



ESCUELA POLITÉCNICA SUPERIOR  
DE SEVILLA

GRADO EN INGENIERÍA ELECTRÓNICA  
TRABAJO FINAL DE GRADO

# Análisis de Necesidades, Diseño y Desarrollo de un Chatbot para la Universidad de Sevilla

Autor: Arturo Olmedo Carpio

Tutor: Antonio Martín Montes

Curso 2020 - 2021

## Resumen

En la actualidad el uso de tecnologías en la vida cotidiana cada vez está más extendido y se usan en múltiples aplicaciones de diferente tipo. Por este motivo los chatbots se han convertido en una herramienta muy utilizada para proporcionar ayuda a los usuarios de todo tipo de aplicaciones y servicios. Estos agentes virtuales ofrecen respuestas e información a las consultas de las diferentes personas de manera automática, proporcionando soporte en cualquier momento y reduciendo los recursos que una corporación o institución educativa necesita en su servicio de atención al cliente. Estos se usan en diversos ámbitos como las compras online, reservar una habitación de hotel, un vuelo, proceso de matriculación en la Universidad, etc.

En este proyecto se explica el funcionamiento de los chatbots y su avance a lo largo de los años para introducir al lector en esta nueva forma de inteligencia artificial, para después estudiar las funciones que pueden realizar y crear un ejemplo real de asistente virtual. Para ello se estudiarán las diferentes plataformas para el desarrollo del bot, repasando los pros y contras de cada una de ellas, para elegir la más conveniente según las necesidades del proyecto.

Una vez desarrollado el bot habrá una primera fase de puesta en preproducción para que diferentes usuarios lo utilicen realizando pruebas, análisis de resultados, corrección de errores, con el fin de mejorar la base de conocimientos de la herramienta siguiendo un proceso mejora continua de Deming.

En nuestro caso, se desarrolla un chatbot para el Área de Atención y Orientación del Estudiante de la Universidad de Sevilla con el fin de reducir los recursos necesarios para atender todas las dudas de los estudiantes y mejorar la experiencia de estos a la hora de realizar cualquier consulta. Ahora mismo con el COVID-19 reducir el contacto es primordial y gracias a implementar un asistente virtual se reducirán las personas que acuden presencialmente a la Universidad sin perder calidad en la atención ofrecida a los estudiantes.

## Contenido

1. INTRODUCCIÓN Y OBJETIVOS.....	11
1.1 Introducción .....	11
1.2 Objetivos .....	11
1.3 Estructura .....	12
2. ESTADO DEL ARTE.....	13
2.1.1 Inicios .....	13
2.1.2 Estado Actual de chatbots a nivel mundial .....	14
2.1.2 Chatbots en entornos universitarios .....	15
2.2. Tecnología de estandarización .....	16
2.3. Comparativa de plataformas.....	17
3.PERSPECTIVA Y FUNCIONES DEL PRODUCTO .....	23
3.1 Descripción general .....	24
3.2 Requisitos .....	25
3.3 Metodología .....	26
3.4 Planificación y costes .....	26
3.4.1 Planificación .....	26
3.4.2 Recursos necesarios .....	29
3.4.3 Costes .....	30
3.4.4 Análisis de riesgo .....	30
4. DISEÑO DE LA ARQUITECTURA DEL CHATBOT .....	32
4.1. Entornos de trabajo.....	32
4.1.2 Chatfuel.....	32
4.1.2 Bases de datos.....	42
4.1.3 Visual Code Studio.....	45
5. DETALLES DE LA IMPLMETANCIÓN.....	46
5.1 Procedimientos asociados .....	46
6. DESARROLLO Y DISEÑO.....	47
6.1 Chatbot.....	47
6.1.1 Bloques y secuencias .....	47
6.1.2 Inteligencia artificial.....	61

---

6.1.3 Roles de usuario .....	62
6.1.4 Acceso al chatbot.....	62
6.2 Páginas web.....	63
6.3 Base de datos.....	91
7. PRUEBAS Y ANÁLISIS DE RESULTADOS .....	96
7.1 Pruebas del servidor de producción .....	96
7.2 Respaldo de información y políticas de seguridad.....	103
7.2.1 Recopilación de información.....	103
7.2.2 Uso de datos personales.....	103
7.2.3 Divulgación de datos personales .....	104
7.2.4 Gestión del cambio .....	104
8. FUTURAS AMPLIACIONES Y MEJORAS .....	105
8.1 Mejoras .....	105
8.2 Conclusiones .....	105
Bibliografía .....	107

## Índice de figuras

Figura 1: El primer chatbot “ELIZA” .....	13
Figura 2: Conversación entre un humano y “ALICE” .....	14
Figura 3: Gráfica del tamaño del mercado de los chatbots en Norte América. ....	15
Figura 4: Gráfica de la cuota de mercado de los chatbots a nivel mundial. ....	15
Figura 5: Elementos básicos de un intent en Dialogflow.....	18
Figura 6: Flujo de conversación en Chatfuel.....	20
Figura 7: Página de IA del agente virtual de Chatfuel. ....	20
Figura 8: Código escrito en lenguaje AIML.....	21
Figura 9: Nodos de dialogo en Watson Conversation de IBM.....	22
Figura 10: Diagrama Gantt de las fases del proyecto .....	29
Figura 11: Página de inicio de Chatfuel para crear un chatbot. ....	32
Figura 12: Apartado “Grow” de Chatfuel. ....	<b>¡Error! Marcador no definido.</b>
Figura 13: Sección Flows de Chatfuel. ....	34
Figura 14: Menú persistente de Chatfuel. ....	35
Figura 15: “Welcome Message” de Chatfuel. ....	36
Figura 16: Tipos de contenido y acciones de los bloques de Chatfuel. ....	36
Figura 17: Ejemplo de galería de Chatfuel.....	37
Figura 18: Ejemplo de respuestas rápidas de Chatfuel .....	37
Figura 19: Ejemplo de guardar datos del usuario en Chatfuel.....	38
Figura 20: Ejemplo de asignar atributos de usuario en Chatfuel.....	38
Figura 21: Inteligencia artificial de Chatfuel. ....	40
Figura 22: Chat en vivo de Chatfuel. ....	40
Figura 23: Apartado para agrupar usuarios de Chatfuel. ....	41
Figura 24: Graficas de análisis de usuarios de Chatfuel.....	42
Figura 25: Ejemplo de conexión con base de datos de MySQL Workbench. ....	43
Figura 26: Ejemplo de crear tablas en MySQL Workbench. ....	44
Figura 27: Ejemplo de datos de una tabla en MySQL Workbench. ....	44
Figura 28: Ejemplo de código en Visual Code Studio. ....	45
Figura 29: Estructura básica de un chatbot.....	46
Figura 30: Redireccionando saludo en función de la hora. ....	50
Figura 31: Ejemplo de saludo del bot. ....	50
Figura 32.: Suscripción a secuencia de menú. ....	51
Figura 33: Ultimo bloque de la secuencia “Default Answer”. ....	51

<i>Figura 34: Menú principal del chatbot.</i>	52
<i>Figura 35: Preguntas para guardar los datos de contacto del usuario.</i>	53
<i>Figura 36: Conexión con página web mediante el método POST.</i>	53
<i>Figura 37: Correo de notificación al usuario.</i>	54
<i>Figura 38: Redirección a otro bloque según atributo "time24h".</i>	54
<i>Figura 39: Diagrama de flujo de conversación de bloques de chat en vivo.</i>	55
<i>Figura 40: Bloque de videos.</i>	56
<i>Figura 41: Opciones del bloque "Todos videos vistos".</i>	57
<i>Figura 42: Bloque "Elegir música 1".</i>	57
<i>Figura 43: Bloque "Elegir música 2".</i>	58
<i>Figura 44: Suscripción a la secuencia "opinión".</i>	59
<i>Figura 45: Formulario para obtener la opinión del usuario.</i>	59
<i>Figura 46: Creación de motor de búsqueda de Google.</i>	60
<i>Figura 47: Buscador de Chatfuel.</i>	60
<i>Figura 48: Inicio de sesión en página web de tutorías.</i>	61
<i>Figura 49: Frases de entrenamiento para los saludos.</i>	61
<i>Figura 50: Frases de entrenamiento para bibliotecas abiertas los fines de semana.</i>	61
<i>Figura 51: Frases de entrenamiento para música.</i>	62
<i>Figura 52: Datos de conexión con la base de datos.</i>	63
<i>Figura 53: Código para recibir datos mediante método POST.</i>	63
<i>Figura 54: Enviar datos mediante método POST en Chatfuel.</i>	64
<i>Figura 55: Cambio de idioma del atributo genero.</i>	64
<i>Figura 56: Eliminar fecha del atributo dateFormatted.</i>	64
<i>Figura 57: Eliminar zona horaria del atributo "dateFormatted".</i>	65
<i>Figura 58: Separar abreviaturas del atributo "locale".</i>	65
<i>Figura 59: Asignar nombre completo a la variable país según su abreviatura.</i>	65
<i>Figura 60: Conexión con base de datos.</i>	65
<i>Figura 61: Crear comando para introducir datos en tabla de la base de datos MySQL.</i>	66
<i>Figura 62: Realizar conexión y notificar al chat mediante JSON.</i>	66
<i>Figura 63: Comando para consultar datos de una tabla de la base de datos.</i>	67
<i>Figura 64: Crear variable para enviar las consultas mediante el chat.</i>	67
<i>Figura 65: Crear mensaje en caso de no haber consultas en la base datos.</i>	67
<i>Figura 66: Ejemplo de etiqueta en HTML.</i>	68
<i>Figura 67: Ejemplo de atributo en HTML.</i>	68

Figura 68: Metadatos de visualización de la página web. ....	72
Figura 69: Título de la pestaña del navegador. ....	72
Figura 70: Utilizar estilos mediante archivos CSS. ....	72
Figura 71: Activar extensiones SDK de Facebook Messenger. ....	72
Figura 72: Atributos básicos del formulario. ....	72
Figura 73: Cuadros de texto del formulario.....	73
Figura 74: Menú desplegable del formulario.....	73
Figura 75: “Checkbox” del formulario.....	74
Figura 76: Enviar datos ocultos mediante el formulario. ....	74
Figura 77: Recibir datos del formulario mediante el método POST.....	74
Figura 78: Crear primera parte del url para enviar datos a Chatfuel. ....	75
Figura 79: Crear segunda parte del url para enviar datos a Chatfuel. ....	75
Figura 80: Comandos “curl” para configurar y realizar la conexión con la página web. ...	76
Figura 81: Código CSS creando la presentación del menú desplegable del formulario. ....	80
Figura 82: Principio del código del formulario de registro.....	81
Figura 83: Código que incluye los archivos JavaScript. ....	81
Figura 84: Menú de la interfaz del rol de administrador.....	81
Figura 85 : Sección de administradores del rol administrador. ....	82
Figura 86 : Código para crear la tabla de la sección de administradores.....	82
Figura 87: Sección de estudiante del rol de administrador. ....	82
Figura 88: Código para la crear la tabla de la sección de estudiantes .....	83
Figura 89: Sección de tutorías del rol de administrador. ....	83
Figura 90: Página para mover reservas de tutorías .....	83
Figura 91: Menú de la interfaz del rol de estudiante. ....	84
Figura 92: Código para crear lista con tutorías disponibles. ....	85
Figura 93: Lista con tutorías reservadas por el alumno. ....	85
Figura 94: Página de tutorías del rol de profesor. ....	86
Figura 95: Código para generar la lista de las tutorías .....	86
Figura 96: Código en JavaScript para enviar datos a archivos PHP. ....	88
Figura 97: Código en PHP para generar el botón que activa la función en JavaScript. ....	88
Figura 98: Código con la conexión a la base de datos. ....	90
Figura 99: Código que recibe los datos del archivo JavaScript. ....	90
Figura 100: Código que realiza la consulta a la base de datos. ....	91
Figura 101: Chat para prueba del bot con usuario 1 .....	96

<i>Figura 102: Chat para prueba del bot con usuario 1 .....</i>	<i>97</i>
<i>Figura 103: Chat para prueba del bot con usuario 1 .....</i>	<i>97</i>
<i>Figura 104: Chat para prueba del bot con el usuario 2.....</i>	<i>98</i>
<i>Figura 105:Chat para prueba del bot con el usuario 2.....</i>	<i>99</i>
<i>Figura 106:Chat para prueba del bot con el usuario 2.....</i>	<i>99</i>
<i>Figura 107:Chat para prueba del bot con el usuario 2.....</i>	<i>100</i>
<i>Figura 108:Chat para prueba del bot con el usuario 2.....</i>	<i>100</i>
<i>Figura 109:Chat para prueba del bot con el usuario 3.....</i>	<i>101</i>
<i>Figura 110:Chat para prueba del bot con el usuario 3.....</i>	<i>102</i>
<i>Figura 111:Chat para prueba del bot con el usuario 4.....</i>	<i>102</i>
<i>Figura 112: Chat para prueba del bot con el usuario 4.....</i>	<i>103</i>
<i>Figura 113:Tareas asociadas a la gestión del cambio.....</i>	<i>104</i>



## **Índice de tablas**

<i>Tabla 1: Ventajas y desventajas de las diferentes plataformas para el desarrollo de chatbots</i>	22
<i>Tabla 2: Requisitos del chatbot del proyecto.</i>	25
<i>Tabla 3: Lista de fases y subtareas del proyecto.</i>	28
<i>Tabla 4: Coste del personal.</i>	30
<i>Tabla 5: Coste del equipo.</i>	30
<i>Tabla 6: Análisis de riesgos.</i>	31
<i>Tabla 7: Atributos que se resetean en el “Welcome Message”</i>	47
<i>Tabla 8: Atributos predefinidos de Chatfuel.</i>	48
<i>Tabla 9: Atributos de fecha y hora de una API externa.</i>	49
<i>Tabla 10: Etiquetas de preámbulo y raíz de HTML</i>	68
<i>Tabla 11: Etiquetas de metadatos de HTML.</i>	68
<i>Tabla 12: Etiquetas de secciones de HTML.</i>	69
<i>Tabla 13: Etiquetas de bloques de contenido de HTML.</i>	69
<i>Tabla 14: Etiquetas de texto de HTML.</i>	70
<i>Tabla 15: Etiquetas de tablas de HTML.</i>	70
<i>Tabla 16: Etiquetas de formularios de HTML.</i>	71
<i>Tabla 17: Propiedades de los caracteres de la página web en lenguaje CSS.</i>	76
<i>Tabla 18: Propiedades del texto de la página web en lenguaje CSS.</i>	77
<i>Tabla 19: Propiedades de los bordes de la página web en lenguaje CSS.</i>	77
<i>Tabla 20: Propiedades de los márgenes exteriores de la página web en lenguaje CSS.</i>	78
<i>Tabla 21: Propiedades de los márgenes interiores de la página web en lenguaje CSS.</i>	78
<i>Tabla 22: Propiedades del fondo de la página web en lenguaje CSS.</i>	78
<i>Tabla 23: Propiedades de las tablas de la página web en lenguaje CSS.</i>	79
<i>Tabla 24: Propiedades de las cajas de la página web en lenguaje CSS.</i>	79
<i>Tabla 25: Archivos y funciones del rol de administrador.</i>	84
<i>Tabla 26: Archivos y funciones del rol de estudiante.</i>	85
<i>Tabla 27: Archivos y funciones del rol de profesor</i>	86
<i>Tabla 28: Funciones de la biblioteca jQuery</i>	88
<i>Tabla 29: Archivos y funciones JavaScript</i>	90
<i>Tabla 30: Descripción de las columnas de tabla "Opiniones" de la base datos</i>	92
<i>Tabla 31: Descripción de las columnas de tabla "Usuarios" de la base de datos</i>	93
<i>Tabla 32: Descripción de las columnas de tabla "usuariostutoria" de la base de datos</i>	93
<i>Tabla 33: Descripción de las columnas de tabla "datosusuario" de la base de datos.</i>	94



---

*Tabla 34: Descripción de las columnas de tabla "horarios" de la base de datos ..... 95*

# 1. INTRODUCCIÓN Y OBJETIVOS

## 1.1 Introducción

En este proyecto se analiza el avance de herramientas software basadas en técnicas de inteligencia artificial denominadas chatbot, su evolución y como se ha extendido su uso en la actualidad. Un chatbot es un programa que actúa como un asistente virtual para facilitar el acceso a la información que se solicite de forma sencilla, como si fuera una persona real. Es decir, el usuario interactúa con el programa mediante un chat y dependiendo de palabras claves o el rumbo de la conversación el bot le proporciona información de diferentes formas: texto, menús, imágenes, audio, redirigiéndolos otras a páginas web, etc.

El desarrollo de chatbots en la industria y empresa moderna, supone diferentes beneficios. A nivel económico es muy rentable ya que se consigue la optimización de los recursos humanos necesarios para dar atención a los clientes. Asimismo, el asistente virtual ofrece diferentes ventajas, estar disponible todo el tiempo mientras que para conseguir esto con trabajadores tendría que haber varios turnos. En cuanto a la experiencia de los usuarios, un asistente virtual permite una atención más eficaz y personalizada ya que la cantidad de información que pueden ofrecer es infinita y el tiempo de respuesta es inmediato. Para las empresas analizar las interacciones que se realizan mediante chat es mucho más sencillo que si se realizaran por ejemplo por teléfono.

Por estos los asistentes virtuales se utilizan cada vez más, un ejemplo de esto son las páginas web de venta online de productos o servicios también llamadas e-commerce. La gran mayoría de tienda físicas ya disponen del servicio de venta online aparte de las que solo venden de esta forma, por lo que hay un porcentaje importante de las ventas mediante internet y más ahora con el COVID19. Al entrar en estas páginas suele abrirse un chat en una esquina que nos ayuda a encontrar el producto o servicio que busca el cliente, haciendo nuestra experiencia más fácil. Con esta pequeña inteligencia artificial se accede más rápido a la información que se necesita mejorando así el servicio de estas páginas de ventas.

Otro ejemplo de este tipo de inteligencias artificiales son asistentes que vienen incorporados en los teléfonos móviles como Siri de Apple o el asistente de Google a los que para hacer aún más simple la interacción con las personas se les ha añadido reconocimiento por voz por lo que ya no hace falta ni escribir, simplemente tener una conversación con ellos para que nos proporcionen ayuda. Este tipo de asistentes también se están empezando a usar en el hogar como Alexa de Amazon, que cada vez estas más extendido y te permite controlar las diferentes funciones de tu casa, si esta domotizada, como encender o apagar luces o cualquier electrodoméstico simplemente usando tu voz.

En este trabajo se estudian las diferentes características de los chatbots y sus aplicaciones, así como su arquitectura, módulos, entornos de trabajo, modos de programación para finalmente desarrollar uno desde cero.

## 1.2 Objetivos

El objetivo general de este proyecto es crear un bot conversacional para el Área de Orientación y Atención al Estudiante de la Universidad de Sevilla. Se diseñará la arquitectura y el flujo de la conversación del asistente mediante una de las plataformas disponibles para ello y se entrenará para que la conversación sea coherente y fluida para obtener la información que el usuario busca.

Centrando en objetivos más específicos de este proyecto se pueden destacar:

- Poner el contexto los chatbot en el ámbito actual repasando su historia y avance hasta la fecha y conocer sus diferentes usos a día de hoy.
- Hacer una comparativa de las aplicaciones más conocidas para su desarrollo exponiendo ventajas e inconvenientes para poder seleccionar la más optima según las necesidades y recursos disponibles.
- Explicar las tecnologías básicas que utilizan los chatbot para comprender mejor su funcionamiento
- Comunicar el chat bot con un back-end mediante una API en lenguaje JSON para acceder o guardar información valiosa.
- Integrar el bot con aplicaciones externas y automatizar tareas como enviar correos con publicidad, guardar información, subir publicaciones en redes sociales, etc.
- Calcular el tiempo y los costes del proyecto para una vez terminado compararlo con los reales para comprobar si se hizo una buena estimación.
- Realizar pruebas sobre el chatbot para corregir posibles errores, analizar las diferentes interacciones con los usuarios para implementar posibles mejoras.

### 1.3 Estructura

Este trabajo consta de 6 apartados que tratan sobre lo siguiente:

- El primero expone una breve introducción sobre el tema a tratar, los chatbots. Se enumeran los objetivos del proyecto y la estructura de este.
- El segundo expone la historia de los chatbots y sus usos desde sus inicios hasta la actualidad, comentando algunos de los chatbots más populares. También se estudian las diferentes plataformas en las que se pueden implementar y la tecnología de procesamiento del lenguaje natural que utilizan para poder entender los mensajes de los usuarios y responder acorde.
- El tercero establece unos objetivos y funcionalidades que tiene que cumplir nuestro chatbot y estimar un tiempo y costes para realizarlo.
- El cuarto explica la arquitectura básica de un chatbot comentando sus diferentes bloques y se analizará más en detalle el framework que se va a utilizar para el desarrollo del bot.
- El quinto explica las funciones u operaciones que realizará el chatbot como enviar fotos, guardar información del usuario, enviar notificaciones, etc.
- El sexto apartado se explica el desarrollo de un chatbot con una aplicación real. Se le da personalidad a este creando el contenido de sus respuestas según la entrada y el tipo de usuario que sea. Finalmente se tiene un chatbot real junto con su manual de usuario.
- El séptimo apartado se prueba el chatbot analizando la forma de desenvolverse de este respecto a varias conversaciones con distintos tipos de usuarios para poder corregir errores e implementar mejoras. Se revisan los tiempos y costes reales para comprobar si se aproximan a los que se estimaron al principio del proyecto.
- Por último, expondremos las conclusiones y trabajos futuros relacionados con el proyecto realizado.

## 2. ESTADO DEL ARTE

Un chatbot es una herramienta software que interactúa con los usuarios, ya sea mediante texto o voz, y mediante pautas y reglas definidas por el desarrollador o aprendidas con el tiempo por el mismo programa (“machine learning”) consigue dar asistencia al usuario como si este estuviera comunicándose con un ser humano. El chatbot mediante el procesamiento del lenguaje natural (PLN) entiende el mensaje recibido y responde acorde a las normas que el programador le impone.

### 2.1.1 Inicios

En 1950, el matemático y uno de los padres de la ciencia de la computación Alan Turing desarrollo la prueba de Turing para determinar la capacidad de una máquina de mostrar un comportamiento inteligente similar al de un humano. En esta prueba una persona que tiene el papel de juez sabe que uno de los dos interlocutores con los que mantiene una conversación escrita es un maquina y el otro una persona. El juez tiene que realizar preguntas a ambos para determinar cuál es la maquina y en caso de no poder distinguir entre la maquina y el humano se considera que la maquina es inteligente y es capaz de “pensar” o mejor dicho de imitar el razonamiento humano. La prueba de Turing se sigue realizando en la actualidad, todos los años se realiza en el Premio Loebner para determinar cuál es la mejor inteligencia artificial, aunque hasta la fecha ninguna ha superado la prueba.

En 1966, el informático Joseph Weizenbaum creo en el MIT el primer chatbot: ELIZA. Este programa reconocía palabras clave de los usuarios con los que interactuaba y según estas elegía sus respuestas. Fue la primera máquina en usar el procesamiento de lenguaje natural y gano gran popularidad por imitar el papel de un psiquiatra. Posteriormente en 1972 la Universidad de Stanford creo a Parry siguiendo los principios de ELIZA, usando como modelo a un paciente con esquizofrenia paranoide. Eliza y Parry se conectaron a través de ARPANET y mantuvieron una conversación.

```

Welcome to
EEEEEE LL      IIII  ZZZZZZ  AAAAA
EE      LL      II   ZZ      AA  AA
EEEEEE LL      II   ZZ      AAAAAA
EE      LL      II   ZZ      AA  AA
EEEEEE LLLLLL  IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
  
```

*Figura 1: El primer chatbot “ELIZA”*

En 1988, el programador británico Rollo Carpenter creo Jabberwacky. Fue el primer chatbot diseñado para imitar la voz humana y estaba dirigido únicamente al entretenimiento. Otro

chatbot a destacar es Chatterbot, que estaba integrado en un juego muy popular de la época llamado Tymimud. Hacia el papel de un jugador virtual y era muy interesante por que realizaba la prueba de Turing ya que se suponía que todos los usuarios de este juego eran personas reales.

En 1995 nace ALICE (Artificial Linguistic Internet Computer Entity) usando un nuevo lenguaje de programación, el AIML (Artificial Intelligence Mark-up Language) desarrollado por Richard Wallace. Este lenguaje fue diseñado específicamente para la creación de chatbots. En este lenguaje si las diferentes entradas del usuario coinciden con los patrones el chatbot respondía con unas plantillas establecidas.

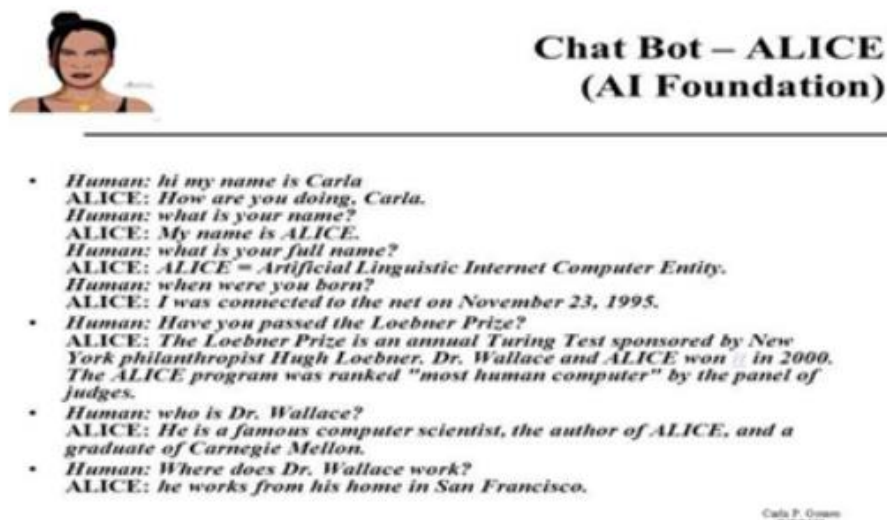


Figura 2: Conversación entre un humano y “ALICE”

Posteriormente grandes compañías han ido incorporando chatbot a sus servicios. Apple creó SIRI como el primer asistente virtual en un teléfono móvil, en el mismo año IBM desarrolló a Watson, y muchos ejemplos más como Cortana de Microsoft, Alexa de Amazon, Google Assistant de Google o Bixby de Samsung.

### 2.1.2 Estado Actual de chatbots a nivel mundial

En la actualidad los chatbot están presentes en múltiples sectores, Figuras 3 y Figura 4:

-En la industria financiera muchos bancos como BBVA o Caixa Bank los usan para agilizar el servicio a sus clientes.

-Los comercios electrónicos dando soporte a los usuarios en las compras y recabando información sobre las preferencias de los clientes y de los problemas a la hora de comprar vía online consiguiendo aumentar las ventas y mejorar la experiencia del usuario

-En salud para fijar citas médicas de los pacientes y procesar los datos clínicos. También se usan para resolver preguntas frecuentes sobre síntomas contestándoles usando su base de datos o rediriéndoles a un médico especialista para agilizar los colapsos de la red sanitaria.

-En el servicio al cliente de múltiples compañías, las llamadas de atención al cliente cuestan millones al año a la empresa. Usando los chatbots para resolver los problemas simples permitiendo a los agentes solucionar los complejos permite ahorrarse a las empresas gran cantidad de recursos.

-En el turismo para realizar las reservas de hoteles, vuelos, restaurantes y dar su opinión sobre el servicio recibido permitiendo mejorar el servicio. Gracias a esto simplifica considerablemente la planificación de un viaje mejorando la experiencia.

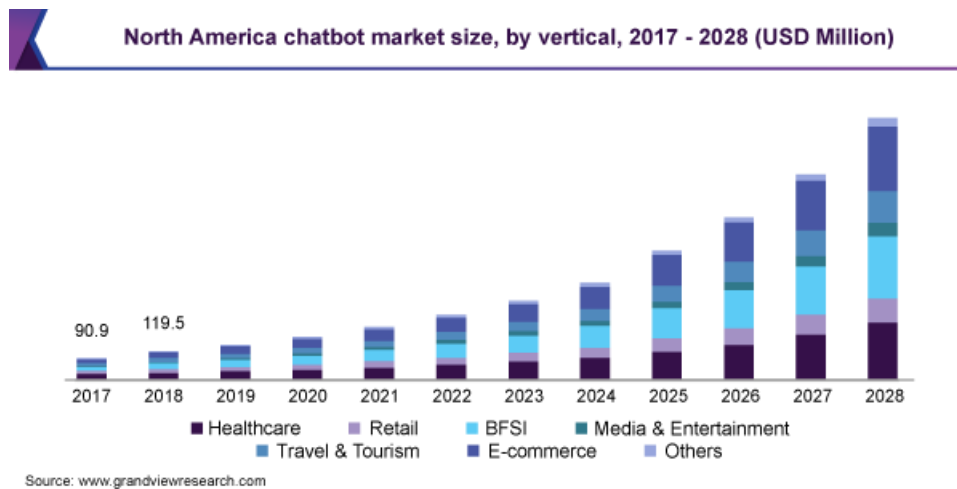


Figura 3: Gráfica del tamaño del mercado de los chatbots en Norte América.

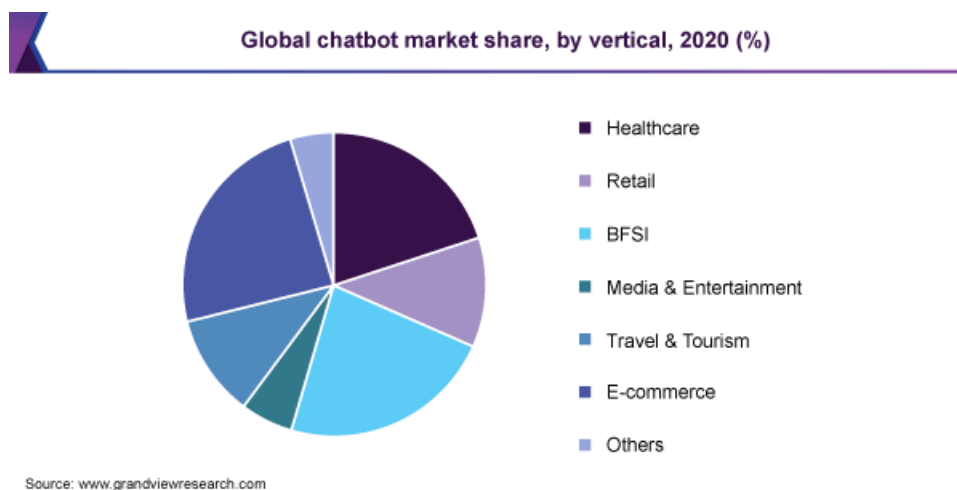


Figura 4: Gráfica de la cuota de mercado de los chatbots a nivel mundial.

### 2.1.2 Chatbots en entornos universitarios

Debido a sus múltiples posibilidades los chatbot también se están implantando en el ámbito universitario. Un ejemplo de esto es MARTHA en la Universidad George Washington, un chatbot que implanto en forma de prueba y el 89% de los usuarios opino que debería quedarse de forma permanente.

En las universidades españolas se está aumentando su uso cada vez más. La Universidad Politécnica de Valencia creo a PAU, un asistente virtual para resolver las consultas de los estudiantes acerca de la las pruebas de acceso a la universidad. PAU en un mes atiende a casi de 7.000 usuarios mejorando la atención hacia alumnos.

En la Universidad Complutense de Madrid han creado a Carol encargada de solventar todas las dudas que tengan los nuevos alumnos a la hora de inscribirse en la universidad como

notas de corte, asignaturas y además es capaz de realizar acciones como las matrículas de los alumnos.

En la Universidad de Murcia se implantó a Lola que el año pasado resolvió más de 28.000 consultas sobre el acceso a la universidad. Para crearla se utilizó la plataforma Dialogflow de Google y tuvo una mención en la “Harvard Business Review” por su gran éxito.

## 2.2. Tecnología de estandarización

Los chatbots necesitan de cierta tecnología para poder comprender el lenguaje de los humanos y convertirlo en un lenguaje interpretable para las máquinas, este proceso se llama Procesamiento del Lenguaje Natural (PLN). Comprender los miles de palabras, la estructura de la oración, el trasfondo y la intención puede llegar a ser bastante complejo. Estos datos conversacionales son datos no estructurados y son difíciles de ordenar y manejar por eso se suele utilizar el aprendizaje automático gracias al cual pueden ir aprendiendo de las interacciones que realizan.

El PLN realiza diferentes tareas para conseguir simplificar la decodificación del mensaje:

- El bot primero normaliza el mensaje poniendo todos los caracteres de igual forma, es decir convirtiendo todos en minúsculas. Después descompone el mensaje completo en grupos o estructuras más pequeñas, separa el mensaje en oraciones y a continuación las oraciones en palabras para facilitar el manejo de la información. Este proceso se llama tokenización.
- El siguiente paso es eliminar las palabras que no tienen información valiosa para comprender el mensaje como algunos pronombres, artículos o conjunciones como por ejemplo “el”, “como” o “y”. A continuación, se halla el lema de todas las formas flexionadas de las palabras, es decir los plurales, conjugaciones y demás se transforman en la forma de la palabra que se encuentra en el diccionario, para ello se usa una base de datos. Posteriormente se asigna una categoría gramatical a las palabras según sean sustantivos, adjetivos, verbos, etc.
- Se utiliza el modelo de Bolsa de palabras en el cual se obvia el orden de las palabras y solo se contabiliza el número de veces que se repite una palabra para determinar la frecuencia de término y designar las palabras con más importancia. Esta bolsa de palabras se representa mediante un vector. A veces los conceptos que permiten distinguir significados están formados por más de una palabra, como por ejemplo “Nueva York” que es un bigrama. Por último, se clasifican las entidades del mensaje que son elementos que existen de verdad como lugares, personas, fechas, empresas, etc, que facilitará muchas funciones del bot como por ejemplo al comprar una entrada de cine si el bot identifica el nombre de la sala de cine, el de la película y la fecha será capaz de comprar la entrada sin problema.

Otra parte importante para el procesamiento de la información son las expresiones regulares, el bot tiene una base de datos donde las almacena y sirven para captar patrones que aparecen en el texto como por ejemplo “Hola, buenos días”. Al tener almacenada esta expresión es más fácil para la máquina “comprender” su significado y responder correctamente.



Este conjunto de procesos es el primer paso del bot para procesar la información del mensaje y convertirla a un lenguaje que él pueda “entender” y manejar para poder responder e interactuar con las personas. Ya se ha realizado el análisis morfológico del mensaje.

A continuación, viene otro elemento muy importante para los bots que es la comprensión del lenguaje natural (CLN) y que va a realizar las siguientes funciones:

- Análisis sintáctico: Analiza la estructura de las oraciones que componen el mensaje según el modelo o reglas que utilicen, es decir si las reglas lingüísticas las ha escrito el desarrollador o las ha aprendido el bot por su cuenta mediante el aprendizaje automático.
- Análisis semántico: Elimina las ambigüedades y determina el significado de las oraciones
- Análisis pragmático: Tiene en cuenta el contexto, la intención de la oración según antecedentes de conversaciones anteriores con el usuario, el tono de este o el lenguaje figurado como la metáfora e ironía para determinar el significado final de la oración.

Gracias a todas estas funciones el bot es capaz de entender el mensaje para poder contestarlo.

La respuesta de este es la última parte de las funciones de los bots, la generación de lenguaje natural (GLN). En este proceso el bot genera el mensaje adecuado para contestar según los datos recabados en la comprensión del texto de la persona con la que interactúa. Estos datos contienen múltiples parámetros para que el bot pueda seleccionar según las reglas lingüísticas o su aprendizaje automático.

No todos los bot utilizan todas estas técnicas, hay entornos de desarrollo que simplifican mucho el proceso y solo con palabras clave y frases de entrenamiento a las que se les asignan respuestas o acciones cumplen las funciones básicas de un bot.

### 2.3. Comparativa de plataformas

A continuación, se van a describir algunas de las plataformas de desarrollo más conocidas de chatbots, estas proporcionan herramientas y facilitan el desarrollo permitiendo crear un bot sin tener grandes conocimientos de programación.

#### Dialogflow

Es la plataforma de la compañía Google desde que la compro en 2016. Esta herramienta para la creación de chatbots cuenta con múltiples funcionalidades para simplificar el proceso:

- Proporciona mecanismo de Procesamiento de Lenguaje Natural para extraer los parámetros de la conversación y Machine Learning que permite que el bot se entrene mediante las conversaciones que tenga con los usuarios o incluso subiendo archivos de texto a la plataforma, consiguiendo entender la misma pregunta de distintas formas.
- Se puede integrar en múltiples canales de comunicación como Facebook, Skype, Telegram y muchos más pudiendo llegar a gran cantidad de usuarios.
- Proporciona mecanismos de reconocimiento de voz por lo que podemos conectarlo con Google Assistant y que no sea necesario escribir para comunicarnos con él.

- Tiene una versión gratuita con la que hacer bot más simples, aunque cuenta con una versión de pago con más funciones.

Ahora se van a explicar los conceptos básicos que usamos para crear a nuestro agente (bot) de Dialogflow.

El primero es el intent que define la intención del usuario, según sus mensajes el agente seleccionara el intent con más coincidencia, como por ejemplo en una la página web de una universidad si el usuario pregunta: “¿Hay alguna biblioteca que este abierta los fines de semana?” el agente seleccionara el intent de bibliotecas. Un intent contiene los siguientes elementos:

- Frases de entrenamiento: Son frases que lo usuarios suelen decir para referirse a ese intent, cuando el agente detecta una de esas frases selecciona ese intent. Para que el agente detecte estas frases no tienen que ser exactamente iguales gracias al procesamiento del lenguaje.
- Acción: Se le asocia una acción a cada intent para el agente la realice cuando una frase coincida con este.
- Parámetros: Son los valores extraídos del mensaje del usuario cuando un intent coincide. Cada parámetro se organiza según su tipo de entidad que determina como se extraen y agrupan los datos. Hay varios de tipo de entidades predeterminadas como fechas, horas, lugares, etc. Pero también puedes crear entidades personalizadas como por ejemplo asignaturas para una agente virtual de una universidad. Se pueden seleccionar las acciones o las respuestas según el conjunto de parámetros que hemos recopilado del mensaje del usuario
- Respuestas: Hay que determinar las respuestas de cada intent según los parámetros obtenidos, pudiendo dar una respuesta, solicitar más datos o terminar la conversación.

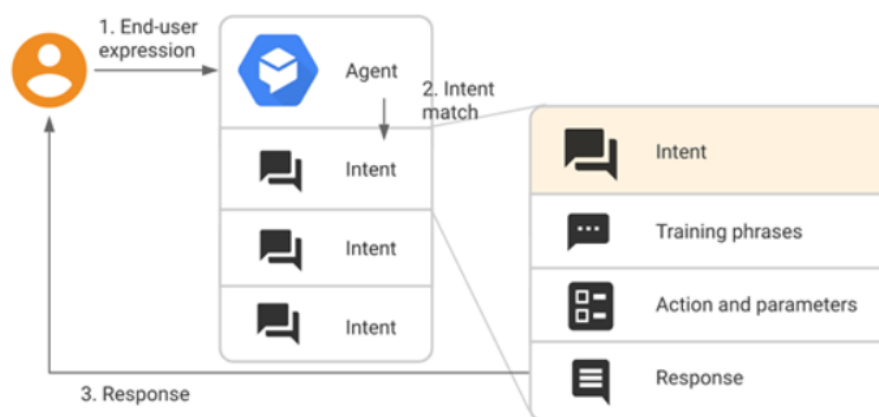


Figura 5: Elementos básicos de un intent en Dialogflow.

Los contextos se le asignan a cada intent y funcionan como los contextos de una conversación real. Estos se activan cuando coincide un intent determinado facilitando que el próximo intent coincida, por ejemplo, un usuario manda el siguiente mensaje: “Quiero hacer la automatrícula” y más tarde en la misma conversación dice “¿Hasta cuándo puedo hacerla?”, el agente gracias al contexto sabe que el usuario está preguntando por la fecha de

la automatrícula. Para poder usar esto eficientemente a los contextos se le dan unos ciclos de vida, para que si la conversación avanza se vayan actualizando los contextos.

## Chatfuel

Esta plataforma está enfocada a realizar un bot más simple sin tener muchos conocimientos de programación. Estas son algunas de sus características:

- Solo permite el desarrollo de bot para Facebook Messenger, pero mediante un plugin se puede introducir en cualquier página web.
- Carece de aprendizaje automático y el procesamiento de lenguaje natural es muy sencillo por lo que está orientado más a una conversación guiada para conseguir objetivos específicos y no a simular la conversación que se tendría con un humano. Aun así, se puede conseguir un gran nivel de comprensión, pero será una tarea mucho más tediosa.
- No dispone de reconocimiento de voz, solo se puede interactuar con el de forma escrita.
- La integración con otras herramientas se realiza mediante el lenguaje JSON. Esto se usa para acceder a la base de datos, backend o APIs externas para que una aplicación externa a Chatfuel realice una función o consulta y le transfiera el resultado. También se puede conectar con una serie de aplicaciones predeterminadas mediante la aplicación Zappier.
- Análisis y seguimiento de las conversaciones con los usuarios, recopilando datos sobre sus preferencias y segmentándolos en diferentes categorías para poder enviarles información personalizada a cada grupo.

En definitiva, es una herramienta muy sencilla al alcance de todos y que puede cumplir con necesidades como publicidad, enviar mensajes masivos a un determinado grupo de clientes, orientar en compras mediante chat, etc.

En esta plataforma el flujo este guiado mediante una arquitectura de bloques, que pueden tener diversas funciones como enviar un mensaje y proponer opciones para responderlas mediante botones para ir dirigiendo la conversación, realizar acciones como asignar valores a atributos, determinar qué acción hacer según unas condiciones, iniciar chat en vivo, llamar por teléfono, etc. Gracias a esto podemos ir creando secuencias de operaciones para conseguir nuestro cometido e ir llamando a estas funciones según el fin que quiera alcanzar nuestro usuario ya sea comprar ropa o ponerse en contacto con un tele operador.

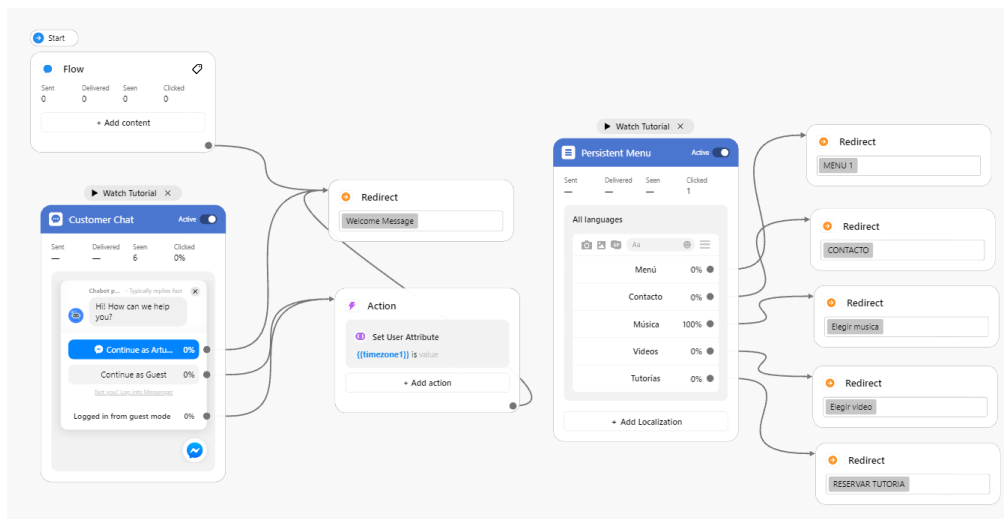


Figura 6: Flujo de conversación en Chatfuel

Aparte de esto se puede ir dotando al bot de inteligencia artificial creando respuestas o llamando a estos bloques si el usuario usa palabras clave para que no sea solo un menú y la conversación parezca más humana. También tiene un mensaje predeterminado para cuando no se reconoce ninguna palabra clave redirigir la conversación.

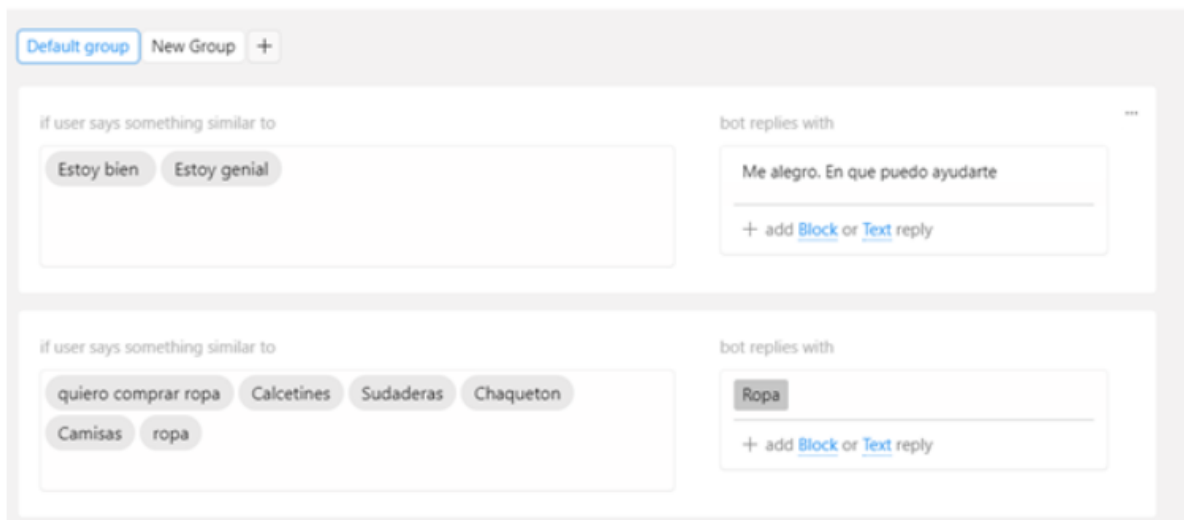


Figura 7: Página de IA del agente virtual de Chatfuel.

## AIML

Es un lenguaje de programación de código libre y usado para crear ALICE, sus siglas significan “Artificial Intelligence Markup Lenguaje”. En él se usan tres conceptos básicos:

- Categorías: Son los temas o unidades básicas de conocimiento. Estas constan de patrones y plantillas
- Patrones: Son palabras clave que el bot identifica los mensajes del usuario para responder con las plantillas

- Plantillas: Son los conjuntos de respuestas asignados para contestar a los mensajes de los usuarios clasificados por patrones.

```

1 <aiml>
2   <category>
3     <pattern>TELL ME A JOKE</pattern>
4     <template>Why did the chicken cross the road?</template>
5   </category>
6   <category>
7     <pattern>* WHY *</pattern>
8     <that>Why did the chicken cross the road?</that>
9     <template>To get to the other side.</template>
10  </category>
11 </aiml>
12

```

*Figura 8: Código escrito en lenguaje AIML*

Aparte de estos elementos básicos hay muchas más funciones típicas del lenguaje de programación como condicionales, bucles, etc. Esto es un lenguaje de programación y no un entorno de trabajo ya que todo el bot se crea a partir de código, cada vez se usa menos por la sencillez que aportan los otros métodos.

### Watson Conversation

Es una plataforma de la empresa IBM que proporciona funciones muy similares a las plataformas de desarrollo anteriores sobre todo a Dialogflow:

- Procesamiento de entrada por voz (Speech to Text) y procesamiento de salida por voz (Text to Speech).
- Procesamiento del lenguaje natural para mantener una conversación estructurada y satisfacer las peticiones de los usuarios.
- Tiene una versión gratuita aparte de la de pago con más funciones.
- Es compatible con muchos canales de comunicación Facebook Messenger, Telegram, Slack..
- Posee machine learning permitiendo que el bot aumente su base de datos constantemente para mejorar la experiencia con el usuario.
- Análisis de las interacciones de los diferentes usuarios con el fin de recopilar datos para mejorar el servicio.

Los conceptos básicos para desarrollar el chatbot se asemejan muchos a los anteriormente vistos:

- Intenciones: Son acciones asignadas a mensajes o peticiones del usuario, es decir a la finalidad que busca el usuario con esos mensajes, como por ejemplo encargar comida a domicilio. A cada intención se añaden frases de ejemplo que el usuario puede usar para que se reconozca.
- Entidades: Son diferentes parámetros que se vinculan a la intención y que aportan información necesaria para satisfacer la necesidad del usuario, como siguiendo el

ejemplo anterior el tipo de comida que quiere encargar, cuando la quiere, la dirección de entrega, etc.

- **Dialogo:** Se crea un flujo de conversación mediante nodos, en los que se define una serie de respuestas para una serie de situaciones como el mensaje de bienvenida. También se puede añadir intenciones a un nodo para así reunir datos (entidades) mediante los slots o conseguir un flujo secuencial añadiendo condicionales para que solo se pueda avanzar a un nodo si primero provenimos de uno anterior.

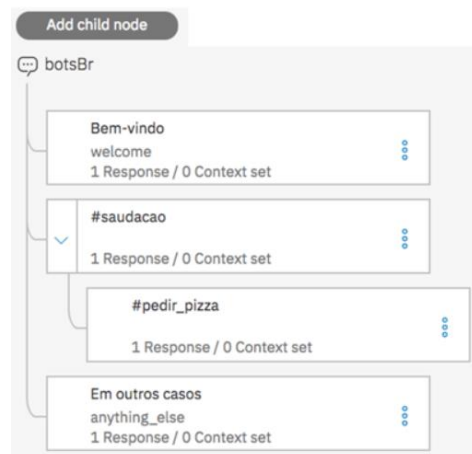


Figura 9: Nodos de dialogo en Watson Conversation de IBM

Hay muchas más plataformas como Manychat, Botisfy, etc. Pero al ser muy similares a las anteriores no se comentan sus características.

	Dialogflow	Chatfuel	Watson Conversation
Ventajas	<ul style="list-style-type: none"> <li>-Dispone de “machine learning” para la mejora continua.</li> <li>-Dispone de mecanismos de “Speech-to-Text” y “Text-to-Speech” para la comunicación por voz.</li> <li>-Es multicanal.</li> <li>-El uso de contextos permite una conversación más fluida.</li> </ul>	<ul style="list-style-type: none"> <li>-La creación de un bot es muy sencilla e intuitiva.</li> <li>-La conexión con APIs externas se puede realizar con unos conocimientos mínimos de JSON.</li> <li>-Dispone de una interfaz gráfica para crear el flujo de la conversación.</li> <li>-Permite hacer seguimiento de los usuarios mediante una herramienta para enviar mensajes a los usuarios que se suscriban.</li> <li>-La versión gratuita incluye todas las funciones.</li> </ul>	<ul style="list-style-type: none"> <li>-Dispone de “machine learning”.</li> <li>-Dispone de mecanismos de “Speech-to-Text” y “Text-to-Speech” para la comunicación por voz, aparte dispone de “Watson Language Translator” capaz de entender 7 idiomas y “Watson Tone Analyzer” para interpretar el tono y sentimiento de la voz.</li> <li>-Es multicanal.</li> <li>-Tiene visión computacional para buscar y clasificar todo el contenido visual.</li> </ul>
Desventajas	<ul style="list-style-type: none"> <li>-Requiere de un mayor conocimiento de programación para conectarlo con APIs externas.</li> <li>-La versión gratuita no incluye todas las funciones.</li> </ul>	<ul style="list-style-type: none"> <li>-No dispone “machine learning” por lo que el entrenamiento puede resultar tedioso.</li> <li>-No dispone de mecanismos de reconocimiento por voz.</li> <li>-Solo dispone de un canal de comunicación que es Facebook Messenger.</li> </ul>	<ul style="list-style-type: none"> <li>-La versión gratuita es muy básica comparada con la de pago.</li> <li>-Se utiliza para desarrollar bot industriales muy complejos.</li> </ul>

Tabla 1: Ventajas y desventajas de las diferentes plataformas para el desarrollo de chatbots.

### 3.PERSPECTIVA Y FUNCIONES DEL PRODUCTO

Actualmente con la pandemia que estamos sufriendo del COVID-19 es recomendable reducir el contacto con otras personas lo máximo posible. Las Universidades debido a su gran concurrencia de estudiantes y profesores han limitado su acceso para minimizar el riesgo de contagio. Por ello los estudiantes tienen difícil acudir presencialmente a resolver dudas sobre la gestión de cualquier asunto relacionado con sus estudios o servicios que ofertan las Universidades. Por eso hay que buscar soluciones para poder dar el mejor servicio posible a los alumnos sin poner en riesgo su salud y ayudar a salir de esta situación lo antes posible.

Para poder solucionar el problema de atender a los alumnos sin que acudan a la Universidad se podría dar atención telefónica o vía email, pero con el gran número de usuarios esto es algo que llevaría mucho tiempo y gran cantidad de recursos humanos ya que la cantidad de alumnos es muy grande. Esto provocaría tiempos de espera demasiado elevados para resolver dudas y provocaría que el servicio no sea muy eficiente.

Pero este problema se podría solucionar en gran medida con el uso de un chatbot para resolver parte de estas situaciones y ayudar al alumno a acceder a la información que necesita. En caso de que la información o trámite que necesite sea muy complejo o demasiado personalizado para su situación se le pondría en contacto con el personal adecuado para ayudarle. Consiguiendo así reducir la carga de trabajo del personal de atención al cliente de las universidades y ofrecer un servicio de calidad.

Aparte de los motivos de la pandemia un chatbot tendría más ventajas como un acceso a la información más sencillo y cercano a los posibles nuevos alumnos de la universidad que tienen una decisión muy importante ante ellos y tienen múltiples caminos que poder elegir. Al implantar un asistente virtual se facilita la elección del camino a seguir y puede marcar la diferencia para que el alumno elija una universidad aumentando así el número de inscripciones de esta. Otra gran ventaja es que el bot puede atender a sus usuarios las 24 horas y a través de redes sociales que casi todos tienen instaladas en sus móviles, consiguiendo estar activo en cualquier lugar y en todo momento.

Se puede actualizar de forma sencilla por el personal docente cada cierto tiempo para incorporar nuevas preguntas frecuentes realizadas por los alumnos o peticiones de estos para ir incorporando nueva información con el paso del tiempo. Gracias al análisis de las conversaciones con los usuarios se pueden obtener datos sobre lo que más se consulta o el perfil de los alumnos que más usan este servicio para cada vez ir implementando nuevas funciones para mejorar el servicio y llegar a más personas.

Por estos motivos el objetivo de este proyecto es la creación de un chatbot para el Área de Atención y Orientación al Estudiante de la Universidad de Sevilla. Muchas universidades ya los están usando y han sido un éxito.

Para crearlo se ha usado como entorno de desarrollo Chatfuel por la facilidad que esta plataforma otorga para crear un chatbot. Frente a otras plataformas una de las ventajas principales es que es gratuita, aunque posee una versión de pago esta ofrece casi las mismas funcionalidades. Su versión gratuita ofrece lo mismo que otras plataformas y su sistema para estructurar la información mediante bloques permite guiar la conversación de una forma muy directa e intuitiva. Esto es muy favorable porque se quiere que el usuario consiga obtener lo que busca en el menor tiempo posible y a parte permite ofrecerle una atención muy personalizada y que pueda cubrir cualquier tipo de necesidad usando muchos bloques y diferentes flujos entre ellos. A parte de esto se puede compartir información con APIs externas mediante el uso de JSON permitiendo gran cantidad de funciones.

Es posible enviar cualquier tipo de información, no solo texto, como imágenes, videos, audios, páginas web, etc. También se pueden crear menús en formas de galerías para que la información sea más atractiva y completa. A continuación, se van a enumerar las diferentes funciones e información que va a tener el chatbot:

- Diferentes menús con las secciones principales en las que el bot puede atender a los usuarios como bibliotecas, deportes, noticias de la universidad, preguntas frecuentes, cursos disponibles...
- Se conecta con APIs para acceder a aplicaciones ya existentes para obtener datos como la fecha actual para determinar el saludo o si está en el horario de atención por alguien del personal. Esto lo hará usando JSON mediante las funciones GET y POST.
- Guarda la información de los usuarios que lo deseen en bases de datos para enviarle noticias nuevas, cursos y resolver dudas.
- Recopila información de la opinión de los usuarios sobre la atención recibida mediante un formulario en una base de datos para ir mejorando los puntos débiles del bot.
- Crear una página web con una interfaz y diferentes roles de usuario para reservar tutorías. Para ello habría que conectar esta página web con la base de datos.
- Pone en contacto al usuario con un trabajador de atención al cliente en caso de que sea necesario.
- Contesta automáticamente a comentarios que se realicen en las publicaciones de la página de Facebook.
- Envía videos y música a los usuarios para que se entretengan mientras son atendidos por alguien del personal.
- Realizara búsquedas en la página de la Universidad de Sevilla para temas que no tenga programados mediante RSS.

### 3.1 Descripción general

El producto final será un chatbot desarrollado con Chatfuel para comunicarse mediante Facebook Messenger solo en forma escrita. Este asistente pone a disposición del usuario contenido de diferentes departamentos de la Universidad de Sevilla.



El bot mediante inteligencia artificial y su arquitectura de bloques conduce la conversación entre sus diferentes opciones para que el usuario obtenga la información deseada. Aparte puede realizar funciones como recopilar datos, enviar contenido multimedia, buscar mediante palabras clave en páginas web externas y enviar nuevas noticias de blogs. Aparte se complementará con una página web para que los estudiantes puedan reservar tutorías y dar su opinión sobre las conversaciones para posibles mejoras.

### 3.2 Requisitos

En este punto analizaremos los requisitos para nuestro chatbot, para poder establecer un orden de prioridades y tener un seguimiento de las funciones resueltas y sin resolver. El análisis de requisitos se define según el Estándar IEEE Std.610[IEE 1990] como “el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos”.

Por lo tanto, un requisito es una característica o función que tiene que realizar el sistema para cumplir con una necesidad del usuario. Entonces para estipularlos se van a plantear diversas situaciones en las que los usuarios necesiten la ayuda de nuestro agente conversacional para hacer una lista de requisitos y organizarlos según su importancia. Con esta lista se puede administrar el tiempo correctamente para dedicarlo solo a estas especificaciones. A continuación, están las tablas con esta clasificación:

ID	Descripción	Prioridad
001	Saludo inicial y presentación del bot	Alta
002	Redirigir la conversación en caso de no disponer de respuesta para un mensaje	Alta
003	Funcionar en múltiples dispositivos (PCs, móviles y tablets)	Alta
004	Poner en contacto con un administrador cuando el bot no sea capaz de solventar el problema	Alta
005	Disponer de un menú para preguntas frecuentes (FAQ)	Alta
006	Preguntar al usuario si las respuestas dadas han servido de ayuda para posteriores mejoras del bot	Alta
007	Entender la misma pregunta realizada de diferentes formas	Alta
008	Guardar registro de conversaciones para su análisis y entrenar al bot para corregir errores	Alta
009	Estar activo en todo momento	Alta
010	Guardar datos de usuario mediante APIs en una base de datos usando la función POST	Alta
011	Acceder a datos de externos o de nuestro backend mediante la función GET	Alta
012	Enviar emails de forma automática a usuarios y administradores	Alta
013	Funcionar en múltiples canales como Facebook Messenger, Instagram, Slack	Media
014	Clasificar a los usuarios según sus respuestas para sus futuras consultas	Media
015	Tener un tiempo de respuesta rápido	Media
016	Enviar imágenes o redireccionar al usuario a ellas mediante la url	Media
017	Mandar notificaciones o publicidad a usuarios determinados cada cierto tiempo	Media
018	Poder acceder a nueva información de blogs o páginas web mediante RSS	Media
019	Buscar información según palabras clave dentro de una página web	Baja

Tabla 2: Requisitos del chatbot del proyecto.

### 3.3 Metodología

En este proyecto se ha seguido el ciclo de Deming de mejora continua, también conocido como ciclo PDCA (Plan-Do-Check-Act). Este sistema se basa en una estrategia de mejora continua de la calidad basada en cuatro fases. La correcta aplicación de estas cuatro fases permite reevaluar continuamente los procesos de un proyecto para así optimizarlo de manera continua. A continuación, se explican cada de las fases del ciclo:

- Plan (Planificar): Se identifica el problema, se establecen los objetivos para solucionarlo y asignan las tareas para lograrlos.
- Do (Hacer): Se realizan las tareas para lograr los objetivos establecidos. Durante esta fase es recomendable realizar un seguimiento del avance del proyecto mediante reuniones periódicas.
- Check (Verificar): Transcurrido un tiempo fijado previamente se analizan los resultados para comprobar si han logrado satisfactoriamente los objetivos.
- Act (Actuar): Según el análisis de la fase anterior se establecen acciones y tareas a realizar para corregir errores e implementar mejoras.

### 3.4 Planificación y costes

Para la gestión del proyecto se ha usado la herramienta Redmine que permite agrupar las tareas en listas según las fases del proyecto, asignarles un tiempo estimado, personal del equipo que las tiene que realizar, hitos a las diferentes fases del proyecto para determinar si se sigue la planificación y crear diagramas Gantt. Esta aplicación web simplifica el seguimiento y la organización del proyecto, sobre todo si se realiza en equipo.

#### 3.4.1 Planificación

Las siguientes listas contienen todas las fases, con las fechas previstas de finalización de cada una de ellas y la persona que las tiene que realizar.

Id	Tipo	Prioridad	Asunto	Asignado a	Descripción
417	Requisitos	Normal	1. Planificación	Arturo Olmedo	En esta fase del proyecto se estudiará la necesidad de implementar un chatbot para el Área de Orientación y Atención al estudiante de la Universidad de Sevilla para mejorar la experiencia del usuario y así mejorar la eficiencia de este departamento de la US. Se determinarán las tareas a realizar y los recursos necesarios para conseguir alcanzar el objetivo del proyecto.
418	Requisitos	Normal	1.1 Recopilar información	Arturo Olmedo	Se recopila toda la información que tiene que proporcionar nuestro asistente y las diferentes funcionalidades que tiene que cubrir.
419	Requisitos	Normal	1.2 Estudio de viabilidad	Arturo Olmedo	Se evalúa la viabilidad del proyecto considerando todos los aspectos legales, económicos y técnicos para considerar si es posible conseguir finalizar el proyecto con los resultados esperados.
420	Requisitos	Normal	1.2.1 Estudio económico	Arturo Olmedo	Se determina el tiempo para realizar el proyecto y sus costes límites para que sea rentable.

421	Requisitos	Normal	1.2.2 Estudio técnico	Arturo Olmedo	Se estudia que se tengan disponibles todos los medios técnicos para realizar el proyecto como pueden ser plataformas de desarrollo, conocimientos, dispositivos, etc..
422	Requisitos	Normal	1.2.3 Estudio social	Arturo Olmedo	Se estudia si la creación del bot tendrá una buena aceptación entre los alumnos de la US.
423	Requisitos	Normal	1.2.4 Análisis de riesgos	Arturo Olmedo	Se determina y clasificaran los posibles riesgos del proyecto según su posibilidad e impacto en el proyecto para poder gestionarlos de forma adecuada.
424	Requisitos	Normal	2. Análisis	Arturo Olmedo	En esta fase del proyecto hay que especificar exactamente que funciones e información tiene que tener el chatbot.
425	Requisitos	Normal	2.1 Objetivos del proyecto	Arturo Olmedo	El asistente virtual tiene que ser capaz de mantener una conversación fluida con el usuario y poner a su disposición la información de la manera más sencilla posible. Para ello dispondrá de menús para las diferentes secciones, usará inteligencia artificial para guiar la conversación, estará conectado a bases de datos para recopilar algunos datos necesarios, se crearán páginas web para poder implementar funcionalidades como reservar tutorías de forma telemática, podrá poner en contacto al usuario con algún miembro del personal en caso de no poder solucionar el problema del usuario y será capaz de enviar videos y audios.
426	Requisitos	Normal	2.2 Determinación de los requerimientos del sistema	Arturo Olmedo	En esta fase se creará una lista con los diferentes requerimientos que deberá cumplir nuestro asistente virtual para asegurar el cumplimiento de estos.
427	Tareas	Normal	3. Diseño	Arturo Olmedo	En esta fase se establecen los detalles de la forma en que nuestro asistente virtual cumplirá los requerimientos que se determinaron en la fase anterior.
428	Tareas	Normal	3.1 Flujo de conversación	Arturo Olmedo	Se diseñará la estructura del flujo de conversación que seguirá el asistente virtual
429	Tareas	Normal	3.2 Bases de datos	Arturo Olmedo	Se establecerán las diferentes bases de datos que se necesitan para nuestro proyecto.
430	Tareas	Normal	3.3 Conexión con aplicaciones externas y páginas web	Arturo Olmedo	Se establecen las aplicaciones externas y páginas web que van a interactuar con nuestro bot.
431	Tareas	Normal	4. Desarrollo del bot	Arturo Olmedo	En esta fase se llevan a cabo todas las tareas planificadas en la fase de diseño para conseguir como resultado final un chatbot funcional.

432	Tareas	Normal	4.1 Creación de bloques de información	Arturo Olmedo	Se construyen los bloques con contienen toda la información y funcionalidades que puede realizar nuestro bot y se establecen las relaciones entre ellos para conseguir el flujo de conversación deseado.
435	Tareas	Normal	4.2 Creación de la inteligencia artificial	Arturo Olmedo	Se crea la inteligencia artificial relacionando frases y palabras clave con respuestas predefinidas o bloques de información para así humanizar al bot y redirigir la conversación.
436	Tareas	Normal	4.3 Creación y conexión con bases de datos	Arturo Olmedo	Se crean las bases de datos con las diferentes tablas que vamos a necesitar y se conecta con el bot para poder guardar y rescatar la información necesaria.
437	Tareas	Normal	4.4 Creación y conexión con páginas web	Arturo Olmedo	Se desarrolla el código de las páginas web y de la conexión de estas con las bases de datos y el bot.
438	Tareas	Normal	4.5 Creación de roles de usuario	Arturo Olmedo	Se crean los roles de usuario del bot y las páginas web para administrarlos en un futuro
413	Tareas	Normal	5. Pruebas	Arturo Olmedo	Se realizan pruebas para comprobar que el bot funciona correctamente, para ello se interacciona con el usando diferentes usuarios y siguiendo diferentes flujos de conversación. También se comprueban las bases de datos y páginas web.
415	Tareas	Normal	6. Implantación	Arturo Olmedo	Una vez que en la fase anterior se comprueba que el bot no tiene errores se pone en funcionamiento. En esta fase se entrega un manual de usuario para que los administradores para facilitar la gestión y uso del bot.
416	Tareas	Normal	7. Mantenimiento y mejora continua	Arturo Olmedo	Con el paso del tiempo la información y funcionalidades que necesitan los usuarios de nuestro bot pueden cambiar por ello hay que llevar un mantenimiento e implementar posibles mejoras en el futuro.

*Tabla 3: Lista de fases y subtareas del proyecto.*

A continuación, se añade un diagrama Gantt en base a estas tareas el que se observan las dependencias entre estas tareas para tener una visión general del proyecto y realizar una correcta gestión del tiempo.

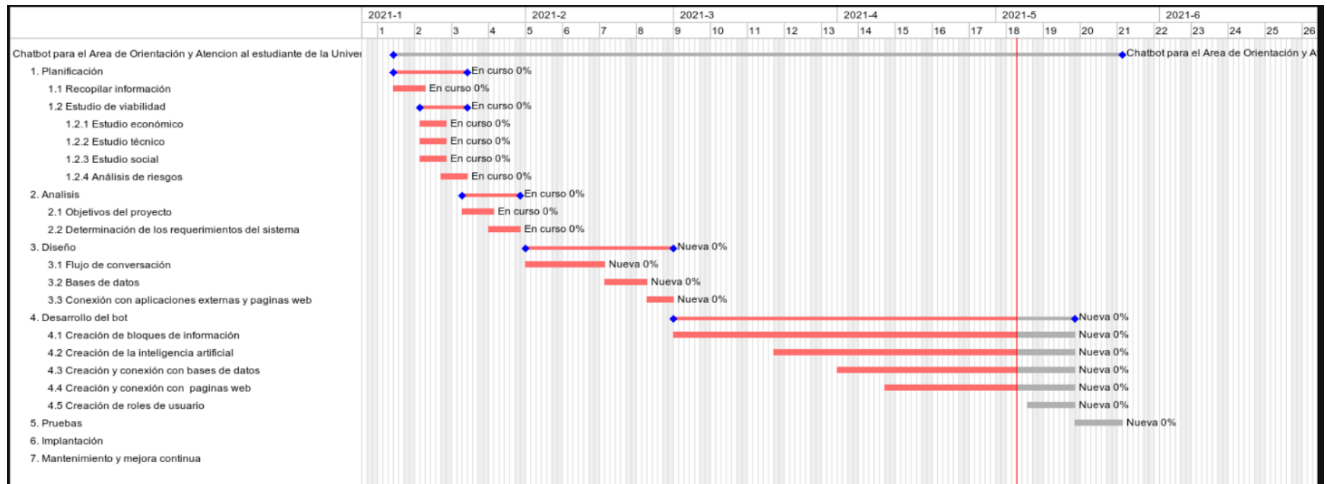


Figura 10: Diagrama Gantt de las fases del proyecto

### 3.4.2 Recursos necesarios

A continuación, se enumeran los recursos necesarios para la correcta realización del proyecto. Estos se clasificarán en diferentes grupos:

#### Hardware

Ordenador con los siguientes requisitos mínimos:

- 2GB de memoria RAM
- 100 GB de memoria física.
- Procesador Petium Dual Core o similar.

#### Software

- Sistema operativo Windows 10.
- MySQL Workbench para la creación de las bases de datos y una base de datos MySQL para guardar toda la información que recoja el chatbot de los usuarios con al menos un 1 GB de capacidad
- Microsoft Visual Studio Code para la creación del código de las páginas web.
- Microsoft Office para la creación de la documentación.
- Host para albergar las páginas web que complementaran el chatbot con su correspondiente dominio para poder llamarlo desde nuestro chatbot.
- Una página de Facebook para incluir nuestro chatbot y añadir novedades e información sobre la Universidad.
- Una cuenta de Dropbox para guardar los archivos multimedia que enviara el bot a los usuarios.
- Una cuenta en algunas de las plataformas disponibles para la creación de chatbots.

#### Personal

- Un programador con conocimientos básicos de HTML, PHP y JavaScript para la creación de estas páginas web y de JSON para que estas se comuniquen con el chatbot.
- Un programador con conocimientos en la creación de chatbots.

## Otros

- Información recabada por la Universidad para introducirla en el chatbot y crear los bloques de contenido.

### 3.4.3 Costes

La estimación del costo-beneficio del proyecto no puede llevarse a cabo de forma exacta, debido al desconocimiento del personal de la Universidad que se encarga de resolver dudas, sus salarios y la cantidad de estudiantes que solicitan ayuda. Pero gracias a los antecedentes de otras empresas y universidades que han implementado chatbots para algunos de sus servicios se estima que reducirá gastos y ofrecerá ventajas como reducir el tiempo de respuesta, aumentar la cantidad de alumnos que pueden ser atendidos, reducir la carga de trabajo del personal al tener que atender solo las consultas muy concretas y un mejor control de todas las consultas de los alumnos. Sin embargo, se puede realizar una estimación de los costos según los recursos necesarios.

Para el desarrollo de este chatbot se han utilizado aplicaciones en sus versiones gratuitas y se va a utilizar el host de la universidad, por lo tanto, el coste del software es nulo.

En cuanto al coste del personal se han estimado que entre todas las fases del proyecto se van a necesitar unas 500 horas. Suponiendo que ingeniero, al ser en prácticas va a cobrar 5 € por hora. Así que el coste del personal es:

Personal	Salario	Horas	Total (€)
Ingeniero de desarrollo	5 € / hora	500	2500

*Tabla 4: Coste del personal.*

El coste del hardware necesario, en este caso un ordenador personal. Suponiendo que el alquiler de un ordenador de gama media cuesta 45 € por mes y se va a utilizar durante cinco meses:

Equipo	Alquiler	Meses	Total (€)
PC	45 € / mes	5	225

*Tabla 5: Coste del equipo.*

Sumando ambos costes se tiene un coste total de 2725 € para el proyecto.

### 3.4.4 Análisis de riesgo

En este apartado se estudia los diferentes riesgos que pueden afectar al proyecto según la probabilidad de que ocurran y el nivel de impacto que tienen sobre el proyecto. También se incluye la gestión de los riesgos con las estrategias de mitigación para en caso de producirse reducir el impacto todo lo posible.

ID	Descripción	Probabilidad	Impacto	Estrategia de mitigación
1	Plataformas para la creación de chatbots desfasadas con el paso del tiempo	Baja	Alto	Tener preparadas alternativas para migrar la información y funciones

				del bot
2	Cambios en el alcance del proyecto	Media	Bajo	Realizar un seguimiento del proyecto
3	Incendio o inundación del servidor que alberga el host de las páginas web	Baja	Alto	Tener copias de seguridad del código de las páginas web y alternativas para contratar otro servidor en el menor tiempo posible
4	Incendio o inundación del servidor que alberga las bases de datos	Baja	Alto	Realizar copias de seguridad periódicamente, pero si el volumen de usuarios es muy grande esto podría suponer mucha cantidad de almacenamiento
5	Cambios en las políticas de privacidad de EU que impida enviar algún contenido de nuestro chatbot como imágenes o videos	Baja	Baja	Buscar otro canal de comunicación para enviarlo o sustituir ese contenido por uno nuevo

*Tabla 6: Análisis de riesgos.*

## 4. DISEÑO DE LA ARQUITECTURA DEL CHATBOT

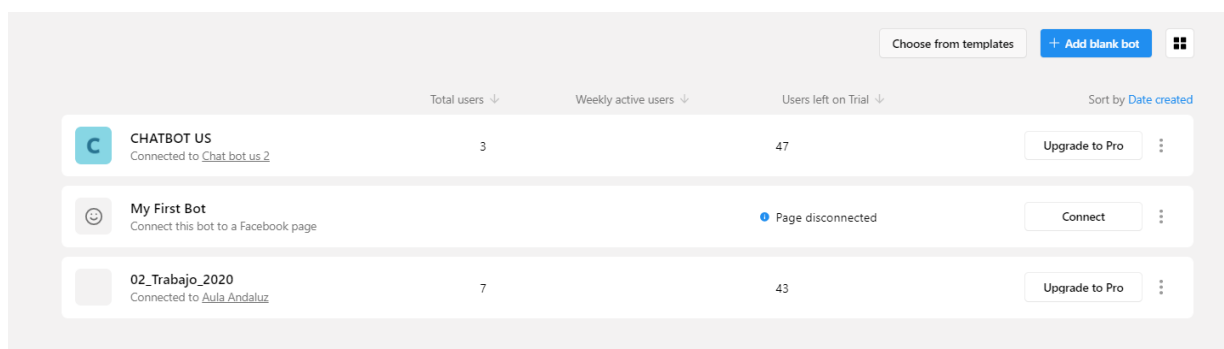
En este apartado se comentan todas tecnologías usadas para crear el chatbot junto con las bases de datos y páginas web.

### 4.1. Entornos de trabajo

#### 4.1.2 Chatfuel

Chatfuel dispone de una gran variedad de funciones para crear un chatbot totalmente operativo de forma intuitiva y sencilla. Aparte de esto, otras de las ventajas por las que se selecciono es porque su versión gratuita tiene las mismas funciones que la de pago siendo ideal para nuestro proyecto. Otro de los motivos es porque dispone de una gran cantidad de contenido docente en internet para usar esta plataforma. A continuación, se van a explicar todos los elementos básicos de Chatfuel.

El único requisito para usar Chatfuel es ser administrador de una página de Facebook, se puede crear de forma gratuita en unos pocos minutos. En la página de inicio de Chatfuel está la opción de crear un bot desde cero o elegir una de las plantillas que ofrece sobre diversos como asistencia para el COVID-19, hoteles, e-commerce, etc. Una vez creado el bot hay que vincularlo con la página de Facebook y se puede empezar a añadir contenido.



*Figura 11: Página de inicio de Chatfuel para crear un chatbot.*

Después se accede al dashboard del bot donde tenemos un menú en el lateral con todas las opciones para administrar y crear el asistente virtual que se van a comentar a continuación.

### Grow

En esta sección se encuentran un resumen de todas las funciones destinadas a aumentar el número de usuarios del bot:

- Crear un link que inicie un chat con el asistente virtual para compartirlo en una página web, algún foro o en cualquier otro medio.
- Dividir a los usuarios del bot en grupos según algunos valores que sean relevantes como los artículos o servicios por los que se interesó cuando interacciono con el bot, para así enviarles publicidad personalizada y captar mejor su atención.
- Crear anuncios en Facebook para vincularlos a un bloque o flujo de Chatfuel para que cuando el usuario visualice un anuncio aparezca la opción de iniciar un chat con el bot



para realizar la compra o proporcionarle ayuda. El inconveniente de esto es que los anuncios de Facebook no son gratuitos.

- Volver a atraer a usuarios inactivos mediante mensajes promocionales como pueden ser ofertas o novedades.
- Responder automáticamente a publicaciones en la página de Facebook en la que esta alojada el bot.

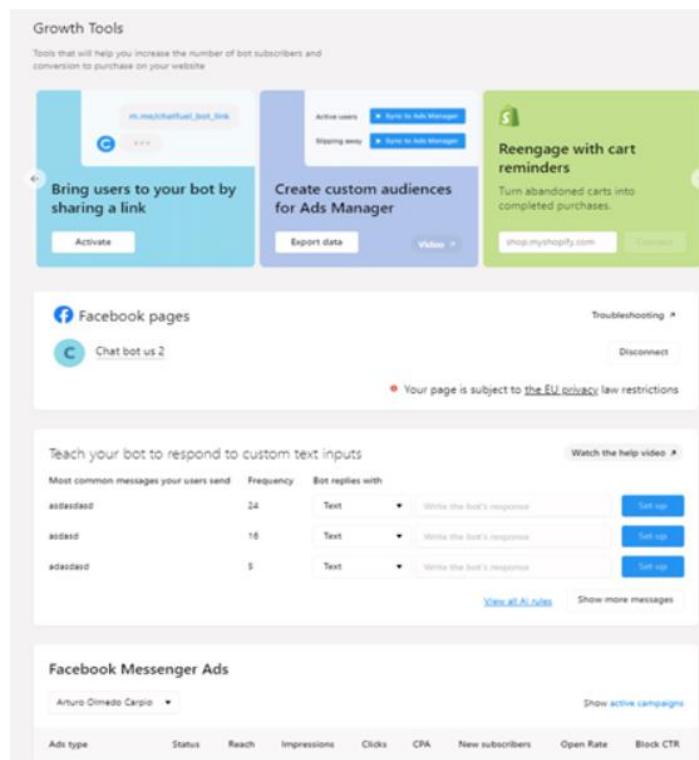


Figura 12: Apartado “Grow” de Chatfuel.

Aparte del resumen de estas funciones la sección Grow nos muestra la página de Facebook a la que está conectada el bot, un resumen de los anuncios de Facebook y los mensajes más comunes que el bot no reconoce, es decir para los que no tiene una respuesta establecida. Esto último es muy útil ya que nos indica el número de veces que se ha recibido ese mensaje y da la opción de añadir una respuesta ya sea con texto o un bloque de información permitiendo eliminar los posibles puntos muertos de la conversación.

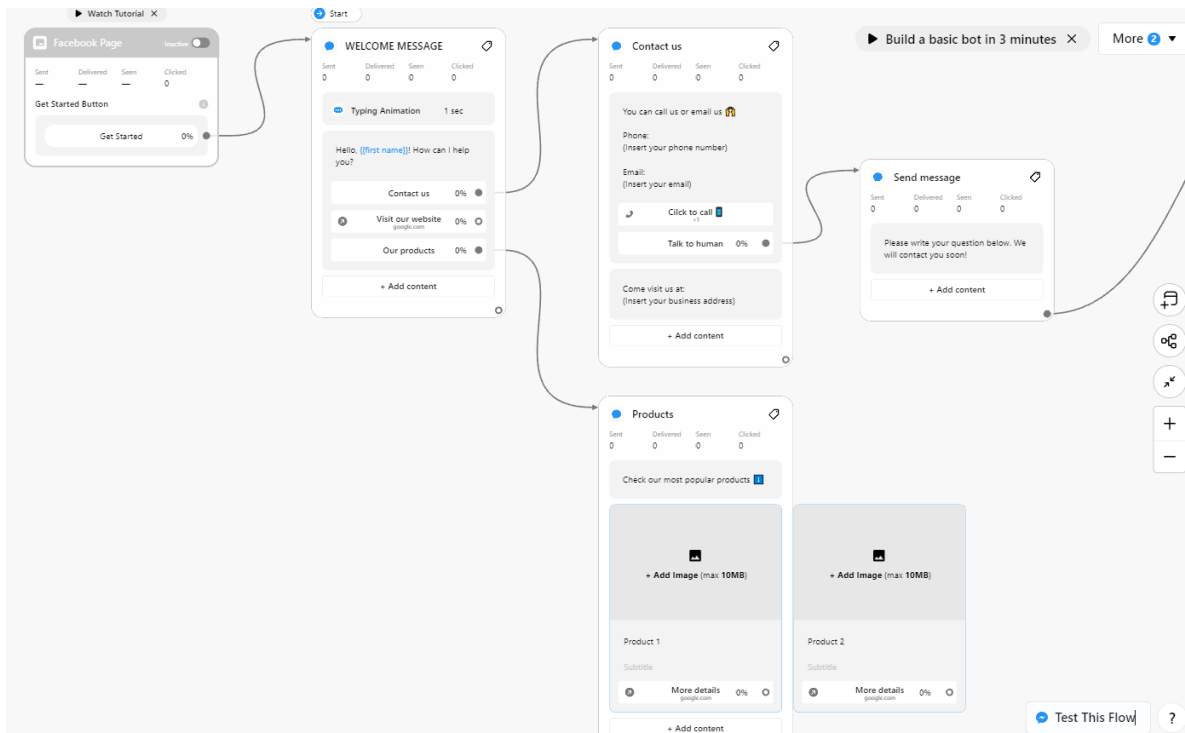
## Automation

Esta es la sección más importante ya que es la que permite dotar de inteligencia y contenido al bot. Esta divide en tres apartados Flows, Blocks y Set Up AI.

### Flows

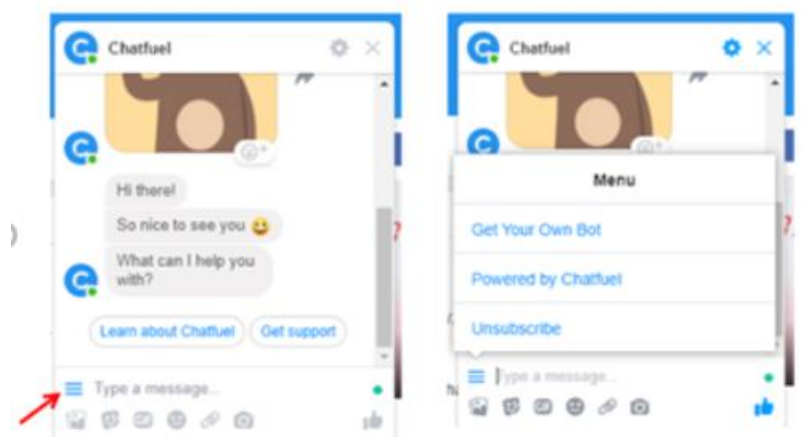
La sección Flows permite añadir todos los bloques de contenido y funciones al bot como dar valor a atributo, añadir condiciones, enviar texto o imágenes y muchas otras funciones más. Flows proporciona casi las mismas funciones que el apartado Blocks, pero acompañadas de una representación gráfica de los bloques que se unen mediante flechas para ver el flujo de la

conversación de manera visual. Durante el desarrollo del bot de este proyecto esta sección estaba en fase beta por lo que la mayoría del bot se ha desarrollado mediante Blocks. Por ello en este apartado solo se explicarán las funciones que solo se puedan implementar en Flows, muchas de ellas mencionadas en el apartado anterior:



*Figura 13: Sección Flows de Chatfuel.*

- Responder automáticamente a publicaciones de Facebook.
- Dirigir mediante anuncios de Facebook hacia un bloque del flujo de la conversación.
- Proporciona un script para añadir en el código de una página web para incluir el chat bot en esta.
- Añadir un menú persistente en el chat con las diferentes opciones que te ofrece el bot, pero esta función no está permitida ahora mismo en Europa por motivos de privacidad.



*Figura 14: Menú persistente de Chatfuel.*

Aparte de esto dispone de varias plantillas con usos típicos como FAQs, atención en cafés, reserva de hoteles, vuelos, etc.

### Blocks

La sección Blocks igual que la anterior se utiliza para crear contenido y dar funciones al bot. Como se ha comentado antes en esta sección se ha creado casi todo el bot del proyecto por lo se explicara sus funciones más en detalle.

El elemento básico que se utiliza para crear contenido en Chatfuel se llama bloque. Los bloques pueden contener información en forma de texto, imágenes, videos o audios y realizar funciones como guardar atributos, conectarse con aplicaciones externas y demás funciones que se explican a continuación. Cada bloque que se crea se relaciona con uno o más bloques para crear el flujo de la conversación, por ejemplo, si dentro de un bloque el bot da la opción al usuario de elegir entre dos opciones cada respuesta conducirá la conversación a un bloque diferente.

Chatfuel tienes dos bloques definidos por defecto que son primordiales para que el bot funcione correctamente, “Welcome Message” y “Default Answer”.

El mensaje de bienvenida (“Welcome Message”) es el mensaje o acción que realiza el bot cuando un usuario envía el primer mensaje. Suele ser un saludo para iniciar la conversación, aunque otro cometido importante es reiniciar los atributos necesarios que guarda el bot en caso de que no fuera la primera interacción. Por ejemplo, si en un chat anterior el bot guarda un atributo para saber si el usuario ha visto las promociones disponibles y en la siguiente hay que cambiar ese valor para poder volver a enseñarle esas promociones.

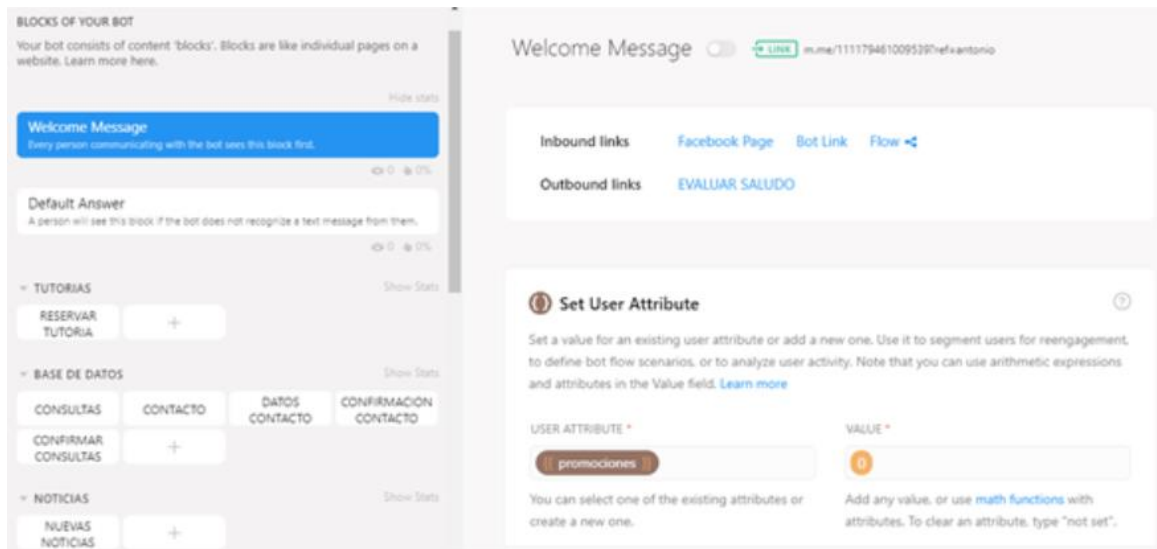


Figura 15: “Welcome Message” de Chatfuel.

La contestación por defecto (“Default Answer”) es el mensaje o acción que el bot usa cuando no reconoce algún mensaje. Chatfuel permite dar inteligencia artificial al bot en la sección “Set Up AI” que se explicara más adelante. Si hay alguna palabra o frase con la que esta inteligencia no encuentre coincidencias el bot responde con el bloque “Default Answer”. Este bloque tiene que intentar reconducir la conversación para que no se quede en un punto muerto ya sea mediante menús, buscadores o iniciando otro chat con algún miembro del personal.

Aparte de los bloques predefinidos se pueden crear todos los bloques que se necesiten para incluir todo el contenido del bot. Estos bloques se clasifican en diferentes grupos según su contenido o cometido para mantener un orden en caso de que haya gran número de ellos. Estas son todas las opciones que hay para bloques:

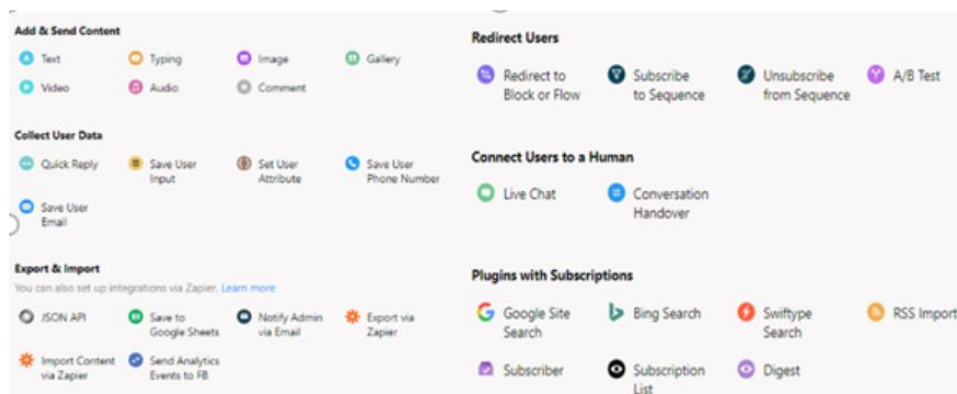


Figura 16: Tipos de contenido y acciones de los bloques de Chatfuel.

- Añadir y enviar contenido: Se puede incluir texto, imágenes, videos, audios, la animación que representa que el bot está escribiendo y galerías. Estas últimas son menús en forma de imágenes y texto que contienen botones que te redireccionan a una url u otro bloque.

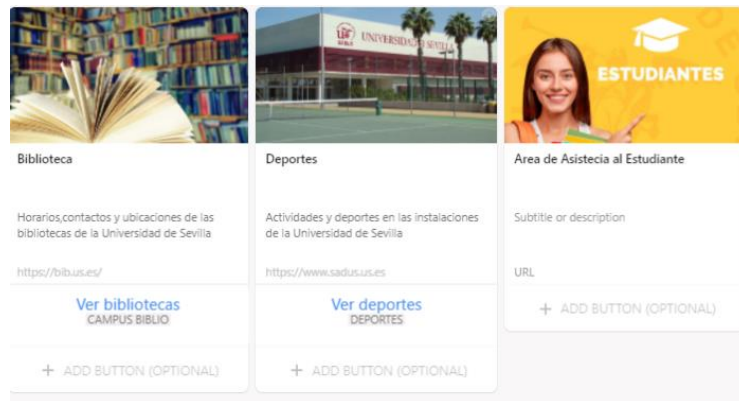


Figura 17: Ejemplo de galería de Chatfuel

Hay varias formas de recopilar datos del usuario:

- Las respuestas rápidas tienen que ir siempre precedidas de un texto, que suele ser una pregunta. Al usuario se le ofrecen varias opciones predefinidas que se guardan en el atributo que se desee. Cada respuesta puede dirigir la conversación a un bloque diferente. Por ejemplo, se puede preguntar por cuál es la facultad del usuario y establecer como respuesta rápida todas las facultades de esa universidad para dirigirlo a un bloque con información como horarios, ubicación, etc.

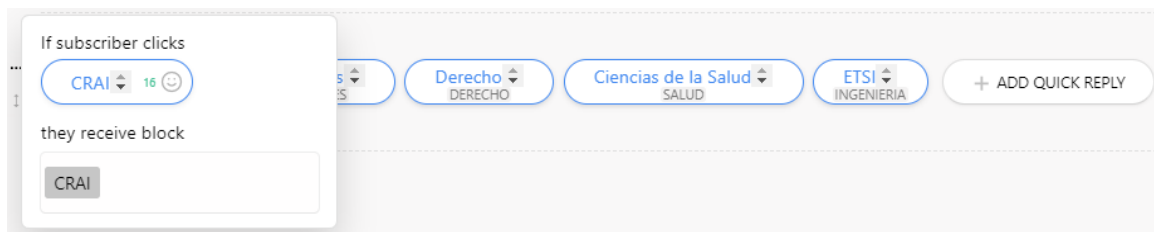


Figura 18: Ejemplo de respuestas rápidas de Chatfuel

- Guardar las entradas del usuario (“Save User Input”) permite hacer diferentes preguntas para guardar la respuesta del usuario en un atributo. Tiene la posibilidad de validar si la respuesta del usuario es un email, teléfono o número, en caso de no coincidir con alguno de estos formatos te permite añadir un texto para indicar que el usuario tiene dar una respuesta valida.

**Save User Input** ?

Ask bot users questions and save their responses to user attributes. You can then utilize attributes in broadcasting user filters, Redirect to Block plugin or export them via Notify Admin via Email or JSON API plugins. [Learn more](#)

MESSAGE TO USER *	VALIDATION	SAVE ANSWER TO ATTRIBUTE *
¿cual es tu correo electronico?	Email	{{email}}
Por favor, dime tu numero de telefono	Phone	{{telefono}}
¿Cual es el motivo de tu consulta?	None	{{consulta}}

+ ADD FIELDS

**Modify Default Plugin Phrases** ▾

Default Phrase	Change to
What you've entered does not look like a phone number. Please try again.	Los datos introducidos no tienen el formato de un número

Figura 19: Ejemplo de guardar datos del usuario en Chatfuel.

- Establecer atributo de usuario (“Set User Attribute”) permite dar un valor al atributo sin que el usuario lo seleccione. Esto es muy útil para crear variables de control para indicar por ejemplo si el usuario ya ha visto un menú.

**Set User Attribute** ?

Set a value for an existing user attribute or add a new one. Use it to segment users for reengagement, to define bot flow scenarios, or to analyze user activity. Note that you can use arithmetic expressions and attributes in the Value field. [Learn more](#)

USER ATTRIBUTE *	VALUE *
{{menu_visto}}	1

You can select one of the existing attributes or create a new one.

Add any value, or use [math functions](#) with attributes. To clear an attribute, type "not set".

Figura 20: Ejemplo de asignar atributos de usuario en Chatfuel

Exportar e importar contenido (“Export and Import”) con aplicaciones externas:

- Guardar los atributos y sus valores en una hoja de Excel de Google.
- Enviar emails de forma automática a algún administrador del bot. Esta opción no permite enviar al usuario un email.
- Exportar e importar información mediante Zappier, que es una aplicación que permite conectar aplicaciones para automatizar procesos como enviar email, subir publicaciones a redes sociales, notificar eventos de un calendario vía SMS, etc.

- Enviar datos mediante JSON API permite intercambiar datos con cualquier API o página web que este programada para usar la funciones POST y GET. Permite enviar o recibir atributos mediante la url. También es posible recibir archivos multimedia como imágenes, videos, PDF, crear botones, etc.
- Enviar eventos de análisis a Facebook (“Send Analytics Events to Facebook”) permite enviar a la herramienta de Facebook notificaciones de cuando algún usuario accede a un bloque, pulsa un botón, etc. Hay 24 eventos predefinidos como suscribirse, pedir un contacto o buscar una localización, también es posible crear tus eventos personalizados. Con esto se consigue un mejor seguimiento de los contenidos a los que acceden los usuarios.

Redirigir usuarios (“Redirect Users”) en este apartado hay varias opciones para conducir la conversación:

- Redirigir a un bloque o flujo (“Redirect to Block or Flow”) como su nombre indica esta opción permite enviar al usuario a otro bloque o flujo. Esta función es imprescindible para que haya varios caminos que el usuario pueda seguir durante la interacción.
- Suscribirse o cancelar una suscripción a una secuencia (“Subscribe to Sequence”) permite crear bloques que se accionan pasado un tiempo después de que ocurra un evento. Por ejemplo, cuando un usuario inicia el chat se puede programar que al cabo de cierto tiempo se inicia un bloque como puede ser preguntar que le ha parecido la ayuda que le ha proporcionado el bot.
- Prueba A/B (“A/B Test”) permite redirigir a un porcentaje de usuarios a diferentes bloques y establecer una meta para analizar cuál de esos bloques es más eficaz para conseguirla. Por ejemplo, se usan dos bloques diferentes para promocionar la venta de algún producto y en ellos se da la opción al usuario a comprarlos, gracias a la prueba de A/B se puede comprobar el número de usuarios y porcentaje de estos que compro el producto en cada bloque para comprobar cual funciona mejor.

Conectar usuarios con un humano (“Conect Users to Human”) permite iniciar un chat con algún administrador del bot. Esta función se suele utilizar para resolver consultas que el bot no puede o en caso de que la conversación alcance un punto muerto.

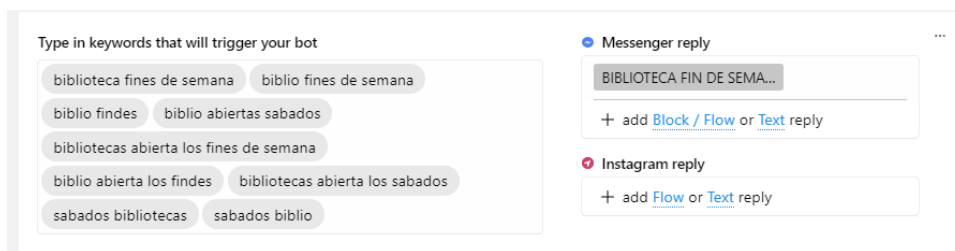
Pluggin con suscripciones (“Pluggin with Subscriptions”) permite utilizar funciones de sitios externos. La más importantes son:

- Los buscadores que permiten buscar información usando palabras clave dentro de sitios web usando motores de búsqueda como Google, Bing o Swiftype.
- Importar contenido mediante RSS (“Really Simple Syndication”) que es un formato XML para compartir contenido en la web. Este formato RSS está diseñado principalmente para web que actualicen su contenido con frecuencia y se pueda compartir esta información de forma sencilla con cualquier página web o aplicación.

Chatfuel permite mostrar las últimas publicaciones en formato RSS de una página en forma de menú en el chat, siendo muy útil para mostrar novedades o noticias.

### Set Up AI

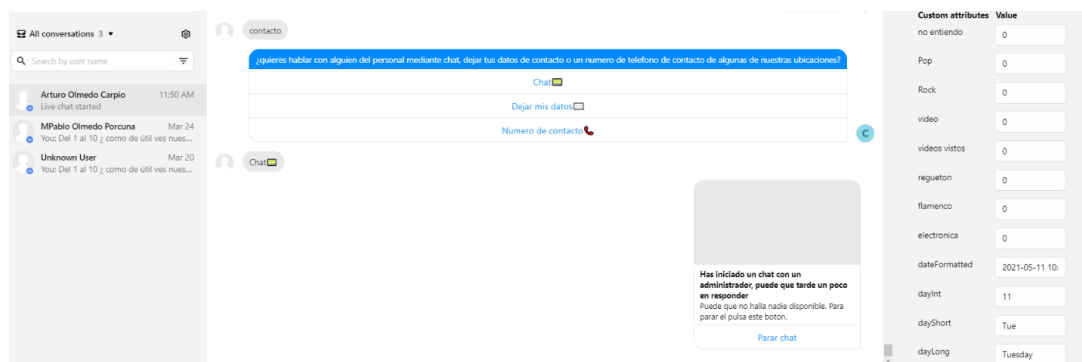
En esta sección se crea la inteligencia artificial del bot mediante palabras o frases de entrenamiento. Se crean grupos formados por varias palabras o frases para que cuando el bot identifique alguna de estas responda con alguna de las respuestas o bloques asociados a ese grupo. Sin esta inteligencia la conversación no puede saltar de un bloque a otro si no están vinculados, obligando a llegar a ciertos bloques para que el usuario pueda obtener el resultado que busca.



*Figura 21: Inteligencia artificial de Chatfuel.*

### Live Chat

Esta sección permite comunicarse mediante un chat con los usuarios que lo han solicitado en algún bloque. Aparte de esto te muestra todas las conversaciones que ha mantenido el bot y todos los atributos actuales de los usuarios.

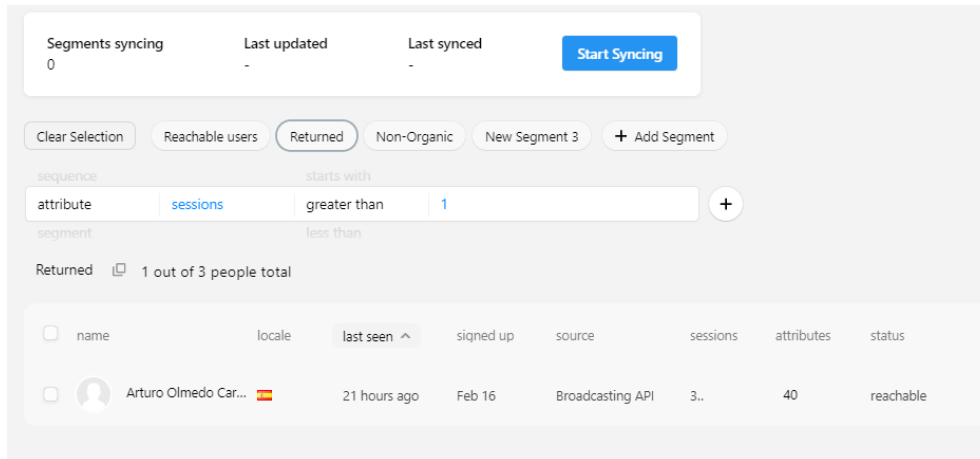


*Figura 22: Chat en vivo de Chatfuel.*

### People

Esta sección permite clasificar a los usuarios del bot en segmentos según condiciones como el valor de atributos, los bloques que han visitado, si hay secuencias activas, etc. Después se pueden usar para realizar acciones específicas para cada segmento como redirigir a diferentes bloques dependiendo del segmento, enviar mensajes personalizados o suscribir a secuencias. También permite sincronizarlo con Facebook Ads para enviar de forma automática publicidad.





*Figura 23: Apartado para agrupar usuarios de Chatfuel.*

### Reengage

Este apartado tiene funciones para captar de nuevo la atención de antiguos usuarios. Dispone de funciones como enviar mensajes difundidos, programar mensajes para determinadas fechas, enviar mensaje cuando se active un disparador y permitir que Zappier envíe mensajes de forma automática cuando por ejemplo hay una nueva publicación en una página de Facebook.

### Configure

Este apartado te permite cambiar opciones como:

- Seleccionar la zona horaria en la que va a operar el bot.
- Administrar el personal del bot añadiendo o eliminando miembros con un rol determinado como administrador, editor u operador de chat en vivo.
- Determinar las urls de las páginas web que se autorizan para utilizar las extensiones de Messenger que permiten varias funciones relacionadas con las webviews.
- Ver la lista de todos los atributos usados en un bot, permitiendo gestionarlos dando valores globales a cada uno de ellos.

### Analyse

Este apartado permite analizar varios datos del bot dando la opción de filtrar por fechas:

- Gráfica con los usuarios totales, nuevos usuarios y usuarios que ha bloqueado al bot.
- Gráfica de actividad de los usuarios.
- Fuentes a través de las cuales los usuarios inician el chat con el bot como pueden ser una página de Facebook, un link o una página web.
- Los elementos más usados en el bot como botones, urls externas o bloques más visitados.

- Las expresiones y la cantidad de veces que la inteligencia artificial del bot no reconoció alguna frase o palabra.
- El porcentaje de usuarios que vuelve a interactuar con el bot pasado el primer día.

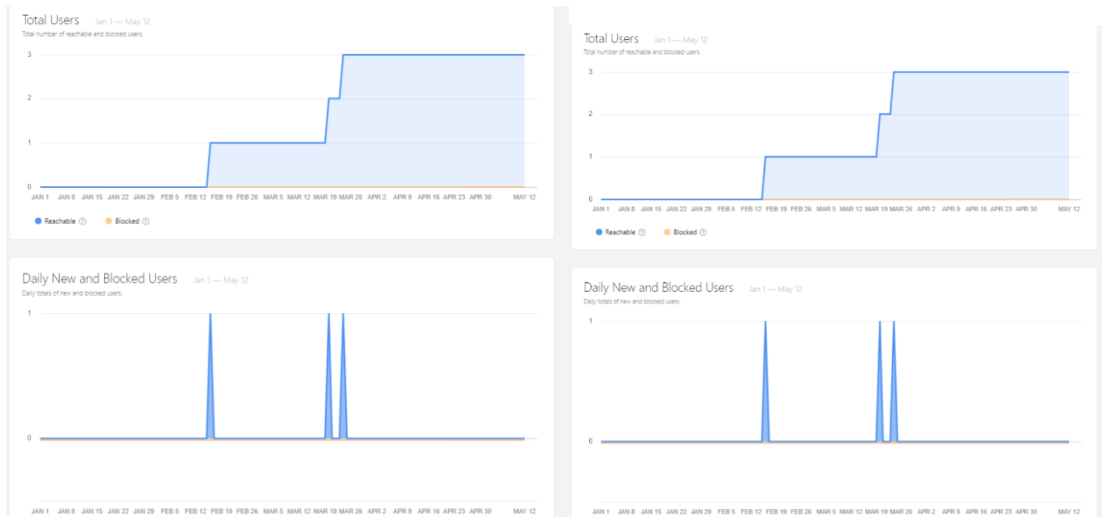


Figura 24: Graficas de análisis de usuarios de Chatfuel

#### 4.1.2 Bases de datos

Para este proyecto se ha creado una base de datos con el fin de almacenar los datos relevantes de los usuarios. Para este cometido se ha elegido MySQL que es “un sistema de gestión de base de datos relacional de código abierto con un modelo cliente-servidor”. Una base de datos es una recopilación de datos estructurados y al ser “relacional” estas se organizan mediante tablas. Estos softwares se conocen por las siglas RDBMS (“Relational Database Management System”) y el más conocido es MySQL. La arquitectura cliente-servidor se basa en una red formada por clientes que solicitan servicios al servidor. El servidor es el proveedor de servicios, que se encarga de enviar la información solicitada de las bases de datos a los clientes siguiendo unas determinadas reglas llamadas protocolo.

Como proveedor del servidor se ha elegido Tonohost por ser gratuito y la capacidad de almacenamiento que ofrece. Para el diseño, gestión y mantenimiento de la base de datos se ha utilizado MySQL Workbench, esta es una herramienta visual que nos permite crear y gestionar tablas de datos sin tener conocimiento de código. Esta aplicación es completamente gratuita y permite realizar de forma sencilla las funciones básicas de establecer la conexión con la base de datos, crear tablas y sus relaciones, insertar datos y hacer consultas.

#### Establecer conexión

En primer lugar, hay conectarse con el host para poder crear una base de datos. Workbench por defecto conecta con el host local, es decir el ordenador está haciendo el papel de servidor virtual. Sin embargo, en este proyecto se ha optado por un proveedor externo que te proporcione la base de datos en un servidor externo para poder acceder a ella de forma remota, debido a que el volumen de los datos no va ser excesivamente grande y se pueden cubrir con una base de datos gratuita. Para realizar la conexión se necesitan los siguientes datos de acceso:

-Nombre de conexión: Se le asigna el nombre deseado a tu conexión para poder organizarlas en caso de ser muchas.

-Método de conexión: Se elije el protocolo para conectarse con la base de datos, el modelo por defecto es el TCP/IP (“Transmission Control Protocol / Internet Protocol”). Hay más opciones como Local Socket para conexiones locales o SSH para intercambiar los datos de forma más segura cifrando la información.

-Nombre del host: Este dato te lo proporciona el proveedor de la base de datos. En caso de realizar una conexión local el host puede variar entre 127.0.0.1 hasta el 127.255.255.255.

-Nombre de usuario: Este dato también te lo proporciona el proveedor, dentro de una misma conexión puede haber varios usuarios con diferentes roles como propietario, editor de tablas, escritor de tablas, etc. Cada nombre de usuario tiene asignado una contraseña.

-Esquema por defecto: Este dato es opcional, solo indica a la base de datos que se quiere acceder. Si se deja en este campo en blanco se puede seleccionar más adelante.

Figura 25: Ejemplo de conexión con base de datos de MySQL Workbench.

### Crear tablas

El siguiente paso es crear una tabla para poder almacenar los datos necesarios. Para ello hay que asignarle un nombre a la tabla y añadirle sus columnas. Cada columna corresponde a un dato y hay que configurar estos atributos:

- Nombre: Es el nombre del dato a almacenar como por ejemplo cliente.
- Tipo de dato y su longitud: Hay varias clases numérica exacta, numérica decimal, fechas, texto y cadenas. Dentro de estas clases hay varios tipos que determinan la longitud y el formato de estos datos. También se puede elegir una longitud personalizada introduciendo el valor de esta entre paréntesis.

Además, a cada columna se le puede asignar unas características:

- “Primary key”: Esta opción indica que el valor de esta columna es el identificador único de cada registro y solo puede haber uno por tabla.

- “Not null”: Indica que este valor no puede ser nulo.
- “Unique index”: Indica que este valor no se puede repetir.
- “Binary”: Indica que este valor tiene que ser binario, es decir con valor “1” o “0”.
- “Unsigned”: Indica que este valor no utiliza signo, es decir no puede ser negativo. Esta opción solo puede utilizarse en datos de tipo numérico.
- “Zero fill”: Indica que todos los espacios sin utilizar se rellenaran con “0” a la izquierda del dato hasta completar la longitud asignada.
- “Autoincremental”: Indica que el valor se aumentara en la cantidad que se indique cada vez que se introduzca un nuevo registro.
- “Generate”: Indica que el valor de esta columna se genera a partir de los valores de otras, como puede ser sumando sus valores o concatenando los textos de varias columnas.

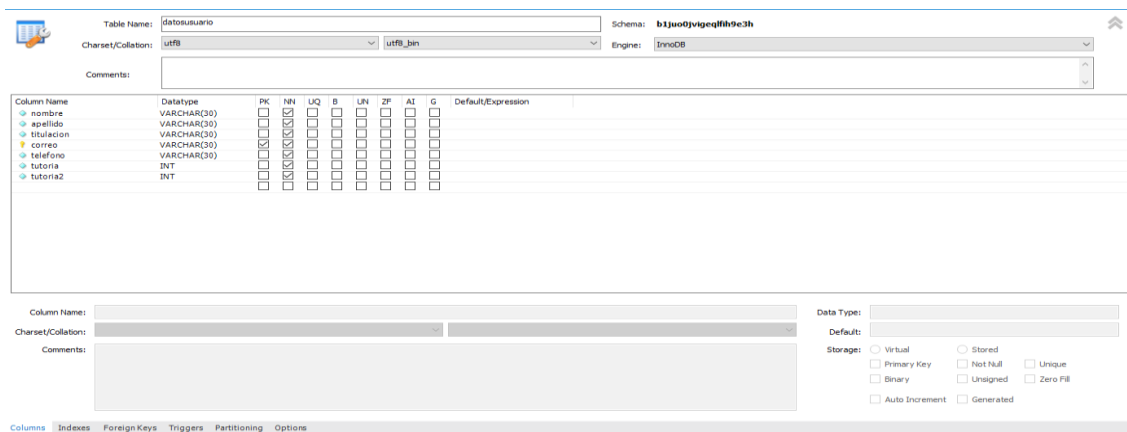


Figura 26: Ejemplo de crear tablas en MySQL Workbench.

Aparte de esto hay funciones más complejas como los índices que sirven para simplificar la búsqueda de datos, las claves foráneas para relacionar los datos de diferentes tablas o los disparadores para realizar funciones de forma automática.

### Insertar y consultar datos

La forma de manejar los datos de la tabla es muy sencilla solo hay que rellenar o eliminar los campos de esta. Además, se pueden usar filtros para buscar el dato deseado de manera más rápida.

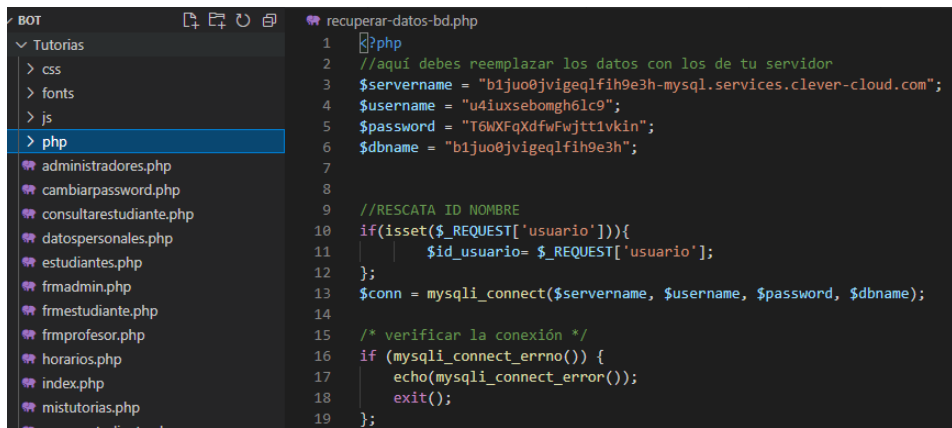
nombre	apellido	titulacion	correo	telefono	tutoria	tutoria2
Arturo	Carpio	Grado en Ingeniería Eléctrica	arturoolm323edo.95@gmail.com	6498131237267	79	0
Manolo	Martin	Grado en Ingeniería Eléctrica	arturoolmeeeedo.95@gmail.com	6495432312323	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 27: Ejemplo de datos de una tabla en MySQL Workbench.

### 4.1.3 Visual Code Studio

Para el desarrollo del código de páginas web en este proyecto se necesita un editor de código. Visual Code Studio es un editor de código fuente desarrollado por Microsoft que admite una gran variedad de lenguajes de programación como C, C++, Python, Java, Ruby, PHP, HTML, CSS y muchos más. Es un programa gratuito que permite funciones como:

- Organizar y guardar proyectos en múltiples lenguajes.
- Buscar y reemplazar expresiones del código.
- Autocompletar el código con funciones y variables.
- Diferenciar las partes del código por colores.



```

1  <?php
2  //aquí debes reemplazar los datos con los de tu servidor
3  $servername = "b1juo0jvigeqlfih9e3h-mysql.services.clever-cloud.com";
4  $username = "u4iuxsebomgh6lc9";
5  $password = "T6wXFqXdfwFwjtt1vkin";
6  $dbname = "b1juo0jvigeqlfih9e3h";
7
8
9  //RESCATA ID NOMBRE
10 if(isset($_REQUEST['usuario'])){
11     $id_usuario= $_REQUEST['usuario'];
12 };
13 $conn = mysqli_connect($servername, $username, $password, $dbname);
14
15 /* verificar la conexión */
16 if (mysqli_connect_errno()) {
17     echo(mysqli_connect_error());
18     exit();
19 };
  
```

Figura 28: Ejemplo de código en Visual Code Studio.

## 5. DETALLES DE LA IMPLMETANCIÓN

### 5.1 Procedimientos asociados

Para lograr implementar correctamente el chatbot hay que tener en cuenta una serie de objetivos y capacidades que hay que cumplir, no solo a nivel de las funciones de bot. También hay que realizar una serie de tareas cuya misión es introducir correctamente el bot en la empresa y en sus clientes. La siguiente imagen muestra la estructura básica de un chatbot orientado a una empresa, mostrando los usuarios del chatbot, los canales que utilizan y un diagrama de flujo con las funciones básicas.

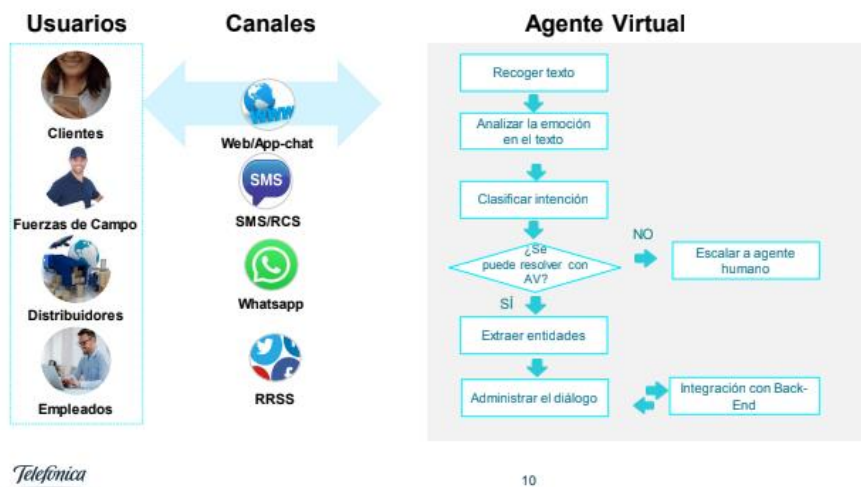


Figura 29: Estructura básica de un chatbot

## 6. DESARROLLO Y DISEÑO

En este apartado se explicarán las funciones explicadas anteriormente, pero aplicadas al proyecto de un chatbot para la Universidad de Sevilla. Para ello se dividen en tres grupos: el chatbot, la base de datos y las páginas web.

### 6.1 Chatbot

Como se ha comentado anteriormente para el desarrollo del agente virtual se ha utilizado la plataforma Chatfuel. Se ha dividido todo el contenido del bot en los siguientes apartados.

#### 6.1.1 Bloques y secuencias

Los bloques y secuencias son los elementos que van a contener toda la información del bot. Se dividen en varios grupos según el tipo de contenido de estos.

##### “Welcome Message”

En este bloque se reinician los atributos necesarios del bot en el momento que el usuario inicia una conversación. Esto sirve para reiniciar variables en el caso de que el usuario cierre el chat y vuelva a interactuar con el bot. La siguiente tabla enumera los nombres de los atributos que se reinician y los atributos predefinidos por Chatfuel y sus funciones:

Atributo	Función
no entiendo	Variable que almacena el número de veces seguidas que el bot recibe un mensaje que no reconoce, para poder dar diferentes respuestas.
video	Variable que almacena el número del último video que visionó el usuario
videos vistos	Variable que almacena el número de videos que ha visualizado el usuario
estilo música	Variable que almacena la siguiente música que el usuario ha seleccionado para escuchar
Rock	Variable de control que almacena si la música Rock ya se ha escuchado
Pop	Variable de control que almacena si la música Pop ya se ha escuchado
Reguetón	Variable de control que almacena si la música Reguetón ya se ha escuchado
Flamenco	Variable de control que almacena si la música Flamenco ya se ha escuchado
Electrónica	Variable de control que almacena si la música Electrónica ya se ha escuchado
menu_visto	Variable de control que almacena si el usuario ya ha visto el menú
búsqueda	Variable que almacena el texto que el usuario quiere buscar en el buscador de Google

Tabla 7: Atributos que se resetean en el “Welcome Message”.

Atributo	Función
<b>messenger user ID</b>	Variable que almacena el id de Facebook Messenger del usuario
<b>first name</b>	Variable que almacena el nombre del usuario
<b>last name</b>	Variable que almacena los apellidos del usuario
<b>gender</b>	Variable que almacena el género del usuario
<b>locale</b>	Variable que almacena el país del usuario
<b>time zone</b>	Variable que almacena la zona horaria del usuario
<b>ref</b>	Variable que almacena la referencia del link mediante el cual se accedió a ese bloque. En caso que se accedería mediante otro método el valor de este atributo estará en blanco
<b>profile picture</b>	Variable que almacena la url de la imagen de perfil de Facebook del usuario
<b>source</b>	Variable que almacena la fuente desde la que se inició el chat con el bot por última vez que puede ser mediante un link, mensaje directo de la página de Facebook o desde una página web
<b>last visited block name</b>	Variable que almacena el nombre del ultimo bloque visitado por el usuario
<b>last visited block id</b>	Variable que almacena el id del ultimo bloque visitado por el usuario
<b>last clicked button name</b>	Variable que almacena el nombre del ultimo botón pulsado por el usuario
<b>sessions</b>	Variable que almacena el número de sesiones que ha realizado el usuario en el chat
<b>signed up</b>	Variable que almacena la fecha que el usuario entro al chat por primera vez
<b>last seen</b>	Variable que almacena la fecha que el usuario entro al chat por última vez
<b>chatfuel user id</b>	Variable que almacena el id del usuario en Chatfuel
<b>original source</b>	Variable que almacena la fuente desde la que se inició el chat con el bot por primera vez que puede ser mediante un link, mensaje directo de la página de Facebook o desde una página web
<b>last user freeform input</b>	Variable que almacena el último mensaje que envió el usuario a través del chat
<b>instagram user id</b>	Variable que almacena el id de la cuenta de Instagram del usuario
<b>instagram name</b>	Variable que almacena el nombre de la cuenta de Instagram del usuario
<b>original ref</b>	Variable que almacena la referencia del link mediante el cual se accedió al chat por primera vez. En caso que se accedería mediante otro método el valor de este atributo estará en blanco

Tabla 8: Atributos predefinidos de Chatfuel.



Una vez hecho esto, el bot se redirige a un bloque llamado “EVALUAR SALUDO” para decidir el saludo inicial según la hora del día que sea. Para ello se hace una llamada a una API externa que envía la fecha y hora actuales en diferentes formatos al bot mediante el método GET que los almacena en los siguientes atributos:

Atributo	Función	Ejemplo
<b>dayInt</b>	Variable que almacena el número del día del mes actual	19
<b>dayShort</b>	Variable que almacena el nombre del día de la semana abreviado	Wed
<b>dayLong</b>	Variable que almacena el nombre del día de la semana	Wednesday
<b>monthInt</b>	Variable que almacena el número del mes	5
<b>monthShort</b>	Variable que almacena el nombre del mes abreviado	May
<b>monthLong</b>	Variable que almacena el nombre del mes	May
<b>year</b>	Variable que almacena el número del año	2021
<b>time12h</b>	Variable que almacena el número de la hora actual en formato 12H.	8
<b>time24h</b>	Variable que almacena el número de la hora actual en formato 24H.	8
<b>timeMin</b>	Variable que almacena el número de minutos de la hora actual	26
<b>timeSec</b>	Variable que almacena el número de segundos de la hora actual	53
<b>timeAmPm</b>	Variable que almacena el periodo del día cuando la hora es en formato 12H	AM
<b>time12hm</b>	Variable que almacena la hora en formato 12 H indicando horas y minutos	8:26 AM
<b>time12hms</b>	Variable que almacena la hora en formato 12 H indicando horas, minutos y segundos	8:26:53 AM
<b>time24hm</b>	Variable que almacena la hora en formato 24 H indicando horas y minutos	8:26
<b>time24hms</b>	Variable que almacena la hora en formato 24 H indicando horas, minutos y segundos	8:26:53
<b>greeting</b>	Variable que almacena el saludo que hay que usar es decir si es por la mañana, tarde o noche	morning
<b>timezoneName</b>	Variable que almacena el nombre de la zona horaria	Europe/London
<b>dateFormatted</b>	Variable que almacena la fecha y la hora actual en formato completo incluyendo la zona horaria	2021-05-19 08:26:53 Europe/London

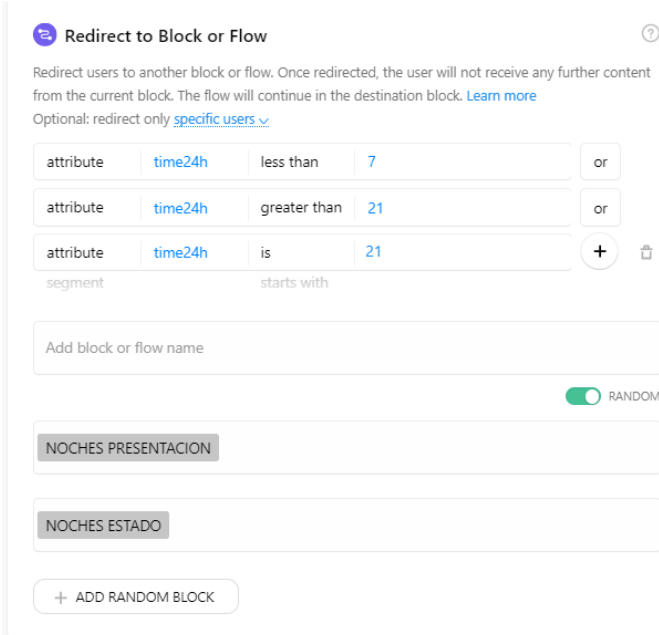
Tabla 9: Atributos de fecha y hora de una API externa.

El atributo que se utiliza para evaluar el saludo es time24h:

-Si este atributo tiene un valor menor que 7 o mayor o igual a 21 se asume que es por la noche y se elige de forma aleatoria entre dos bloques para dar las buenas noches.

-Si este atributo tiene un valor menor que 13 o mayor o igual a 7 se asume que es por la mañana y se elige de forma aleatoria entre dos bloques para dar los buenos días.

-Si este atributo tiene un valor menor que 21 o mayor o igual que 13 se asume que es por la tarde y se elige de forma aleatoria entre dos bloques para dar las buenas tardes.



**Redirect to Block or Flow**

Redirect users to another block or flow. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. [Learn more](#)

Optional: redirect only [specific users](#)

attribute **time24h** less than **7** or

attribute **time24h** greater than **21** or

attribute **time24h** is **21** +

segment starts with

Add block or flow name

RANDOM

NOCHES PRESENTACION

NOCHES ESTADO

+ ADD RANDOM BLOCK

Figura 30: Redireccionando saludo en función de la hora.

Dentro de los bloques de los saludos se utiliza el atributo “first name” para incluir el nombre del usuario en el saludo y emoticonos para humanizar al bot.



Figura 31: Ejemplo de saludo del bot.

Aparte en este bloque se suscribe al usuario a una secuencia para que pasados 15 minutos se le recuerde la opción de ver el menú. En el momento que el usuario acceda al bloque del menú cambia de valor el atributo “menu\_visto” y esta secuencia no realiza ninguna acción.

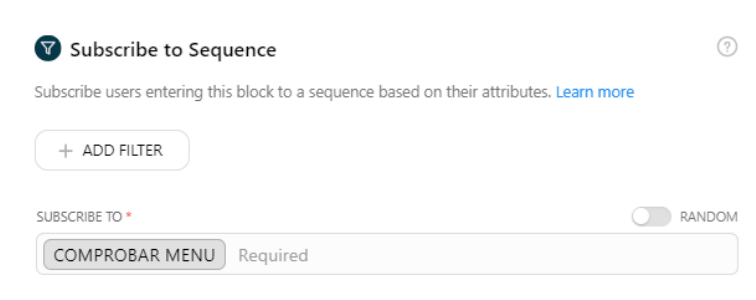


Figura 32.: Suscripción a secuencia de menú.

### “Default Answer”

Este bloque como se comentó anteriormente sirve para responder cuando el bot no reconoce el mensaje del usuario. En nuestro bot se utiliza el parámetro “no entiendo” para elegir entre varios bloques y que la respuesta no sea la misma. Al entrar en este bloque se le suma 1 al valor de este parámetro y posteriormente dependiendo de este valor se redirige el flujo a diferentes bloques:

-Si el valor es 1 se selecciona de manera aleatoria entro dos bloques, uno que simplemente le indica al usuario que no se comprendió el mensaje y otro que aparte de esto le redirige al bloque donde se encuentra el menú.

-Si el valor es 2 se le comunica al usuario que no se ha vuelto a entender el mensaje y se le insta a que utilice otras palabras para solicitar la información que está buscando.

-Si el valor es 3 se le da la opción de usar un buscador que encontrara las coincidencias con su último mensaje (atributo “last user freeform input”) en la página de la US o de contactar con algún miembro del personal. Dentro de la opción de contactar hay varias opciones que se comentaran más adelante. Aparte de esto en este bloque se resetea el atributo “no entiendo” para que la próxima vez que el bot no comprenda algo se repita el mismo proceso.

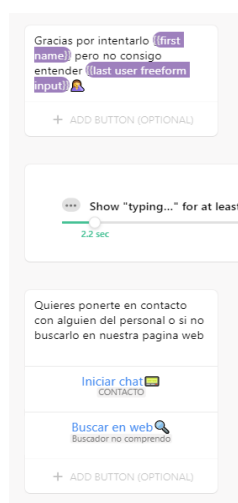


Figura 33: Ultimo bloque de la secuencia “Default Answer”.

## Menús

Dentro del chat el bot ofrece un menú para facilitar la información al usuario de forma más sencilla y visual. El menú está formado por imágenes y botones que te redirigen a otros bloques que pueden ser sub menú o a páginas web mediante la url. En el menú de nuestro bot hay las siguientes secciones:



*Figura 34: Menú principal del chatbot.*

-Bibliotecas: En esta sección se muestran la localización, horarios y teléfono de contacto de todas las bibliotecas de la US. Están agrupadas por campus para que sea más fácil buscarlas.

-Deportes: En esta sección se muestra toda la información y servicios relacionados con deportes que ofrece la universidad. Este departamento llamado SADUS (“Servicios de Actividades Deportivas de la Universidad de Sevilla”) pone a disposición de los alumnos múltiples instalaciones y organiza diversas actividades. El bot redirige al usuario mediante urls a las página web de las instalaciones y como alquilarlas, horarios, tarifas, ofertas, etc.

-Cursos: En esta sección se muestran los cursos del CFP (“Centro de Formación Permanente”), redirige a los usuarios a la página web donde aparecen las ofertas de los cursos, los teléfonos de contacto, localización, horarios e información general del centro. Aparte de esto hay otra sección conectada a la página del CFP mediante RSS que muestra las novedades en el chat en forma de menú.

-Área de Asistencia y Orientación al estudiante: En esta sección a una serie de preguntas y respuestas frecuentes (FAQ) realizadas por los alumnos.

-Tutorías: En esta sección se redirige al usuario a una página web donde puede reservar tutorías. En esta página web hay tres roles, estudiante, profesor y administrador, para poder gestionar las tutorías de los diferentes profesores. Se explicará más en profundidad en el apartado de las páginas web.

-Noticias: En esta sección se muestran las últimas noticias y novedades de US. El bot está conectado a la página de universidad para mostrar las últimas publicaciones de contenido RSS en forma de menú.

## “Contacto”

Como se comentó anteriormente el bot da la opción al usuario de dejar los datos de contacto. Para ello el bot pregunta por el email, teléfono y motivo de la consulta del usuario y guarda las preguntas en atributos para almacenarlo en una base de datos. Para ello se realizan las siguientes acciones:

-Utilizar la función “Save User Input”, pregunta por el e mail, teléfono y motivo de consulta.

**Save User Input**

Ask bot users questions and save their responses to user attributes. You can then utilize attributes in broadcasting user filters, Redirect to Block plugin or export them via Notify Admin via Email or JSON API plugins. [Learn more](#)

MESSAGE TO USER *	VALIDATION	SAVE ANSWER TO ATTRIBUTE *
¿cual es tu correo electronico?	Email	{{email}}
Por favor, dime tu numero de telefono	Phone	{{telefono}}
¿Cual es el motivo de tu consulta?	None	{{consulta}}

Figura 35: Preguntas para guardar los datos de contacto del usuario.

-Enviar un correo electrónico de notificación a los administradores con todos los datos del usuario mediante la función “Notify Admin via Email”.

-Conectar con la API que nos proporciona la fecha explicada anteriormente para poder guardar la hora en la que se realizó la consulta.

-Enviar mediante el método POST los atributos a nuestra página web que será la encargada de guardar toda la información en la base datos. Los atributos que se envían son el nombre, apellidos, email, número de teléfono, motivo de la consulta, genero, fecha, país e id de la cuenta de Facebook Messenger.

**JSON API**

Create integrations with your own server or other 3rd party systems. For example, you can generate dynamic content in Messenger or export data to a CRM. [Learn more](#).

Type: **URL**  
 POST

Headers  
 + Add header

Send  
 Full JSON profile | JSON | **URL encoded**

nombre	first name
apellido	last name
email	email
telefono	telefono
consulta	consulta
genero	gender
fecha	dateFormatted
lugar	locale
usuario	messenger user id

Figura 36: Conexión con página web mediante el método POST.

-Enviar un correo electrónico de notificación al usuario mediante la plataforma Zappier. Zappier envía de forma automática un correo al email que proporcione el usuario con sus datos de contacto y su consulta comunicándole que se resolverá su duda lo antes posible.

Body Type

Plain

If using the HTML option, you must add any and all formatting (paragraphs, lists, etc) directly using HTML. [Learn more.](#)

Body (required)

Hola **1. First Name:** Arturo **1. Last Name:** Olmedo Carpio ,

Tu datos de contacto se han guardado correctamente:

Email: **1. Correo:** arturoolmedo.95@gmail.com  
 Telefono: **1. Telefono:** 675876543  
 Motivo de consulta: **1. Consulta:** nada

En breves se pondran en contacto contigo.

Muchas gracias.  
 Un saludo

Asistente Virtual de la Universidad de Sevilla

Figura 37: Correo de notificación al usuario.

### “Chat en vivo”

En esta sección se inicia un chat entre el usuario y los administradores del bot. En caso de que no haya ningún administrador disponible se le da la opción al usuario de escuchar música o ver videos mientras espera. Estos son los bloques que hay en este grupo:

-Chat en vivo 1: Se consulta la fecha y la hora mediante una API externa para determinar si puede haber algún administrador disponible:

- Primero se comprueba el mes utilizando el atributo “monthLong”, si este es Julio o Agosto se dirige la conversación a otro bloque para comunicar que es periodo de vacaciones.
- Después se comprueba el día utilizando el atributo “dayLong”, si este es sábado o domingo se dirige la conversación a otro bloque para comunicar que los fines de semana no hay administradores disponibles.
- Por último, se comprueba la hora utilizando el atributo “time24h”, si esta fuera del horario de 9:00 a 14:00 y de 16:00 a 20:00 se dirige la conversación a otro bloque para comunicar el horario de atención al cliente.

**Redirect to Block or Flow**

Redirect users to another block or flow. Once redirected, the user will not receive any further content from the current block. The flow will continue in the destination block. [Learn more](#)

Optional: redirect only [specific users](#)

attribute	time24h	less than	9	and
attribute	time24h	greater than	20	and
attribute	time24h	is	14, 15	+
segment		starts with		

FUERA HORARIO

Figura 38: Redirección a otro bloque según atributo “time24h”.

Si no se da ninguno de estos casos se inicia un chat con algún administrador mediante la función “Live Chat” y se le pregunta al usuario si tuvo respuesta en el chat para en caso de no haber nadie del personal disponible redirirlo al bloque “Esperar personal”

-Esperar personal: En este bloque se notifica a los administradores mediante un correo electrónico que hay un usuario esperando para iniciar un chat en vivo y se da la opción a los usuarios a elegir entre ver un video o escuchar música mientras esperan o dejar directamente sus datos de contacto.

-Esperar personal 2: Este bloque es al que se redirige al usuario si no ha visualizado todos los videos. En él se indica que ha transcurrido muy poco tiempo de espera y da la opción de escuchar música, dejar los datos de contacto o iniciar el chat en vivo.

-Chat en vivo video y chat en vivo música: En estos bloques se inicia el chat en vivo cuando el usuario proviene de ver los videos o escuchar la música. Al final se le pregunta al usuario si ha sido atendido, en caso de que no se les redirige a los bloques de condición.

-Condición esperar videos y Condición esperar música: Se redirige a los siguientes bloques si no se han visto todos los videos o escuchado toda la música, esto se consigue usando la condición de si se han visitado los bloques videos vistos o canciones oídas que se explican más adelante. En caso contrario se redirige al bloque Esperar personal 3.

-Esperar personal video y Esperar personal música: En estos bloques se da la opción de escuchar música y ver videos o dejar los datos de contacto.

-Esperar personal 3: En este bloque se finaliza el intento de iniciar el chat en vivo y se redirige al bloque para dejar los datos de contacto del usuario.

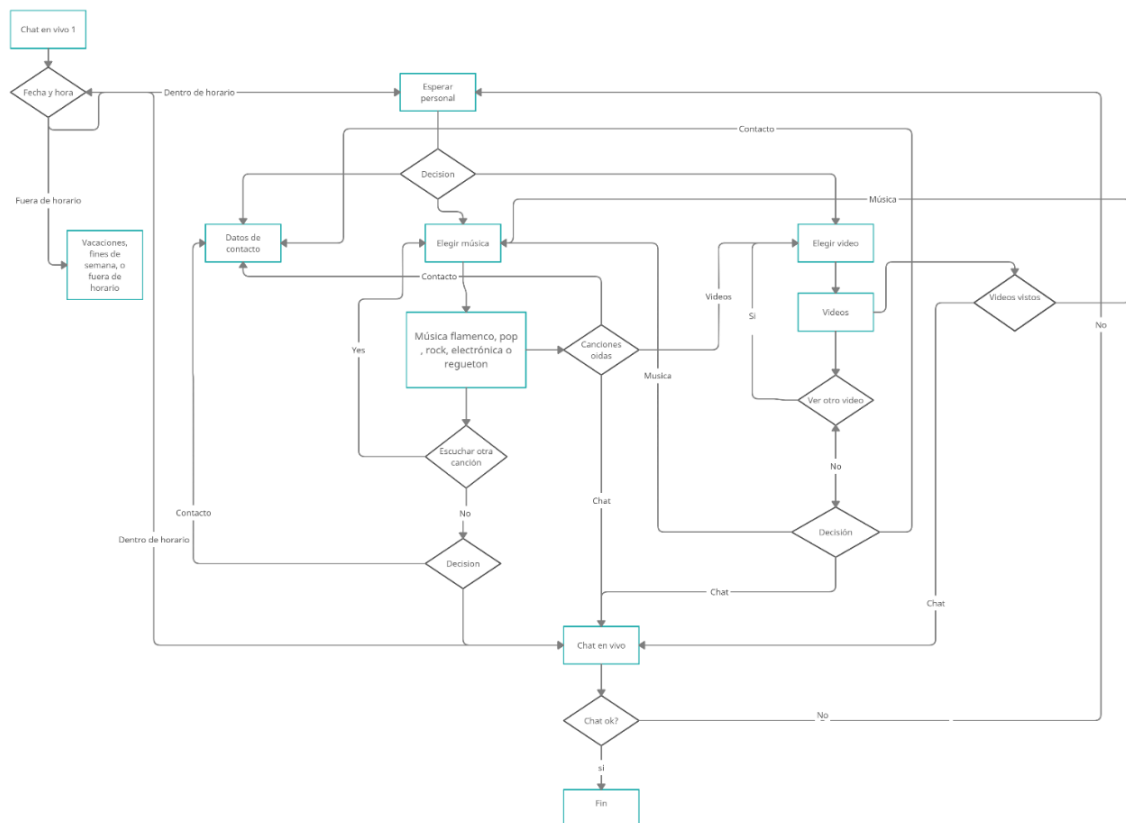


Figura 39: Diagrama de flujo de conversación de bloques de chat en vivo.

## “Videos”

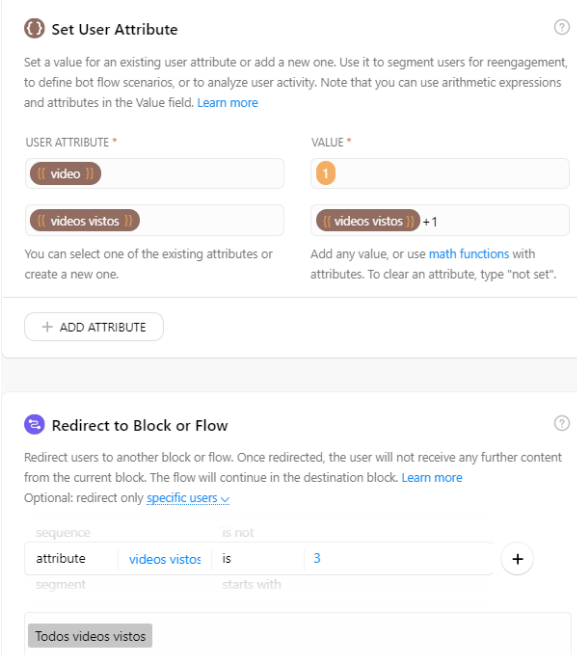
En este grupo se encuentran los bloques necesarios para elegir y visionar los videos para esperar el chat en vivo. Consta de los siguientes bloques:

-Elegir video: En este bloque se redirige la conversación a los bloques donde se encuentran los videos. El primer video se selecciona de forma aleatoria y ya el resto sigue esta secuencia:

- Si se ha visionado el video 3 se redirige al bloque del video 2.
- Si se ha visionado el video 2 se redirige al bloque del video 1.
- Si se ha visionado el video 1 se redirige al bloque del video 3.

Para conseguir esto se utiliza un condicional con el atributo “video” al que se le asigna un valor en los siguientes bloques.

-Video 1, 2 o 3: En estos bloques se encuentran videos de la Universidad de Sevilla almacenados en Dropbox y que se reproducen a través de una webview. Cuando el usuario entra a un bloque se le asigna al atributo “video” el valor 1, 2 o 3 según corresponda y se le suma 1 al atributo “videos vistos”. A continuación, hay una condición para redirigir la conversación a otro bloque si el valor de “videos vistos” es 3.



The screenshot shows two configuration blocks in a chatbot interface:

- Set User Attribute:** This block is used to set a value for an existing user attribute or add a new one. It has two columns: 'USER ATTRIBUTE' and 'VALUE'. Under 'USER ATTRIBUTE', there are two dropdown menus with 'video' and 'videos vistos' selected. Under 'VALUE', there are two input fields with '1' and 'videos vistos + 1' entered. A '+ ADD ATTRIBUTE' button is located below the columns.
- Redirect to Block or Flow:** This block is used to redirect users to another block or flow. It has a condition builder with 'attribute' selected, 'videos vistos' in the middle, and 'is' as the operator, followed by the value '3'. Below the condition, there is a dropdown menu with 'Todos videos vistos' selected.

Figura 40: Bloque de videos.

-Todos videos vistos: Este bloque sirve dar la opción al usuario de dejar los datos de contacto, escuchar música o iniciar el chat en vivo, pero ya no permite volver a los videos.



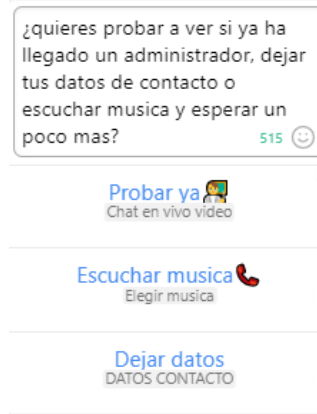


Figura 41: Opciones del bloque “Todos videos vistos”.

## “Música”

El funcionamiento de este grupo de bloques es muy similar al de los videos, pero con la diferencia que la música la elige el usuario no es al azar. Para conseguir esto tenemos dos bloques de elegir música:

-Elegir música 1: Es este bloque se le indica al usuario los tipos de música de los que dispone el bot y se le pregunta cual prefiere mediante la funcionalidad “Save User Input”. Después se le redirige al bloque de música que haya seleccionado.

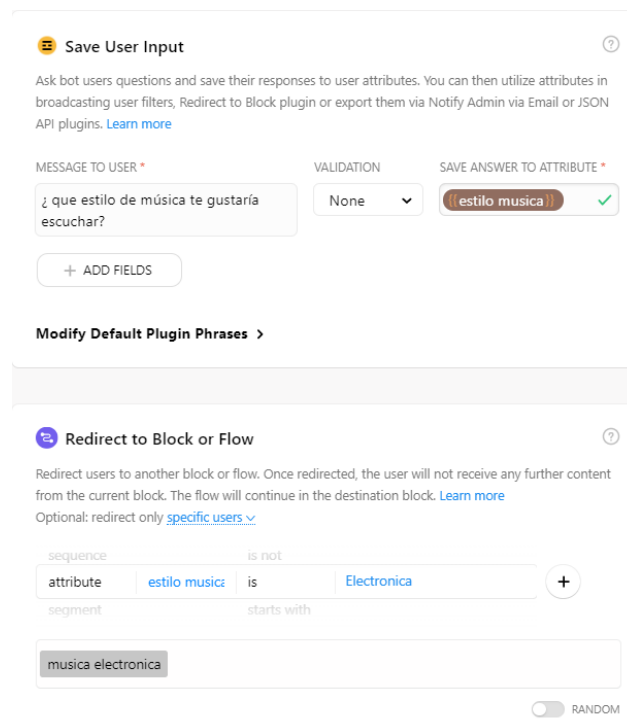
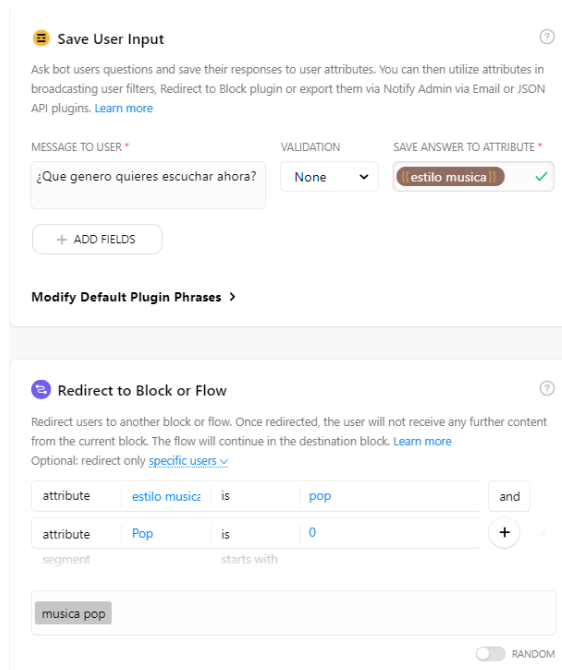


Figura 42: Bloque “Elegir música 1”.

-Elegir música 2: Este bloque es igual que el anterior, pero se comprueba mediante condicionales el atributo “estilo de música” y otro atributo con el nombre del estilo como por ejemplo pop. El primer atributo almacena el ultimo estilo seleccionado por el usuario y el segundo atributo toma el valor 1 o 0 si ya se ha escuchado ese estilo. Si el usuario selecciona un estilo ya escuchado se redirige la conversación “Canciones escuchadas” para que se elija

otra canción. Aparte también se comprueba si se han escuchado todas las canciones para redirigir al usuario al bloque “Todas canciones escuchadas” que funciona igual que el bloque análogo de los videos.



The image shows a configuration interface for a chatbot. It features two main blocks:

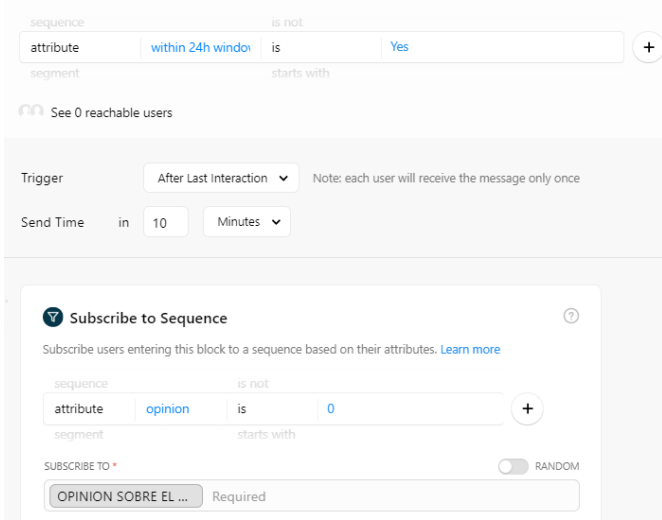
- Save User Input:** This block is used to ask users questions and save their responses as user attributes. It includes a message field with the text "¿Que genero quieres escuchar ahora?", a validation dropdown set to "None", and a "SAVE ANSWER TO ATTRIBUTE" field set to "estilo musica".
- Redirect to Block or Flow:** This block is used to redirect users to another block or flow based on specific conditions. It shows a logic rule: "attribute 'estilo musica' is 'pop' and attribute 'Pop' is '0'". The result of this rule is "musica pop".

Figura 43: Bloque “Elegir música 2”.

Las canciones se envían igual que los videos, mediante enlaces de Dropbox que se abre mediante una webview.

### “Opinión”

Pasado un tiempo el bot pregunta automáticamente al usuario que le ha parecido su experiencia. Para ello se utiliza la función “Reengage” de Chafuel, mediante la cual se añade un disparador para suscribir al usuario a la secuencia “Opinión sobre el bot”. El disparador se activa 10 minutos después de la última interacción si el atributo “opinión” tiene valor 0, es decir si el usuario ha dejado su opinión anteriormente.



sequence is not  
attribute within 24h window is Yes +

segment starts with

See 0 reachable users

Trigger After Last Interaction Note: each user will receive the message only once

Send Time in 10 Minutes

**Subscribe to Sequence** ?

Subscribe users entering this block to a sequence based on their attributes. [Learn more](#)

sequence is not  
attribute opinion is 0 +

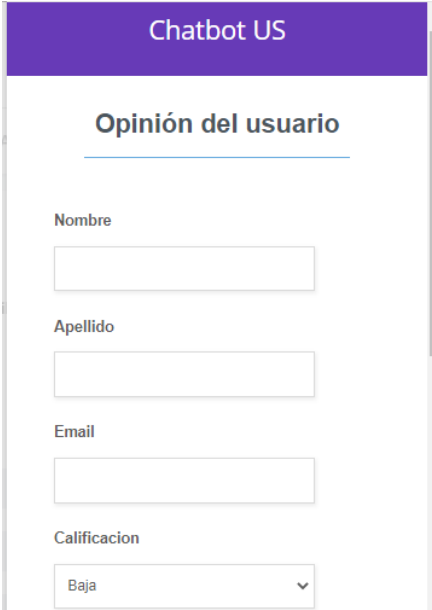
segment starts with

SUBSCRIBE TO \* RANDOM

OPINION SOBRE EL ... Required

Figura 44: Suscripción a la secuencia “opinión”.

Una vez suscrito a la secuencia, esta redirige al usuario al bloque “Opinión usuario” que le da la opción al usuario de rellenar un formulario que se abre en una webview. Los datos del formulario se guardan en la base de datos para poder analizarlos. Esta página web se explicará en detalle más adelante.



Chatbot US

**Opinión del usuario**

Nombre

Apellido

Email

Calificación

Baja

Figura 45: Formulario para obtener la opinión del usuario.

Por último, se redirige al usuario a diferentes bloques según la calificación que le dio al bot, cada bloque tiene una respuesta acorde a la opinión del usuario.

### “Buscador”

Este grupo de bloques permite realizar búsquedas en la página web oficial de la Universidad de Sevilla. Para ello lo primero es crear un motor de búsqueda en Custom Search API de Google con las características que se necesite. En este caso se ha creado un motor que busque resultados en español en la página web de la US.

Idioma 🗕

español ⌵

---

Sitios web en los que


buscar

1 - 1 de 1 < >

<input type="checkbox"/> Sitio	Etiqueta	Disponible en la API Site Restricted JSON <span>🗕</span>
<input type="checkbox"/> <a href="https://www.us.es/">https://www.us.es/</a>		✓

Figura 46: Creación de motor de búsqueda de Google.

Después hay que introducir en Chatfuel el ID del motor de búsqueda y el token de la API. El ID se encuentra en el apartado de configuración del motor de búsqueda y el token en el apartado de credenciales de Custom Search API dentro de Google Cloud Plataform. Una vez configurado esto solo hay que introducir los datos que se desean buscar.

 Google Site Search 🗕

Use this plugin to display search results for a request. You can use a specific request or a user attribute saved using Save User Input, Quick Reply or Set User Attribute plugins. [Learn more](#)

INPUT:

SEARCH REQUEST \*

✓

Use this format: `{{User Attribute}}` to add saved user attributes

SUBSCRIPTION TITLE \*

✓

Use this format: `{{User Attribute}}` to add saved user attributes

GOOGLE API KEY: \*

✓

To generate a token [use this link](#) (Credentials -> Create credentials -> API Key)

SEARCH ENGINE ID (AKA CX OR CSE ID): \*

✓

Generated using [this link](#)

IMAGE TITLE

Use this format: `{{User Attribute}}` to add saved user attributes

**Modify Default Plugin Phrases >**

SETTINGS:

Show no more than

⌵

Let user subscribe to this search result

Figura 47: Buscador de Chatfuel.

Aparte de esto Chatfuel permite cambiar las frases que tiene por defecto para funciones como suscribirse, mostrar más resultados, avanzar a la siguiente página, etc. El resultado se presenta como un menú en el que puedes ir avanzando por los diferentes resultados de la búsqueda.

## “Tutorías”

En este bloque se redirecciona al usuario a una página web donde puede reservar tutorías con los profesores. Dentro de esta página hay varios roles y funciones que se explicaran más adelante.



Figura 48: Inicio de sesión en página web de tutorías.

### 6.1.2 Inteligencia artificial

En el apartado “Set Up AI” se crean frases de entrenamiento para que cuando el bot las reconozca redirija la conversación a bloques de determinados. Estas frases se han agrupado en diferentes grupos según su temática:

-Funciones básicas: Se encuentran las frases relacionadas con los bloques de saludos, despedidas, estados de ánimo, menú, deportes, noticias y chat en vivo.

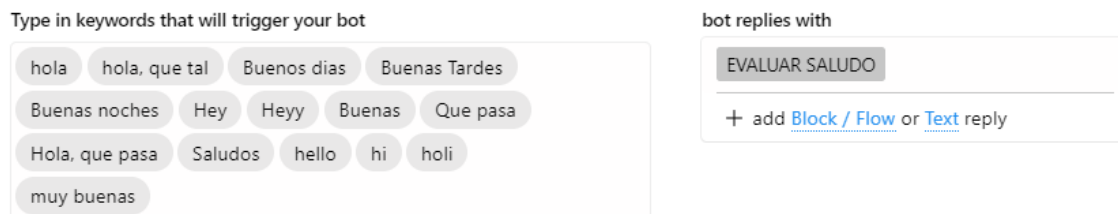


Figura 49: Frases de entrenamiento para los saludos.

-Bibliotecas: Se encuentran las frases relacionadas con los bloques de todos los campus y bibliotecas. El bot reconoce el nombre o el campus de la biblioteca y envía al usuario a ese bloque. También puede indicar que busca una biblioteca abierta los fines de semana y se le redirige a un bloque con las bibliotecas con ese horario.

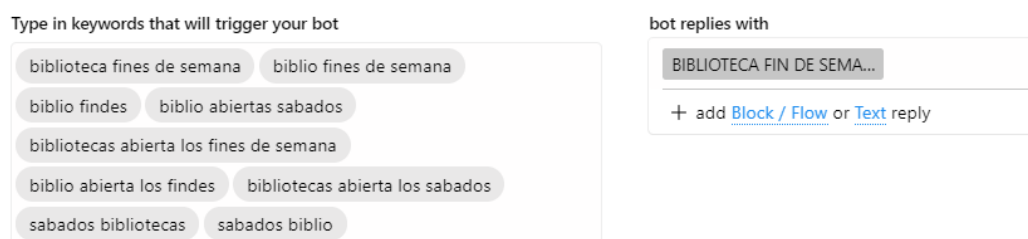
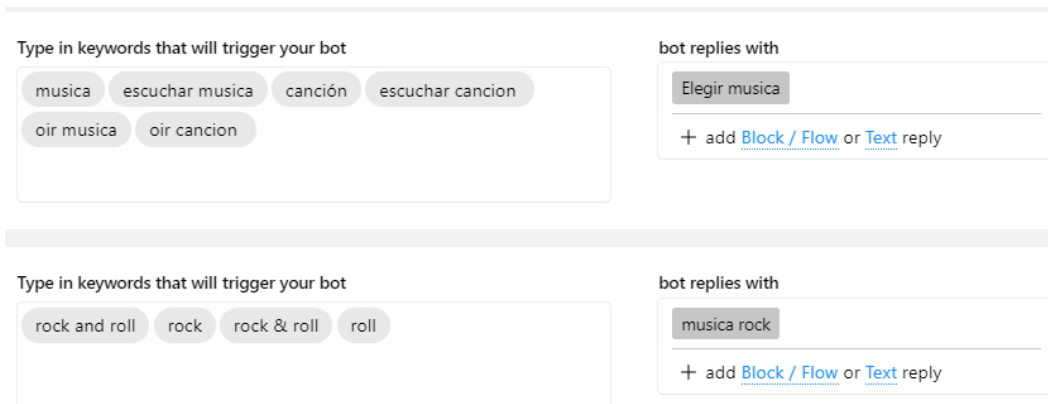


Figura 50: Frases de entrenamiento para bibliotecas abiertas los fines de semana.

-Buscador: Se encuentran las frases relacionadas con el buscador.

-Multimedia: Se encuentran las frases relacionadas con los bloques de videos y música, redirige al usuario si indica un estilo de música como rock o pop.



The image shows two examples of chatbot configuration. Each example consists of a 'Type in keywords that will trigger your bot' section and a 'bot replies with' section.

**Example 1:**

- Keywords:** musica, escuchar musica, canción, escuchar cancion, oir musica, oir cancion
- Reply:** Elegir musica
- Options:** + add Block / Flow or Text reply

**Example 2:**

- Keywords:** rock and roll, rock, rock & roll, roll
- Reply:** musica rock
- Options:** + add Block / Flow or Text reply

*Figura 51: Frases de entrena miento para música.*

-Cursos: Se encuentran las frases de entrenamiento relacionadas con los bloques de los cursos del Centro de Formación Permanente.

-Bases de datos: Se encuentran las frases relacionadas con los bloques que accedan a la base de datos como dejar los datos de contacto, acceder a consultas anteriores o reservar tutorías.

### 6.1.3 Roles de usuario

Para el chatbot de la Universidad son necesarios tres roles de usuario con diferentes permisos:

-Administradores("Admin"): Tienen control total. Son los encargados de crear y modificar el contenido y funciones del bot. También gestionan los miembros del equipo encargado del bot.

-Editores("Editor"): Son los encargados del mantenimiento y posibles mejoras del bot a lo largo del tiempo.

-Operadores de chat en vivo("Live Chat Operator"): Son los encargados de resolver las consultas que el bot no pueda solucionar. No tienen permiso para modificar o crear contenido.

Aparte de estos roles que son los principales para este proyecto existen otros dos:

- Comercializadores("Marketer"): Son los encargados de enviar mensajes difundidos a los usuarios con publicidad o promociones.

-Espectadores ("Viewer"): Solo tienen permiso para ver el contenido del bot.

### 6.1.4 Acceso al chatbot

Para acceder a la ventana del chatbot se pueden utilizar tres métodos diferentes:

- Mediante la página de Facebook con el siguiente enlace → <https://www.facebook.com/profile.php?id=100072321903945> o buscando en Facebook la página con el título: "Chatbot de la Universidad de Sevilla".
- Mediante el siguiente enlace que te redirige al bloque inicial "Welcome Message" → <https://m.me/107080458367032?ref=inicio> .

- A través de la ventana de chat que aparece en la esquina inferior izquierda de la página para reservar tutorías. Una vez abierto el chat tienes que continuar como invitado o iniciar sesión en Facebook. Este es el enlace de la página web : <https://chatbotus.000webhostapp.com/bot/Tutorias/>

## 6.2 Páginas web

En este apartado se explica en detalle el código de las páginas web utilizado en este proyecto. Los lenguajes de programación utilizados son HTML, PHP, JavaScript, CSS, JSON y MySQL. Para ello se dividirán las páginas web dependiendo de su utilidad y se explican sus variables, funciones utilizadas, marcadores y contenido. Las páginas web se han dividido en los siguientes grupos.

### Datos de contacto

Para introducir y rescatar de la base de datos la información que el usuario envía a través del chat se utilizan dos archivos en formato PHP. Uno de ellos se encarga de recibir los datos y almacenarlos y el otro realiza el proceso contrario. A continuación, se explica el código de ambos archivos

#### Recibir y almacenar datos

Este archivo es el encargado de recibir los datos de Chatfuel mediante la función POST, procesarlos e introducirlos en la tabla correspondiente de la base de datos. El primer paso es abrir la etiqueta de código PHP para que el navegador reconozca que lenguaje se está usando(<?php) y crear las variables con los datos de acceso de la base de datos para realizar la conexión más adelante.

```
<?php
//Introduzco datos del servidor de la base de datos
$servername = "b1juo0jvigeqlfih9e3h-mysql.services.clever-cloud.com";
$username = "u4iuxsebomgh6lc9";
$password = "T6WXFqXdfwFwjtt1vkin";
$dbname = "b1juo0jvigeqlfih9e3h";
```

Figura 52: Datos de conexión con la base de datos.

Después se recibe los datos del chat mediante la función POST, para comprobar si la variable está definida se utiliza la función isset(). Al usar la función POST los datos enviados se almacenan en un array asociativo, es decir los datos se agrupan en valores asociados a nombres de variables que se envían a través de Chatfuel.

```
//Obtengo los datos de la función POST enviados por Chatfuel
if(isset($_POST['nombre'])){
    $nombre= $_POST['nombre'];
};

if(isset($_POST['apellido'])){|
    $apellido= $_POST['apellido'];
};
```

Figura 53: Código para recibir datos mediante método POST.

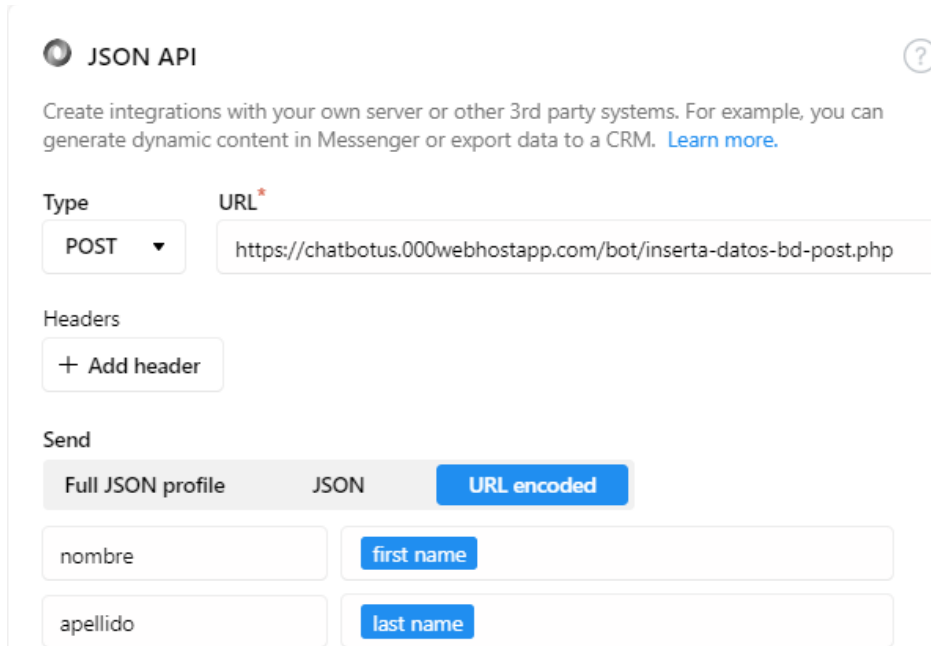


Figura 54: Enviar datos mediante método POST en Chatfuel.

El siguiente paso es procesar algunas variables para conseguir el formato deseado para almacenarlos:

-El idioma del género del usuario enviado por Chatfuel está en inglés para cambiarlo a español se utiliza la función `str_replace($buscar, $reemplazar, $cadena, int $cuenta)` de PHP. Esta función busca dentro de `$cadena` el valor `$buscar` y los sustituye por el valor de `$reemplazar`, las veces que lo haya sustituido quedan guardadas en `$cuenta`. Si el campo `$cuenta` está vacío, ese valor no se almacena en ninguna variable.

```
$genero = str_replace('female', 'femenino', $genero);
```

Figura 55: Cambio de idioma del atributo género.

-El formato del atributo `dateFormatted` es el siguiente “2021-05-19 08:26:53 Europe/London” pero para almacenarlo es conveniente separar la fecha de la zona horaria. Para conseguirlo se utiliza la función `substr($variable, int $comienzo, int $longitud)` para devolver de la cadena `$variable` todos los caracteres a partir del número `$comienzo` con un tamaño máximo `$longitud`. En nuestro caso se guardan los caracteres de la cadena a partir del número 20 que coincide con la zona horaria.

```
//Aislo la zona horaria en una variable propia
$zona_hora = substr($fecha, 20);
```

Figura 56: Eliminar fecha del atributo `dateFormatted`.

Después se realiza la operación contraria, eliminar la zona horaria para almacenar la fecha. Para esto se utiliza la función `preg_replace($buscar, $reemplazar, $cadena, int $limite, int $cuenta)` que funciona igual que `str_replace` pero se pueden buscar expresiones regulares y establecer un límite de sustituciones. En nuestro caso se sustituyen por un espacio todos los caracteres que no sean números, “-” o “:” para eliminar la zona horaria.



```
//Elimino la zona horaria de la fecha
$fecha = preg_replace('/^[^0-9_, :; \-]/', '', $fecha);
```

Figura 57: Eliminar zona horaria del atributo “dateFormatted”.

-El formato del idioma y el país del atributo locale es el siguiente “es\_ES” donde la primera abreviatura es el idioma y la segunda el país. Para separar ambas abreviaturas se utiliza la función strstr(\$cadena, \$busqueda, \$antes) que reconoce la primera coincidencia en la variable \$cadena con la variable \$busqueda y devuelve el resto caracteres a partir de esa coincidencia. Si la variable \$antes tiene el valor “true” los caracteres devueltos son antes de la coincidencia.

```
$pais = strstr($lugar, '_');
$pais = str_replace('_', '', $pais);
$id idioma = strstr($lugar, '_', true);
```

Figura 58: Separar abreviaturas del atributo “locale”.

Por último, hay que convertir las abreviaturas a nombres completos usando la función switch que dependiendo del valor de la abreviatura dará a la variable un valor.

```
switch($pais) {
case "ES":
$pais="España";
break;
case "UK":
$pais="Reino Unido";
break;
case "US":
$pais="Estados Unidos";
```

Figura 59: Asignar nombre completo a la variable país según su abreviatura.

Una vez ajustado el formato de las variables hay que realizar la conexión con la base de datos y guardar los valores de las variables en la tabla adecuada. Para realizar la conexión con el servidor MySQL se utiliza la función mysqli (\$servername, \$username, \$password, \$dbname) con las variables creadas al principio del archivo con los valores del nombre del servidor, usuario, contraseña y base de datos. Si ocurre algún error al conectar se notifica por pantalla.

```
// Conecta a BD
$conn = new mysqli($servername, $username, $password, $dbname);

// Comprobación de errores
if ($conn->connect_error) {
echo'error de conexion\n';
die("Ups problemas al conectar con BD: " . $conn->connect_error);
}
```

Figura 60: Conexión con base de datos.

Después hay que comprobar que las variables no estén vacías e indicar en lenguaje mySQL la tabla y columnas de esta donde introducir los valores elegidos. Para ello se utiliza el comando

“INSERTO nombretabla (nombrecolumna1, nombrecolumna2) VALUES (valor1, valor 2)”.

```
//me aseguro de que existan datos
if($nombre!="" && $apellido!="" && $email!="" && $telefono!="" && $consulta!="" && $genero!="" && $fecha!="" && $zona_hora!="" && $pais!="" && $idioma!="" && $usuario!="" ){
//Creo la query para introducir los datos en la tabla deseada
$sql = "INSERT INTO Usuarios (Nombre, Apellido, Email, telefono, Consulta, Genero, Fecha, Zona_horaria, Pais, Idioma, Id_chat)
VALUES ('$nombre', '$apellido', '$email', '$telefono', '$consulta', '$genero', '$fecha', '$zona_hora', '$pais', '$idioma', '$usuario')";
}
```

Figura 61: Crear comando para introducir datos en tabla de la base de datos MySQL.

Por último, se realiza la conexión con la base de datos y se envía un mensaje al chat usando JSON indicando si la conexión se realizó correctamente o hubo errores.

```
//Guardo los datos
if ($conn->query($sql) === TRUE) {
    //Genero JSON a renderizar con alerta de datos grabados
    $json='{
        "messages": [
            {
                "text": "Tus datos se han guardado correctamente '.$nombre.'"},
            {"text": "Te he enviado un correo notificandolo a tu email"},
            {"text": "Gracias por tu paciencia"}
        ]
    }';
} else {
    //Genero JSON a renderizar con alerta de error
    $json='{
        "messages": [
            {
                "text": "Error al grabar"
            }
        ]
    }';
}
//Cierro conexión
$conn->close();

//Envío el JSON al chat
echo ($json);
?>
```

Figura 62: Realizar conexión y notificar al chat mediante JSON.

### Recuperar datos

En este apartado solo se explican las nuevas funciones utilizadas para recuperar los datos ya que muchas partes del código son iguales que en el apartado anterior.

La conexión con la base de datos se hace de la misma forma lo único que cambia es el comando que se utiliza para realizar la consulta que tiene este formato: “SELECT nombrecolumna1, nombrecolumna2 FROM nombrebasededatos.nombretabla WHERE nombrecolumna3=.\$valor”. Con este comando se rescatan de la base de datos las columnas deseadas (nombrecolumna1 y nombrecolumna2) de una cierta tabla(nombretabla de la base de datos nombrebasededatos) en las que el valor de una columna(nombrecolumna3) coincida con el indicado en la consulta(.\$valor).

```
$consulta = "SELECT Consulta, Fecha FROM b1juo0jvigeqlfih9e3h.Usuarios where Id_Chat=".$id_usuario;
```

Figura 63: Comando para consultar datos de una tabla de la base de datos.

Se realiza la consulta con la función `mysqli_query($conn,$consulta)` y si el resultado no es nulo se utiliza la función `mysqli_fetch_row($resultado)` para hacer un bucle. Esta función almacena los valores una fila de la tabla en un array enumerado, si se repite la función se almacenan los datos de la siguiente fila hasta que estén todas y el valor devuelto es nulo. Con esta función se crea un bucle en el que mientras el valor devuelto no sea nulo se crea un JSON con estos valores. A partir de la segunda fila de resultado hay que ir dividiendo el JSON en cada interacción para que mantenga este formato: `{ "messages": [ { "text": "texto1" }, { "text": "texto2" } ] }`. Para ello se utiliza la función `explode($busqueda, $cadena)` que divide la cadena cuando encuentra una coincidencia con búsqueda y almacena las diferentes partes resultantes en un array numerado. Gracias a esta función conseguimos que el principio del código JSON sea siempre el mismo y se mantenga el formato.

```
if ($result = mysqli_query($conn, $consulta)) {
    $i=0;
    $a=1;
    while ($row = mysqli_fetch_row($result)) {
        //Almaceno resultado de consultas en un array numerado
        $consultas_previas[$i] = $row[0];
        $fecha[$i]= $row[1];
        if($i==0){
            $json='{
                "messages": [
                    {"text": "Consulta '.$a.': ' . $consultas_previas[$i]. ' / Fecha: ' . $fecha[$i]. ' "
                }
            ]';
        }
        // Segunda parte de JSON en caso de mas de una consulta
        else {
            //Divido JSON
            $explode = explode("[", $json );
            $json2 = '[{"text": "Consulta '.$a.': ' . $consultas_previas[$i]. ' / Fecha: ' . $fecha[$i]. ' "
            ],';
            // Creo el JSON con mas de una consulta para mantener el formato adecuado
            $json = $explode[0].$json2.$explode[1];
        }
        //Se le suma 1 al valor de estas variables para almacenar los datos de la consulta en otra posición del array
        $i= $i + 1;
        $a= $a + 1;
    }
}
```

Figura 64: Crear variable para enviar las consultas mediante el chat.

Por último, se vacía la variable de los resultados, se genera un JSON con un mensaje en caso de no tener ninguna coincidencia en la tabla al realizar la consulta, se cierra la conexión y se envía el JSON al chat.

```
/// Liberar la serie de resultados
mysqli_free_result($result);
};
// Mensaje en caso de haber coincidencias en la tabla
if (empty($consultas_previas)) {
    $json = '{
        "messages": [
            {"text": "No tienes consultas en nuestra base de datos"}
        ]
    }';
};
// Cerrar la conexión
mysqli_close($conn);
echo($json);
?>
```

Figura 65: Crear mensaje en caso de no haber consultas en la base de datos.

## Formulario de opinión

Para crear y procesar el formulario de opinión se han utilizado dos archivos PHP. El primero de ellos es el formulario en sí y combina código en PHP, HTML y JavaScript con llamadas a archivos externos en formato CSS para el estilo de la página. El segundo archivo procesa y guarda la información del formulario, este solo usa código en lenguaje PHP y es muy similar a los vistos anteriormente.

### Formulario

El lenguaje HTML utiliza etiquetas para definir las características del contenido del código, cada etiqueta define unas propiedades diferentes a cada elemento.



Figura 66: Ejemplo de etiqueta en HTML.

Aparte a cada elemento se le puede añadir información que no sea visible en el resultado final de la página web mediante los atributos.



Figura 67: Ejemplo de atributo en HTML

Las siguientes tablas explican las principales etiquetas utilizadas en HTML:

Preámbulo y elemento raíz	
Etiqueta	Descripción
!DOCTYPE	Etiqueta para definir el tipo de documento
html	Etiqueta que engloba todo el documento
<!-- ... -->	Etiqueta para incluir un comentario que solo es visible en el código fuente

Tabla 10: Etiquetas de preámbulo y raíz de HTML

Metadatos	
Etiqueta	Descripción
head	Etiqueta para crear el encabezado del documento
title	Etiqueta para crear el título del documento que aparece en la pestaña del navegador
link	Etiqueta para crear enlaces a otros archivos como puede ser los archivos de estilo CSS
meta	Etiqueta para crear metadatos sobre el documento
style	Etiquetas para incluir hoja de estilos en el documento

Tabla 11: Etiquetas de metadatos de HTML.

Secciones	
Etiqueta	Descripción
<b>body</b>	Etiqueta para crear el cuerpo principal del documento.
<b>article</b>	Etiqueta para crear un artículo, es decir para crear documentos que aparezcan en la página, pero se puedan ver de forma independiente.
<b>section</b>	Etiquetas para crear secciones, es decir clasificar en apartados el contenido principal.
<b>nav</b>	Etiquetas para agrupar enlaces a otras páginas web o partes de la misma.
<b>aside</b>	Etiqueta para crear contenido secundario que aparece en la lateral.
<b>h1 a h6</b>	Etiquetas para crear encabezados (títulos) según su importancia.
<b>header</b>	Etiquetas para crear el contenido inicial de la página u otro elemento de la página.
<b>footer</b>	Etiquetas para crear el contenido general de la página u otro elemento que suele ir al final (pie).
<b>address</b>	Etiqueta para incluir información sobre el autor.

*Tabla 12: Etiquetas de secciones de HTML.*

Bloques de contenido	
Etiqueta	Descripción
<b>p</b>	Etiqueta para crear un párrafo
<b>hr</b>	Etiqueta para crear un separador para indicar cambio de temática. Los navegadores la representan como una línea horizontal.
<b>div</b>	Etiqueta para agrupar elementos en diferentes divisiones. Suelen utilizarse para dar estilos diferentes a cada división.
<b>main</b>	Etiqueta para crear el contenido principal de la página web.
<b>figure</b>	Etiqueta para incluir una imagen o representación.
<b>figcaption</b>	Etiqueta para crear el pie de una imagen

*Tabla 13: Etiquetas de bloques de contenido de HTML.*

Texto (en línea)	
Etiqueta	Descripción
<b>br</b>	Etiqueta para introducir un salto de línea
<b>wbr</b>	posible salto de línea
<b>a</b>	Etiqueta para crear enlaces a otras páginas web, suele utilizar el atributo href que contiene la url de destino
<b>span</b>	Etiqueta para crear texto genérico, suele ir acompañado del atributo class para especificar la función de la etiqueta.

<b>strong</b>	Etiqueta para dar importancia a un texto
<b>em</b>	Etiqueta para dar énfasis a un texto
<b>sub</b>	Etiqueta para crear un subíndice
<b>sup</b>	Etiqueta para crear un super índice
<b>abbr</b>	Etiqueta para crear una abreviatura
<b>dfn</b>	Etiqueta para crear una definición
<b>mark</b>	Etiqueta para resaltar texto
<b>small</b>	Etiqueta para crear un comentario
<b>time</b>	Etiqueta para introducir en su atributo date-time la información de la fecha y la hora, pero en formato apto para las maquinas

*Tabla 14: Etiquetas de texto de HTML.*

Tablas	
Etiqueta	Descripción
<b>table</b>	Etiqueta para crear tablas
<b>caption</b>	Etiqueta para crear leyendas explicando el contenido de las tablas
<b>tbody</b>	Etiqueta para crear el cuerpo de la tabla, es decir el contenido de las filas de la tabla que no sean ni cabecera ni pie
<b>thead</b>	Etiqueta para agrupar las celdas de cabecera de las tablas, es decir el título de las columnas
<b>tfoot</b>	Etiqueta para crear el pie de la tabla, similar a la cabecera, pero en la parte de abajo de la tabla
<b>tr</b>	Etiqueta para definir cada fila de la tabla
<b>td</b>	Etiqueta para definir cada celda de una fila
<b>th</b>	Etiqueta para crear celdas de cabecera, los navegadores las remarcan en negrita

*Tabla 15: Etiquetas de tablas de HTML.*

Formularios	
Etiqueta	Descripción
<b>form</b>	Etiqueta para crear un formulario que permite al usuario introducir y enviar datos al servidor. Los atributos más importantes son action que indica el archivo que se va a encargar de procesar los datos y method que indica el método para enviar esos datos (POST o GET).
<b>label</b>	Etiquetas para asociar un texto a un elemento de control como puede ser un botón, cuadro de texto, etc.

<b>input</b>	Etiquetas para crear un elemento de control como puede ser un botón, caja de texto, etc. El atributo más importante es type, según el valor de type se puede crear lo diferentes elementos de control(submit, reset, button, text, password, number, search, tel, url, email, date, time, color,etc).
<b>button</b>	Etiquetas para crea botones que aparte de texto pueden llevar otro tipo de contenido como imágenes
<b>select</b>	Etiquetas para crear menús con diferentes opciones para seleccionar
<b>option</b>	Etiquetas para crear las diferentes opciones de un menú
<b>textarea</b>	Etiquetas para crear un área donde introducir texto.
<b>progress</b>	barra de progreso
<b>meter</b>	indicador
<b>fieldset</b>	Etiquetas para agrupar grupos de controles, los navegadores muestran una caja alrededor.
<b>legend</b>	Etiquetas para añadir una leyenda a los grupos de controles

*Tabla 16: Etiquetas de formularios de HTML.*

A continuación, se va explicar la parte principales del código del formulario. La primera parte está escrita en lenguaje PHP y es la encargada de recibir y guardar los datos que se envían a través de Chatfuel. Por otro lado, las siguientes cabeceras indican al navegador web que no hay que guardar el cache en absoluto.

```
header('Cache-Control: no-cache, no-store, must-revalidate');
header('Pragma: no-cache');
header('Expires: 0');
```

*Figura 67: Configuración del cache en HTML*

Para recibir los datos del chat se utiliza la misma estructura que en los archivos explicados anteriormente con la diferencia que se utiliza la función `$_REQUEST['nombreatributo']` que sirve para recibir datos con el método POST y GET.

```
if(isset($_REQUEST['bot_id'])){
    $bot_id= $_REQUEST['bot_id'];
};
```

*Figura 68: Recibir datos por el método POST*

El resto del código está escrito en HTML y corresponde al formulario en sí. La primera parte es la cabecera que consta de varias partes:

- Metadatos: En estos se especifican los caracteres usados, la compatibilidad con diferentes versiones de navegadores y la visualización en pagina responsivas para que la página web se vea correctamente independientemente del tamaño del dispositivo.

```
<head>
<!-- Metadatos -->
  <!-- Caracteres utilizados -->
  <meta charset="utf-8">
  <!-- Compatibilidad con navegadores -->
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!-- Visualización en diferentes dispositivos -->
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

Figura 68: Metadatos de visualización de la página web.

-Título: Es el título que aparece en la pestaña del navegador.

```
<!-- Título de la pestaña -->
<title>Cuestionario opinión</title>
```

Figura 69: Título de la pestaña del navegador.

-Links a archivos CSS: Enlace con los archivos de estilo en formato CSS para dar diferentes presentaciones a las partes de la página web.

```
<!-- Link a archivos CSS para modales -->
<link rel="stylesheet" href="https://chatbotus.000webhostapp.com/bot/demo.css">
<link rel="stylesheet" href="https://chatbotus.000webhostapp.com/bot/form-basic.css">
```

Figura 70: Utilizar estilos mediante archivos CSS.

-Extensiones SDK de Messenger: Es una parte de código en JavaScript proporcionada por Facebook para activar funciones predefinidas en la aplicación de Messenger de Facebook.

```
<!-- Messenger extensions SDK -->
<script type="text/javascript">
window.name = "messenger_ref";

(function(d, s, id){
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) {return;}
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/es_ES/messenger.Extensions.js";
  fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'Messenger');
window.extAsyncInit = function() {
  // the Messenger Extensions JS SDK is done loading
};
</script>
```

Figura 71: Activar extensiones SDK de Facebook Messenger.

El resto del código es el contenido del formulario:

-Se crea el formulario indicando su nombre, estilo del archivo CSS que se va utilizar mediante el atributo class, el método y el archivo al que se envían los datos.

```
<form name="form" class="form-basic" method="post" action="procesa-form-webview.php">
```

Figura 72: Atributos básicos del formulario.



-Se crean los diferentes apartados de la interfaz para que el usuario introduzca los datos. Hay cuadros para introducir texto o emails, menús desplegables con varias opciones y recuadros para seleccionar una de las opciones.

```
<div class="form-row">
  <label>
    <span>Apellido</span>
    <input type="text" name="apellido">
  </label>
</div>

<div class="form-row">
  <label>
    <span>Email</span>
    <input type="email" name="email">
  </label>
</div>
```

Figura 73: Cuadros de texto del formulario.

```
<div class="form-row">
  <label>
    <span>Calificacion</span>
    <select name="calificacion">
      <option>Baja</option>
      <option>Normal</option>
      <option>Buena</option>
      <option>Muy buena</option>
    </select>
  </label>
</div>
```

Figura 74: Menú desplegable del formulario.

```

<div class="form-row">
  <label>
    <span>Suscribirse para noticias a través del chat</span>
    <input type="checkbox" name="noticias" checked>
  </label>
</div>

<div class="form-row">
  <label><span>Etapas de estudios</span></label>
  <div class="form-radio-buttons">

    <div>
      <label>
        <input type="radio" name="estudios" value="Cursando estudios en la US">
        <span>Cursando estudios en la US</span>
      </label>
    </div>

    <div>
      <label>
        <input type="radio" name="estudios" value="Estudios ya cursados en la US">
        <span>Estudios ya cursados en la US</span>
      </label>
    </div>

    <div>
      <label>
        <input type="radio" name="estudios" value="Pensando en cursar estudios en la US">
        <span>Pensando en cursar estudios en la US</span>
      </label>
    </div>
  </div>
</div>

```

Figura 75: "Checkbox" del formulario.

-Por último, se envían a través del formulario los datos recibidos del chat y el bloque de Chatfuel al que se redirige al usuario, estos datos se envían al otro archivo PHP de forma oculta para que el usuario no pueda verlos ni modificarlos. Aparte se añade el botón para enviar el formulario y activar la función JavaScript de las extensiones SDK de Messenger para cerrar el chat al pulsar el botón.

```

</div>

<div class="form-row">
  <input name="user_id" type="hidden" value="<?=$user_id?">
  <input name="lugar" type="hidden" value="<?=$lugar?">
  <input name="genero" type="hidden" value="<?=$genero?">
  <input name="fecha" type="hidden" value="<?=$fecha?">
  <input name="bot_id" type="hidden" value="<?=$bot_id?">
  <input name="token" type="hidden" value="<?=$token?">
  <input name="block_name" type="hidden" value="respuesta-webview">
  <button type="submit" onClick="MessengerExtensions.requestCloseBrowser();">Envia</button>
</div>

```

Figura 76: Enviar datos ocultos mediante el formulario.

### Procesar formulario

Este archivo PHP es el encargado de procesar y guardar los datos del formulario del usuario. Al ser muy similar a archivos anteriores solo se explican las partes diferentes del código. El archivo empieza recibiendo los datos del formulario por el método POST y guardando en un array llamado data para enviarlos mediante la url posteriormente.

```

if(isset($_POST['block_name'])){
    $block_name= $_POST['block_name'];
};

if(isset($_POST['nombre'])){
    $nombre= $_POST['nombre'];
    $data["nombre_webview"] = $nombre;
}

```

Figura 77: Recibir datos del formulario mediante el método POST.

Se ha creado la función `notify_chatfuel` que es la encargada de generar la url que va enviar el valor de todas las variables a Chatfuel:

- Primero crea la url con los atributos por defecto que envía Chatfuel sin que el usuario tenga que rellenarlos.

```
function notify_chatfuel($user_id, $bot_id, $token, $block_id, $block_name, $data=array()) {
    //echo("bot id:". $bot_id);

    if ($block_name!=""){
        $brcast_url = "https://api.chatfuel.com/bots/$bot_id/users/$user_id/send?chatfuel_token=$token&chatfuel_block_name=$block_name";
    }
}
```

Figura 78: Crear primera parte del url para enviar datos a Chatfuel.

-Segundo se añaden las variables del formulario que se almacenaron en el array `data`, para ello se utiliza el condicional “for” hasta que se introduzcan todos los valores del array que se averigua usando la función “count”. Para que los valores de las variables se envíen correctamente hay que sustituir los espacios por “%20” y utilizar este formato cada vez que se añade una variable a la url: `variable1=valor1&variable2=valor2`. Para averiguar el nombre de cada valor del array se utiliza la función “key” que devuelve el nombre de la variable a la que apunta el puntero interno del array, para avanzar el puntero a la siguiente posición se utiliza la función “next”.

```
//Condicional para introducir todas las variables del array en la url
for ($i = 0; $i < count($data); $i++) {
    // Nombre de la variable en el array
    $key=key($data);
    //Valor de la variable del array
    $val=$data[$key];
    //Sustituye espacios
    $val = str_replace(' ', '%20', $val);
    //Si el valor de val no es nulo añadir a la url
    if ($val <> ' ') {
        $brcast_url = $brcast_url."&".$key."=".$val;
        echo $key ." = ". $val ." <br> ";
    }
    //Avanza puntero del array
    next($data);
}
```

Figura 79: Crear segunda parte del url para enviar datos a Chatfuel.

-Por último, se utilizarán los comandos de Cliente url(curl) para verificar conexiones y transferir datos mediante url. Estos son los comandos utilizados y sus funciones:

- `curl_init($url)`: Inicia una nueva sesión y devuelve el manipulador curl para utilizar con el resto de funciones.
- `curl_setopt ( $curl , $option , $value )`: Configura los parámetros de la conexión url. El valor `$curl` es el devuelto por la función `curl_init`, los valores de `$option` dependen del parámetro a configurar y `$value` es el valor que se le da a ese parámetro. En este caso se ha configurado los parámetros de `CUSTOMREQUEST` para seleccionar el método para enviar los datos, `TIMEOUT` para indicar el tiempo máximo para realizar las funciones curl y `CONNECTTIMEOUT` para indicar el tiempo máximo de espera hasta realizar la conexión.
- `curl_exec($curl)`: Estable la conexión con la url.
- `curl_close($curl)`: Cierra la conexión con la url.

```

//Funciones curl
//Iniciar sesión
$ch = curl_init($brcast_url);
//Metodo de envio de datos
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
//Tiempo máximo de funciones curl
curl_setopt($ch, CURLOPT_TIMEOUT, 5);
//Tiempo máximo de espera para conexión
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 5);

//Realizar conexión
$result = curl_exec($ch);
//Cerrar conexión
curl_close($ch);
    
```

Figura 80: Comandos “curl” para configurar y realizar la conexión con la página web.

El final del código de este archivo guarda las variables en la base de datos, este proceso es igual que en los archivos anteriores por lo que no se explicara para este archivo.

#### Archivos CSS de estilo del formulario

Estos archivos son los encargados de crear las diferentes presentaciones y estilos de las partes de la página web. En el archivo HTML del formulario se les asignaba a las divisiones diferentes nombres al atributo “class”. Estos mismos nombres aparecen en el archivo CSS para dotar de diferentes características a los elementos de la página web. Las siguientes tablas explican las principales propiedades que se utilizan en los archivos CSS:

Tipo de letra (fuente)	
Propiedad	Descripción
font-family	Propiedad que permite elegir el estilo de las letras, es decir la fuente.
font-size	Propiedad que permite elegir el tamaño de las letras.
font-weight	Propiedad que permite elegir el grosor de las letras.
font-style	Propiedad que permite elegir la inclinación de las letras, pueden ser itálica, oblicua o normal.
font-variant-caps	Propiedad que permite mostrar las letras en diferentes formas de versalitas, es decir varía el tamaño y la forma de las letras según sea mayúsculas o minúsculas
font-kerning	Propiedad que permite aplicar kerning a las letras, es decir elegir la distancia entre dos letras contiguas aprovechando la forma de estas.

Tabla 17: Propiedades de los caracteres de la página web en lenguaje CSS.

Texto	
Propiedad	Descripción
<b>color</b>	Propiedad que permite elegir el color del texto.
<b>direction</b>	Propiedad que permite elegir la dirección del texto.
<b>letter-spacing</b>	Propiedad que permite elegir el espacio entre caracteres.
<b>line-height</b>	Propiedad que permite elegir el espacio entre líneas.
<b>text-align</b>	Propiedad que permite elegir la alineación horizontal del texto, puede ser izquierda, derecha, centrada o justificada.
<b>text-decoration</b>	Propiedad que permite elegir el rayado del texto, puede ser ninguno, subrayado, sobre rayado o tachado.
<b>text-indent</b>	Propiedad que permite elegir el sangrado del texto.
<b>text-shadow</b>	Propiedad que permite elegir el sombreado del texto.
<b>vertical-align</b>	Propiedad que permite elegir la alineación vertical del texto.
<b>word-spacing</b>	Propiedad que permite elegir el espacio entre palabras.

Tabla 18: Propiedades del texto de la página web en lenguaje CSS.

Bordes	
Propiedad	Descripción
<b>border</b>	Propiedad que permite elegir las características de los cuatro bordes a la vez.
<b>border-top</b>	Propiedad que permite elegir las características del borde superior.
<b>border-right</b>	Propiedad que permite elegir las características del borde derecho.
<b>border-bottom</b>	Propiedad que permite elegir las características del borde inferior.
<b>border-left</b>	Propiedad que permite elegir las características del borde izquierdo.
<b>border-color</b>	Propiedad que permite elegir el color de los bordes.
<b>border-width</b>	Propiedad que permite elegir el grosor de los bordes.
<b>border-style</b>	Propiedad que permite elegir el estilo de los bordes.
<b>border-radius</b>	Propiedad que permite elegir que las esquinas de los bordes sean redondeadas.

Tabla 19: Propiedades de los bordes de la página web en lenguaje CSS.

Margen exterior	
Propiedad	Descripción
margin	Propiedad que permite elegir los cuatro márgenes exteriores a la vez
margin-top	Propiedad que permite elegir el margen superior.
margin-right	Propiedad que permite elegir el margen derecho.
margin-bottom	Propiedad que permite elegir el margen inferior.
margin-left	Propiedad que permite elegir el margen izquierdo.

Tabla 20: Propiedades de los márgenes exteriores de la página web en lenguaje CSS.

Margen interior	
Propiedad	Descripción
padding	Propiedad que permite elegir los cuatro márgenes interiores a la vez
padding-top	Propiedad que permite elegir el margen superior.
padding-right	Propiedad que permite elegir el margen derecho.
padding-bottom	Propiedad que permite elegir el margen inferior.
padding-left	Propiedad que permite elegir el margen izquierdo.

Tabla 21: Propiedades de los márgenes interiores de la página web en lenguaje CSS.

Fondos	
Propiedad	Descripción
background-color	Propiedad que permite elegir el color del fondo.
background-image	Propiedad que permite elegir una de imagen de fondo.
background-position	Propiedad que permite elegir la posición de la imagen de fondo.
background-repeat	Propiedad que permite elegir la repetición de la imagen de fondo.
background-size	Propiedad que permite elegir el tamaño de la imagen de fondo.

Tabla 22: Propiedades del fondo de la página web en lenguaje CSS.

Tablas	
Propiedad	Descripción
<b>border-collapse</b>	Propiedad que permite elegir el modo que se muestran los bordes de las celdas.
<b>border-spacing</b>	Propiedad que permite elegir la separación entre las celdas.
<b>caption-side</b>	Propiedad que permite elegir la posición de la leyenda.
<b>empty-cells</b>	Propiedad que permite elegir si se muestran o no los bordes de las celdas vacías.

Tabla 23: Propiedades de las tablas de la página web en lenguaje CSS.

Modelo de caja / Tamaño	
Propiedad	Descripción
<b>box-sizing</b>	Propiedad que permite elegir el comportamiento de los bordes y márgenes interior y exterior de una caja.
<b>width</b>	Propiedad que permite elegir la anchura de la caja.
<b>min-width</b>	Propiedad que permite elegir la anchura mínima de la caja.
<b>max-width</b>	Propiedad que permite elegir la anchura máxima de la caja.
<b>height</b>	Propiedad que permite elegir la altura de la caja.
<b>min-height</b>	Propiedad que permite elegir la altura mínima de la caja.
<b>max-height</b>	Propiedad que permite elegir la altura máxima de la caja.

Tabla 24: Propiedades de las cajas de la página web en lenguaje CSS.

Con estas propiedades se crea la presentación de cada parte de la página web, para ello se le asigna características a cada elemento o sección creados en el archivo HTML. En este ejemplo se seleccionan las siguientes características para el menú:

-Fondo de color blanco expresado en hexadecimal.

-Color del texto gris expresado en hexadecimal.

-El borde y los márgenes interior y exterior se añaden al tamaño definido para el elemento.

-El ancho máximo es de 240 píxeles.

-La sombra de la caja tiene un desplazamiento horizontal de 1 píxel, un desplazamiento vertical de 2 píxeles, esta difuminada 4 píxeles, la sombra es del mismo tamaño del elemento y del color negro.

-Tiene un margen interior de 12 píxeles arriba y abajo y de 8 píxeles a la izquierda y derecha.

-El borde es de 1 píxel de grosor de estilo sólido y color blanco.

```

.form-basic select{
  background-color: #ffffff;
  color: #5f5f5f;
  box-sizing: border-box;
  max-width: 240px;
  box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);
  padding: 12px 8px;
  border: 1px solid #dbdbdb;
}

```

Figura 81: Código CSS creando la presentación del menú desplegable del formulario.

Las características del estilo del resto de elementos se crean de forma similar, pero utilizando otras propiedades de las tablas anteriores.

## Tutorías

En este apartado se explica los archivos para crear las páginas webs necesarias para administrar las tutorías de los profesores. Es el más complejo y consta de varios archivos PHP, JavaScript, CSS e iconos agrupados de la siguiente forma:

-En la carpeta principal se encuentran los archivos PHP que contienen código HTML y JavaScript y tienen representación gráfica, el usuario puede ver e interactuar con ellos, es decir las páginas web en sí. Aparte aquí están las siguientes subcarpetas que contienen el resto de archivos.

-php: Esta carpeta contiene el resto de archivos PHP que son los encargados de realizar procesar la información y realizar funciones como acceder a la base de datos.

-js: Esta carpeta contiene todos los archivos JavaScript que encargan de enviar información mediante el método post cuando se pulsa un botón o se llama a una función. La información que introduce el usuario se envía desde las páginas web (archivos con código HTML) a los archivos PHP de procesado de información. Aparte de estos archivos en esta carpeta esta la biblioteca multiplataforma JQuery, que contiene una serie de métodos y funciones predefinidas en JavaScript muy utilizados en el desarrollo de páginas y web, y un archivo que contiene todas las funciones de la biblioteca Bootstrap que se explica a continuación.

-css: Esta carpeta contiene todos los archivos CSS de la biblioteca Bootstrap. Esta herramienta está formada por una biblioteca de archivos CSS y JavaScript que contienen múltiples modales y funciones para mejorar la interfaz del sitio web.

-fonts: Esta carpeta contiene todos los iconos de la biblioteca Bootstrap.

A continuación, se explica el código que sea necesario ya que mucha parte de él es similar al explicado anteriormente y la función de cada archivo.

-”Index.php”: Este archivo contiene la página de inicio de sesión y registro de la página web. Estas dos funciones se realizan mediante un formulario en HTML como el explicado en las páginas anteriores con la única diferencia es que los datos no se envían directamente al archivo PHP que los procesa, sino que se envían al archivo JavaScript y función de este es la encargada de enviarlos al archivo PHP que los almacena en la base de datos.



```

<!--////////// Llamada a función JS con los datos del formulario //-->
<form role="form" id="frmestudiante" name="frmestudiante" onsubmit="Registrarestudiante(); return false">
  <div class="col-lg-12">

    <div class="form-group">
      <label>Nombre</label>
      <input name="nombre" class="form-control" required>
    </div>
  </div>

```

Figura 82: Principio del código del formulario de registro.

Aparte incluye los archivos JS que contienen las funciones y las bibliotecas JQuery y Bootstrap.

```

<!--////////// Incluir bibliotecas JQuery y Bootstrap y archivos con funciones JS //-->
<script src="js/jquery-2.2.3.min.js"></script>
<script type="text/javascript" src="js/main.js"></script>
<script src="js/bootstrap.min.js"></script>
<script type="text/javascript">

    function ventananuevo(){
        $('#modal').modal('show');
    }
</script>

```

Figura 83: Código que incluye los archivos JavaScript.

-" frmestudiante.php", "frmprofesor.php" y "frmadministrador.php": Estos archivos contienen la página al iniciar sesión de los diferentes perfiles: estudiante, profesor y administrador. En estas páginas aparece una interfaz con un menú con las diferentes opciones que tiene cada rol. Cuando se accede a una de estas secciones esta interfaz se mantiene añadiendo el contenido de la sección.



Figura 84: Menú de la interfaz del rol de administrador.

A continuación, se explica cada una de las secciones de cada rol.

### Administrador

El primer rol es el de administrador que tiene permisos para acceder y modificar cualquier dato de las tutorías. Este rol tiene las siguientes secciones:

-Administradores y Profesores: Permite agregar o eliminar administradores y profesores de la página web. Este archivo accede a la tabla de la base de datos donde se guardan todos los usuarios y muestra todos los que tienen el rol de la sección mediante un bucle "while". Los nombres de los usuarios se muestran en la columna de una tabla y en la otra aparece un botón para eliminarlos, cuando se pulsa el botón se llama a una de las dos funciones de Java que envía el nombre del usuario mediante el atributo "id" al archivo eliminaradministrador.php o eliminarprofesor.php dependiendo del rol y se elimina de la base de datos. Aparte debajo de la tabla aparece un botón para agregar usuarios mediante un formulario que sigue el mismo proceso para enviar los datos al archivo agregaradmin.php o agregarprofesor.php dependiendo del rol para guárdalos en la base de datos.

Bienvenido Administrador pedro   Administradores   Profesores   Estudiantes   Tutorías   Cambiar contraseña   **CERRAR SESION**

LISTA DE ADMINISTRADORES

NOMBRE	ACCIONES
Adrian Rodriguez	<a href="#">- Eliminar</a>
Pedro Serrano	<a href="#">- Eliminar</a>

[+ Agregar](#)

Figura 85 : Sección de administradores del rol administrador.

```
<?php
//////Llamada a archivo php para la conexión con la base de datos//////
require('php/conexion.php');
//////Acceso a la base de datos para crear la tabla con los nombres de los administradores
$result=mysqli_query($conexion,'SELECT correo FROM usuariotutoria where tipo="admin"');
while ($usuarios=mysqli_fetch_array($result)){
    $nombre=$usuarios['correo'];
    echo "<tr><td id='id:$nombre' class='cam_editable'>".$usuarios['correo'].</td>";
    //////Botón para eliminar administrador//////
    echo"<td id='$nombre' button='true'><button type='button' class='btn btn-danger'><span class='glyphicon glyphicon-minus'></span> Eliminar</button></td>";
    echo"</tr>";
}
?>
```

Figura 86 : Código para crear la tabla de la sección de administradores

-Estudiantes: Permite agregar, eliminar o modificar los datos de los estudiantes. Este archivo sigue el mismo proceso que las secciones anteriores para mostrar, eliminar y agregar los estudiantes con la diferencia que los datos son enviados a los archivos agregarestudiante.php y eliminarestudiante.php. La función de modificar los datos permite cambiar cualquier dato del estudiante menos su correo, que es el identificador único del usuario, mediante cuadros de texto editables. Al modificar alguno de estos cuadros de texto se llama a una función JS que envía ese dato mediante el método post al archivo modificarestudiante.php que lo cambia en la base de datos.

Bienvenido Administrador pedro   Administradores   Profesores   **Estudiantes**   Tutorías   Cambiar contraseña   **CERRAR SESION**

LISTA DE ESTUDIANTES

NOMBRE	APELLIDO	TITULACION	CORREO	TELEFONO	ACCIONES
Arturo	Carpio	Grado en Ingeniería Eléctrica	arturooim323edo.95@gmail.com	6498131237267	<a href="#">- Eliminar</a>
Manolo	Martin	Grado en Ingeniería Eléctrica	arturoomeeeedo.95@gmail.com	6495432312323	<a href="#">- Eliminar</a>

[+ Agregar](#)

Figura 87: Sección de estudiante del rol de administrador.

```
<?php
////////Llamada a archivo php para la conexión con la base de datos////////
require('php/conexion.php');
////////Acceso a la base de datos para crear la tabla con los nombres de los administradores////////
$result=mysqli_query($conexion,"SELECT correo FROM usuariotutoria where tipo="usuario");
while ($usuarios=mysqli_fetch_array($result)){
    $id=$usuarios['correo'];
    $result2=mysqli_query($conexion,"SELECT * FROM datosusuario where correo='$id'");
    $dato=mysqli_fetch_array($result2);
    //////////Creación de las celdas de la tabla con contenido editable////////
    echo "<td id='nombre:$id' class='cam_editable' contenteditable='true'>".$dato['nombre']."</td>";
    echo "<td id='apellido:$id' class='cam_editable' contenteditable='true'>".$dato['apellido']."</td>";
    echo "<td id='titulacion:$id' class='cam_editable' contenteditable='true'>".$dato['titulacion']."</td>";
    echo "<td id='correo:$id' class='cam_editable'>".$dato['correo']."</td>";
    echo "<td id='telefono:$id' class='cam_editable' contenteditable='true'>".$dato['telefono']."</td>";
    //////////Botón para eliminar administrador///
    echo"<td id='$id' button='true'><button type='button' class='btn btn-danger'><span class='glyphicon glyphicon-minus'></span> Eliminar</button></td>";
    echo"</tr>";
}
?>
```

Figura 88: Código para la crear la tabla de la sección de estudiantes

-Tutorías: Muestra el horario de tutorías de cada profesor y permite agrega y eliminar horas a este horario, para ello utiliza el mismo método anterior en el que envía los datos a los archivos nuevohorario.php y eliminarhora.php. También muestra si algún estudiante ha reservado alguna hora de tutoría y permite mover o cancelar la reserva. Al pulsar el botón de “Mover” redirige al administrador a una nueva página con los horarios de tutorías libres para que pueda seleccionar uno y se envían los datos al archivo mover moverestudiante.php para modificarlos en la base de datos.

Bienvenido Administrador Pedro Serrano							
Administradores		Profesores		Estudiantes		Tutorías	
Cambiar contraseña						CERRAR SESION	
HORARIO							
DIA	HORA INICIO	HORA FIN	ESTUDIANTE	MOVER ESTUDIANTE	CANCELAR TUTORIA	PROFESOR	BORRAR HORAS
Lunes	16:00	16:45	arturoolm323edo.95@gmail.com			Antonio Martin	
Jueves	11:30	12:15				Paco Lopez	
Viernes	11:00	12:00				Paco Lopez	

Agregar

Figura 89: Sección de tutorías del rol de administrador.

Bienvenido Administrador Pedro Serrano					
Administradores		Profesores		Estudiantes	
Tutorías				Cambiar contraseña	
CERRAR SESION					
HORARIO					
DIA	HORA INICIO	HORA FINAL	Profesor	ACCION	
Jueves	11:30	12:15	Paco Lopez		
Viernes	11:00	12:00	Paco Lopez		

Figura 90: Página para mover reservas de tutorías

La siguiente tabla resume las funciones del rol de administrador indicando los archivos que se utilizan para ello y las tablas de la base de datos con las que conectan estos archivos:

Archivo PHP de origen	Funciones	Archivo JS	Archivo PHP de destino	Tabla de base de datos
administradores.php	Añadir administrador	main.js	agregaradministrador.php	usuariostutorias
	Eliminar administrador		eliminaradmin.php	usuariostutorias
profesores.php	Añadir profesor	main.js	agregarprofesor.php	usuariostutorias
	Eliminar profesor		eliminarprofesor.php	usuariostutorias
estudiantes.php	Añadir estudiante	mainestudiante.js	agregarestudiante.php	usuariostutorias y datosusuario
	Eliminar estudiante		eliminarestudiante.php	usuariostutorias y datosusuario
	Modificar estudiante		modificaestudiante.php	datosusuario
tutorias.php	Añadir horario tutoría	mainintutoriaadmin.js	nuevohorario.php	horarios
	Eliminar horario tutoría		eliminarhora.php	horarios
	Mover reserva de tutoría		moverestudiante.php	horarios y datosusuario
	Eliminar reserva de tutoría		cancelartutoria.php	horarios y datosusuario

Tabla 25: Archivos y funciones del rol de administrador.

### Estudiante

Este rol solo tiene permisos para modificar sus datos de usuario y poder reservar o cancelar horas de tutorías. La interfaz de la página web es similar a la anterior, pero cambiando las secciones del menú que se explican a continuación:

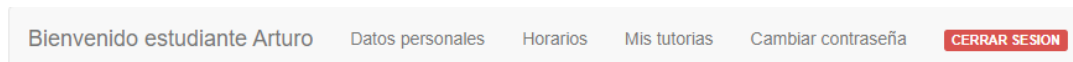


Figura 91: Menú de la interfaz del rol de estudiante.

-Datos personales: En esta sección se muestran los datos de alumno y permite modificarlos mediante cuadros de texto editables. Cuando se modifican el valor de estos cuadros se llama a

una función JS que envía el nombre de la variable y su nuevo valor al archivo `modificarestudiante.php`.

-Horarios: En esta sección se muestran todos los horarios de tutorías que hay sin reservar en una tabla ordenados por profesor y horas. Cada hora dispone de un botón para reservarla, al pulsarlo se llama a una función JS que envía el “id” de la hora de tutoría y el nombre del usuario al archivo `apartartutoria.php` que se encarga de guardar esta información en la base de datos. Solo se permite un máximo de 2 horas de tutorías por alumno así que si el alumno ya las tiene reservadas en vez de la tabla se muestra un mensaje de notificación.

```

//////Acceso a la base de datos para comprobar las horas de tutoria que tiene reservado el alumno//////
$result=mysqli_query($conexion,"SELECT tutoria FROM datosusuario where correo='$user'");
$result2=mysqli_query($conexion,"SELECT tutoria2 FROM datosusuario where correo='$user'");
$tutoria=mysqli_fetch_array($result);
$tutoria2=mysqli_fetch_array($result2);
$codtutoria=$tutoria['tutoria'];
$codtutoria2=$tutoria2['tutoria2'];
//////Si el alumno no tiene 2 horas reservadas//////
if ($codtutoria==0 || $codtutoria2==0 ){
    //////Acceso a la base de datos para crear la tabla con los horarios disponibles de las tutorias ordenados por profesor y horas//////
    $result=mysqli_query($conexion,"SELECT * FROM horarios ORDER BY profesor , id_dia, horainicio");
    while ($horarios=mysqli_fetch_array($result)){
        $id=$horarios['id'];
        if ($horarios['estudiante']==null){
            echo "<tr><td id='dia:$id' class='cam_editable'>".$horarios['dia']."</td>";
            echo "<td id='horas:$id' class='cam_editable'>".$horarios['horainicio']."</td>";
            echo "<td id='horas2:$id' class='cam_editable'>".$horarios['horafinal']."</td>";
            //echo "<td id='estudiante:$id' class='cam_editable'>".$horarios['estudiante']."</td>";
            echo "<td id='profesor:$id' class='cam_editable'>".$horarios['profesor']."</td>";
            echo "<td id='$id' name='$user' button='true'><button type='button' class='btn btn-success'><span class='glyphicon glyphicon-pencil'></span> Apartar</button></td>";
        }
    }
}
    
```

Figura 92: Código para crear lista con tutorías disponibles.

- Mis tutorías: En esta sección se muestran las tutorías que tiene reservadas el alumno y da la opción a eliminarlas mediante un botón, al pulsarlo se envían los datos mediante una función JS al archivo `cancelartutoria.php` que se encarga de eliminarlos de la base de datos.

Bienvenido estudiante Manolo	Datos personales	Horarios	Mis tutorías	Cambiar contraseña	CERRAR SESION
HORARIO					
DIA	HORA INICIO	HORA FIN	PROFESOR	ACCION	
Jueves	11:30	12:15	Paco Lopez	✕ Cancelar Cita	

Figura 93: Lista con tutorías reservadas por el alumno.

La siguiente tabla resume las funciones del rol de estudiante indicando los archivos que se utilizan para ello y las tablas de la base de datos con las que conectan estos archivos:

Archivo PHP de origen	Funciones	Archivo JS	Archivo PHP de destino	Tabla de base de datos
<code>datospersonales.php</code>	Modificar datos del estudiante	<code>main.js</code>	<code>modificarestudiante.php</code>	<code>datosusuario</code>
<code>horarios.php</code>	Reservar tutoría	<code>maintutoria.js</code>	<code>apartartutoria.php</code>	horarios y <code>datosusuario</code>
<code>mistutorias.php</code>	Cancelar reserva de tutoría	<code>maintutoria.js</code>	<code>cancelartutoria.php</code>	horarios y <code>datosusuario</code>

Tabla 26: Archivos y funciones del rol de estudiante.

## Profesor

Este rol solo tiene permisos para realizar acciones relacionadas con sus tutorías como son agregar y eliminar horas de tutoría a su horario, consultar los datos de los estudiantes que han reservado alguna tutoría y cancelar estas reservas. El menú tiene la misma presentación que en los roles anteriores.

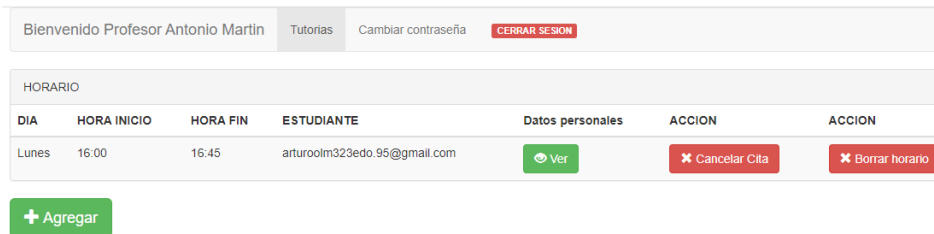


Figura 94: Página de tutorías del rol de profesor.

El funcionamiento de los botones de esta página es el mismo que en las secciones de horarios de tutoría de los otros roles, lo único que varía es el botón “ver”, que permite ver los datos del alumno. Al pulsar este botón se envía el correo del estudiante mediante una función JS usando el método POST al archivo consultarestudiante.php que muestra los datos del usuario, pero en este caso no se pueden modificar.

```

////////llamada a archivo php para la conexión con la base de datos////////
require('php/conexion.php');
$user=$_SESSION['consultado2'];
/////Acceso a la base de datos para crear la tabla con los datos del estudiante/////
$result=mysqli_query($conexion,"SELECT * FROM datosusuario where correo='$user'");
while ($usuarios=mysqli_fetch_array($result)){
    echo "<td id='nombre:$user' class='cam_editable'>".$usuarios['nombre']."</td>";
    echo "<td id='apellido:$user' class='cam_editable'>".$usuarios['apellido']."</td>";
    echo "<td id='titulacion:$user' class='cam_editable'>".$usuarios['titulacion']."</td>";
    echo "<td id='correo:$user' class='cam_editable'>".$usuarios['correo']."</td>";
    echo "<td id='telefono:$user' class='cam_editable'>".$usuarios['telefono']."</td>";
    echo "</tr>";
    }
    
```

Figura 95: Código para generar la lista de las tutorías

La siguiente tabla resume las funciones del rol de profesor indicando los archivos que se utilizan para ello y las tablas de la base de datos con las que conectan estos archivos:

Archivo PHP de origen	Funciones	Archivo JS	Archivo PHP de destino	Tabla de base de datos
tutoriasprofesor.php	Añadir horario tutoría	mainprofesor.js	nuevohorarioprofesor.php	horarios
	Eliminar horario tutoría		eliminarhora.php	horarios
	Cancelar reserva de tutoría		cancelartutoriaprofesor.php	horarios y datosusuario
	Ver datos del estudiante		consultardatos.php	datosusuario

Tabla 27: Archivos y funciones del rol de profesor

## Archivos JavaScript

Estos archivos son los encargados de enviar variables mediante el método POST cuando hay un evento como pulsar un botón a los archivos PHP que conectan con la base de datos. En la siguiente tabla se explican las funciones de la biblioteca jQuery utilizadas en el proyecto:

Función	Explicación	Parámetros
<code>\$(document).ready(function())</code>	Ejecuta función cuando la página web se carga por completo	document(opcional): Indica el DOM de la página web actual. function(obligatorio): Función a ejecutarse
<code>\$(id).blur(function())</code>	Ejecuta la función cuando el elemento con ese id pierde el foco que obtuvo al ser clicado por el cursor o mediante el tabulador.	id(obligatorio): identificador del elemento. function(opcional): Función a ejecutarse
<code>\$(this).attr(id)</code>	Devuelve el valor del atributo de elemento seleccionado con ese id.	selector(obligatorio): Elemento a examinar. El selector "this" indica que solo afecta al elemento que activo el evento. id: identificador del atributo a examinar.
<code>\$(this).text(id)</code>	Devuelve el texto del elemento seleccionado.	selector(obligatorio): Elemento a examinar. El selector "this" indica que solo afecta al elemento que activo el evento. id: identificador del atributo a examinar.
<code>\$(id).click(function())</code>	Ejecuta la función cuando el elemento con ese id es clicado.	id(obligatorio): identificador del elemento. function(opcional): Función a ejecutarse.
<code>\$(id).change(function())</code>	Ejecuta la función cuando el elemento con ese id cambia de valor.	id(obligatorio): identificador del elemento. function(opcional): Función a ejecutarse.
<code>document.getElementById(id).value</code>	Devuelve el valor del atributo con el id indicado	id(obligatorio): identificador del atributo.
<code>window.location.href = archivo</code>	Redirige a la url del archivo indicado.	archivo(obligatorio): nombre del archivo al que redirigir.

<b>alert(mensaje)</b>	Muestra un cuadro de alerta con un mensaje y un botón de "OK"	mensaje(obligatorio): Mensaje a mostrar en el cuadro.
<b>console.log(mensaje)</b>	Muestra un mensaje en consola.	mensaje(obligatorio): Mensaje a mostrar en consola.
<b>\$.post(archivo, variable, función)</b>	Envía datos al archivo indicado mediante el método post y realiza una función.	archivo(obligatorio): nombre del archivo al que enviar los datos. variable(opcional): Especifica la variable a enviar. función(opcional): Indica la función a realizar cuando se realice el envío.
<b>\$(selector).serialize(tipo)</b>	Devuelve los valores de los campos de un formulario en el formato adecuado para enviarlos mediante una llamada AJAX(método POST o GET)	selector(obligatorio): Elemento a examinar. tipo(opcional): Indica el tipo de campos a devolver.

Tabla 28: Funciones de la biblioteca jQuery

A continuación, se muestra un ejemplo de función JavaScript que se activa al pulsar un botón y envía los datos para eliminar un administrador, el resto de funciones de JavaScript tiene una estructura similar.

```

    /// Se activa la función al pulsar el boton "td[button=true]" en la página web
    $("td[button=true]").click(function(){
        // Cuadro con mensaje de confirmación
        if (confirm("esta seguro")) {
            // Guardar el valor del atributo
            var field_id=$(this).attr("id");
            console.log('id:'+field_id);
            //Enviar mediante el metodo post
            $.post('php/eliminaradmin.php', {
                "var_1": field_id},function(respuesta){
                //Si el envio se realizo correctamente se recarga la pagina
                if (respuesta=="true")
                    window.location.reload(true);
                // Si no se muestra el mensaje de alerta
                else
    
```

Figura 96: Código en JavaScript para enviar datos a archivos PHP.

```

    /////Botón para eliminar administandor///
    echo"<td id='$nombre' button='true'><button type='button' class='btn btn-danger'><span class='glyphicon glyphicon-minus'></span> Eliminar</button></td>";
    
```

Figura 97: Código en PHP para generar el botón que activa la función en JavaScript.



Por último, se muestra esta tabla con todos los archivos JavaScript y la funciones que contienen:

Archivo JavaScript	Función	Archivo PHP destino
<b>main.js</b>	Modificar los datos de un estudiante con el rol de estudiante.	modificarestudiante.php
	Eliminar administrador con el rol de administrador.	eliminaradmin.php
	Registrar nuevo administrador con el rol de administrador.	agregaradministrador.php
	Registrar nuevo estudiante desde página de inicio.	agregarestudiante.php
	Registrar nuevo profesor con el rol de administrador.	agregarprofesor.php
	Iniciar sesión con cualquier rol.	iniciarsesion.php
<b>mainestudiante.js</b>	Modificar los datos de un estudiante con el rol de administrador	modificarestudiante.php
	Eliminar estudiante con el rol de administrador.	eliminarestudiante.php
	Registrar nuevo estudiante con el rol de administrador.	agregarestudiante.php
<b>mainmover.js</b>	Mover reserva de tutoría de un estudiante con el rol de administrador.	moverestudiante.php
<b>mainprofesor.js</b>	Consultar los datos de un alumno que haya reservado una tutoría con el rol de profesor	consultardatos.php
	Cancelar una reserva de tutoría de un alumno con el rol de profesor.	cancelartutoriaprofesor.php
	Eliminar horas del horario de tutoría con el rol de profesor	eliminarhora.php
	Añadir horas del horario de tutoría con el rol de profesor	nuevohorarioprofesor.php
<b>maintutoria.js</b>	Reservar una hora de tutoría con el rol de alumno.	apartartutoria.php
	Cancelar una reserva de tutoría con el rol de alumno.	cancelartutoria.php

maintutoriaadmin.js	Consultar los datos de un alumno para mover una tutoría con el rol de administrador	consultardatos.php
	Cancelar una reserva de tutoría de un alumno con el rol de administrador.	cancelartutoria.php
	Eliminar horas del horario de tutoría de un profesor con el rol de administrador.	eliminarhora.php
	Añadir horas del horario de tutoría de un profesor con el rol de administrador.	nuevohorario.php
change.js	Modificar la contraseña de inicio de sesión con cualquier rol.	modificarpasword.php
bootstrap.js	Biblioteca de estilos y modelos de Bootstraps	No aplica
bootstrap.min.js	Biblioteca de estilos y modelos de Bootstraps	No aplica
jquery-2.2.3.min.js	Biblioteca de funciones jQuery.	No aplica
npm.js	Módulo de CommonJS	No aplica

Tabla 29: Archivos y funciones JavaScript

### Archivos PHP que conectan con base de datos

Estos archivos son los encargados de recibir las variables de las funciones JavaScript y conectar con la base de datos. Las funciones de todos estos archivos están enumeradas en la tabla anterior y la estructura del código de estos es la siguiente:

-Primero realizan la conexión con la base datos.

```

<?php
//Si se ha recibido algun dato mediante el método POST se conecta con la base de datos
if(!empty($_POST)){
$conexion=(mysqli_connect("b1juo0jvigeqlfih9e3h-mysql.services.clever-cloud.com","u4iuxsebomgh6lc9","T6wXfQxdfwfwjttivkin","b1juo0jvigeqlfih9e3h"));
//mysqli_select_db($conexion,'b1juo0jvigeqlfih9e3h1') or die ("no se encuentra la bd");
//Si hay un error de conexión se notifica
if (mysqli_connect_errno()) {
    echo "Error al conectar a la base de datos";
    exit();
};
    
```

Figura 98: Código con la conexión a la base de datos.

-Segundo reciben los datos del archivo JavaScript mediante la el método POST.

```

///Se reciben los datos de JS mediante el método POST
$nombre=$_POST['nombre'];
$password=$_POST['password'];
$tipo='admin';
$id=null;
    
```

Figura 99: Código que recibe los datos del archivo JavaScript.

-Por último, realizan la acción con la base de datos, ya sea una consulta, guardar, actualizar o borrar datos de alguna tabla.

```

///Se realiza la consulta a la base de datos
$consultarusuario="SELECT * FROM usuariotutoria where correo='$nombre'";
$resultadousuario=mysqli_query($conexion,$consultarusuario);
$busquedausuario=mysqli_fetch_array($resultadousuario);
/// Si no hay coincidencia con la consulta se guarda el nombre de usuario
if(empty($busquedausuario)){
$insertar="INSERT INTO usuariotutoria (correo, password, tipo) VALUES ('$nombre', '$password', '$tipo')";
mysqli_query($conexion,$insertar) or die ("NO se pudo insertar");
mysqli_close($conexion);
echo"true";
/// Si el nombre de usuario ya existe se notifica
}else{
if(!empty($busquedausuario)){
echo "el nombre de usuario ya esta registrado";
}
}
}
  
```

Figura 100: Código que realiza la consulta a la base de datos.

### 6.3 Base de datos

En este apartado se explican las características y funciones de las columnas de cada tabla de la base de datos. La primera tabla es la encargada de guardar la información del formulario de opinión completado por los usuarios, los datos que almacena son los siguientes:

Columna	Tipo de dato	Características especiales	Función
<b>idOpiniones</b>	Numérico (INT)	Identificador único (Primary Key); No puede ser nulo (Not Null); Suma uno al valor cada nuevo registro (Auto Increment)	Identificador de cada opinión de los usuarios.
<b>Id_chat</b>	Cadena de caracteres (VARCHAR)	No aplica	Identificador de Messenger del usuario enviado por Chatfuel
<b>Nombre</b>	Cadena de caracteres (VARCHAR)	No aplica	Nombre del usuario
<b>Apellido</b>	Cadena de caracteres (VARCHAR)	No aplica	Apellido del usuario
<b>Email</b>	Cadena de caracteres (VARCHAR)	No aplica	Email del usuario
<b>Calificación</b>	Cadena de caracteres (VARCHAR)	No aplica	Nota que dio el usuario al chatbot
<b>Opinión</b>	Texto (TEXT)	No aplica	Opinión del usuario sobre el chatbot

<b>Suscrito</b>	Cadena de caracteres (VARCHAR)	No aplica	Indica si el usuario desea suscribirse a que le enví noticias a través del chat
<b>Genero</b>	Cadena de caracteres (VARCHAR)	No aplica	Genero de usuario
<b>Fecha</b>	Cadena de caracteres (VARCHAR)	No aplica	Fecha en la que se realizó el formulario
<b>Zona_horaria</b>	Cadena de caracteres (VARCHAR)	No aplica	Zona horaria del usuario
<b>Pais</b>	Cadena de caracteres (VARCHAR)	No aplica	País del usuario
<b>Idioma</b>	Cadena de caracteres (VARCHAR)	No aplica	Idioma del usuario

Tabla 30: Descripción de las columnas de tabla "Opiniones" de la base datos

La segunda tabla almacena la información de las consultas de los usuarios cuando tienen alguna duda que el chatbot no es capaz de resolver:

Columna	Tipo de dato	Características especiales	Función
<b>idUsuarios</b>	Numérico (INT)	Identificador único (Primary Key); No puede ser nulo (Not Null); Suma uno al valor cada nuevo registro (Auto Increment)	Identificador de cada consulta realizada por los usuarios.
<b>Id_chat</b>	Cadena de caracteres (VARCHAR)	No aplica	Identificador de Messenger del usuario enviado por Chatfuel
<b>Nombre</b>	Cadena de caracteres (VARCHAR)	No aplica	Nombre del usuario
<b>Apellido</b>	Cadena de caracteres (VARCHAR)	No aplica	Apellido del usuario
<b>Email</b>	Cadena de caracteres (VARCHAR)	No aplica	Email del usuario
<b>Teléfono</b>	Numérico (INT)	No aplica	Teléfono del usuario
<b>Consulta</b>	Texto (TEXT)	No aplica	Consulta del usuario

<b>Genero</b>	Cadena de caracteres (VARCHAR)	No aplica	Genero de usuario
<b>Fecha</b>	Cadena de caracteres (VARCHAR)	No aplica	Fecha en la que se realizó la consulta
<b>Zona_horaria</b>	Cadena de caracteres (VARCHAR)	No aplica	Zona horaria del usuario
<b>Pais</b>	Cadena de caracteres (VARCHAR)	No aplica	País del usuario
<b>Idioma</b>	Cadena de caracteres (VARCHAR)	No aplica	Idioma del usuario

*Tabla 31: Descripción de las columnas de tabla "Usuarios" de la base de datos*

La tercera tabla almacena los datos de inicio de sesión de la página web para reservar tutorías:

Columna	Tipo de dato	Características especiales	Función
<b>correo</b>	Cadena de caracteres (VARCHAR)	Identificador único (Primary Key); No puede ser nulo (Not Null);	Correo o nombre para iniciar sesión.
<b>password</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Contraseña para iniciar sesión
<b>tipo</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Rol de usuario, puede ser administrador, profesor o estudiante

*Tabla 32: Descripción de las columnas de tabla "usuariostutoria" de la base de datos*

La cuarta tabla almacena los datos de los alumnos de la página web de reserva de tutorías:

Columna	Tipo de dato	Características especiales	Función
<b>Nombre</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Nombre del alumno
<b>Apellido</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Apellido del alumno
<b>Titulación</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Titulación del alumno

<b>Correo</b>	Cadena de caracteres (VARCHAR)	Identificador único (Primary Key); No puede ser nulo (Not Null) ;	Correo del alumno para iniciar sesión.
<b>Teléfono</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Teléfono del alumno.
<b>Tutoria1</b>	Numérico (INT)	No puede ser nulo (Not Null)	Número que almacena el identificador del horario de tutoría si el alumno realiza una reserva
<b>Tutoria2</b>	Numérico (INT)	No puede ser nulo (Not Null)	Número que almacena el identificador del horario de tutoría si el alumno realiza una reserva

Tabla 33: Descripción de las columnas de tabla "datosusuario" de la base de datos

La última tabla almacena todo lo referente a los horarios de tutorías de los profesores.

Columna	Tipo de dato	Características especiales	Función
<b>id</b>	Numérico (INT)	Identificador único (Primary Key); No puede ser nulo (Not Null) ; Suma uno al valor cada nuevo registro(Auto Increment)	Identificador único de una tutoría
<b>horainicio</b>	Numérico (INT)	No puede ser nulo (Not Null)	Hora de inicio de la tutoría
<b>horafinal</b>	Numérico (INT)	No puede ser nulo (Not Null)	Hora de final de la tutoría
<b>día</b>	Cadena de caracteres (VARCHAR)	No puede ser nulo (Not Null)	Día de la tutoría
<b>id_dia</b>	Numérico (INT)	No puede ser nulo (Not Null)	Número del día de la semana de la tutoría para poder ordenarlos en la página web



<b>estudiante</b>	Cadena de caracteres (VARCHAR)	No aplica	Estudiante que ha reservado la tutoría
<b>profesor</b>	Cadena de caracteres (VARCHAR)	No aplica	Profesor que hace la tutoría

*Tabla 34: Descripción de las columnas de tabla "horarios" de la base de datos*

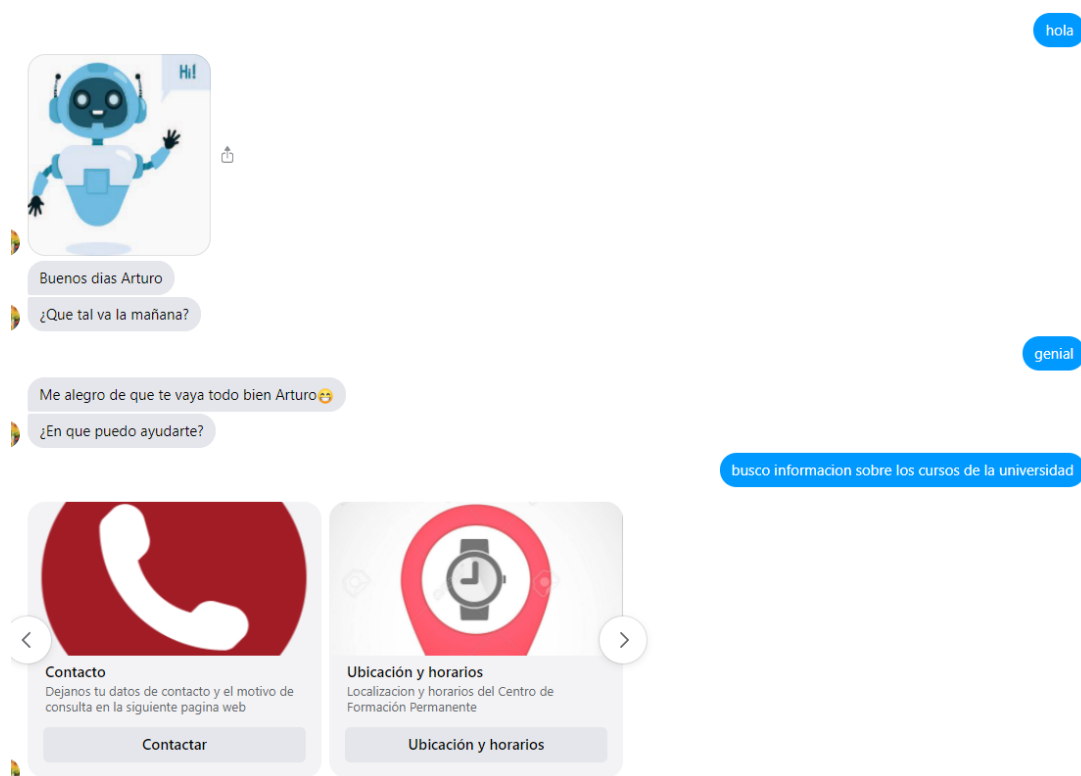
## 7. PRUEBAS Y ANÁLISIS DE RESULTADOS

### 7.1 Pruebas del servidor de producción

Para realizar las pruebas del chatbot se simulan varias conversaciones con usuarios que buscan diferentes tipos de información, con el fin de detectar y resolver posibles errores para que el bot no tenga puntos muertos.

#### Primer usuario

El primer usuario que se simula busca información sobre los cursos y las instalaciones deportivas de la universidad preguntando por estos temas directamente sin acceder al menú. El usuario pregunta por los cursos después de que el bot lo salude y accede al menú con los enlaces a los apartados del Centro de Formación Permanente.



*Figura 101: Chat para prueba del bot con usuario 1*



Ubicación y horarios

Nuestros horarios son:  
 -De Lunes a Viernes 9:30h a 13.30h  
 Horario reducido: de Lunes a Viernes 10:00h a 13:00h del 16 de junio al 15 de septiembre (agosto cerrado)  
 Semana Santa, Feria de Sevilla y Navidad

Estamos en Paseo de las Delicias s/n  
 Pabellón de México41013 Sevilla



gracias

De nada Arturo, me alegro de haberte ayudado 😊

No dudes en preguntarme si necesitas algo mas

Figura 102: Chat para prueba del bot con usuario 1

Por último, el usuario busca información para realizar deporte y el bot le conduce al menú con la información del SADUS donde se encuentran todos los enlaces relacionados con las instalaciones deportivas de la universidad.

me gustaría hacer deporte este año



**Instalaciones**  
 Conoce nuestras diversas instalaciones

Ver instalaciones



**Horarios**  
 Calendario de horarios de las diferentes instalaciones de la US

Horarios

Ver instalaciones



**CDU Los Bermejales**  
 El Campus Deportivo Universitario los Bermejales

Ver Bermejales



**CED Pirotecnia**  
 Centro Educativo Deportivo Pirotecnia

Ver Pirotecnia

gracias por la informacion

hasta luego

Me alegro de haber hablado con tigo 😊

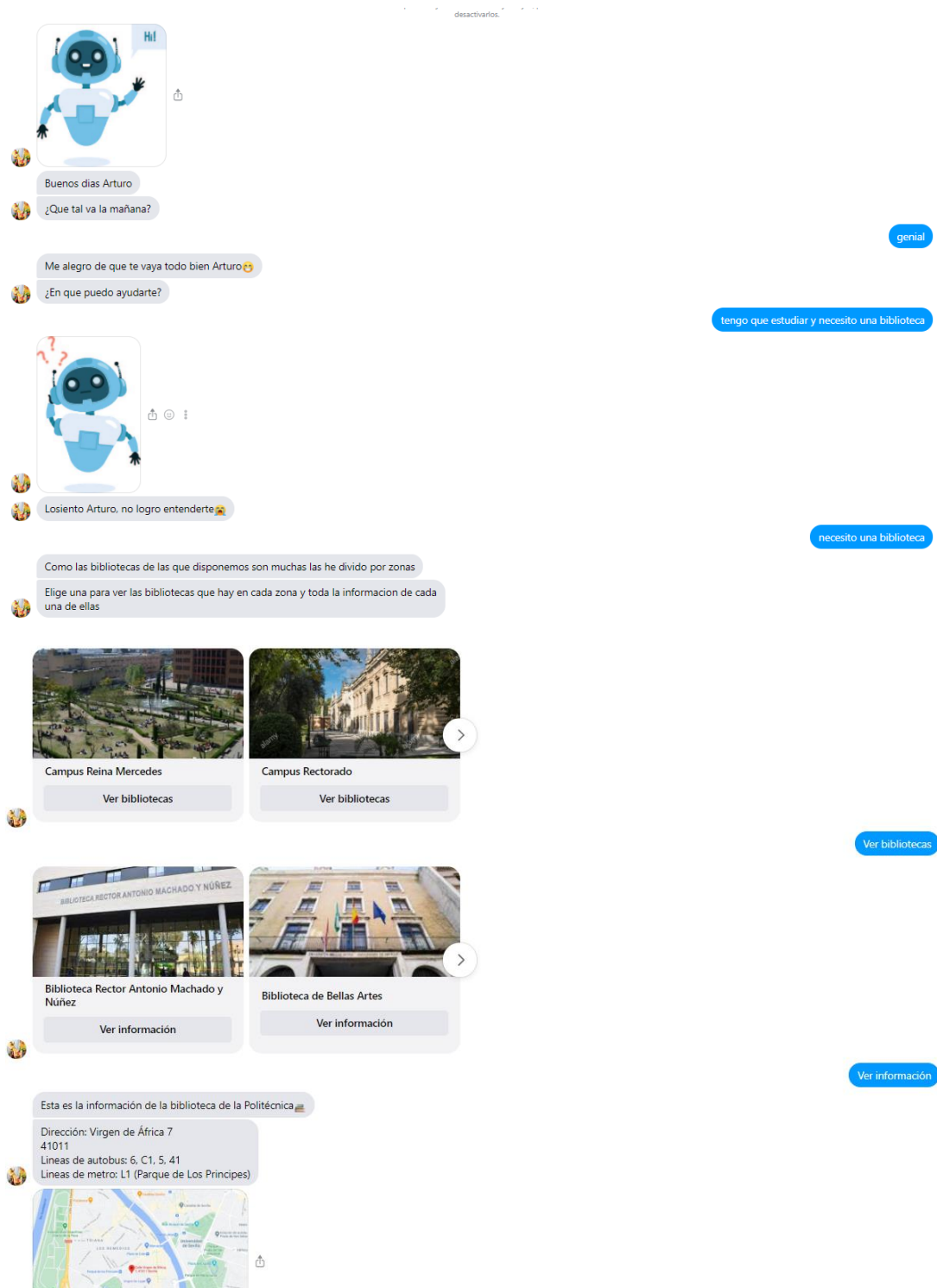
Cuando lo necesites aqui estoy

Hasta luego 😊

Figura 103: Chat para prueba del bot con usuario 1

## Usuario 2

El segundo usuario que se simula busca información sobre las bibliotecas disponibles de la universidad. El bot lo conduce al menú con las diferentes bibliotecas agrupadas por campus y el usuario accede a la información de los horarios y ubicación de una de ellas.

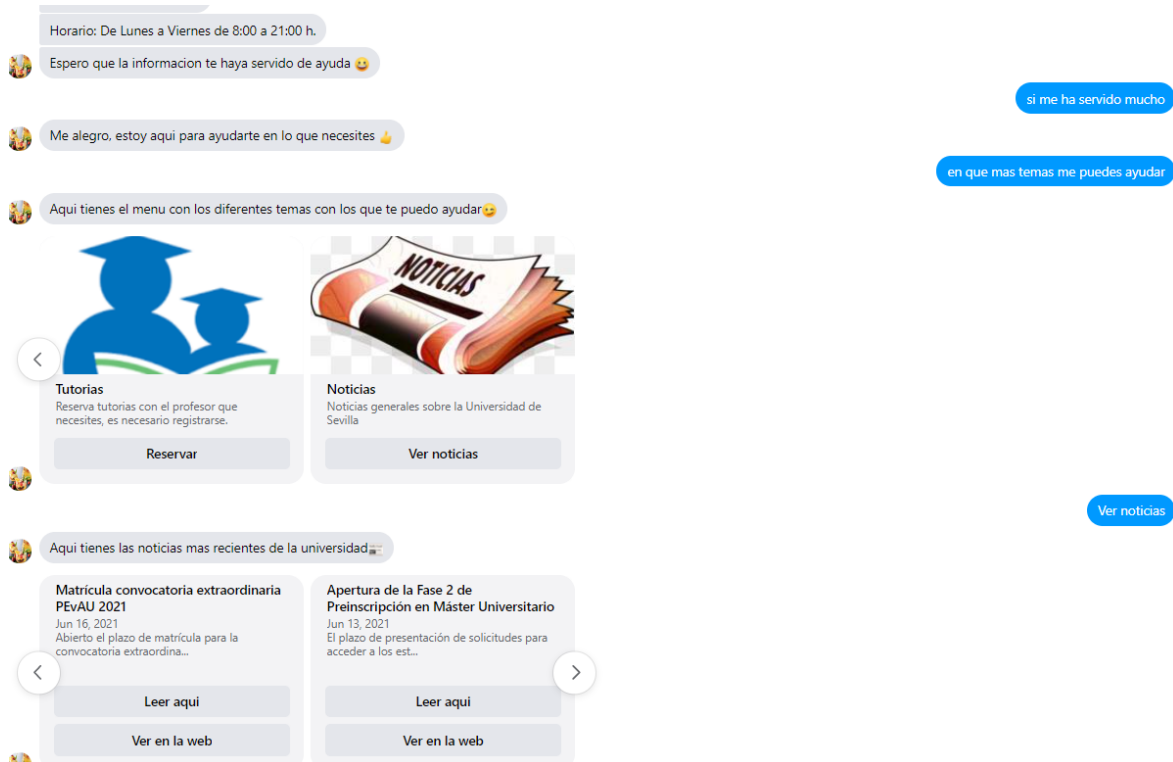


The chat interface shows the following sequence of messages and actions:

- User:** "Buenos dias Arturo"
- Bot (Arturo):** "¿Que tal va la mañana?"
- User:** "Me alegro de que te vaya todo bien Arturo 😊"
- Bot (Arturo):** "¿En que puedo ayudarte?"
- User:** "tengo que estudiar y necesito una biblioteca"
- Bot (Arturo):** "genial."
- User:** "Losiento Arturo, no logro entenderte 😞"
- Bot (Arturo):** "Como las bibliotecas de las que disponemos son muchas las he dividido por zonas. Elige una para ver las bibliotecas que hay en cada zona y toda la informacion de cada una de ellas"
- User:** "necesito una biblioteca"
- Bot (Arturo):** Presents two options:
  - Campus Reina Mercedes:** "Ver bibliotecas"
  - Campus Rectorado:** "Ver bibliotecas"
- User:** "Ver bibliotecas"
- Bot (Arturo):** Presents two library options:
  - Biblioteca Rector Antonio Machado y Núñez:** "Ver información"
  - Biblioteca de Bellas Artes:** "Ver información"
- User:** "Ver información"
- Bot (Arturo):** "Esta es la información de la biblioteca de la Politécnica 📄"
  - Dirección: Virgen de África 7 41011
  - Lineas de autobus: 6, C1, S, 41
  - Lineas de metro: L1 (Parque de Los Principes)
  - Mapa de ubicación de la biblioteca.

Figura 104: Chat para prueba del bot con el usuario 2

Después el usuario consulta por los diferentes temas en los que el bot le puede ayudar y es redirigido al menú principal donde accede a la sección de novedades de la universidad.




Horario: De Lunes a Viernes de 8:00 a 21:00 h.

Espero que la información te haya servido de ayuda 😊


Me alegro, estoy aquí para ayudarte en lo que necesites 🙌

Aquí tienes el menú con los diferentes temas con los que te puedo ayudar 😊



**Tutorías**  
Reserva tutorías con el profesor que necesites, es necesario registrarse.

Reservar



**Noticias**  
Noticias generales sobre la Universidad de Sevilla

Ver noticias

Aquí tienes las noticias más recientes de la universidad 📰:

**Matrícula convocatoria extraordinaria PEvAU 2021**  
Jun 16, 2021  
Abierto el plazo de matrícula para la convocatoria extraordinaria...

Leer aquí

Ver en la web

**Apertura de la Fase 2 de Preinscripción en Máster Universitario**  
Jun 13, 2021  
El plazo de presentación de solicitudes para acceder a los est...

Leer aquí

Ver en la web

Ver noticias

Figura 105: Chat para prueba del bot con el usuario 2

Una vez consultadas las novedades, el usuario quiere información sobre las asignaturas de su carrera, pero el bot no tiene asociado la palabra “asignatura” con ningún bloque y ofrece al usuario la opción de usar el buscador en la web de la universidad.



Ver en la web

Ver en la web

Leer aquí

Se ha publicado Resolución Rectoral por la que se establece el procedimiento de acceso, matrícula y desarrollo de las enseñanzas por esta vía.

quiero saber las asignatura de las carreras

Upps, no entiendo lo que me quieres decir Arturo

Preguntamelo de otra forma por favor

Entiendo mejor las frases cortas con palabras clave 😊

asignaturas

Gracias por intentarlo Arturo pero no consigo entender: "asignaturas" 🙄

Quieres ponerte en contacto con alguien del personal o si no buscarlo en nuestra página web

Iniciar chat 🗨️

Buscar en web 🔍

Buscar en web 🔍

Figura 106: Chat para prueba del bot con el usuario 2

La información que se encuentra en la web no es la que busca el usuario y se le da la opción de contactar con algún miembro del personal, pero al estar en verano están de vacaciones y no se puede iniciar un chat en vivo. Para mantener la calidad del servicio el bot permite al

usuario dejar sus datos de contacto, que se guardan en la base de datos, para que pasadas las vacaciones alguien de personal contacte con él y resuelva su duda.



Figura 107: Chat para prueba del bot con el usuario 2

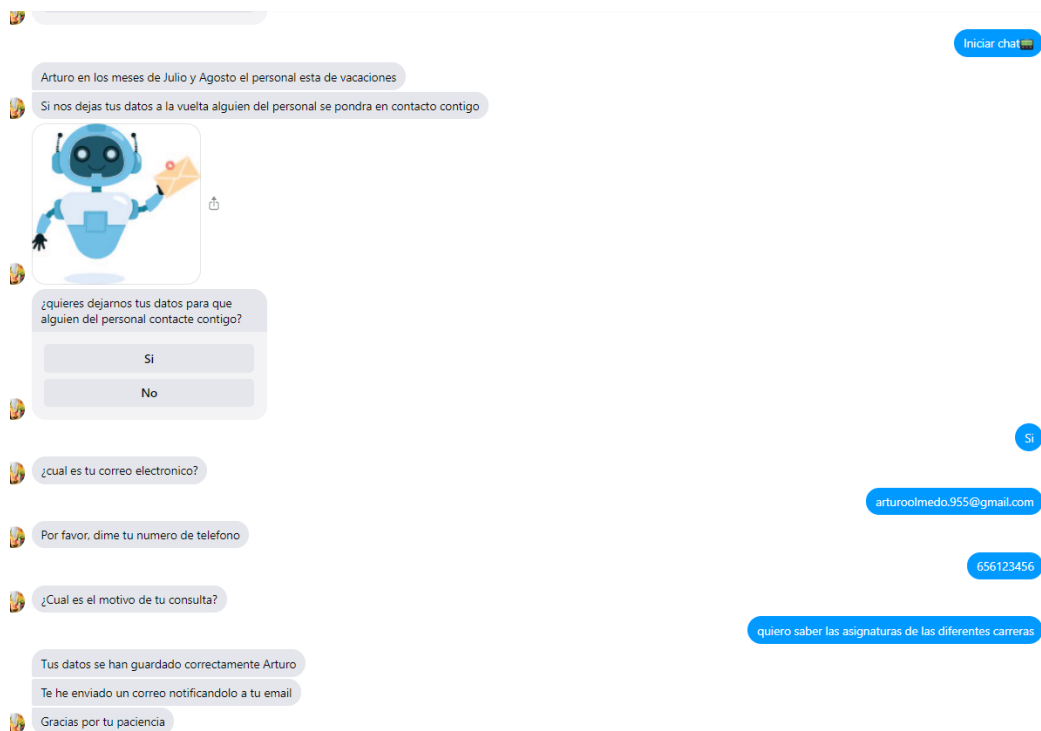
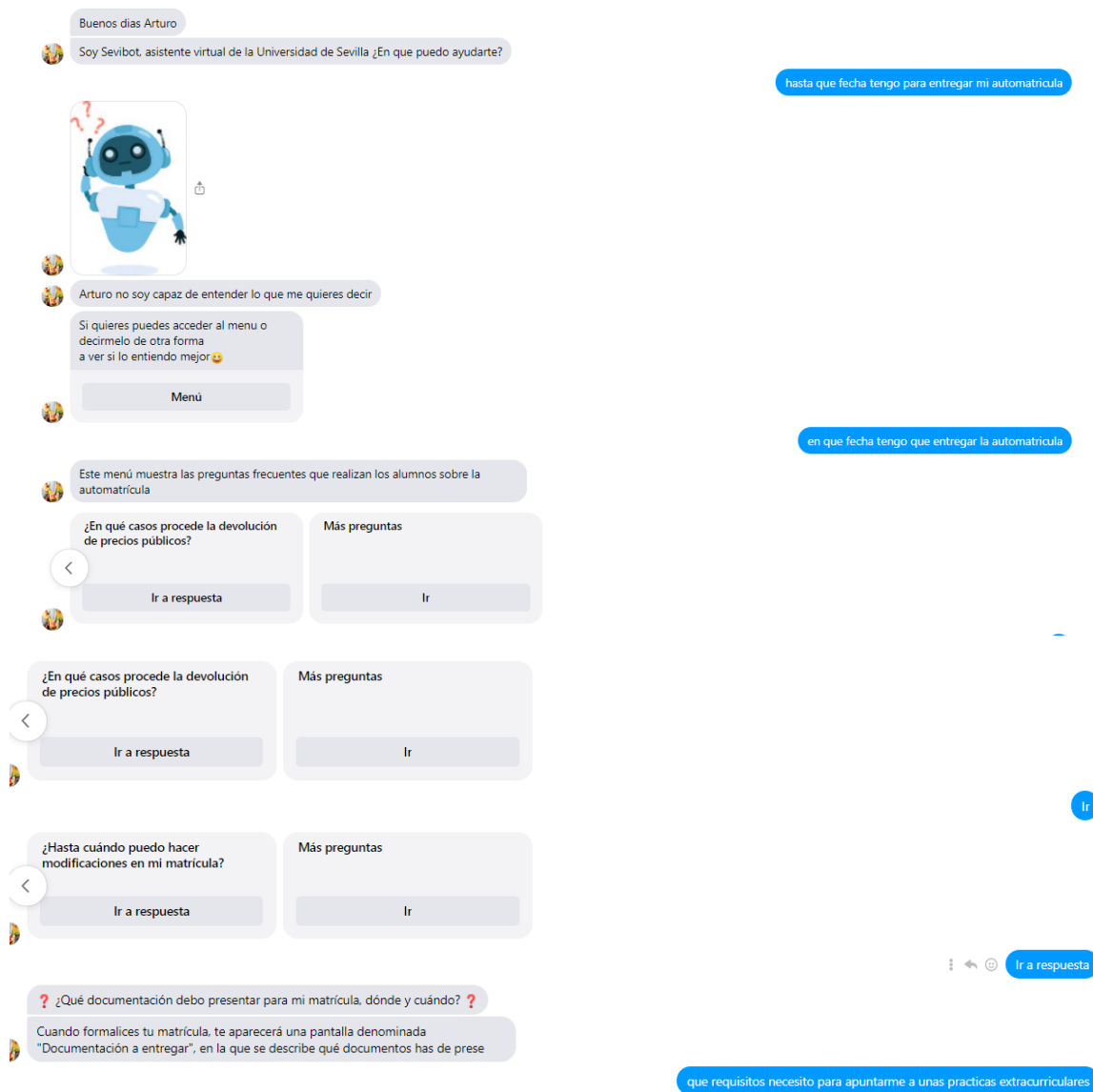


Figura 108: Chat para prueba del bot con el usuario 2

### Usuario 3

El tercer usuario que se simula busca información sobre la fecha de entrega de la automatrícula, como el bot no reconoce la frase inicial del usuario le contesta con una de sus frases por defecto para que el usuario simplifique la frase para buscar la información. Con la segunda frase del usuario el bot reconoce el tema “automatrícula” y conduce al usuario hasta el menú con las preguntas frecuentes sobre ese tema.



Buenos días Arturo

Soy Sevibot, asistente virtual de la Universidad de Sevilla ¿En que puedo ayudarte?

hasta que fecha tengo para entregar mi automatrícula

Arturo no soy capaz de entender lo que me quieres decir

Si quieres puedes acceder al menu o decirmelo de otra forma a ver si lo entiendo mejor 😊

Menú

Este menú muestra las preguntas frecuentes que realizan los alumnos sobre la automatrícula

¿En qué casos procede la devolución de precios públicos? Más preguntas

Ir a respuesta Ir

¿En qué casos procede la devolución de precios públicos? Más preguntas

Ir a respuesta Ir

¿Hasta cuándo puedo hacer modificaciones en mi matrícula? Más preguntas

Ir a respuesta Ir

¿Qué documentación debo presentar para mi matrícula. dónde y cuándo? ?

Cuando formalices tu matrícula, te aparecerá una pantalla denominada "Documentación a entregar", en la que se describe qué documentos has de prese

que requisitos necesito para apuntarme a unas practicas extracurriculares

Figura 109: Chat para prueba del bot con el usuario 3

Por último, el usuario busca información sobre las prácticas en empresas externas y el bot lo conduce al menú de preguntas frecuentes de las practicas siguiendo un procedimiento similar al anterior.

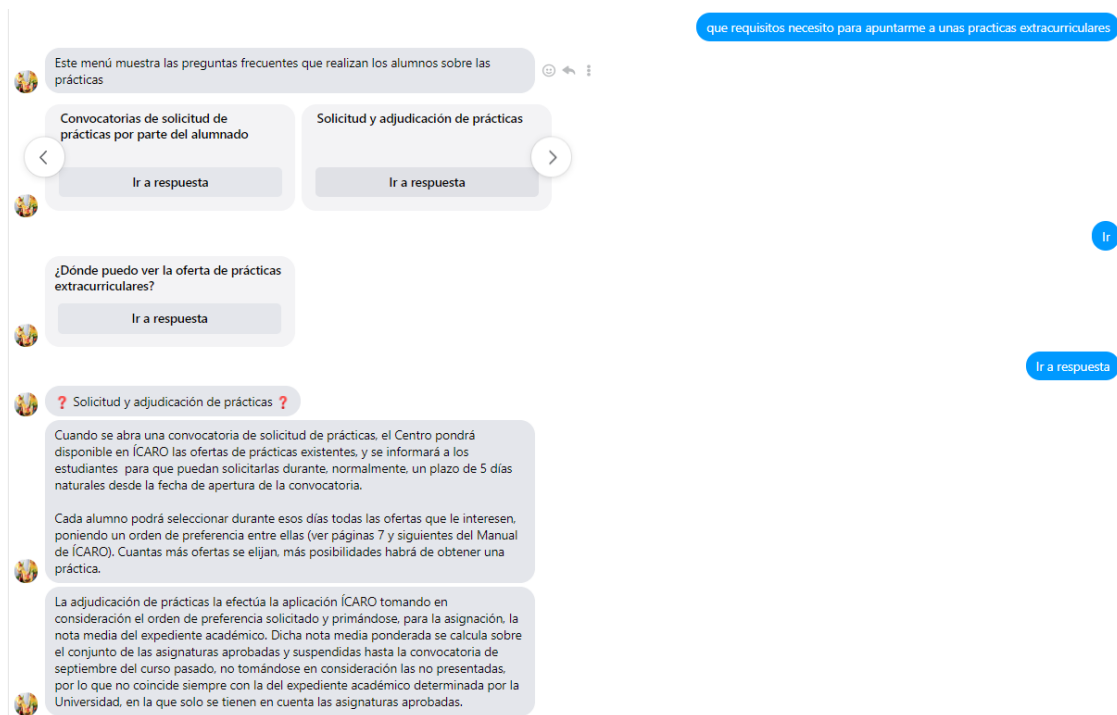


Figura 110: Chat para prueba del bot con el usuario 3

#### Usuario 4

El cuarto usuario que se si mula está interesado en reservar una tutoría de uno de sus profesores y accede a la página web para ello mediante el enlace que le proporciona el bot. Después el usuario que ya había interactuado con el bot antes sabe que puede escuchar música mientras espera a alguien del personal, aunque estén de vacaciones.



Figura 111: Chat para prueba del bot con el usuario 4

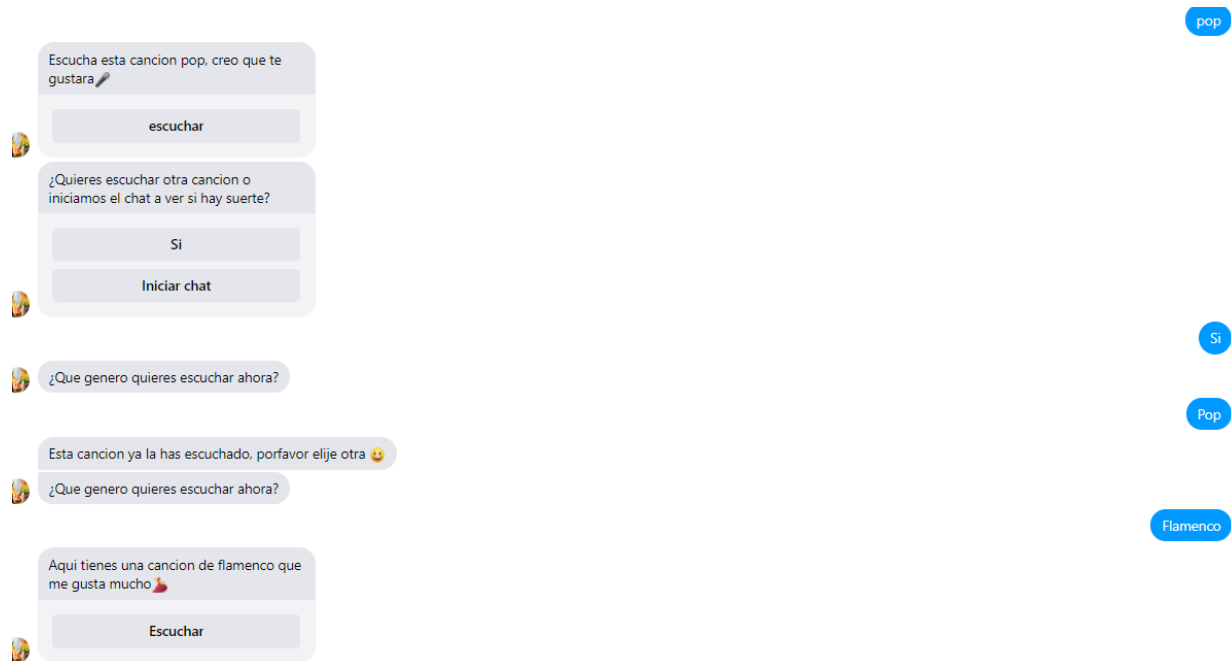


Figura 112: Chat para prueba del bot con el usuario 4

## 7.2 Respaldo de información y políticas de seguridad

### 7.2.1 Recopilación de información

El chatbot de este proyecto solo recopilara de los usuarios la información estrictamente necesaria para poder realizar sus funciones:

- Los datos del usuario proporcionados por Facebook que son nombre y apellido de la cuenta, ID de Messenger del usuario, fuente por la que accedió al chat, genero, país de localización y zona horaria.
- Los datos recopilados mediante el formulario de opinión que son nombre, apellido, email, opinión sobre el bot y etapa de estudios del usuario.
- Los datos de contacto para resolver dudas mediante el personal de la atención al cliente que son email, teléfono y motivo de consulta.
- Los datos para registraste en la página web para reservar tutorías que son nombre, apellido, email, grado que se está estudiando en la universidad y una contraseña para la cuenta.

### 7.2.2 Uso de datos personales

Los datos personales de los usuarios recopilados mediante Chatfuel se utilizan para los siguientes propósitos:

- Verificar la identidad de los usuarios en la plataforma de Facebook.
- Proporcionar los datos necesarios para el personal de atención al cliente para poder contactar con los usuarios.
- Analizar el nivel de satisfacción de los usuarios con el servicio del chatbot.

- Crear perfiles de usuarios con los datos relevantes del estudiante para poder reservar tutorías.

### 7.2.3 Divulgación de datos personales

Los datos personales proporcionados por los usuarios no se divulgarán entre terceros, solo estarán disponibles para profesores, administradores del chat y personal de atención al cliente.

### 7.2.4 Gestión del cambio

Hay una serie de procedimientos para introducir el chatbot de forma adecuada y realizar una correcta gestión del cambio, para preparar a una empresa para la implementación del chatbot. Hay varias tareas a realizar como la comunicación a las personas afectadas, la formación de los agentes o el soporte y seguimiento del chatbot.



Figura 113: Tareas asociadas a la gestión del cambio

Esta serie de tareas aplicadas a este proyecto son las siguientes:

- Informar a los profesores y personal de la universidad sobre la integración de un chatbot.
- Formar al personal encargado de administrar el contenido del bot y al personal encargado de atender a los usuarios en caso de que el bot no sea capaz de atender sus necesidades.
- Comunicar a los alumnos la introducción del nuevo chatbot mediante la página web de la universidad, los profesores y emails divulgativos.
- Seguimiento de las conversaciones del bot para solucionar problemas y errores.
- Analizar las opiniones de los usuarios para posibles mejoras.



## 8. FUTURAS AMPLIACIONES Y MEJORAS

### 8.1 Mejoras

Una vez creado el bot se plantean una serie de mejoras y ampliaciones de contenido para implementar más adelante y mejorar el servicio que se les dará a los usuarios en el futuro. Estas son las posibles mejoras planteadas:

-Aumentar la base de datos de la inteligencia artificial con las frases que el bot no reconoce de las conversaciones que mantiene con los usuarios para así eliminar los puntos muertos de las conversaciones. Para ello se utiliza una herramienta de Chatfuel que te indica las frases que condujeron al bot al bloque “Default Answer” ordenadas según con la frecuencia que se utilizaron. Para mejorar aún más la inteligencia artificial conectar el chatbot con Dialogflow mediante Janis para usar funciones de su inteligencia artificial como:

- Permiten utilizar contextos que perduran en la conversación durante una serie de interacciones, es decir si el bot reconoce que el usuario está hablando sobre ropa este contexto permanecerá un número de interacciones y cualquier frase que el bot no reconozca como otro contexto se le asignará a la ropa como por ejemplo si usa una frase indicando el color.
- Dispone de machine learning mediante el cual el bot aprende solo según los usuarios interactúen con él. También permite introducir textos en Word o mediante urls para que el bot aumente la base de datos de su inteligencia artificial, pero esta función solo está permitida con textos en inglés.

-Mejorar la página web de tutorías añadiendo un video chat que permita realizar las tutorías de forma telemática, reduciendo el contacto y el tiempo que los estudiantes necesitan para resolver sus dudas con los profesores.

-Añadir más secciones de preguntas frecuentes como:

- Explicación del contenido y salidas laborales de las asignaturas de cada grado que oferta la universidad.
- Resolución de dudas sobre todos los temas relacionados con el COVID-19, como si va haber exámenes presenciales, medidas a tomar cuando se asiste a algún centro de la US, etc.

-Añadir una sección con información sobre colegios mayores y residencias para facilitar a los estudiantes que no residan en Sevilla a encontrar un lugar para poder realizar sus estudios.

### 8.2 Conclusiones

Tras finalizar el proyecto se ha comprobado que utilizar un chatbot mejora el servicio al cliente en cualquier empresa proporcionando las siguientes ventajas:

- Reduce los costos del servicio de atención al cliente quitándole carga de trabajo al personal.
- Reduce el tiempo de respuesta a los usuarios y permite resolver cualquier duda mientras este en su base de datos.
- Ofrece atención 24 horas al día permitiendo al usuario resolver sus problemas en cualquier momento.

- Permite conocer mejor a los usuarios para mejorar el servicio gracias a sus herramientas de análisis de las conversaciones. Esto sería muy tedioso si la atención fuera vía telefónica o en persona.

Aunque también la implementación de un chatbot tiene inconvenientes como:

- Solo pueden resolver dudas sobre temas simples, aunque puede redirigir a los usuarios con un agente cuando el bot no sea capaz de resolver un problema.
- Se necesita un trabajo inicial para el desarrollo del bot que puede resultar complejo.
- Requiere un continuo mantenimiento para que no quede obsoleto.
- Hay que formar al personal e informar a los usuarios de este nuevo servicio.

Para la creación de chatbots hay múltiples plataformas disponibles para conseguir este cometido, según el propósito del bot es conveniente elegir una determinada plataforma. Para el caso particular del bot de la universidad se ha seleccionado Chatfuel por ser muy intuitivo a la hora de desarrollar, ser gratuito y la integración con APIs externas es muy sencilla. Gracias a esta herramienta se añade el contenido y se dota de funciones al bot como:

- Los diferentes menús con toda la información que proporciona el bot.
- Se añade contenido multimedia como imágenes, videos y audios.
- Se conecta con aplicaciones externas como base de datos o páginas web.
- Envío automático de emails de notificación tanto a usuarios como personal.
- Chat en vivo con agentes encargados de resolver dudas que el bot no tiene capacidad para solucionar.

Aparte para complementar el bot se han creado páginas web con información y funciones que no eran recomendables realizar mediante un chat como reservar tutorías o realizar un cuestionario de opinión.

En conclusión, un bot siempre va a proporcionar una ventaja a tu empresa a largo plazo mejorando la experiencia del usuario y reduciendo costos. Ofreciendo múltiples ventajas a los usuarios siempre que se establezca correctamente el propósito, alcance y funciones que tiene que cumplir este para poder seleccionar correctamente la plataforma para su desarrollo y gestionar correctamente el cambio para su implementación tanto formando a el personal como informando a los usuarios consiguiendo mejorar considerablemente el servicio de atención al cliente.

## Bibliografía

[1] La línea de tiempo de la historia de los Chatbots: antes, ahora y mañana.

<https://planetachatbot.com/linea-tiempo-historia-de-chatbots-antes-ahora-y-manana/>

[2] Procesamiento del lenguaje natural. Qué es y por qué es importante

[https://www.sas.com/es\\_ar/insights/analytics/what-is-natural-language-processing-nlp.html#nlphistory](https://www.sas.com/es_ar/insights/analytics/what-is-natural-language-processing-nlp.html#nlphistory)

[3] Procesamiento del Lenguaje Natural (NLP).

<https://www.aprendemachinelearning.com/procesamiento-del-lenguaje-natural-nlp/>

[4] Curso gratis de Chatfuel.

<https://botifica.com/curso-gratis-chatfuel/>

[5] Aprende a crear Chatbots con Chatfuel y Dialogflow como un profesional.

<https://www.udemy.com/course/curso-chatfuel-gratis/>

[6] Chatbots: La guía definitiva (2020).

<https://www.artificial-solutions.com/es/chatbots/>

[7] Documentación de Dialogflow.

<https://cloud.google.com/dialogflow/es/docs/quick/setup?hl=es>

[8] Documentación de Chatfuel.

<https://docs.chatfuel.com/>

[9] Documentación de Watson.

<https://developer.ibm.com/es/technologies/conversation/tutorials/desarrollando-chatbots-con-watson-conversation/>

[10] Manual de MySQL Workbench

<https://dev.mysql.com/doc/workbench/en/>

[11] Manual de PHP.

<https://www.php.net/manual/es/index.php>

[12] Manual de HTML.

<https://desarrolloweb.com/manuales/manual-html.html>

[13] Políticas de privacidad de chatbots.

<https://botonchat.com/politica-de-privacidad/>

[14] Ciclo de Deming

[https://es.wikipedia.org/wiki/Ciclo\\_de\\_Deming](https://es.wikipedia.org/wiki/Ciclo_de_Deming)