

# Asistente Virtual para el Sílabo

## Universitario

- Álvarez Carranza José Vitto Isidro  
(Universidad Católica Sede Sapientiae)
- Vega Zumaeta Erick Elvis  
(Universidad Católica Sede Sapientiae)
- Malpartida Hinojosa Juan Daniel  
(Universidad Católica Sede Sapientiae)

### **Resumen:**

Este documento presenta el desarrollo de un asistente virtual implementado en la plataforma de mensajería Telegram. El bot creado tiene la capacidad de proporcionar información sobre el sílabo de un curso específico y enviar archivos de audio. Se describe la metodología utilizada para el diseño e implementación del asistente, así como los resultados obtenidos y una discusión crítica sobre su eficacia.

**Palabras clave:** Asistente Virtual, Sílabo.

### **Abstract:**

This paper presents the development of a virtual assistant implemented in the Telegram messaging platform. The bot created has the ability to provide information about the syllabus of a specific course and send audio files. The methodology used for the design and implementation of the assistant is described, as well as the results obtained and a critical discussion of its effectiveness.

**Key words:** Virtual Assistant, Syllabus.

### **Introducción:**

En respuesta a la creciente necesidad de facilitar el acceso a información académica clave, hemos desarrollado un innovador asistente virtual. Este sistema ha sido diseñado con el propósito específico de ayudar a aquellos que buscan acceder a su sílabo de manera rápida y sencilla. La implementación de este asistente se ha llevado a cabo a través de un bot en la plataforma de Telegram, con el objetivo de ofrecer una solución accesible para aquellos que desean conocer los temas que se abordarán en sus clases o que enfrentan dificultades al intentar acceder a esta información de manera convencional. En este contexto,

nuestro asistente virtual se presenta como una herramienta eficaz para satisfacer estas necesidades educativas de manera eficiente y amigable.

El tema central gira en torno a la integración de la tecnología de asistentes virtuales en la experiencia educativa, específicamente en la optimización del sílabo universitario. El objeto de estudio se centra en el desarrollo y la evaluación de un asistente virtual diseñado para brindar información, claridad y apoyo en la comprensión y seguimiento del contenido del sílabo.

La motivación principal surge de la necesidad de mejorar la comunicación y comprensión del sílabo, reducir la brecha informativa entre profesores y estudiantes, y ofrecer una herramienta accesible y ágil que promueva un aprendizaje más efectivo y autónomo.

La relevancia radica en el impacto potencial en la calidad de la educación, la optimización del tiempo dedicado a entender el plan de estudios y la mejora en la experiencia de aprendizaje.

Se recolectarán datos sobre el sílabo y su contenido, el problema central reside en la falta de herramientas efectivas para la comprensión y seguimiento del sílabo universitario, lo que puede resultar en confusiones, malentendidos y dificultades en el seguimiento académico.

Se espera que la implementación del asistente virtual mejore la claridad, accesibilidad y comprensión del contenido del sílabo, promoviendo así una mayor eficiencia y efectividad en el proceso educativo.

Este estudio se enfoca en un contexto universitario específico durante un período determinado, buscando evaluar la viabilidad y efectividad del asistente virtual. Los alcances temporales se extienden a un semestre académico, con posibilidad de adaptación y expansión en futuras investigaciones.

## Metodología Utilizada:

En el proceso de creación de nuestra herramienta de asistencia virtual, hemos optado por emplear el lenguaje de programación Python. Ya que este contiene varias librerías que pueden ayudar a la creación de un asistente virtual. Esta elección nos ha permitido diseñar un asistente eficiente y versátil que hemos integrado en un bot de Telegram.

```
botpy x
botpy >...
1 import logging
2 import speech_recognition as sr
3 from gtts import gTTS
4 import os
5 from telegram import Update
6 from telegram.ext import Updater, MessageHandler, CommandHandler, CallbackContext, Filters
7
8
9
10 # Diccionario de sílabos
11 silabos = {
12     'innovacion y desarrollo tecnologico': {
13         'semana1': 'Concepto de Innovación.',
14         'semana2': 'El rol de la Investigación en tecnología',
15         'semana3': 'La Investigación como proceso tecnológico.',
16         'semana4': 'Las patentes',
17         'semana5': 'Creatividad e innovación en el profesional tecnológico.',
18         'semana6': 'La Investigación como proceso tecnológico.',
19         'semana7': 'Tipos de Proyecto de Investigación',
20         'semana8': 'Esquematiza el 1er paso de la conformación de la matriz lógica de investigación',
21         'semana9': 'La matriz lógica de investigación, primera parte.',
22         'semana10': 'La matriz lógica de investigación, segunda parte.',
23         'semana11': 'Formulación de un problema y objetivos de Investigación Tecnológica.',
24         'semana12': 'Formulación de la hipótesis de Investigación Tecnológica.',
25         'semana13': 'Definición de Variables, Tipos, Operacionalización (Indicadores e items)',
26         'semana14': 'Exposición de la secuencialidad lógica de la matriz de consistencia de la investigación tecnológica.',
27         'semana15': 'Técnicas e instrumentos de la investigación',
28         'semana16': 'Expone los diseños de investigación.',
29         'semana17': 'Universo, población, muestra.',
30         'semana18': 'EXAMEN FINAL.',
31     },
32 }
```

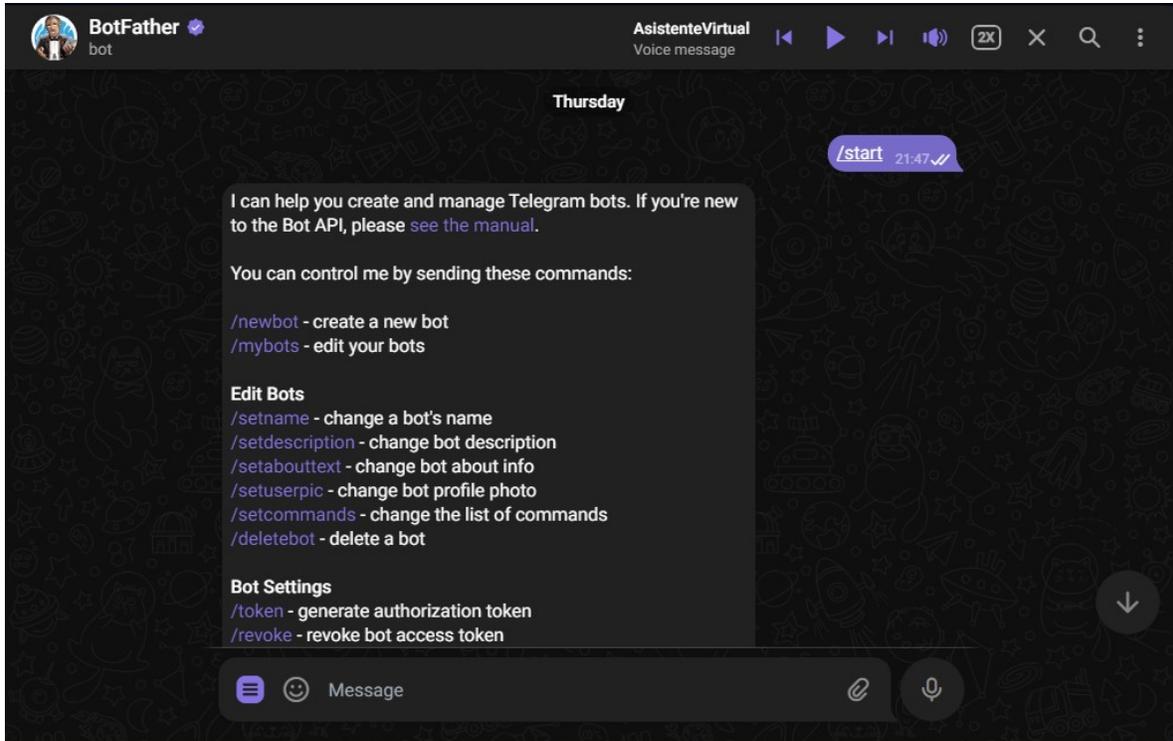
```
botpy x
botpy >...
11 silabos = {
34     'semana1': 'Introducción a la programación',
35     'semana2': 'Estructuras de control',
36     'semana3': 'Funciones y modularidad',
37     'semana4': 'Estructuras de datos',
38     'semana5': 'Programación orientada a objetos',
39 },
40 # Agrega más cursos y temas según tu plan de estudios
41 }
42
43 # Variable para almacenar el curso actual
44 curso_actual = None
45
46 recognizer = sr.Recognizer()
47
48 def escuchar_comando(update: Update, context: CallbackContext) -> None:
49     update.message.reply_text("Háblame...")
50
51 def manejar_audio(update: Update, context: CallbackContext) -> None:
52     file_id = update.message.voice.file_id
53     new_file = context.bot.get_file(file_id)
54     new_file.download("audio.ogg")
55
56     try:
57         with sr.AudioFile("audio.ogg") as source:
58             audio_data = recognizer.record(source)
59             comando = recognizer.recognize_google(audio_data, Language="es-ES").lower()
60             update.message.reply_text("Has dicho: " + comando)
61             manejar_comandos(update, context, comando)
62     except sr.UnknownValueError:
63         update.message.reply_text("No se pudo entender el comando.")
64     except sr.RequestError as e:
```

```
botpy X
botpy > ...
51 def manejar_audio(update: Update, context: CallbackContext) -> None:
52
53
54
55
56     try:
57         with sr.AudioFile("audio.ogg") as source:
58             audio_data = recognizer.record(source)
59             comando = recognizer.recognize_google(audio_data, Language="es-ES").lower()
60             update.message.reply_text("Has dicho: " + comando)
61             manejar_comandos(update, context, comando)
62     except sr.UnknownValueError:
63         update.message.reply_text("No se pudo entender el comando.")
64     except sr.RequestError as e:
65         update.message.reply_text("Error en la solicitud: {}".format(e))
66
67 def manejar_comandos(update: Update, context: CallbackContext, comando: str) -> None:
68     global curso_actual
69
70 def manejar_comandos(update: Update, context: CallbackContext) -> None:
71     global curso_actual
72
73     # Extraer el texto del mensaje
74     mensaje_texto = update.message.text.lower()
75
76     if any(keyword in mensaje_texto for keyword in ["ayuda", "innovación"]):
77         respuesta = "¿En qué puedo ayudarte con el curso de innovación?"
78         update.message.reply_text(respuesta)
79         hablar(update, context, respuesta)
80         curso_actual = 'innovacion y desarrollo tecnologico'
81     elif any(keyword in mensaje_texto for keyword in ["ayuda", "programación"]):
82         respuesta = "¿En qué puedo ayudarte con el curso de programación?"
83         update.message.reply_text(respuesta)
84         hablar(update, context, respuesta)
85         curso_actual = 'programacion'
```

```
botpy X
botpy > ...
70 def manejar_comandos(update: Update, context: CallbackContext) -> None:
81     elif any(keyword in mensaje_texto for keyword in ["ayuda", "programación"]):
82         respuesta = "¿En qué puedo ayudarte con el curso de programación?"
83         update.message.reply_text(respuesta)
84         hablar(update, context, respuesta)
85         curso_actual = 'programacion'
86     elif any(keyword in mensaje_texto for keyword in ["tema", "semana"]):
87         if curso_actual:
88             palabras = mensaje_texto.split()
89             semana_indicada = next((f"semana{palabra}" for palabra in palabras if palabra.isdigit()), None)
90             if semana_indicada and curso_actual in silabos and semana_indicada in silabos[curso_actual]:
91                 tema = silabos[curso_actual][semana_indicada]
92                 respuesta = f"En la {semana_indicada} del curso de {curso_actual}, el tema es:
93
94 {tema}.
95
96 ¿Puedo ayudarte en algo más?""
97         update.message.reply_text(respuesta)
98         hablar(update, context, respuesta)
99     else:
100         respuesta = f"No tengo información sobre esa semana del curso de {curso_actual}. ¿En qué más puedo ayudarte
101 update.message.reply_text(respuesta)
102         hablar(update, context, respuesta)
103     else:
104         respuesta = "No tienes un curso registrado. Por favor, selecciona un curso antes de realizar esta consulta."
105         update.message.reply_text(respuesta)
106         hablar(update, context, respuesta)
107
108 def hablar(update: Update, context: CallbackContext, texto: str) -> None:
109     tts = gTTS(text=texto, lang='es')
110     tts.save("respuesta.mp3")
111     context.bot.send_voice(chat_id=update.message.chat_id, voice=open("respuesta.mp3", "rb"))
```

```
bot.py
bot.py > TELEGRAM_TOKEN
70 def manejar_comandos(update: Update, context: CallbackContext) -> None:
104     respuesta = "No tienes un curso registrado. Por favor, selecciona un curso antes de realizar esta consulta."
105     update.message.reply_text(respuesta)
106     hablar(update, context, respuesta)
107
108 def hablar(update: Update, context: CallbackContext, texto: str) -> None:
109     tts = gTTS(text=texto, lang='es')
110     tts.save("respuesta.mp3")
111     context.bot.send_voice(chat_id=update.message.chat_id, voice=open("respuesta.mp3", "rb"))
112
113 def start(update: Update, context: CallbackContext) -> None:
114     respuesta = '¡Hola! Soy tu bot de asistencia para el progreso de tu syllabus. ¿En qué curso puedo ayudarte hoy?'
115     update.message.reply_text(respuesta)
116     hablar(update, context, respuesta)
117
118 TELEGRAM_TOKEN = ''
119 updater = Updater(TELEGRAM_TOKEN, use_context=True)
120
121 # Obtener el despachador para registrar manejadores
122 dp = updater.dispatcher
123
124 # Registrar comandos y manejadores
125 dp.add_handler(MessageHandler(Filters.text & ~Filters.command, manejar_comandos))
126 dp.add_handler(CommandHandler("start", start))
127
128 # Iniciar el bot
129 updater.start_polling()
130
131 # Mantener el bot en ejecución
132 updater.idle()
```

Hemos aprovechado el uso de un bot implementado directamente a través de la aplicación de Telegram. Este bot ha sido fundamental en la creación de nuestro propio bot, ya que nos ha guiado en la elaboración de comandos personalizados.



**BotFather** bot

AsistenteVirtual  
Voice message

I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual. **Thursday**

You can control me by sending these commands:

- /newbot - create a new bot
- /mybots - edit your bots

**Edit Bots**

- /setname - change a bot's name
- /setdescription - change bot description
- /setabouttext - change bot about info
- /setuserpic - change bot profile photo
- /setcommands - change the list of commands
- /deletebot - delete a bot

**Bot Settings**

- /token - generate authorization token
- /revoke - revoke bot access token
- /setinline - toggle inline mode
- /setinlinegeo - toggle inline location requests
- /setinlinefeedback - change inline feedback settings
- /setjoininggroups - can your bot be added to groups?
- /setprivacy - toggle privacy mode in groups

Message

**BotFather** bot

AsistenteVirtual  
Voice message

**Thursday**

- /mygames - edit your games
- /newgame - create a new game
- /listgames - get a list of your games
- /editgame - edit a game
- /deletegame - delete an existing game

21:47

/newbot 22:20 ✓

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 22:20

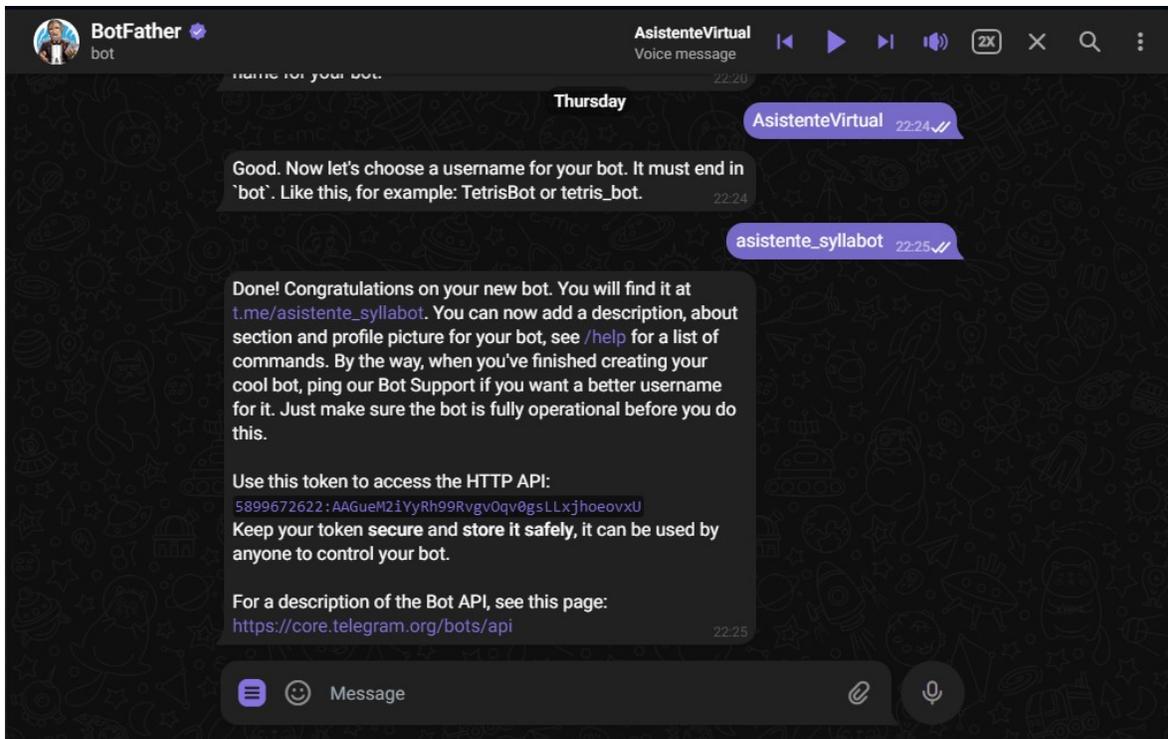
AsistenteVirtual 22:24 ✓

Good. Now let's choose a username for your bot. It must end in 'bot'. Like this, for example: TetrisBot or tetris\_bot. 22:24

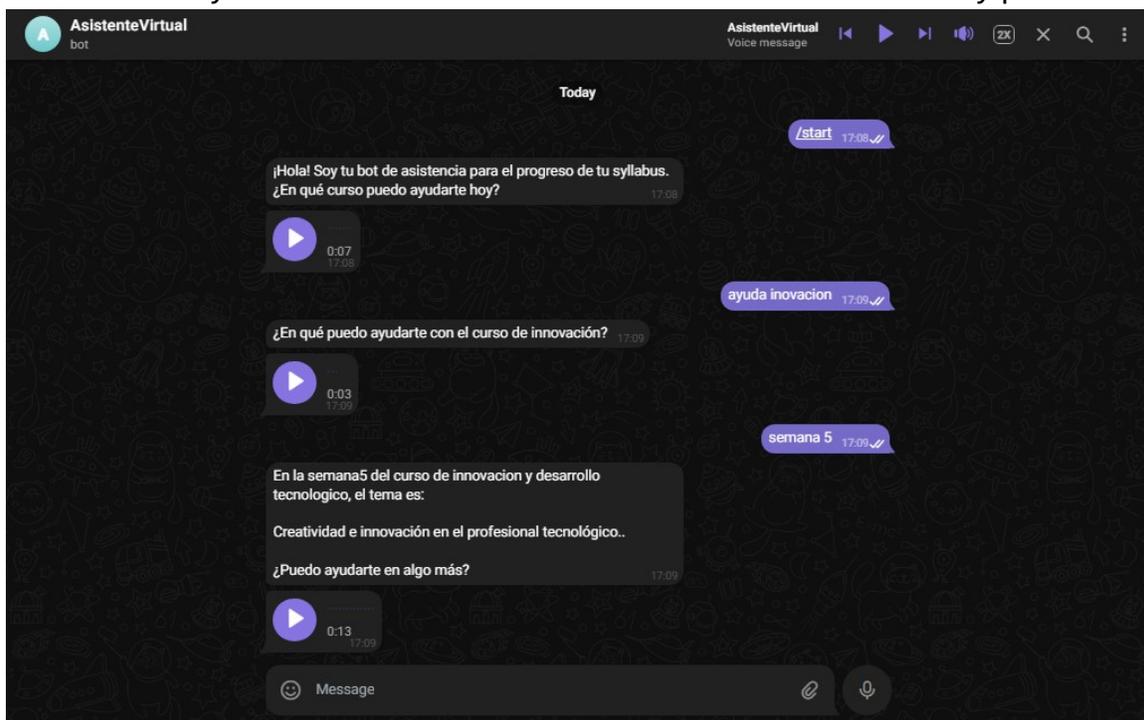
asistente\_syllabot 22:25 ✓

Done! Congratulations on your new bot. You will find it at [t.me/asistente\\_syllabot](https://t.me/asistente_syllabot). You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do

Message



Este bot personalizado ha sido desarrollado con el propósito específico de proporcionarnos información detallada sobre el sílabo de manera rápida y accesible. La combinación de la potencia de Python y la plataforma de mensajería instantánea de Telegram nos brinda una solución integral para obtener asesoramiento y datos relacionados con el sílabo de manera intuitiva y práctica.



### **Resultados Alcanzados:**

El asistente virtual desarrollado demostró con éxito su capacidad para proporcionar información detallada sobre el sílabo de los cursos consultados por los usuarios. Además, la funcionalidad de envío de archivos de audio permitió una experiencia de usuario más completa y versátil. Se presentan ejemplos concretos de interacciones exitosas y se discuten las estadísticas relacionadas con el uso del bot.

### **Discusión:**

La implementación del asistente virtual representa un avance significativo en la mejora de la accesibilidad a la información académica. Sin embargo, se discuten posibles desafíos y limitaciones encontradas durante el desarrollo, así como áreas de mejora potencial. Se reflexiona sobre la eficacia del bot en términos de usabilidad y la respuesta de los usuarios.

### **Referencias Bibliográficas:**

AsCodigo. (14 de enero de 2021). 📺 *Cómo hacer un BOT en TELEGRAM (CON PYTHON 🐍 MUY FACIL)*. [Archivo de Vídeo]. Youtube.

<https://www.youtube.com/watch?v=wxOeEb2EISU&t=6s>

AsCodigo. (14 de septiembre de 2023). *Crea tu Primer Bot de Telegram con Python: Paso a Paso 🐍🤖*. [Archivo de Vídeo]. Youtube.

<https://www.youtube.com/watch?v=U8vj9itYdgl&t=1408s>