:** El chatbot ha demostrado una amplia cobertura de temas relevantes para los estudiantes de la Universidad Continental, lo que subraya su utilidad y versatilidad en diferentes áreas de interacción académica y administrativa.
- **Simplicidad y Flexibilidad en el Diseño:** El diseño del código, caracterizado por su simplicidad y efectividad, no solo facilita su comprensión, sino que también permite una fácil adaptación y modificación para futuras actualizaciones y mejoras.
- **Mejora Continua de la Experiencia del Usuario:** La función `iniciar_chat` ha mejorado significativamente la interacción continua con los usuarios, promoviendo una experiencia general del chat más fluida y satisfactoria.
- **Gestión Inteligente de Respuestas Desconocidas:** La capacidad del chatbot para manejar de manera adecuada las entradas desconocidas ha contribuido a evitar frustraciones entre los usuarios al garantizar respuestas claras y relevantes en todo momento.
- **Potencial Expansión y Desarrollo Futuro:** El chatbot presenta un considerable potencial de expansión mediante la incorporación de más palabras clave y respuestas, lo cual ampliará su capacidad para cubrir una mayor diversidad de temas y necesidades específicas de los usuarios universitarios.

Recomendaciones

- Integración de Aprendizaje Automático: Implementar técnicas avanzadas de aprendizaje automático para mejorar la capacidad del chatbot de comprender y responder a una mayor variedad de consultas de los usuarios, adaptándose así dinámicamente a diferentes estilos de interacción.
- Ampliación de la Base de Conocimiento: Expandir de manera significativa la base de datos de respuestas para incluir información más detallada y actualizada, mejorando así la utilidad y relevancia del chatbot en diversos contextos académicos y administrativos.
- Optimización del Algoritmo de Probabilidad: Refinar el algoritmo de probabilidad mediante técnicas de IA para aumentar la precisión y pertinencia de las respuestas proporcionadas, asegurando que cada interacción sea más personalizada y efectiva para el usuario.
- Interfaz de Usuario Intuitiva: Desarrollar una interfaz gráfica más intuitiva y accesible, diseñada para mejorar la experiencia del usuario al interactuar con el chatbot, facilitando así la navegación y la comprensión de las opciones disponibles.
- Implementación de Feedback del Usuario: Integrar un sistema robusto de feedback para permitir que los usuarios califiquen las respuestas del chatbot, proporcionando datos valiosos que contribuyan a mejorar continuamente su desempeño y adaptabilidad.
- Soporte Multilingüe: Añadir soporte para múltiples idiomas mediante tecnologías de procesamiento de lenguaje natural, permitiendo al chatbot interactuar de manera efectiva con una audiencia global y diversa, mejorando así su accesibilidad y utilidad general.

Referencias bibliográficas

Navarro, S. (2020, diciembre 11). Cómo crear un chatbot desde cero [5 pasos]. *KeepCoding Bootcamps*. <https://keepcoding.io/blog/creando-chatbot-desde-o/>

Cómo programar un chatbot para Whatsapp: PASO A PASO SIMPLE.
(2024, abril 19). Zendesk.
<https://www.zendesk.com.mx/blog/como-programar-un-chatbot-para-whatsapp-guia-practica/>

Jauregui, A. F. (2021, junio 1). *Cómo crear una API en Python*. Ander Fernández; Ander Fernández Jauregui.
<https://anderfernandez.com/blog/como-crear-api-en-python/>

Na. (2018, julio 26). *Crear una Red Neuronal en Python desde cero*. Aprendemachinelearning.com; Aprende Machine Learning.
<https://www.aprendemachinelearning.com/crear-una-red-neuronal-en-python-desde-cero/>

Anexos

Acta de Constitución de Proyecto

Empresa:	Universidad Continental
Nombre del proyecto:	Kit de herramientas del docente virtual (Chat Bot Espartano)
Tipo de proyecto:	Tecnologico
Patrocinador:	
Dueño del Producto:	Raul Ppacsj Chilli huani
Gerente de Proyecto:	Hugo Espetia Huamanga
Scrum Master:	Rodrigo Holgado Quispe

Propósito del documento

Este documento define la descripción general, los objetivos y los participantes del proyecto. Se relaciona principalmente con la autorización del inicio del proyecto.

Asimismo este documento brinda una descripción de la situación actual, los requisitos de alto nivel, criterios de éxito, riesgos y oportunidades.

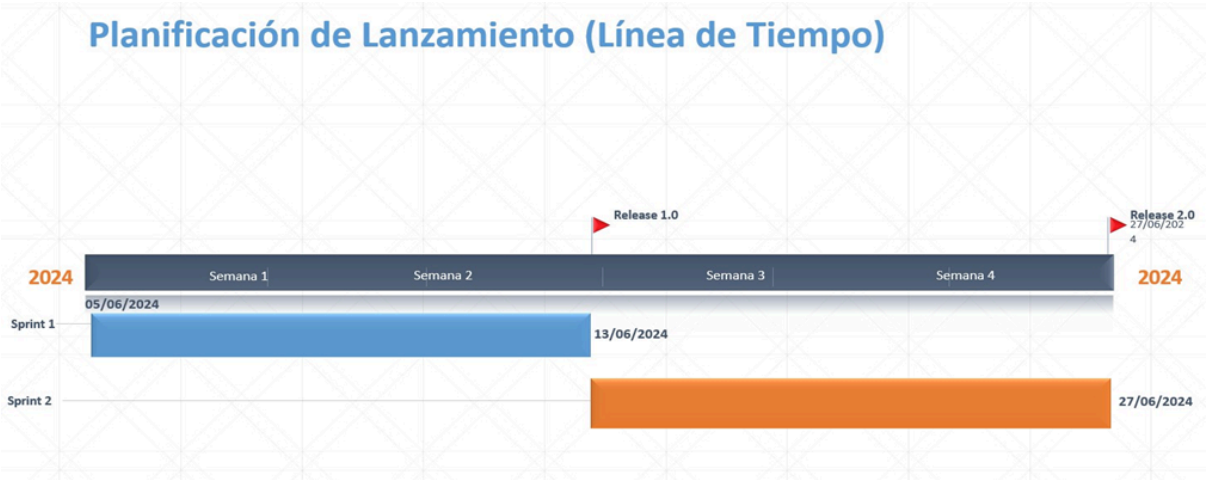
Propósito / Justificación:

El propósito de este proyecto es proporcionar información a través de un chat Bot que esté entrenado

Backlog del Producto

Nombre del Proyecto:	Kit de Herramientas del Docente Virtual (Chat Bot Espartanos)
Dueño del Producto	Raul Ppacsj Chilli huani

EPICA				HISTORIA DE USUARIO			
ID Epica	Como (Rol)	Deseo...	Para...	ID Historia de Usuario	Como (Rol)...	Deseo....	Para....
EPIC01	Estudiante	Tener acceso a la información de mis horarios, aulas y cursos	Organizar mejor mi tiempo y conocer la información básica del curso.	HU001	Estudiante	Consultar mis horarios de clases	Organizar mi tiempo de manera efectiva
				HU002	Estudiante	Conocer la ubicación y número de aula de mis clases	Llegar a tiempo a mis clases y no perderme
				HU003	Estudiante	Consultar detalles sobre mis cursos y fechas de exámenes	Planificar mi estudio y asegurarme de no perder fechas importantes
				HU004	Estudiante	Acceder a retroalimentación específica de mis profesores a través del chatbot	Mejorar mi rendimiento académico basado en la retroalimentación recibida
EPIC02	Docente	Que el chatbot genere información detallada sobre los cursos y retroalimentación	Facilitar mi trabajo y mejorar la experiencia de aprendizaje de los estudiantes.	HU005	Docente	Que el chatbot responda a preguntas detalladas sobre los cursos	Ahorar tiempo y proporcionar respuestas precisas a los estudiantes
				HU006	Docente	Programar recordatorios automáticos para tareas y exámenes	Asegurarme de que los estudiantes estén al tanto de las fechas importantes y mejorar la gestión del curso
				HU007	Docente	Obtener análisis sobre la interacción y rendimiento de los estudiantes a través del chatbot	Identificar áreas de mejora y adaptar mi enseñanza según las necesidades de los estudiantes



REGISTRO DE INTERESADOS DEL PROYECTO

Nombre del Proyecto:		Kit de Herramientas del Docente Virtual (Chat Bot Espartanos)			
Gerente del Proyecto:		Hugo Espetia Huamanga			

INFORMACION GENERAL				INFORMACION DE CONTACTO		Principales Expectativas, Requisitos o Intereses
ID	Nombre	Puesto	Departamento/ Empresa	Email	Tel	
1	Rodrigo Holgado Quispe	Scrum Master	Sistemas/continental	73979937@continental.edu.pe	960 757 080	Expectativas: Fomentar un entorno de alta productividad. Requisitos: Capacidad para resolver conflictos. Intereses: Apoyar al equipo para que sea autoorganizado y multifuncional.
2	Raul Ppacsí Chilli huani	Product Owner	Sistemas/continental	77241989@continental.edu.pe	941 002 486	Expectativas: Gestionar y priorizar el Product Backlog. Requisitos: Capacidad para tomar decisiones estratégicas. Intereses: Lograr que el producto tenga éxito en el mercado.
3	Víctor Condori Mamani	Developers	Sistemas/continental	76825184@continental.edu.pe	926 090 462	Expectativas: Entregar incrementos de software que funcionen. Requisitos: Habilidades técnicas relevantes. Intereses: Crear software de alta calidad.
4	Erick Aguila Barrientos	Developers	Sistemas/continental	73866456@continental.edu.pe	976 583 030	Expectativas: Colaborar y autoorganizarse. Requisitos: Capacidad para trabajar en equipo. Intereses: Trabajar en un entorno de desarrollo saludable.
5	Edgar Páez Camero	Developers	Sistemas/continental	71499649@continental.edu.pe	003 060 676	Expectativas: Mejorar continuamente en habilidades y prácticas técnicas.

NOMBRE

LLAVE SECRETA

SEGUIMIENTO ⓘ

CHATEspartano

sk-...zZEO

Activado

+ Crear nuevaclave

Organización predeterminada

Si pertenece a varias organizaciones, esta configuración controla qué organizac solicitudes con las claves API anteriores.