



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Desarrollo y explotación de  
un chatbot de FAQs sobre  
una asignatura de Trabajo  
Fin de Grado  
UBU-Chatbot**



Presentado por Alfredo Asensio Vázquez  
en Universidad de Burgos — julio de 2021  
Tutor: Dr. Raúl Marticorena Sánchez



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	2
A.3. Estudio de viabilidad . . . . .	6
<b>Apéndice B Especificación de Requisitos</b>	<b>11</b>
B.1. Introducción . . . . .	11
B.2. Objetivos generales . . . . .	11
B.3. Catalogo de requisitos . . . . .	12
B.4. Especificación de requisitos . . . . .	13
<b>Apéndice C Especificación de diseño</b>	<b>19</b>
C.1. Introducción . . . . .	19
C.2. Diseño de datos . . . . .	19
C.3. Diseño procedimental . . . . .	23
C.4. Diseño arquitectónico . . . . .	24
<b>Apéndice D Documentación técnica de programación</b>	<b>25</b>
D.1. Introducción . . . . .	25
D.2. Estructura de directorios . . . . .	25
D.3. Manual del programador . . . . .	26

D.4. Compilación, instalación y ejecución del proyecto . . . . .	41
D.5. Pruebas del sistema . . . . .	41
<b>Apéndice E Documentación de usuario</b>	<b>45</b>
E.1. Introducción . . . . .	45
E.2. Requisitos de usuarios . . . . .	45
E.3. Instalación . . . . .	47
E.4. Manual del usuario . . . . .	52
<b>Bibliografía</b>	<b>67</b>

---

# Índice de figuras

---

A.1. Cronograma del proyecto . . . . .	5
B.1. Diagrama de casos de uso. . . . .	13
C.1. Entidades utilizadas en el agente. . . . .	19
C.2. Diseño de <i>intents</i> del agente para la modalidad online (Parte 1). . . . .	20
C.3. Diseño de <i>intents</i> del agente para la modalidad online (Parte 2). . . . .	21
C.4. Primera página del documento recopilatorio de preguntas y res- puestas. . . . .	22
C.5. Diagrama de secuencia. . . . .	23
C.6. Diagrama de despliegue. . . . .	24
D.1. Creación de cuenta para acceder a la herramienta online. . . . .	26
D.2. Creación de agente en Dialogflow. . . . .	27
D.3. Configuración básica del agente. . . . .	27
D.4. Panel de <i>intents</i> . . . . .	28
D.5. Creación de un <i>intent</i> . . . . .	28
D.6. Implementación de una Entidad. . . . .	30
D.7. Edición de JSON para una <i>Rich Response</i> con Notepad++. . . . .	31
D.8. Añadiendo <i>Rich Response</i> a un <i>intent</i> . . . . .	32
D.9. Panel de <i>fulfillments</i> de Dialogflow. . . . .	33
D.10. Menú de integraciones de Dialogflow. . . . .	34
D.11. Ventana modal para la integración web. . . . .	34
D.12. Creación de página en el curso de UBUVirtual. . . . .	35
D.13. Insertar el código HTML. . . . .	35
D.14. Web para la creación del espacio de trabajo. . . . .	36
D.15. Modal de la integración de Slack. . . . .	37
D.16. Selección de espacio de trabajo para la integración. . . . .	38

D.17.Chatbot integrado correctamente en Slack. . . . .	39
D.18.Personalización del CSS de la interfaz conversacional. . . . .	39
D.19.Custom Element para el <i>chatbot</i> de Dialogflow. . . . .	40
D.20.Uso del control de versiones desde la interfaz del IDE. . . . .	40
D.21.Validación del agente. . . . .	42
E.1. Email de confirmación una vez recibido acceso . . . . .	48
E.2. Pantalla para unirse al espacio de trabajo de Slack . . . . .	49
E.3. Confirmación para crear la cuenta y acceder al espacio de trabajo. . . . .	50
E.4. Espacio de trabajo de Slack . . . . .	51
E.5. Visualización del chatbot inicialmente minimizado. . . . .	52
E.6. Ventana del chatbot desplegada. . . . .	53
E.7. Respuesta a saludo. . . . .	54
E.8. Pregunta bien formulada. . . . .	55
E.9. Distintas formulaciones. . . . .	56
E.10.Pregunta mal formulada. . . . .	57
E.11.Cerrar chatbot. . . . .	58
E.12.Chatbot cerrado. . . . .	59
E.13.Página del chatbot desde Safari en iOS. . . . .	60
E.14.Ventana del chatbot en Safari iOS. . . . .	61
E.15.Página chatbot desde Safari en iOS. . . . .	62
E.16.Respuesta del chatbot y botón de cierre de teclado desde Safari en iOS. . . . .	63
E.17.Volver a la página anterior desde Safari en iOS. . . . .	64
E.18.Página principal del espacio de trabajo de Slack. . . . .	65
E.19.Página del chatbot en Slack. . . . .	66

---

# Índice de tablas

---

A.1. Costes humanos. . . . .	6
A.2. Costes hardware. . . . .	7
A.3. Costes de licencias. . . . .	7
A.4. Costes de redes y comunicación. . . . .	8
A.5. Costes infraestructura. . . . .	8
A.6. Costes totales. . . . .	8
A.7. Licencias software. . . . .	9
B.1. CU-01 Expandir la interfaz conversacional. . . . .	14
B.2. CU-02 Formular una pregunta. . . . .	15
B.3. CU-03 Formular una pregunta. (Slack) . . . . .	16
B.4. CU-04 Ocultar interfaz conversacional. . . . .	17





## *Apéndice A*

---

# Plan de Proyecto Software

---

## A.1. Introducción

Para el correcto desarrollo del proyecto es imprescindible establecer una planificación adecuada. En esta planificación se evalúa la viabilidad de la solución propuesta y, una vez se ha determinado que es viable, se establecen una serie de etapas que va a seguir la implementación. También se fija el calendario con los plazos para cada una de estas etapas.

Este Plan de Proyecto Software se divide en dos apartados:

- **Planificación temporal:** establece los plazos de desarrollo del proyecto. Se establece una fecha de inicio y fin de proyecto, así como el intervalo de fechas previstas para cada desarrollo incremental. Las fechas marcadas han sufrido adaptaciones a lo largo del desarrollo y las cargas de trabajo de los intervalos no son semejantes. La disponibilidad del alumno ha marcado el volumen de trabajo de cada una de las fases.
- **Estudio de viabilidad:** antes de comenzar a desarrollar el proyecto es necesario determinar si es viable.
  - **Viabilidad económica:** realizar una estimación de los posibles costes y beneficios. Se analizan todo tipo de costes: humanos, software, hardware, instalaciones...
  - **Viabilidad legal:** determinar si este proyecto cumple todas las leyes y regulaciones.

## A.2. Planificación temporal

En el desarrollo del proyecto se siguió la metodología ágil Scrum [2], estudiada a lo largo de la carrera. No se aplicó de manera estricta esta metodología, ya que se dieron varios factores que no lo hicieron posible:

- No había un equipo de trabajo como tal, únicamente el alumno y tutor. En ocasiones puntuales colaboró un tercero: el presidente del tribunal.
- Al compaginar estudio y trabajo los *sprints* en algunas ocasiones no fueron tan breves como se hubiera deseado, aunque nunca excedieron el máximo de un mes que establece esta metodología.

De la misma manera, se puede determinar que se utilizó esta metodología basándonos en los siguientes puntos:

- Desarrollo incremental del producto software por medio de *sprints*.
- Duración general de dos semanas para los sprints.
- Reuniones entre alumno y tutor al final de cada *sprint*, en las que se analizaba el trabajo realizado y se establecían los objetivos del siguiente sprint.

### Sprint 0 (23/10/20 - 06/11/20)

El inicio de este *sprint* lo marcó la primera reunión con Raúl Marticorena, en la que se dieron las indicaciones de lo que se buscaba con el proyecto y se establecieron los primeros objetivos.

Se estableció LaTeX como herramienta a utilizar para la documentación de la memoria y anexos.

Las primeras tareas fueron: estudiar las FAQ de la asignatura a incorporar en la base de datos del *chatbot* y determinar qué herramienta se iba a utilizar.

### Sprint 1 (07/11/20 - 20/11/20)

Se establece Dialogflow como motor NLP elegido y se crea el proyecto. También se crea el espacio de trabajo de *Slack*. Se crea un documento en el que ir recopilando las preguntas y respuestas de manera conjunta con Carlos López. En la reunión de final de *sprint* se estudian las herramientas disponibles para trabajar con LaTeX.

Como objetivos se opta por analizar las posibles integraciones y estudiar la viabilidad de añadir hipervínculos en las respuestas del *chatbot*.

## **Sprint 2 (21/11/20 - 27/11/20)**

Del *sprint* anterior se tomó la decisión de realizar la integración externa en Slack. Se empieza a trabajar introduciendo *intents* a la base de datos del *chatbot*. Como metas se establecen el trabajo en la memoria estudiando los trabajos relacionados y la descarga de un Moodle local para probar la integración del *chatbot*.

## **Sprint 3 (28/11/20 - 11/12/20)**

Se trabajó en la integración con UBUVirtual y en la reunión del final del *sprint* se probó conjuntamente a integrarlo, de manera que quedase oculto a todos los usuarios. Se logró dicha integración de una primera versión muy básica del *chatbot*, sin ningún tipo de personalización CSS ni soporte de hipervínculos.

## **Sprint 4 (12/12/20 - 15/01/21)**

Este *sprint* tuvo una duración algo mayor debido a los exámenes del primer cuatrimestre. Durante este periodo se siguió trabajando en la base de datos de *intents*. Por otra parte, se personalizó el código CSS del *chatbot*, dotándolo de un mayor atractivo y pasando a ser visible para los usuarios el día 12 de enero. Se estudió y probó el funcionamiento de las *Rich Responses* para ambas integraciones.

## **Sprint 5 (16/01/21 - 29/01/21)**

En este *sprint* se comenzaron a recibir los primeros *logs* de conversaciones reales llevadas a cabo por otros estudiantes con nuestro *chatbot*. Se trabajó en analizar estas preguntas ampliando la base de datos de *intents* con las que no se contaba anteriormente y arreglando los fallos que surgían en el algoritmo de asignación de *intents*. Se implementaron respuestas para *Slack* y las *Rich Responses*.

## **Sprint 6 (30/01/21 - 05/03/21)**

Tuvo una duración mayor a un mes por motivos de disponibilidad. En este tiempo se centraron los esfuerzos en la implementación de las *Rich Responses*. Se integró en *Slack* y fueron añadiéndose respuestas para esta implementación.

### **Sprint 7 (06/03/21 - 02/04/21)**

Se continúa trabajando en el análisis de *logs*, arreglando todas las conversaciones. Se continúa creando *intents* e implementando *Rich Responses* para todos aquellos que contienen un hipervínculo o una lista. Empieza a cobrar mayor importancia la documentación del proyecto en la que se empiezan a invertir más horas y se centra el foco de las reuniones con el tutor.

### **Sprint 8 (03/04/21 - 07/05/21)**

En este periodo se continuó ampliando la base de datos del NLP realizando revisiones periódicas de los *logs* y añadiendo las nuevas preguntas sin respuesta al documento para que Raúl Marticorena en colaboración con Carlos López les diesen respuesta.

Se realiza un estudio numérico en el que todas las conversaciones son analizadas y se determina para cada pregunta de los usuarios si se contesta de manera correcta, incorrecta, o si la pregunta era incoherente. Por medio de Excel se realiza un estudio numérico y gráfico de los resultados obtenidos. En la reunión se comenta este gráfico analizando los resultados obtenidos hasta la fecha, se comenta la situación del proyecto y las tareas restantes, que giraban principalmente en torno a la documentación.

### **Sprint 9 (08/05/21 - 21/05/21)**

Se continúa con el análisis de *logs*, ampliación de base de datos de *intents* y documentación de la memoria. Se estudian los conceptos teóricos necesarios para la comprensión del trabajo y se realiza su documentación en la memoria. En la reunión se comentan aspectos relativos a estos.

### **Sprint 10 (22/05/21 - 09/06/21)**

Se incentiva a personas ajenas al proyecto a utilizar el *chatbot* con el objetivo de obtener una última gran recopilación de *logs* para poder analizar en vista de que se está en las fases finales del proyecto. Pruebas de compatibilidad del *chatbot* con distintos navegadores web y de *smartphones*. Se sigue trabajando en la documentación y la reunión gira en torno a esta.

### **Sprint 11 (10/06/21 - 18/06/21)**

En esta semana se dedica la mayor parte del tiempo a la documentación de la memoria. Se añaden nuevas preguntas a la base de datos del NLP

generadas a partir de los *logs*.

También se genera e integra el logo del proyecto en el *chatbot* de UBUVirtual. En la reunión se comentan algunos problemas encontrados en la integración de Slack y se fija como objetivo arreglarlos.

## Sprint 12 (19/06/21 - 24/06/21)

Los esfuerzos se centraron en la elaboración de la memoria y anexos. Se finaliza la implementación de las respuestas en la versión de Slack y se actualizan las de UBUVirtual para ambas modalidades. Con esta última actualización se da por cerrados los incrementos en la base de datos del *NLP*. También se actualiza el estudio numérico con las últimas conversaciones registradas.

## Sprint 13 (25/06/21 - 02/07/21)

Se realizan correcciones en la memoria y se trabaja en la elaboración de los anexos. En la reunión de final del *sprint* se comentan aspectos relativos a ambos documentos.

## Cronograma

En la Figura A.1 se muestra el cronograma seguido por el proyecto. La implementación se refiere a todo aquello relacionado con los componentes de Dialogflow.

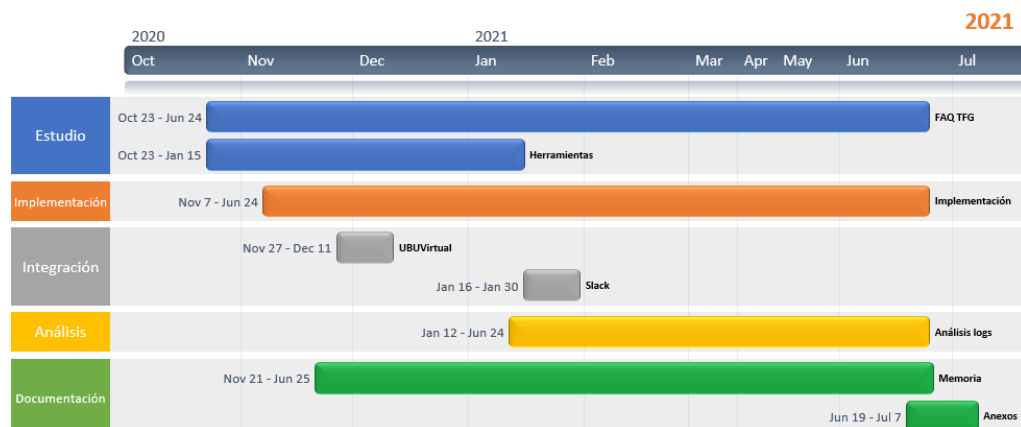


Figura A.1: Cronograma del proyecto

### A.3. Estudio de viabilidad

#### Viabilidad económica

Este apartado va a analizar los costes y posibles beneficios del proyecto vistos desde una perspectiva empresarial; se tendrán en cuenta los recursos humanos, así como todos los gastos derivados que se generarían en una empresa real.

#### Costes humanos

Se va a extrapolar el total de horas dedicadas al proyecto -se estiman 300- al equivalente en tiempo que supondría para una jornada completa de 8h/diarias. Contando días festivos esto produce una equivalencia aproximada de dos meses de trabajo a jornada completa.

El salario mensual bruto de un Ingeniero Informático junior se sitúa en torno a los 18.000 € brutos anuales, por lo que se va a suponer un sueldo mensual de 1500 € brutos.

Los costes de la Seguridad Social para la empresa son los siguientes: 23,6 % costes comunes, 0,2 % fogasa, 0.6 % formación profesional y una cotización de entre el 5,5-6,7 %, que vamos a suponer del 5,6 % para redondear, sumando en total un 30 %.

Para el trabajador la Seguridad Social supone un 4,7 % [5]. El IRPF es de un 12 % para este tramo. [3]

La empresa paga un 30 % de estos 1500 € a la Seguridad Social. De estos 1500€ el trabajador paga a la seguridad social un 4,7 % y el IRPF lo que supone 250,5 € (70,5 + 180). Por lo que el sueldo neto será de 1249,5 €.

En la Tabla A.1 se calcula el coste humano total.

Concepto	Coste
Salario mensual bruto	1500 €
Cotización seguridad social	450 €
Total mensual	1950 €
Total 2 meses	3900 €

Tabla A.1: Costes humanos.

### Costes hardware

En la Tabla A.2 se muestra el gasto total en componentes físicos. Se asume un periodo de amortización para el hardware de 3 años (36 meses) y que han sido utilizados 2 meses.

Concepto	Coste	Coste amortizado
Ordenador portátil	800 €	44,5 €
Smartphone	300 €	16,7 €
Ratón	40 €	44,5 €
Monitor	100 €	5,5 €
Teclado	15 €	0,8 €
Total	1255 €	70 €

Tabla A.2: Costes hardware.

### Costes licencias

La Tabla A.3 recoge el coste total de las licencias software utilizadas. Se asume un periodo de amortización para las licencias software de 6 años (72 meses) y que han sido utilizadas 2 meses.

Concepto	Coste	Coste amortizado
Windows 10 Home [13]	135 €	3,8 €
Dialogflow Free Trial [8]	0 €	0 €
Adobe Photoshop [1]	12 €/mes	12 €/mes
Total	27,8 €	27,8 €

Tabla A.3: Costes de licencias.

### Costes redes y comunicación

La Tabla A.4 recoge el coste total de la conexión a Internet.

Concepto	Coste
Internet [15]	30 € /mes
Total	60 €

Tabla A.4: Costes de redes y comunicación.

### Costes infraestructura

La Tabla A.5 recoge el coste de alquiler aproximado de una oficina pequeña en Zaragoza. El precio de alquiler del metro cuadrado en Zaragoza es en el momento de rectar este documento de 8.4 € [11].

Concepto	Coste
Alquiler oficina [11]	450 € /mes
Total	900 €

Tabla A.5: Costes infraestructura.

### Costes totales

En la Tabla A.6 se muestra el coste total del proyecto..

Tipo coste	Coste
Humano	3000 €
Hardware	70 €
Licencias	27,8 €
Redes y comunicación	60 €
Infraestructura	900 €
Total	4057,8 €

Tabla A.6: Costes totales.



## Beneficios

Para obtener beneficios podrían implementarse *chatbots* similares para otras universidades o cualquier entidad que podría mejorar su negocio con la incorporación de este elemento para la respuesta de preguntas frecuentes. El precio aproximado podría rondar los 4000 € por licencia, siendo el mantenimiento un servicio opcional con un precio aproximado de 800 € al mes.

Por lo que con la venta de una única licencia sin mantenimiento se habría amortizado el desarrollo.

## Viabilidad legal

En la Tabla A.7 se recoge la información relativa a las licencias software de los programas utilizados en el proyecto.

Software	Licencia	Fuente
Dialogflow	Google APIs Terms of Service	[7]
Notepad++	GNU GPL	[10]
Visual Studio Code	MIT License	[14]
Slack	Slack API Terms of Service	[16]
JSON	The JSON License	[12]

Tabla A.7: Licencias software.

En el caso de Dialogflow la licencia es distinta para la versión de prueba que estamos utilizando que para las versiones de pago y cuya especificación se encuentra en el enlace de la tabla anterior. Al no tener un fin comercial nuestro proyecto, podemos utilizarla.

La licencia JSON es libre, simple, sin *copyleft* y permisiva.  
GNU GPL es libre, abierta y con *copyleft*.  
MIT License es abierta, son *copyleft* y permisiva.



## Apéndice *B*

---

# Especificación de Requisitos

---

### B.1. Introducción

En este apartado se detallan los objetivos generales del proyecto, los requisitos y la especificación de los mismos. Se analizan tanto los requisitos funcionales como los no funcionales.

- Requisitos funcionales: comportamientos específicos que debe tener el sistema. Se relacionan con los casos de uso.
- Requisitos no funcionales: especifican los criterios a seguir, restricciones y condiciones que impone el cliente.

### B.2. Objetivos generales

- Implementar un *chatbot* que responda las preguntas frecuentes de la asignatura Trabajo de Fin de Grado tanto para la modalidad online como presencial.
- Ser capaces de dar respuesta al mayor número de preguntas posible.
- Integrar el *chatbot* en UBUVirtual y Slack.
- Diseñar una interfaz conversacional que resulte agradable a los usuarios y respete la imagen corporativa.
- Capturar los *logs* de las conversaciones para analizarlos y extraer resultados.

## B.3. Catalogo de requisitos

### Requisitos funcionales

- **RF-1 Interacción textual:** la aplicación debe permitir al usuario interactuar con ella mediante texto.
- **RF-2 Reconocimiento de preguntas:** la aplicación debe ser capaz de reconocer las preguntas que introduce el usuario.
- **RF-3 Respuesta a las preguntas:** la aplicación debe ser capaz de responder a las preguntas que le formule el usuario.
- **RF-4 Información mediante hipervínculos:** la aplicación debe adjuntar como hipervínculos las redirecciones a otras páginas.
- **RF-5 Minimizar la interfaz conversacional:** la aplicación debe permitir minimizar la interfaz conversacional.
- **RF-6 Expandir la interfaz conversacional:** la aplicación debe permitir expandir la interfaz conversacional.
- **RF-7 Informe de error:** la aplicación debe informar de un error al usuario cuando su mensaje de entrada no sea un texto válido o no haya sido posible interpretarlo.

### Requisitos no funcionales

- **RNF-1 Rendimiento:** la aplicación tiene que tener un tiempo de respuesta bajo.
- **RNF-2 Usabilidad:** la aplicación debe ser intuitiva y fácil de entender y utilizar.
- **RNF-3 Colores corporativos:** la aplicación debe mantener los colores corporativos de UBUVirtual.
- **RNF-4 Disponibilidad:** la aplicación debe estar disponible el mayor tiempo posible.
- **RNF-5 Mantenibilidad:** la aplicación debe ser fácilmente modificable.
- **RNF-6 Registro de actividad:** la aplicación debe almacenar las interacciones llevadas a cabo con los usuarios.
- **RNF-7 Portabilidad:** la aplicación debe poder ejecutarse en distintas plataformas.
- **RNF-8 Compatibilidad de navegadores:** la aplicación debe ser compatible para los navegadores más importantes, incluyendo también los navegadores de *smartphones*.

## B.4. Especificación de requisitos

### Diagrama de casos de uso

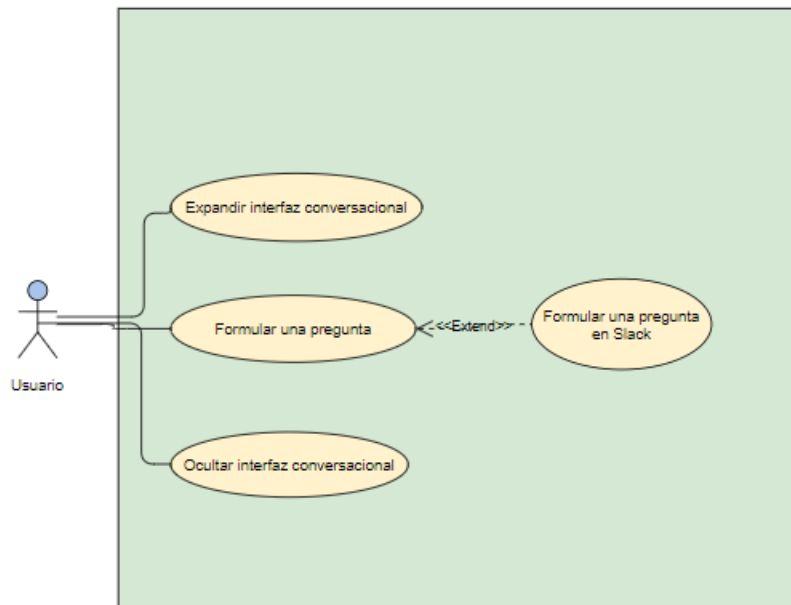


Figura B.1: Diagrama de casos de uso.

El actor -usuario- va a ser el estudiante, profesor o cualquier persona con acceso a la asignatura que comience una nueva sesión con el *chatbot*.

## Casos de uso

<b>CU-01</b>	<b>Expandir la interfaz conversacional.</b>
<b>Requisitos asociados</b>	RF-6
<b>Descripción</b>	Permite al usuario expandir la interfaz conversacional del <i>chatbot</i> .
<b>Precondición</b>	El usuario está logeado en UBUVirtual y dentro de la asignatura Trabajo Fin de Grado, en el apartado <i>Chatbot de preguntas frecuentes</i> . El <i>chatbot</i> esta minimizado.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página en que aparece el icono del <i>chatbot</i>.</li> <li>2. El usuario hace clic en el botón circular con el logo del proyecto.</li> <li>3. Se expande la interfaz conversacional.</li> </ol>
<b>Postcondición</b>	La interfaz conversacional queda expandida y el usuario puede interactuar con el <i>chatbot</i> .
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.1: CU-01 Expandir la interfaz conversacional.

<b>CU-02</b>	<b>Formular una pregunta.</b>
<b>Requisitos asociados</b>	RF-1, RF-2, RF-3, RF-4, RF-7
<b>Descripción</b>	El usuario introduce de manera textual una pregunta al <i>chatbot</i> .
<b>Precondición</b>	El usuario está logeado en UBUVirtual y dentro de la asignatura Trabajo Fin de Grado, en el apartado <i>Chatbot de preguntas frecuentes</i> . El <i>chatbot</i> está expandido.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario escribe y envía por medio de la interfaz conversacional una pregunta al <i>chatbot</i>.</li> <li>2. El programa analiza la pregunta introducida e intenta reconocerla.</li> <li>3. Se responde al usuario en modo texto y por medio de la interfaz conversacional a su pregunta.</li> <li>4. El <i>chatbot</i> queda a la espera de recibir nuevas preguntas.</li> </ol>
<b>Postcondición</b>	Se devuelve un mensaje con la respuesta a la pregunta.
<b>Excepciones</b>	Si el <i>chatbot</i> no es capaz de reconocer la pregunta introducida por el usuario o está mal formulada se informa al usuario por medio de un mensaje de que no ha sido posible entenderle.
<b>Importancia</b>	Alta

Tabla B.2: CU-02 Formular una pregunta.

CU-03	Formular una pregunta. (Slack)
<b>Requisitos asociados</b>	RF-1, RF-2, RF-3, RF-4, RF-7
<b>Descripción</b>	El usuario introduce de manera textual una pregunta al <i>chatbot</i> .
<b>Precondición</b>	El usuario está logeado en el espacio de trabajo de Slack y en la aplicación <i>UBU Asistente Virtual</i> .
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario escribe y envía por medio de la interfaz conversacional una pregunta al <i>chatbot</i>.</li> <li>2. El programa analiza la pregunta introducida e intenta reconocerla.</li> <li>3. Se responde al usuario en modo texto y por medio de la interfaz conversacional a su pregunta.</li> <li>4. El <i>chatbot</i> queda a la espera de recibir nuevas preguntas.</li> </ol>
<b>Postcondición</b>	Se devuelve un mensaje con la respuesta a la pregunta.
<b>Excepciones</b>	Si el <i>chatbot</i> no es capaz de reconocer la pregunta introducida por el usuario o está mal formulada se informa al usuario por medio de un mensaje de que no ha sido posible entenderle.
<b>Importancia</b>	Alta

Tabla B.3: CU-03 Formular una pregunta. (Slack)



<b>CU-04</b>	<b>Ocultar interfaz conversacional.</b>
<b>Requisitos asociados</b>	RF-5
<b>Descripción</b>	Permite al usuario ocultar la interfaz conversacional del <i>chatbot</i> .
<b>Precondición</b>	El usuario está logeado en UBUVirtual y dentro de la asignatura Trabajo Fin de Grado, en el apartado <i>Chatbot de preguntas frecuentes</i> . El <i>chatbot</i> esta expandido.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario hace clic en el botón circular con el logo del proyecto.</li> <li>2. La interfaz conversacional queda oculta.</li> </ol>
<b>Postcondición</b>	La interfaz conversacional queda oculta.
<b>Excepciones</b>	-
<b>Importancia</b>	Alta

Tabla B.4: CU-04 Ocultar interfaz conversacional.



## Apéndice C

---

# Especificación de diseño

---

### C.1. Introducción

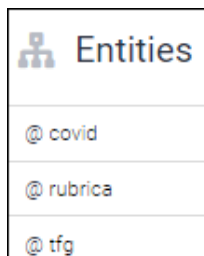
En este apartado se explica con detalle como se han estructurado los datos, los procedimientos y la arquitectura que se ha utilizado.

### C.2. Diseño de datos

Los datos se estructuran en torno a los *intents* y *entities*.

#### Entities

Hemos considerado las tres entidades mostradas en la figura C.1, al darse el caso de que había sinónimos de estos que se repetían en muchas ocasiones.




 Entities
@ covid
@ rubrica
@ tfg

Figura C.1: Entidades utilizadas en el agente.

## Intents

Aquí es donde almacenamos nuestras frases de entrenamiento y respuestas. Disponemos de un total de 205 archivos JSON para 103 intents diferentes en la modalidad online, y 195 archivos para 98 *intents* diferentes en la modalidad presencial. En las Figuras C.2 y C.3 se presenta el esquema utilizado para la versión online.

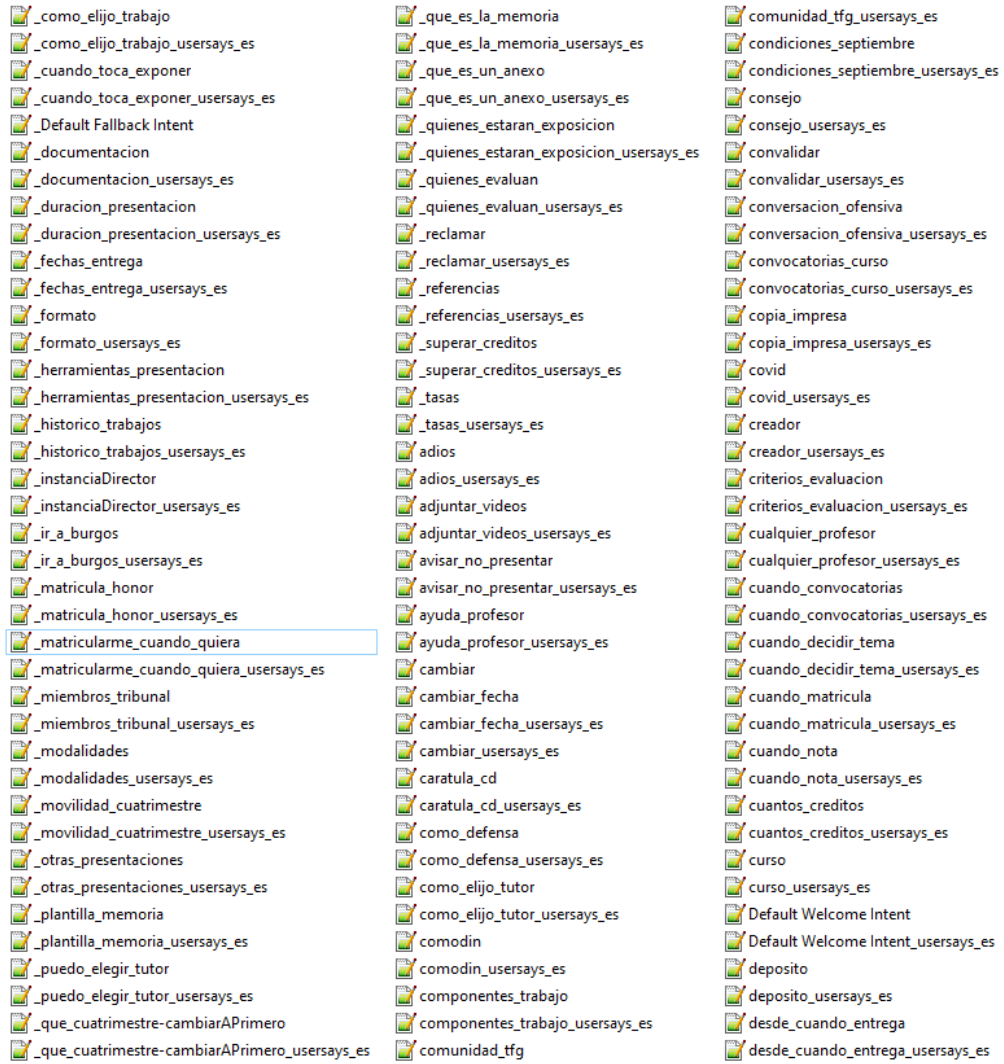


Figura C.2: Diseño de *intents* del agente para la modalidad online (Parte 1).

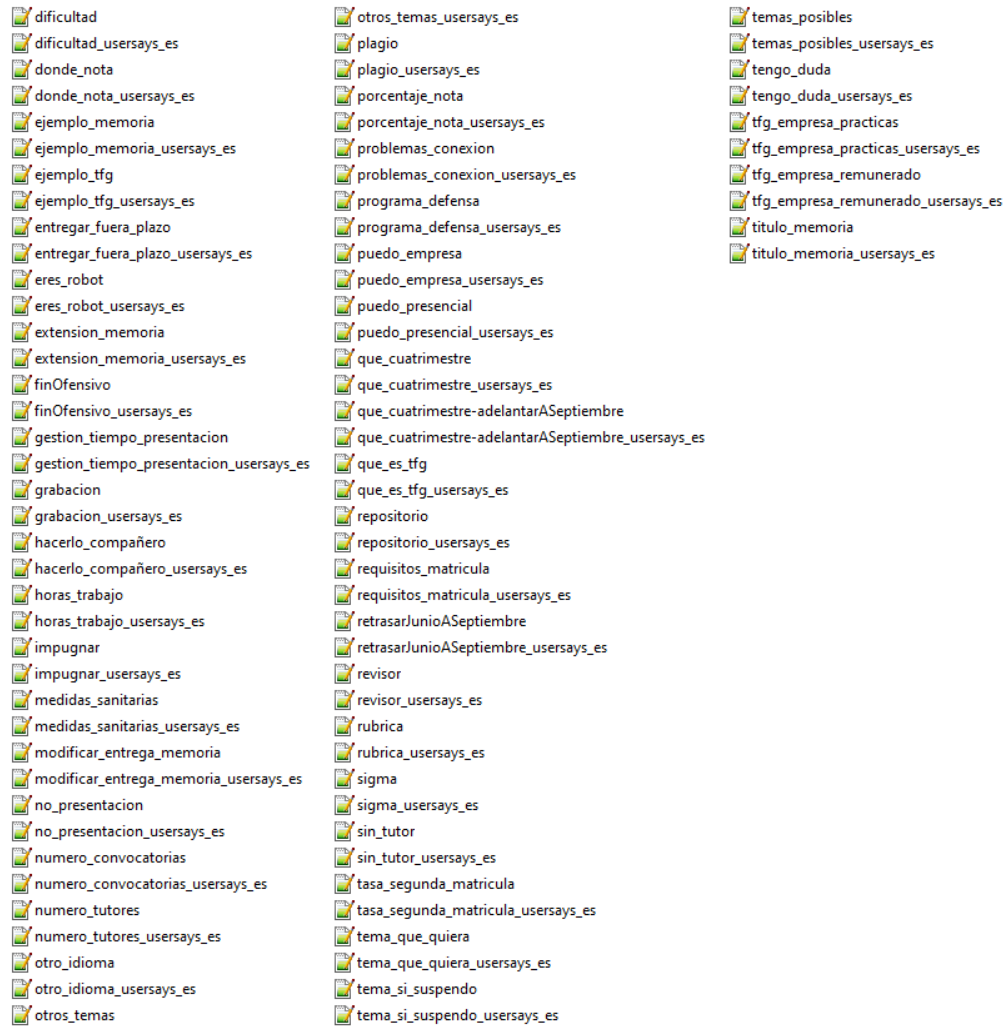


Figura C.3: Diseño de *intents* del agente para la modalidad online (Parte 2).

Se ha utilizado la nomenclatura de comenzar el nombre del *intent* con un símbolo `_` para determinar aquellos cuya respuesta es una *rich response*.

Para hacer legible y mantenible nuestra base de datos de preguntas y respuestas se dispone de un documento de texto en el que se recopila toda la información. De esta manera una persona ajena al proyecto podrá comprender la estructura rápidamente.

Intent	Ejemplo mensaje usuario	Respuesta
adios	Adiós	Espero haber sido de utilidad. Un saludo :)
adjuntar_videos	¿Se pueden adjuntar videos en el depósito del TFG??	No se pueden adjuntar videos en el depósito. Se subirá un documento con los enlaces a dichos videos que deberán estar colgados en YouTube.
ayuda_profesor	¿Me ayudará algún profesor?	El principal apoyo para ayudarte con tu TFG será tu tutor o tutora. En caso de que haya algún problema concreto que necesite la consulta a otros profesores él o ella te lo indicará.
cambiar	¿Puedo cambiar de tutor o tema?	El estudiante que quiera realizar un cambio de tutor y/o tema deberá solicitarlo al Tribunal por escrito, de manera motivada, en el plazo máximo de un mes desde la fecha de la publicación de la asignación. El Tribunal estará obligado a contestar al alumno por escrito en un plazo máximo de siete días hábiles. En dicho escrito se motivará la resolución del TFG/TFM a la petición del alumno.
Como_defensa	¿Cómo se hace la defensa del TFG?	La defensa se realiza oralmente, apoyándose en medios audiovisuales y demostraciones prácticas, durante un periodo de tiempo de 10 a 15 minutos. Posteriormente se realiza un turno de preguntas por parte del tribunal a responder por el alumnado.
como_elijo_trabajo	¿Cómo elijo un tema?	<ul style="list-style-type: none"> <li>• Antes de comenzar la convocatoria los tutores/estudiantes/empresas (consulta modalidades A, B, C de ofertas de TFG) recogen las propuestas de TFG</li> <li>• El tribunal publica la lista con la oferta de trabajos en las primeras semanas del semestre</li> <li>• Los alumnos sin TFG de ese semestre se ponen en contacto con los tutores de los TFG de su interés.</li> <li>• El tutor confirma a un alumno que le asigna un trabajo</li> <li>• El alumno notifica la asignación en el foro</li> <li>• El tribunal publica la asignación</li> <li>• Si pasado un mes del inicio del semestre un alumno no ha notificado la asignación, se le asignará uno de los trabajos disponibles</li> </ul>
como_elijo_tutor	¿Cómo elijo tutor?	En la modalidad C de oferta de TFG por el alumno una buena opción es identificar los conceptos teóricos principales que se necesitan en el TFG y buscar a los profesores de las guías docentes de las asignaturas del Grado que te los enseñaron.
comodin	*PREGUNTA AMBIGUA *	No he entendido, ¿puedes precisar más la pregunta?
componentes_trabajo	¿De que consta el trabajo?	Para la entrega se habilitan en UBUVirtual, en la convocatoria correspondientes las subidas de ficheros correspondientes (memoria, anexos, fuentes y otros enlaces de interés).
Consejo	Dame un consejo	Realiza una elección de TFG relacionado con asignaturas que te hayan gustado del Grado para estar motivado. Autofórmate y busca bibliografía específicas para resolver los problemas del TFG. Consulta TFG de cursos anteriores, tanto desarrollos en Github, como la memoria descriptiva. Planificate bien el tiempo de tu TFG. El tiempo para realizar un TFG depende de muchos factores y ahora tú gestionas tu propio proceso de aprendizaje.
Convalidar	¿Se puede convalidar el TFG?	No, el TFG no se puede reconocer.
conversacion_ofensiva	*CUALQUIER INSULTO*	Siento si no te he sido de utilidad. Dame otra oportunidad, estoy aquí para aprender y ayudarte.
convocatorias_curso	¿Cuántas convocatorias tengo por curso?	Cada curso el estudiante que se matricula del TFG tiene derecho a dos convocatorias de evaluación.
Covid	¿Hay algo a tener en cuenta a raíz del Covid?	Dado el carácter online de la asignatura no hay que tener en cuenta aspectos relacionados con el Covid. Si surgiera alguna cuestión adicional por causas sobrevenidas, se comunicarán en el tablón de la asignatura con antelación suficiente.
criterios_evaluacion	¿Cuáles son los criterios de evaluación?	La calificación se establece por consenso entre los miembros del tribunal. Todos los TFG tienen como tutor o cotutor a un miembro del tribunal. Además, el tribunal se basa una guía de evaluación detallada de tres dimensiones publicada en la plataforma virtual. La evaluación de un TFG puede ser vista desde tres perspectivas distintas: ¿qué documentación genera el estudiante? ¿Cómo ha generado esa documentación aplicando libremente conocimientos adquiridos en el Grado? ¿Cómo presenta su producto y resuelve dudas a una audiencia externa?
cuando_convocatorias	¿Cuándo son las convocatorias?	Por defecto la asignatura TFG está asignada al segundo cuatrimestre y las convocatorias de evaluación son en mayo y junio. Aunque también se puede adelantar a septiembre, enero o febrero.
cuando_decidir_tema	¿Cuándo tengo que	El tema se decide el primer mes de cada semestre: septiembre y febrero.

Figura C.4: Primera página del documento recopilatorio de preguntas y respuestas.

### C.3. Diseño procedimental

El proceso de interaccion es siempre el mismo: el usuario formula una pregunta desde el *chatbot* de UBUVirtual o Slack, esta pregunta se envía al agente de Dialogflow que está alojado en un servidor.

A continuación, este mensaje se envía a la API Natural Language de Google Cloud, la cuál mediante un algoritmo asigna un *intent* de salida.

En el agente de Dialogflow se consulta la respuesta del *intent* asignado y se devuelve al usuario por medio de la interfaz por la que esté operando.

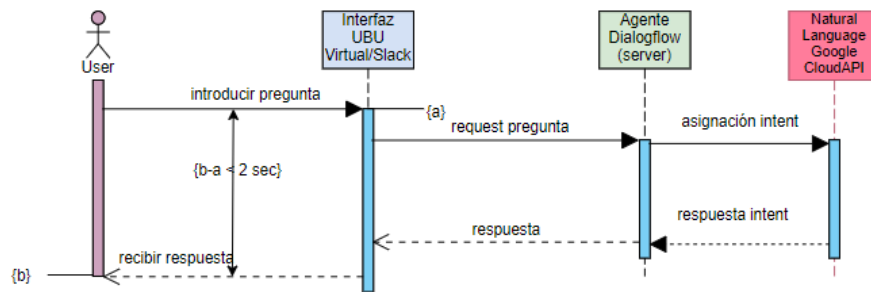


Figura C.5: Diagrama de secuencia.

## C.4. Diseño arquitectónico

El agente de Dialogflow está alojado en un servidor de Google Cloud. Dicho servidor reside en Estados Unidos, y Dialogflow nos lo proporciona de manera gratuita al utilizar una versión reducida.

El usuario accede por medio de su navegador al *chatbot* de la asignatura, el cuál está integrado en UBUVirtual. Es decir, en primer lugar accede al servidor en el que se aloja la plataforma Moodle de la Universidad.

La arquitectura que sigue la interacción es la del modelo cliente-servidor: el usuario introduce una pregunta en el *chatbot* (cliente) la cuál envía una petición al servidor (agente de Dialogflow) que la procesa y devuelve una respuesta al cliente.

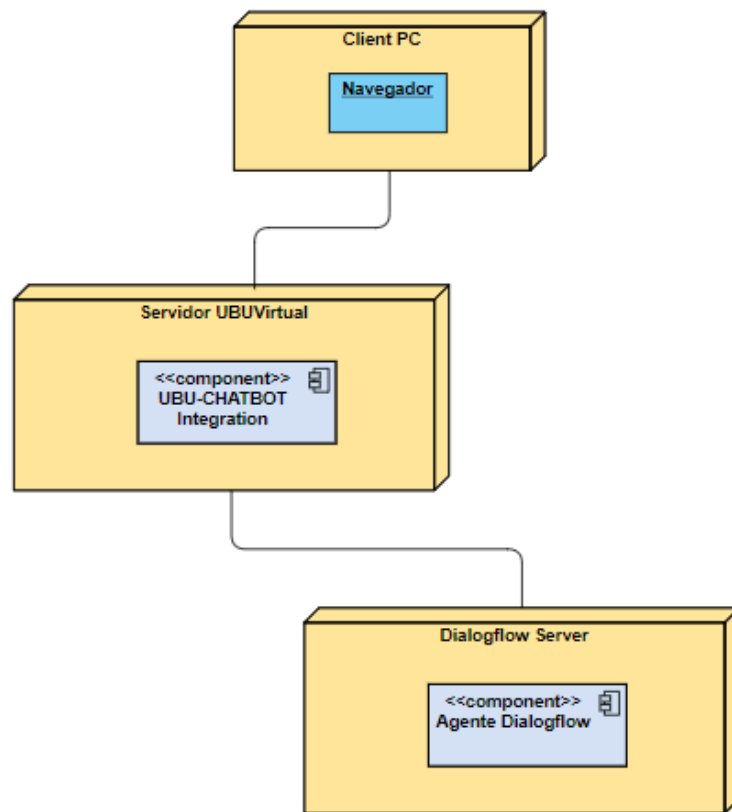


Figura C.6: Diagrama de despliegue.



## Apéndice *D*

---

# Documentación técnica de programación

---

### D.1. Introducción

En este apartado se va a detallar la estructura de directorios del proyecto y los conceptos necesarios para la programación del sistema así como para su instalación y ejecución. Finalmente se va a detallar el proceso de pruebas utilizado.

### D.2. Estructura de directorios

La estructura de directorios del proyecto es la siguiente:

- `/`: logo del proyecto, fichero de la licencia y un documento en el que se recopilan las estadísticas de los registros de las conversaciones.
- `/chatbotOnline/`: configuración del agente y documento informativo de las preguntas implementadas para la modalidad online.
- `/chatbotOnline/entities/`: entidades del agente para la modalidad online.
- `/chatbotOnline/intents/`: *intents* del agente para la modalidad online.
- `/chatbotPresencial/`: configuración del agente y documento informativo de las preguntas implementadas para la modalidad presencial.
- `/chatbotPresencial/entities/`: entidades del agente para la modalidad presencial.

- `/chatbotPresencial/intents/`: *intents* del agente para la modalidad presencial.
- `/docs/`: documentación del proyecto.
- `/docs/img/`: imágenes utilizadas en la documentación.
- `/docs/ltx/`: documentación en formato  $\text{\LaTeX}$ .
- `/scripts/`: ficheros `.html` necesarios para la integración del *chatbot* en UBUVirtual.

### D.3. Manual del programador

Las herramientas utilizadas en el proyecto han sido las siguientes:

#### Herramientas utilizadas

- Dialogflow
- Notepad++
- Visual Studio Code

#### Dialogflow

El primer paso para la creación del proyecto es crear una cuenta de Google con la que acceder a la herramienta web Dialogflow. Se muestra en la Figura D.1.

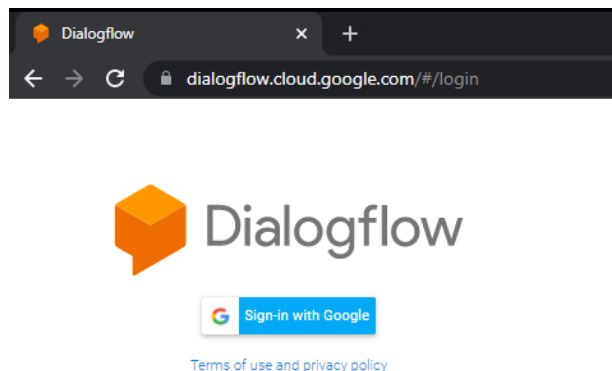


Figura D.1: Creación de cuenta para acceder a la herramienta online.

Posteriormente se hace clic en la rueda dentada que permite crear un nuevo agente, como se muestra en la Figura D.2

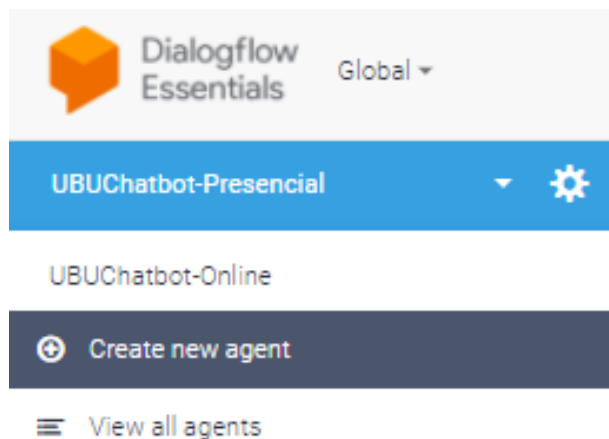


Figura D.2: Creación de agente en Dialogflow.

Deberemos introducir la información básica de nuestro proyecto como se nos pide en la Figura D.3

The image shows a form titled 'EjemploCreación' with a 'CREATE' button. The form has four sections: 1. 'DEFAULT LANGUAGE' with a dropdown set to 'Spanish - es' and a note 'Primary language for your agent. Other languages can be added later.' 2. 'DEFAULT TIME ZONE' with a dropdown set to '(GMT+1:00) Europe/Madrid' and a note 'Date and time requests are resolved using this timezone if not provided in the API requests.' 3. 'GOOGLE PROJECT' with a dropdown set to 'Create a new Google project' and a note 'Enables Cloud functions, Actions on Google and permissions management.' 4. 'AGENT TYPE' with a toggle switch 'Set as Mega Agent' and a note 'Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. mega agent).'

Figura D.3: Configuración básica del agente.

El agente habrá sido creado y ya se puede comenzar a implementar el *chatbot*.

## Intents

El pilar básico del proyecto son los *intents*: nuestra base de datos de preguntas y respuestas. Al listado de *intents* ya creados y la opción de crear nuevos se accede desde el panel de navegación de la parte izquierda de la pantalla, como se muestra en la Figura D.4.

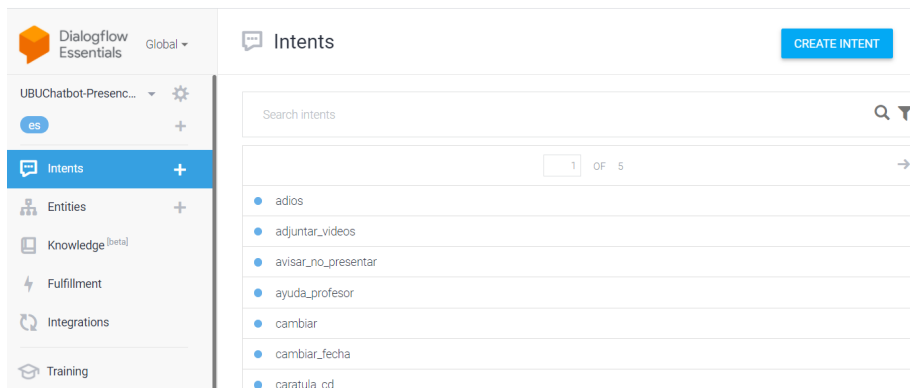


Figura D.4: Panel de *intents*

Al pulsar sobre la opción *Create Intent* se despliega el panel de creación mostrado en la figura D.5.

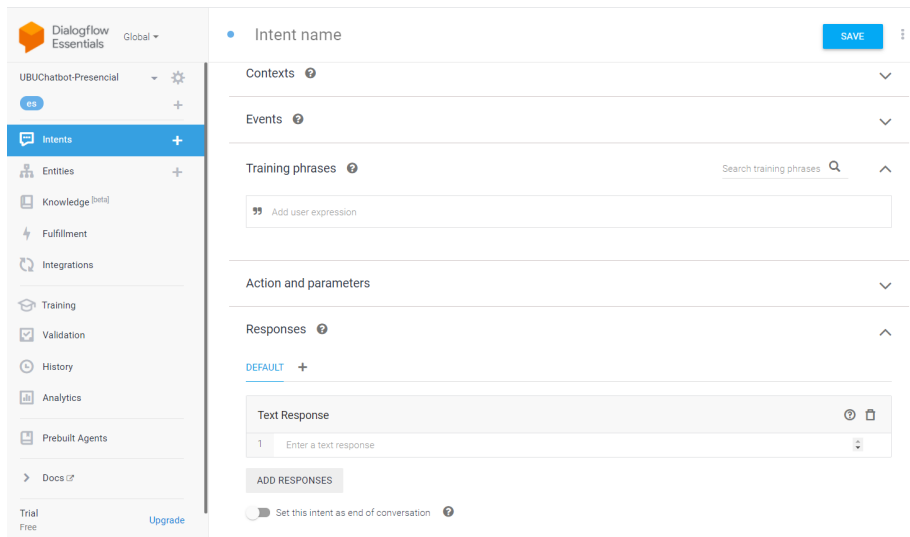


Figura D.5: Creación de un *intent*

En este panel debemos introducir los siguientes elementos:

- **Contexto:** contexto de entrada y de salida. El de entrada establece qué contextos deben estar activos para poder acceder a este *intent* y el de salida qué contextos se activan una vez accedido a este *intent* y por cuántos turnos de conversación.
- **Eventos:** eventos que disparan el *intent*. Siempre tendremos uno para el evento *WELCOME* que se corresponde con el mensaje inicial que lanzamos.
- **Frases de entrenamiento:** listado de frases que cuando se detecten provocarán que el agente acceda a este *intent*. Es recomendable poner el mayor número de frases posibles para mejorar la tasa de éxito del algoritmo de asignación.
- **Acciones y parámetros:** algunas palabras de las frases de entrenamiento pueden tratarse de Entidades. En este apartado estableceremos dichas relaciones y el tipo de parámetro que es cada uno.
- **Respuestas:** las respuestas que devolverá nuestro *chatbot* para cada integración cuando se acceda a este *intent*. Es aquí donde se pueden configurar las *Rich Responses*, lo cual se explica en el apartado [D.3](#).
- **Fulfillment:** al marcar esta opción activamos los *fulfillment* -llamadas a código externo- para este *intent*.

### Entities

Las Entidades son un concepto importante dentro de un proyecto de Dialogflow. Una entidad es un tipo de dato.

Vamos a verlo con un ejemplo: tenemos una entidad llamada *tfg* la cual incluye todos los sinónimos con los que el usuario puede nombrar al Trabajo de Fin de Grado. Esto se va a utilizar en la implementación de los *intents*. Cuando escribamos en una frase de entrenamiento las palabras *trabajo fin de grado* -o cualquier sinónimo- estableceremos que es un parámetro de tipo *tfg*, ya que es como hemos definido esta entidad. Esto producirá que detecte cualquier frase que tenga alguno de los valores introducidos en dicha entidad.

De esta manera evitamos tener que escribir una frase de entrenamiento para cada sinónimo de una palabra.

Para crear una Entidad o ver el listado de las que tenemos accedemos a la sección *Entities* del menú de navegación lateral. En la Figura D.6 se muestra el proceso de creación de una Entidad.

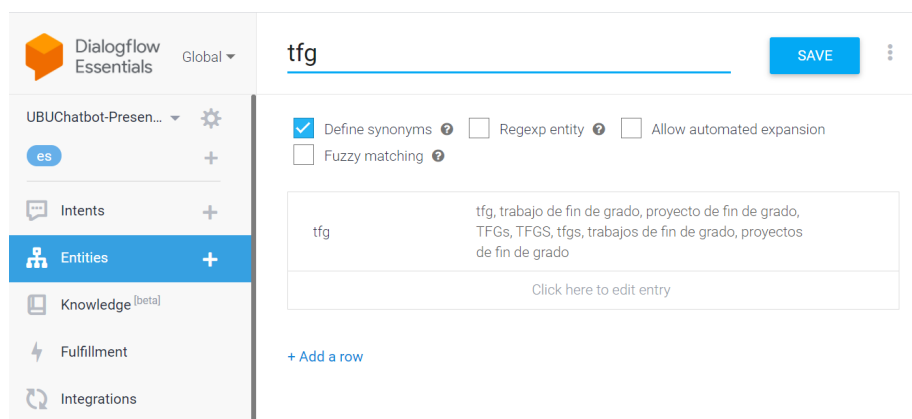


Figura D.6: Implementación de una Entidad.

Dentro de una entidad se pueden introducir distintos valores, cada uno de ellos con sus sinónimos. Por ejemplo, podríamos tener una entidad *Vehículo* en la que cada fila se correspondiera a un tipo de vehículo. Posteriormente, cuando se disparase un *intent* que contuviese algún valor de esta Entidad se recogería la información tanto de que es una entidad Vehículo como que su valor es coche/moto/camión. Esto es importante ya que podría sernos de utilidad saber el tipo de vehículo si por ejemplo la respuesta que tuviésemos que dar fuese distinta en función del tipo de vehículo.

### ***Rich Responses***

Para implementar las *Rich Responses* -que permiten funciones como el formateo de texto, los hipervínculos o las *chip responses*- debemos generar un objeto JSON en el que definir las variables, comportamientos y textos. Consultamos la documentación de Dialogflow Messenger [4] para consultar las distintas opciones e implementaciones.

A la hora de implementarlas se ha utilizado el editor de textos Notepad++ ya que soporta el formato JSON.

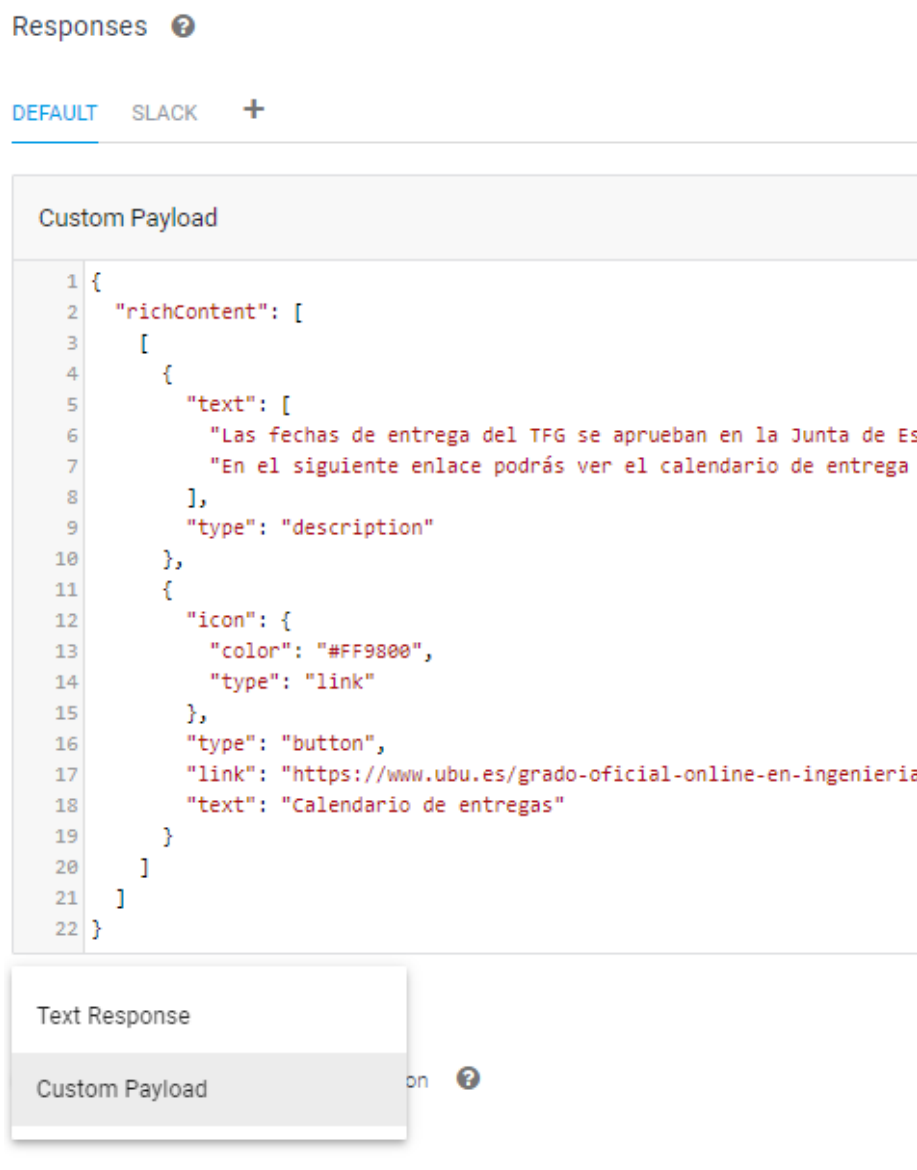


```
1 {
2   "richContent": [
3     [
4       {
5         "text": [
6           "Las fechas de entrega del TFG se aprueban en la Junta de Escu",
7           "En el siguiente enlace podrás ver el calendario de entrega de",
8         ],
9         "type": "description"
10      },
11      {
12        "icon": {
13          "color": "#FF9800",
14          "type": "link"
15        },
16        "type": "button",
17        "link": "https://www.ubu.es/grado-oficial-online-en-ingenieria-",
18        "text": "Calendario de entregas"
19      }
20    ]
21  ]
22 }
23 }
```

Figura D.7: Edición de JSON para una *Rich Response* con Notepad++.

En la Figura D.7 se muestra el editor con el JSON para un ejemplo de respuesta que responde a las fechas de entrega. En este caso se está respondiendo un mensaje de texto junto a un hipervínculo, el cual tiene una descripción e imagen asociada entre otras características.

Una vez hemos implementado el JSON con la respuesta, debemos ir al *intent* en el que la queremos implantar. Vamos al apartado de respuestas y hacemos clic en *add responses*. Previamente podemos eliminar la respuesta por defecto que teníamos si no queremos que las concatene. Seleccionamos *Custom Payload* como se muestra en la Figura D.8.

Figura D.8: Añadiendo *Rich Response* a un *intent*.

Una vez hemos seleccionado esta opción pegamos nuestro código de la misma forma que se ha hecho en la imagen anterior. Guardamos y la respuesta ya estará implementada.



## Fulfillments

Los *fulfillment* son una respuesta en código que pueden disparar algunos *intents*. En este proyecto no ha sido necesario utilizarlos.

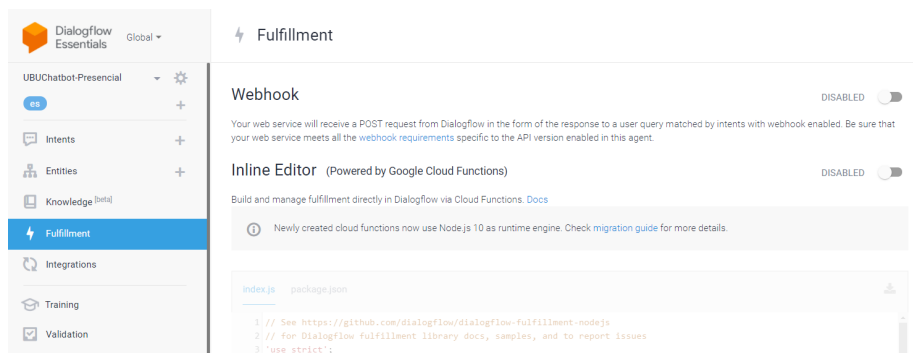


Figura D.9: Panel de *fulfillments* de Dialogflow.

Existen dos posibilidades:

- *Inline Editor*: es una versión simplificada que permite introducir el código directamente en Dialogflow. Exige introducir una cuenta de facturación.
- *Webhook*: url de la plataforma en la que esté alojado nuestro código. Se realizará una petición POST cuando se dispare un *intent* con *fulfillment*. Podríamos habilitar las Cloud Functions del proyecto de Google asociado al agente, pero también requeriría cuenta de facturación. Otra posible opción es utilizar una plataforma en la nube como Heroku para alojar nuestro código. Esta solución también tiene restricciones en su versión gratuita, ya que si no introducimos una tarjeta de crédito solo vamos a disponer de 550 horas al mes. [9]

## Integrations

Para realizar las integraciones accederemos a la herramienta *Integrations* en el menú lateral de Dialogflow. Se muestra en la Figura D.10.

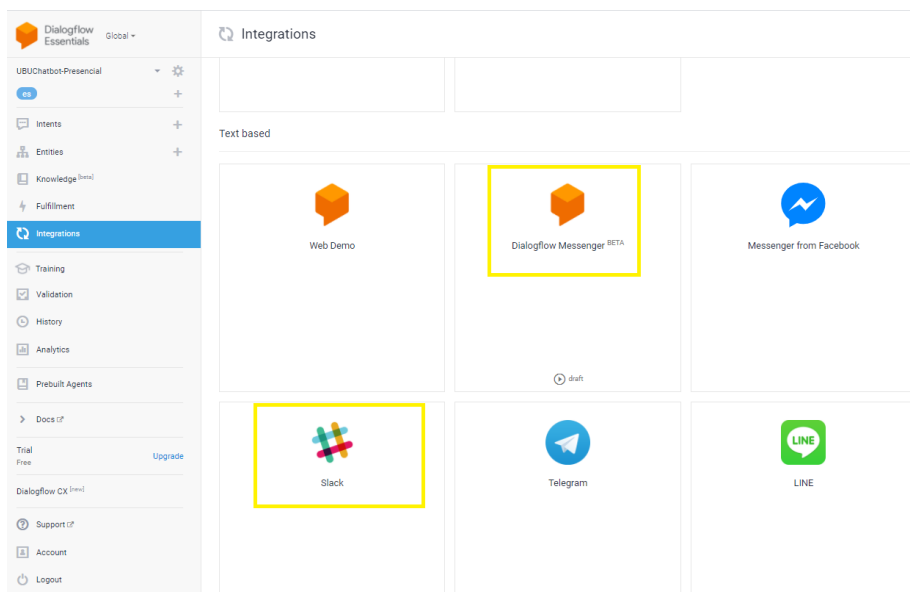


Figura D.10: Menú de integraciones de Dialogflow.

Para la integración con UBUVirtual hacemos clic en Dialogflow Messenger, ya que es la única integración web que permite las *rich responses*. Se nos desplegará una ventana modal como la mostrada en la Figura D.11.

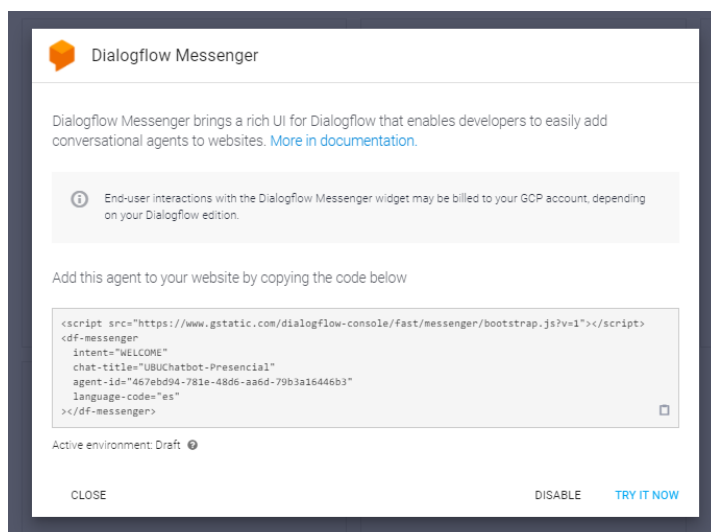


Figura D.11: Ventana modal para la integración web.

Copiamos el código que nos suministra. Deberemos añadir este código HTML en UBUVirtual. En el apartado D.3 se explica cómo modificarlo con un editor como Visual Studio Code para personalizar la apariencia del *chatbot*.

Para añadir el HTML a UBUVirtual necesitaremos tener permisos de administrador. Accedemos al curso en UBUVirtual y creamos un recurso de tipo página. Ver Figura D.12.

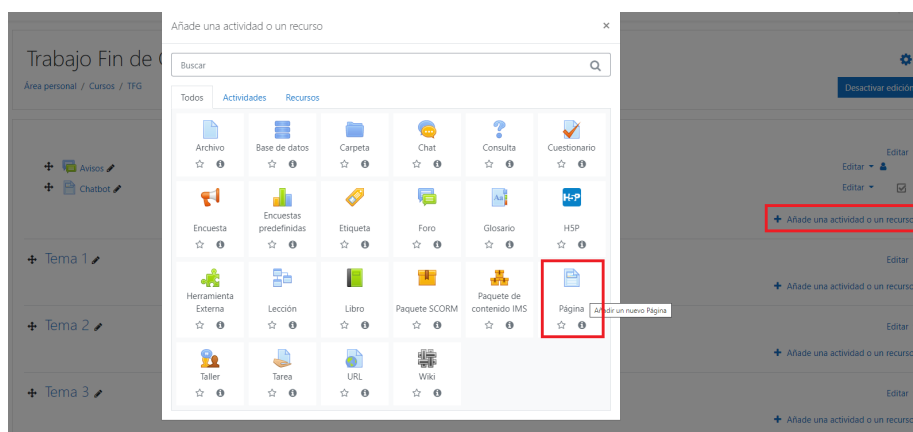


Figura D.12: Creación de página en el curso de UBUVirtual.

Introducimos nuestro código y guardamos los cambios. Ver Figura D.13. Con esto el *chatbot* habrá quedado integrado en UBUVirtual.

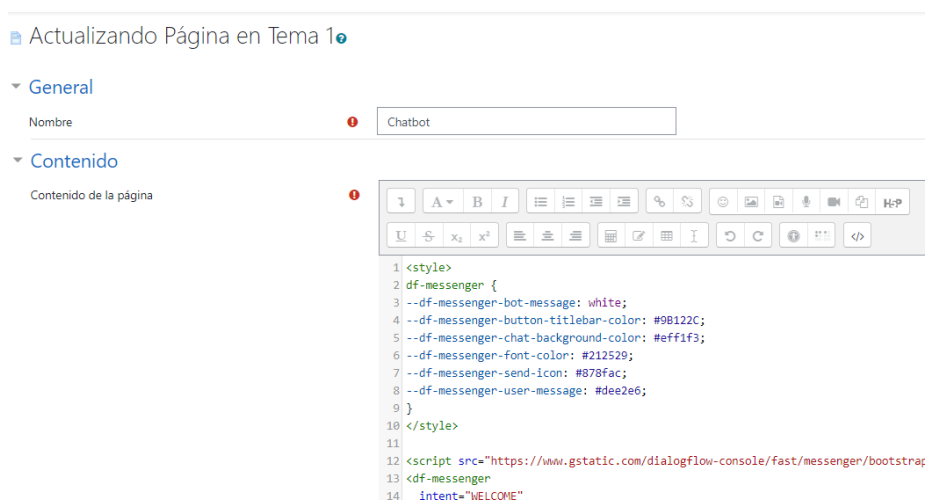


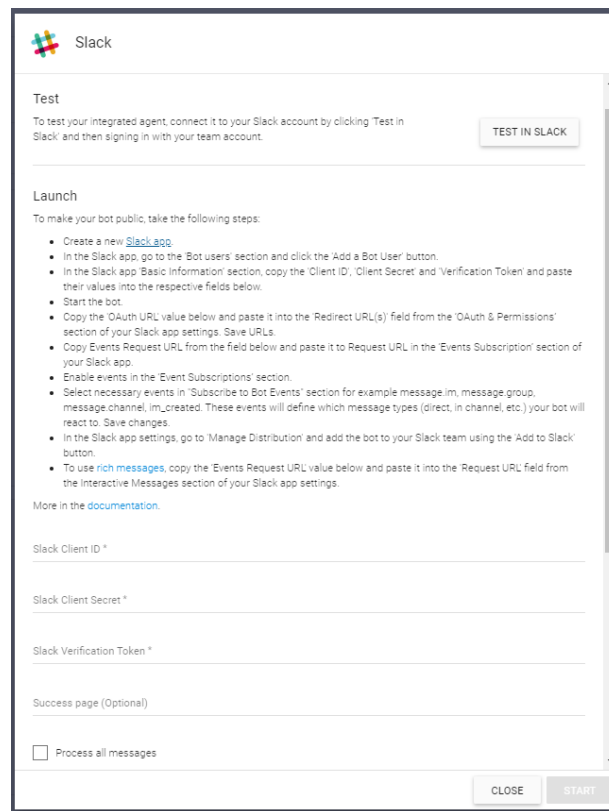
Figura D.13: Insertar el código HTML.

Para la integración con Slack en primer lugar necesitamos crear un espacio de trabajo. Para ello accedemos a la web de Slack [17] mostrada en la Figura D.14. Introducimos un email y seguimos los pasos introduciendo nuestros datos.



Figura D.14: Web para la creación del espacio de trabajo.

Una vez creado el espacio de trabajo volvemos a Dialogflow y dentro de *Integrations* hacemos clic en Slack para que se despliegue el modal mostrado en la Figura D.15.



The image shows a modal window titled "Slack" with a Slack logo. It contains two main sections: "Test" and "Launch".

**Test**

To test your integrated agent, connect it to your Slack account by clicking 'Test in Slack' and then signing in with your team account.

**Launch**

To make your bot public, take the following steps:

- Create a new [Slack app](#).
- In the Slack app, go to the 'Bot users' section and click the 'Add a Bot User' button.
- In the Slack app 'Basic Information' section, copy the 'Client ID', 'Client Secret' and 'Verification Token' and paste their values into the respective fields below.
- Start the bot.
- Copy the 'OAuth URL' value below and paste it into the 'Redirect URL(s)' field from the 'OAuth & Permissions' section of your Slack app settings. Save URLs.
- Copy Events Request URL from the field below and paste it to Request URL in the 'Events Subscription' section of your Slack app.
- Enable events in the 'Event Subscriptions' section.
- Select necessary events in 'Subscribe to Bot Events' section for example message.im, message.group, message.channel, im\_created. These events will define which message types (direct, in channel, etc.) your bot will react to. Save changes.
- In the Slack app settings, go to 'Manage Distribution' and add the bot to your Slack team using the 'Add to Slack' button.
- To use [rich messages](#), copy the 'Events Request URL' value below and paste it into the 'Request URL' field from the Interactive Messages section of your Slack app settings.

More in the [documentation](#).

Slack Client ID \*

Slack Client Secret \*

Slack Verification Token \*

Success page (Optional)

☐ Process all messages

CLOSE START

Figura D.15: Modal de la integración de Slack.

Pulsamos sobre el botón *TEST IN SLACK* y seleccionamos el espacio de trabajo que acabamos de crear, tal y como se muestra en la Figura D.16.



Figura D.16: Selección de espacio de trabajo para la integración.

Confirmamos y comprobamos como al acceder ahora a Slack nos aparece el *chatbot* ya integrado como una aplicación. Ver Figura D.17.



Figura D.17: Chatbot integrado correctamente en Slack.

## Visual Studio Code

Con este editor de código se ha programado el código HTML5 de la integración de UBUVirtual y también se utilizó para el control de versiones, ya que nos facilita mucho su uso.

```
<> integracion_moodle_presencial.html > script > <function>
1  <style>
2  df-messenger {
3  --df-messenger-bot-message: white;
4  --df-messenger-button-titlebar-color: #9B122C;
5  --df-messenger-chat-background-color: #eff1f3;
6  --df-messenger-font-color: #212529;
7  --df-messenger-send-icon: #878fac;
8  --df-messenger-user-message: #dee2e6;
9  }
10 </style>
```

Figura D.18: Personalización del CSS de la interfaz conversacional.

En la Figura D.18 se muestra el código CSS en el que se personalizan los colores del *chatbot*. El código hexadecimal de los colores se ha obtenido desde

el inspector de elementos del navegador Chrome, extrayendo los valores utilizados en UBUVirtual para mantener la estética de la página.

También se incluye en el fichero HTML que importamos en UBUVirtual la personalización de los valores por defecto del chatbot mediante el *Custom Element* que nos proporciona Dialogflow.

Los *Custom Elements* son una característica del estandar *Web Components* que permite crear elementos personalizados que encapsulan la funcionalidad en una página HTML. [6]

```

63 <df-messenger
64   intent="WELCOME"
65   chat-title="UBU-CHATBOT"
66   agent-id="467ebd94-781e-48d6-aa6d-79b3a16446b3"
67   language-code="es"
68   chat-icon="https://raw.githubusercontent.com/aav0038
69 ></df-messenger>

```

Figura D.19: Custom Element para el *chatbot* de Dialogflow.

En la Figura D.19 vemos la configuración básica de nuestro agente: *intent* inicial, nombre, id del agente, idioma e imagen del icono del chat.

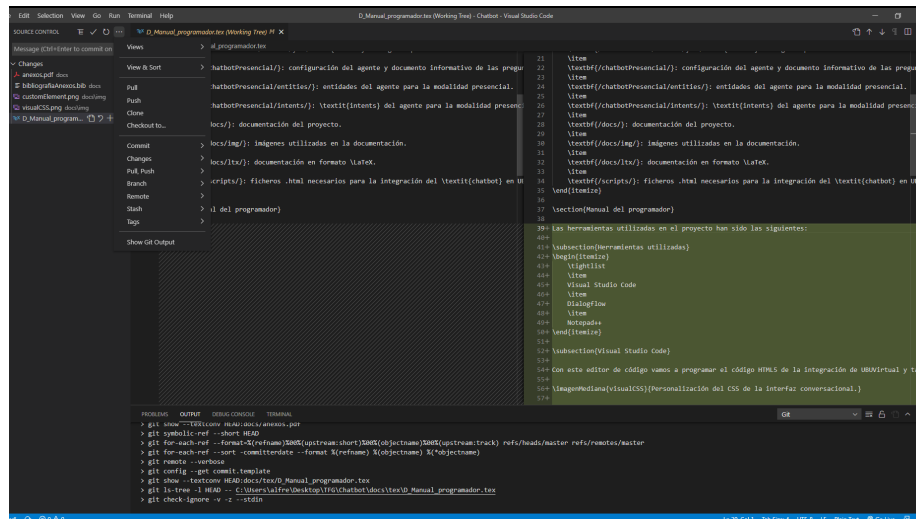


Figura D.20: Uso del control de versiones desde la interfaz del IDE.

Finalmente en la Figura D.20 se muestra la herramienta *Source Control* integrada en el IDE que se ha utilizado para trabajar con el control de versiones de GitHub del proyecto.



## D.4. Compilación, instalación y ejecución del proyecto

El proyecto está alojado en los servidores de Dialogflow e integrado en UBUVirtual por lo que no es necesaria su compilación ni instalación.

Para ejecutarlo es tan sencillo como acceder al recurso disponible en UBUVirtual donde automáticamente se desplegará y podrá comenzar a utilizarse.

## D.5. Pruebas del sistema

Al tratarse de un proyecto ciertamente particular no se han realizado las pruebas de código habituales de sistema tradicionales.

### Pruebas de compatibilidad

Para las pruebas de compatibilidad se accedió a las plataformas UBU-Virtual y Slack por medio de distintos navegadores. Con estas pruebas también se ratifica que la integración se ha realizado correctamente. A Safari se accedió desde un *smartphone* con IOs. En la tabla D.5 se marca como superado aquellas plataformas en las que el *chatbot* funcionó sin ningún problema.

	UBUVirtual	Slack
Chrome	✓	✓
Firefox	✓	✓
MS Edge	✓	✓
Opera	✓	✓
Safari	✓	✓
IEExplorer	✗	✗

Tabla D.5: Compatibilidad con distintos navegadores.

Internet Explorer resultó ser el único navegador incompatible con el *chatbot*.

## Pruebas de funcionalidad

Para comprobar la funcionalidad del proyecto se determinan varias maneras:

- Introducción manual de preguntas en el *chatbot* integrado en UBUVirtual o Slack o directamente desde la consola de pruebas de Dialogflow. Se comprueba que la respuesta ofrecida por el programa responde a la pregunta.
- Análisis manual de *logs* en el apartado *History* de Dialogflow. De forma similar a la descrita en el apartado anterior con la ayuda de que se marcaran con un símbolo de *warning* aquellas conversaciones que hayan tenido algún *fallback*.
- Validación de la implementación generada al realizar el entrenamiento del agente en la herramienta *Validation*.

En la figura D.21 se muestra la validación del sistema, en la que se observan distintas advertencias -recomendaciones todas ellas- y ningún error.

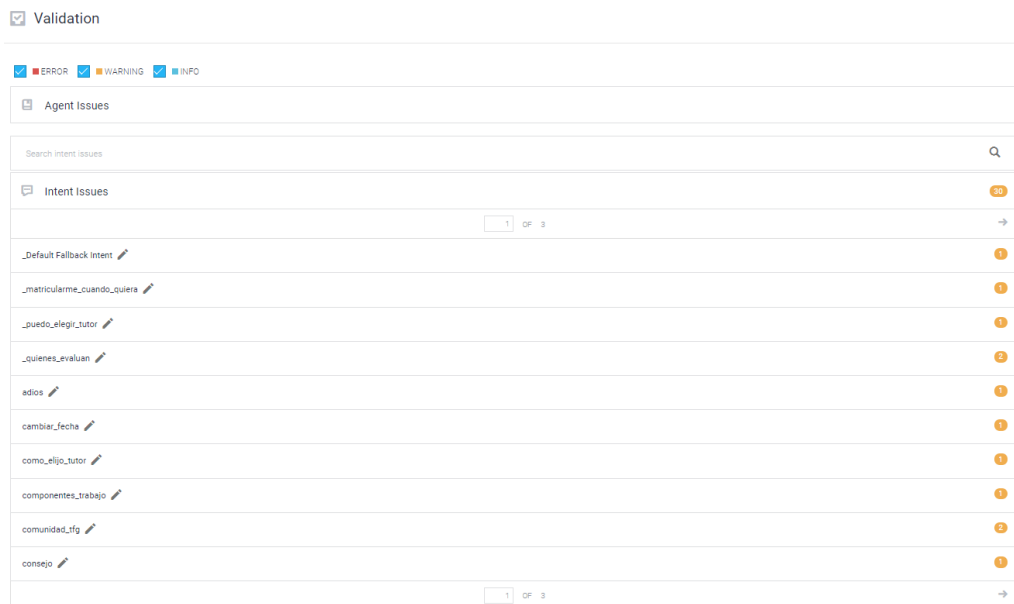


Figura D.21: Validación del agente.

Finalmente para comprobar que nuestro sistema es preciso se ha realizado un estudio numérico del análisis de todas las conversaciones llevadas a cabo por los usuarios desde que se integrase en la plataforma UBUVirtual.

Dado que al tratarse de lenguaje natural no es posible hacer test automáticos para comprobar que las salidas del programa responden correctamente a la entrada introducida por el usuario, se ha tenido que realizar manualmente. Obteniendo para el mes de junio un 55,36 % de respuestas correctas sobre un total 56 preguntas formuladas.

Una tasa de éxito que cumple el objetivo del proyecto; que no es el de responder como haría un humano, sino el de reducir la carga de trabajo de los profesores y ayudar a los alumnos de manera considerable.



## *Apéndice E*

---

# Documentación de usuario

---

## E.1. Introducción

En este apartado se explican los requerimientos de la aplicación para ser ejecutada tanto en la plataforma Moodle como en Slack. Se indica también el proceso de instalación en ambas plataformas. Finalmente, en el manual de usuario se dan las indicaciones para utilizar correctamente la aplicación.

## E.2. Requisitos de usuarios

En esta sección se indican los requisitos para utilizar el chatbot en cada plataforma para la versión de la modalidad online.

### Plataforma UBU Virtual

- Conexión a Internet.
- Navegador web en cualquier dispositivo: PC, Smartphone, Tablet, etc. Compatible con Chrome, Safari, Firefox, Edge y Opera entre otros. Incompatible con Internet Explorer.
- Cuenta activa en la plataforma UBUVirtual.
- Permiso de acceso al curso ‘TRABAJO FIN DE GRADO (Grupo 90) On-line’ (id 11707).

## **Slack**

- Conexión a Internet.
- Navegador web en cualquier dispositivo: PC, Smartphone, Tablet, etc. Compatible con Chrome, Safari, Firefox, Edge y Opera. [18]
- Cuenta de Slack y permiso de acceso al espacio de trabajo ubuchat-bot.slack.com.

## E.3. Instalación

### Plataforma UBUVirtual

No es necesario ningún proceso de instalación. Únicamente necesitaremos acceder a la asignatura en la plataforma UBUVirtual con un navegador compatible y tener JavaScript activado.

El enlace de acceso al chatbot es el siguiente: <https://ubuvirtual.ubu.es/mod/page/view.php?id=3132682>

### Slack

Se puede acceder a esta plataforma mediante la versión web o la aplicación de escritorio.

En caso de que quisiéramos obtener la versión de escritorio, deberíamos acceder a <https://slack.com/intl/es-es/downloads> y seleccionar la versión compatible con nuestro sistema operativo, pero este paso no es necesario.

Como paso previo necesitaremos una invitación al espacio de trabajo que nos deberá haber enviado alguno de los miembros. En caso de que no se te haya facilitado esta invitación, escribe un email al administrador a su correo [aav0038@alu.ubu.es](mailto:aav0038@alu.ubu.es) indicando tu correo de Google y te la hará llegar.

Te llegará un email a la bandeja de entrada con el título ‘Alfredo te ha invitado a trabajar en Slack’. En caso de no recibirlo se debe comprobar la bandeja de correo no deseado.

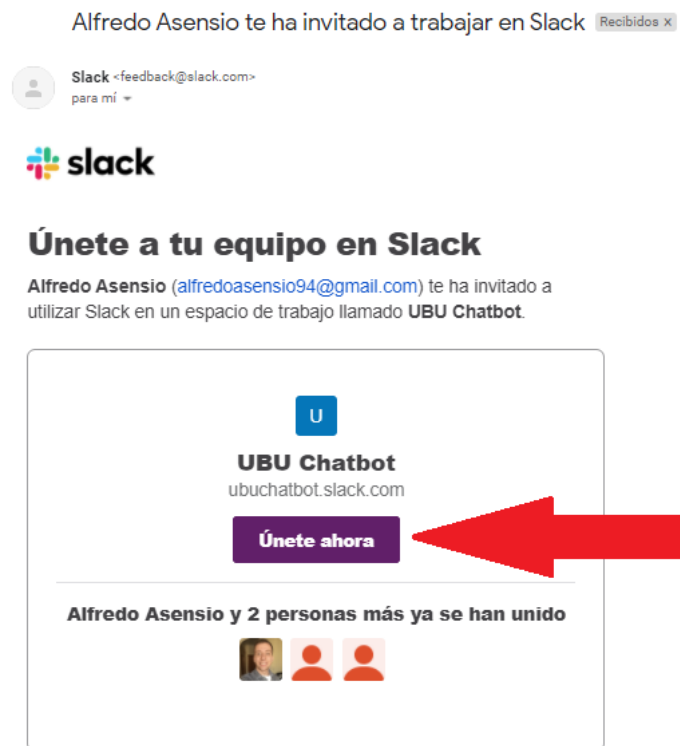


Figura E.1: Email de confirmación una vez recibido acceso

Para acceder a la aplicación, lo haremos por medio del botón morado de ‘Únete ahora’ disponible en el email de confirmación como se muestra en la Figura E.1.

Nos llevará a una pantalla de login en la que al haber introducido un correo de Google, haremos clic en ‘Continuar con Google’ como se muestra en la Figura E.2.





## Unirse a UBU Chatbot en Slack



Alfredo Asensio y 1 otro se han unido



Nombre completo

Contraseña



¿Podemos escribirte con noticias, ofertas y consejos para usar Slack?

☐ ¡Por supuesto!

☐ Prefiero que no.

Si continúas, aceptas nuestras [Condiciones de servicio al usuario](#), la [Política de privacidad](#) y la [Política de cookies](#).

Figura E.2: Pantalla para unirse al espacio de trabajo de Slack

Seleccionaremos nuestra cuenta de Google y haremos clic en el botón morado ‘Crear cuenta’ como se muestra en la Figura E.3.

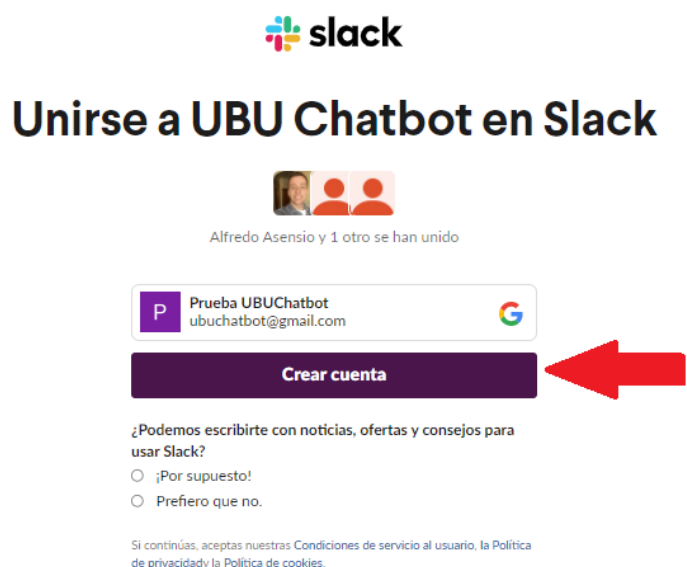



Figura E.3: Confirmación para crear la cuenta y acceder al espacio de trabajo.

Tras unos segundos de carga habremos accedido al espacio de trabajo ‘UBU Chatbot’.


Alternativamente y en posteriores conexiones, se puede acceder a través del siguiente enlace que nos llevará al espacio de trabajo en el que está integrado el chatbot:


<https://ubuchatbot.slack.com>



## Conéctate a UBU Chatbot

ubuchatbot.slack.com

 Conectarse con Google

 Conectarse con Apple

---

Dirección de correo

Contraseña

Conectarse

☒ Recordarme en este dispositivo

[¿Olvidaste tu contraseña? Te ayudamos a conectarte](#)

[¿Buscas otro espacio de trabajo? Encuentra tus espacios de trabajo](#)

Figura E.4: Espacio de trabajo de Slack

Se nos pedirá credenciales para acceder de la manera que se muestra en la Figura E.4 y lo haremos de la forma anteriormente explicada.

## E.4. Manual del usuario

### Plataforma UBU Virtual

#### Desde PC

Para la versión de ordenador, una vez hemos accedido a la página de UBUVirtual donde está disponible el chatbot, veremos como en la parte inferior derecha aparece un icono redondo con el logo del proyecto: un oso con un bocado de texto sobre un fondo granate. A los segundos aparece un mensaje de bienvenida junto a este, tal y como se muestra en la Figura E.5.

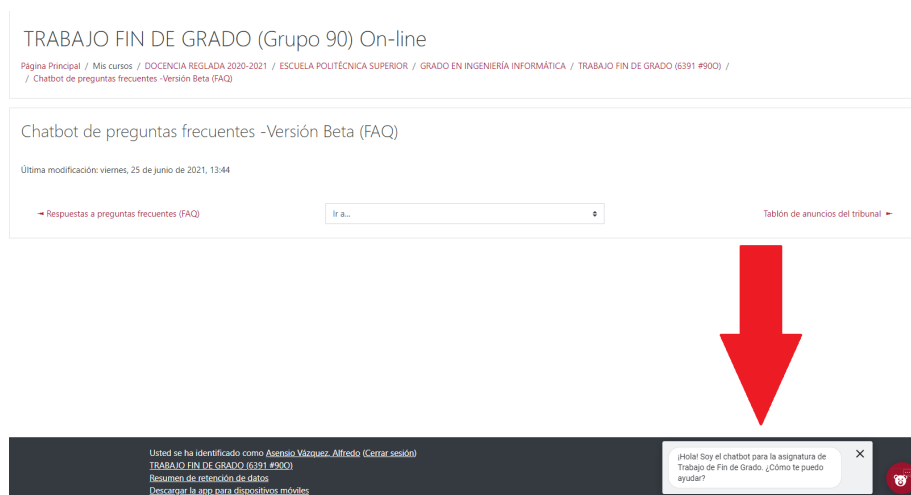


Figura E.5: Visualización del chatbot inicialmente minimizado.

Para abrir el chatbot haremos clic en el círculo del logo o en el mensaje de bienvenida. Se nos desplegará el chat, en el que aparecerá el mensaje de bienvenida y se habilitará una entrada de texto en la que deberemos hacer clic para empezar a introducir nuestros mensajes. Ver Figura E.6.

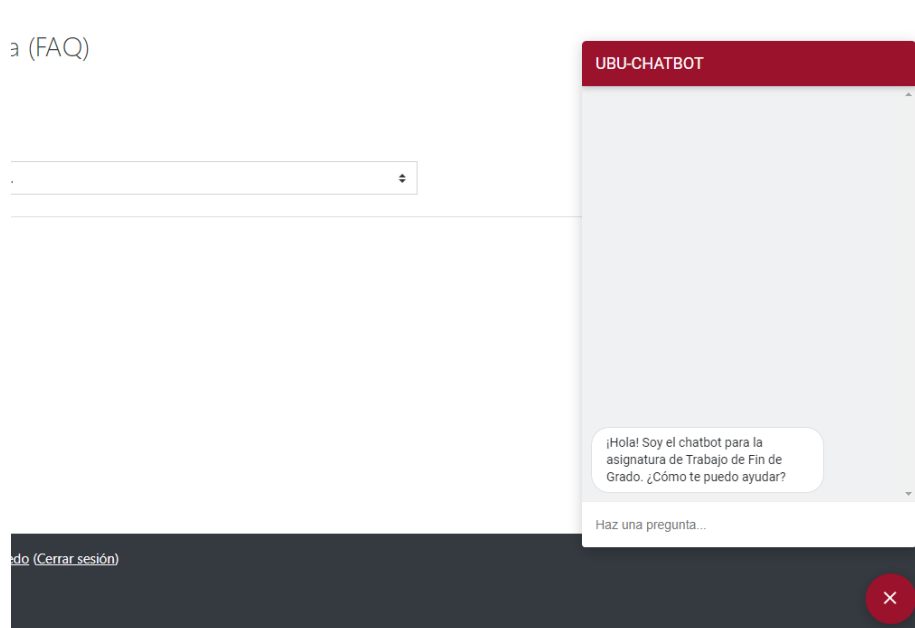


Figura E.6: Ventana del chatbot desplegada.

Podemos comenzar realizando cualquier pregunta. Es importante tener en cuenta que el chatbot esta diseñado para dar respuesta a **estudiantes online del TFG**. Es decir, si le hacemos preguntas como si fuésemos profesores de la asignatura no las va a saber responder, así como si le hacemos preguntas sobre otras asignaturas. Además, todas las respuestas las va a dar enfocadas a la modalidad online, que pueden no coincidir en muchos casos con la que se daría para presencial. Comenzamos por ejemplo saludando al chatbot, aunque es un paso innecesario.



Figura E.7: Respuesta a saludo.

Como vemos en la Figura E.7 al no haber introducido una pregunta nos devolverá el mismo saludo inicial y nos volverá a pedir que le introduzcamos una pregunta.

En este caso vamos a formular la primera pregunta. Podemos hacer cualquier pregunta relativa al Trabajo de Fin de Grado.

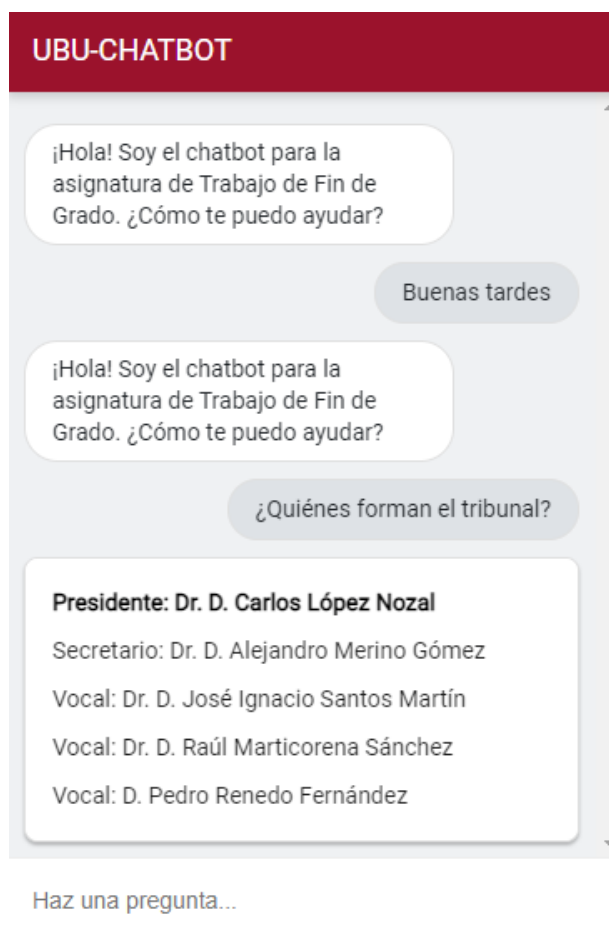


Figura E.8: Pregunta bien formulada.

En este primer ejemplo mostrado en la Figura E.8 se ha formulado la pregunta con signos de interrogación y sin fallos ortográficos. Esto no es un requisito necesario para que el chatbot sea capaz de responder. Podemos formular las preguntas de muchas maneras diferentes.

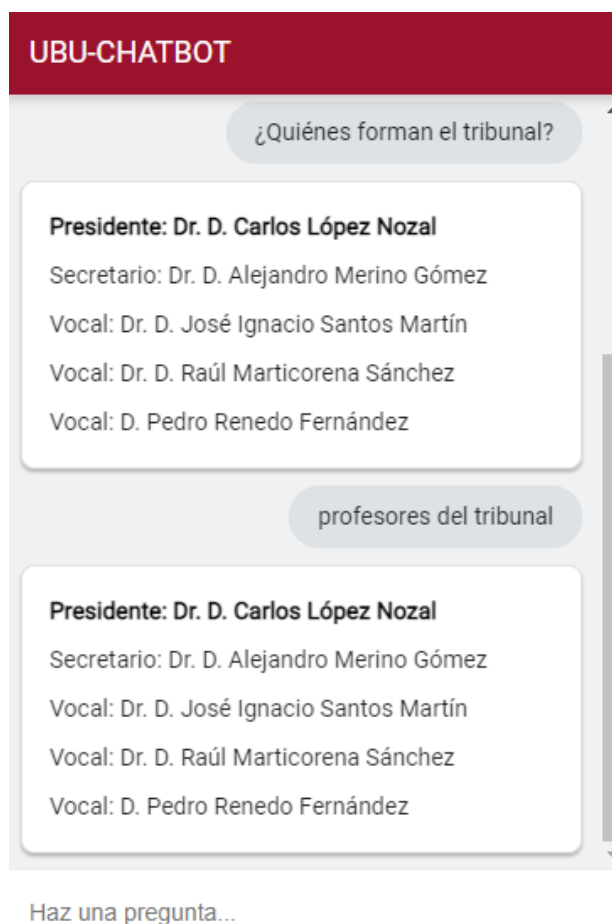


Figura E.9: Distintas formulaciones.

Como se ve en la Figura E.9, distintas formulaciones de la misma pregunta han producido la misma respuesta, ya que el chatbot ha sido capaz de comprenderlas.

No obstante, hay que seguir una serie de buenas prácticas para que el chatbot funcione de la mejor manera posible:

- Solo una pregunta por mensaje.
- El mensaje ha de ser lo más concreto posible.
- La pregunta no debe dar lugar a ambigüedades. Ejemplo: si escribimos “memoria” no le estamos dando información suficiente para saber qué información queremos conocer acerca de la memoria.
- No hacer preguntas personales. Si le preguntamos cuál es nuestra hora de defensa no va a saber responderlo. Hay que recordar que no estamos



identificados (el chatbot no sabe qué alumno somos) y solo se dan respuestas generales a todos los alumnos.

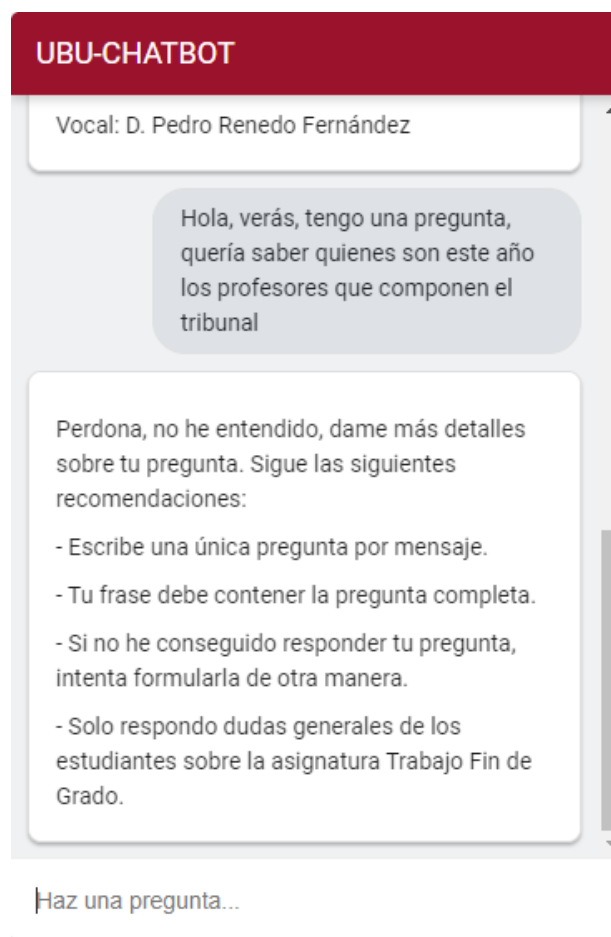


Figura E.10: Pregunta mal formulada.

En la Figura E.10 vemos un ejemplo de pregunta mal formulada. Buscaba el mismo objetivo que las anteriores, pero en este caso el chatbot no ha sido capaz de identificar qué información quería conocer el alumno. Esto se debe a que la pregunta no ha sido concreta y ha introducido información que no tenía nada que ver con la pregunta. En este caso la respuesta ha sido un mensaje de error en el que se aportan indicaciones para formular mejor la pregunta y aumentar así las probabilidades de éxito.

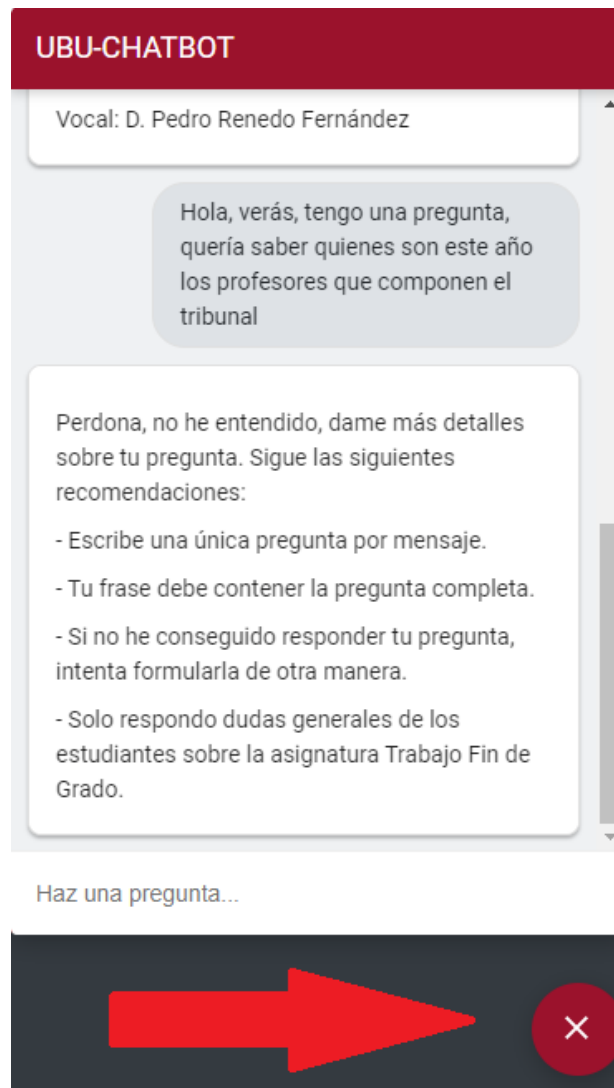


Figura E.11: Cerrar chatbot.

La conversación continuará indefinidamente. Cuando deseemos cerrar la conversación simplemente hacemos clic en el círculo con la 'x' en la parte inferior del chatbot, señalado con la flecha roja en la Figura E.11.



Figura E.12: Chatbot cerrado.

Al hacer clic en el botón de cerrar se volverá al estado inicial en el que la conversación no está desplegada. En este caso, el mensaje de bienvenida ya no aparecerá y únicamente se verá el círculo con el logo del proyecto con el que haciendo clic podremos retomar la conversación cuando queramos. Ver Figura E.12

Si cambiamos de página o refrescamos se reiniciará la conversación, borrándose los mensajes anteriores.

## Smartphone

El funcionamiento en Smartphone es muy similar al de PC, primero abrimos la página en la que está integrado el chatbot.

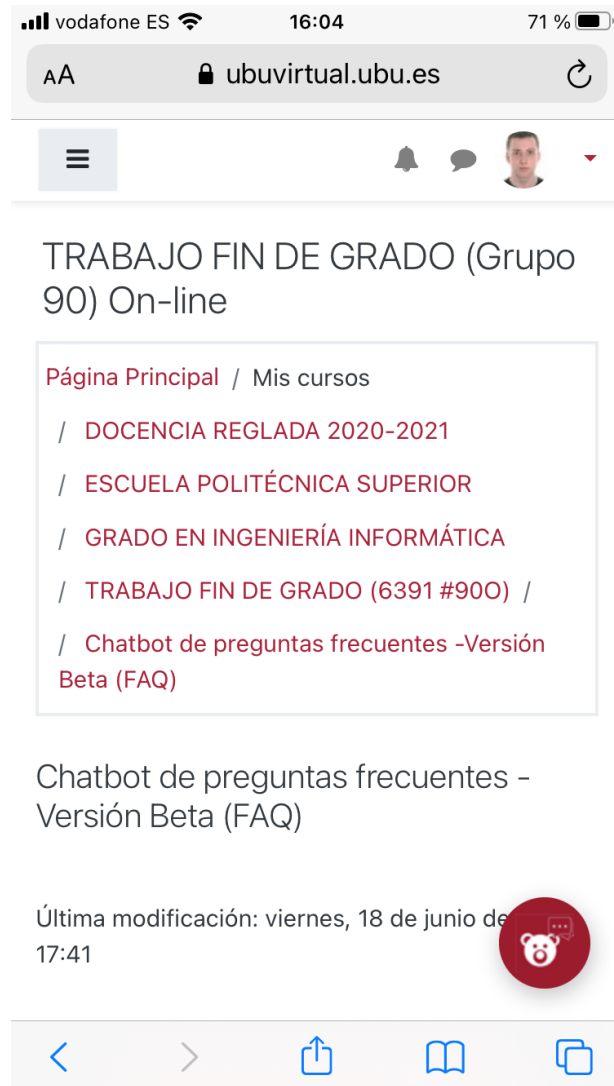


Figura E.13: Página del chatbot desde Safari en iOS.

En la Figura E.13 podemos ver como en esta versión también aparece un pequeño icono con el logo del proyecto, esta vez sin que se despliegue el mensaje de bienvenida. Hacemos clic en dicho botón.

Se nos habrá desplegado una ventana de chat algo diferente a la de versión PC, ya que en este caso ocupa toda la pantalla. Ver Figura E.14



Figura E.14: Ventana del chatbot en Safari iOS.

Hacemos clic en el cuadro de texto inferior con el texto sombreado “Haz una pregunta...”.

Al clicar nos aparecerá el teclado del teléfono móvil como se muestra en la Figura E.15, y ya podremos teclear nuestra pregunta. Una vez la tengamos escrita hacemos clic en el botón ‘intro’ del teclado. Este botón puede ser distinto en otros dispositivos móviles.



Figura E.15: Página chatbot desde Safari en iOS.

El funcionamiento y recomendaciones para realizar las preguntas es el mismo que el explicado en la versión de PC, por lo que es recomendable leer dicha sección.

Una vez hemos hecho clic en ‘intro’ y enviado nuestro mensaje recibiremos la respuesta del chatbot.

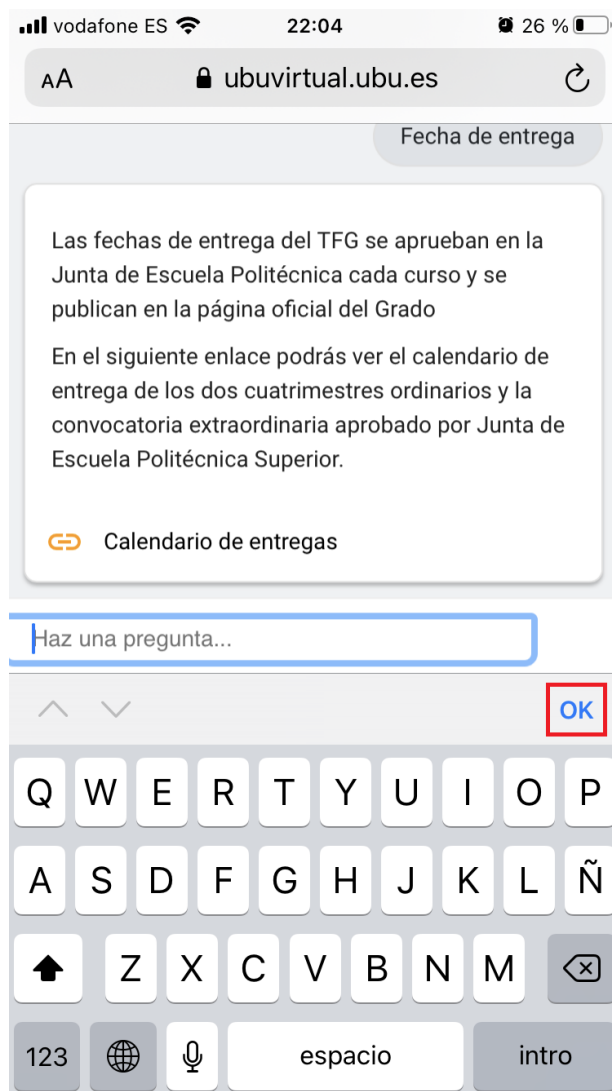


Figura E.16: Respuesta del chatbot y botón de cierre de teclado desde Safari en iOS.

El zoom de la pantalla puede haber sufrido modificaciones, por lo que se recomienda ajustarlo a nuestro gusto. Para cerrar el chat deberemos primero hacer clic en el botón de ‘ok’ marcado en el cuadro rojo en la Figura E.16, lo cual cerrará el teclado.

Una vez se nos ha cerrado el teclado nos aparecerá una fila de opciones en la parte inferior de la pantalla. Hacemos clic en la flecha de volver marcada en rojo en la Figura E.17 para salir definitivamente del chat y volver a la página anterior.



Figura E.17: Volver a la página anterior desde Safari en iOS.

Deslizando la pantalla hacia la izquierda también se podía haber hecho este proceso sin necesidad de cerrar previamente el teclado. Nos devolverá a la página principal de la asignatura.



## Slack

Accedemos al espacio de trabajo en el que está integrado el chatbot como se ha explicado en el apartado de instalación.

URL: <https://ubuchatbot.slack.com>

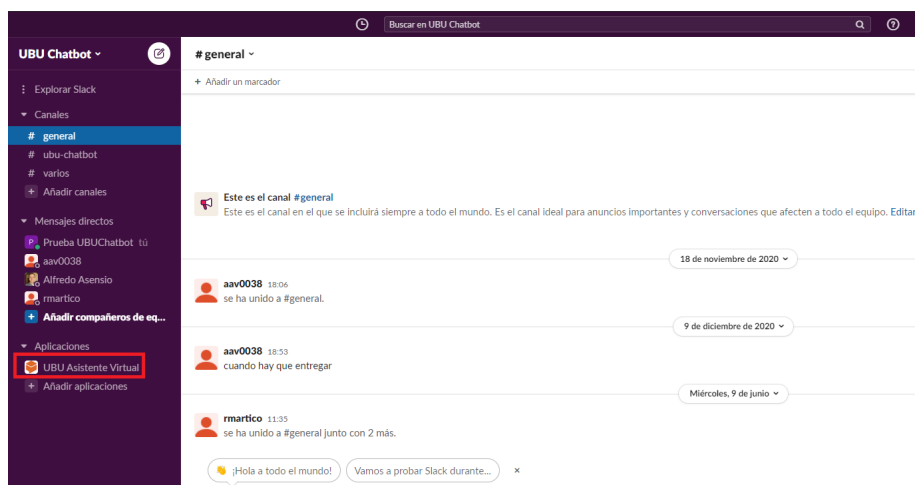


Figura E.18: Página principal del espacio de trabajo de Slack.

Hacemos clic dentro de la sección de ‘Aplicaciones’ en ‘UBU Asistente Virtual’, marcado en la Figura E.18 con un rectángulo rojo.

En la página que se nos habrá abierto nos aparecerá el mensaje de bienvenida del chatbot igual que se muestra en la figura E.19. Ya podemos empezar a introducir nuestras preguntas de la misma forma que en el chatbot de UBUVirtual. Es recomendable seguir las mismas recomendaciones que se han dado en la primera parte del apartado E.4 .

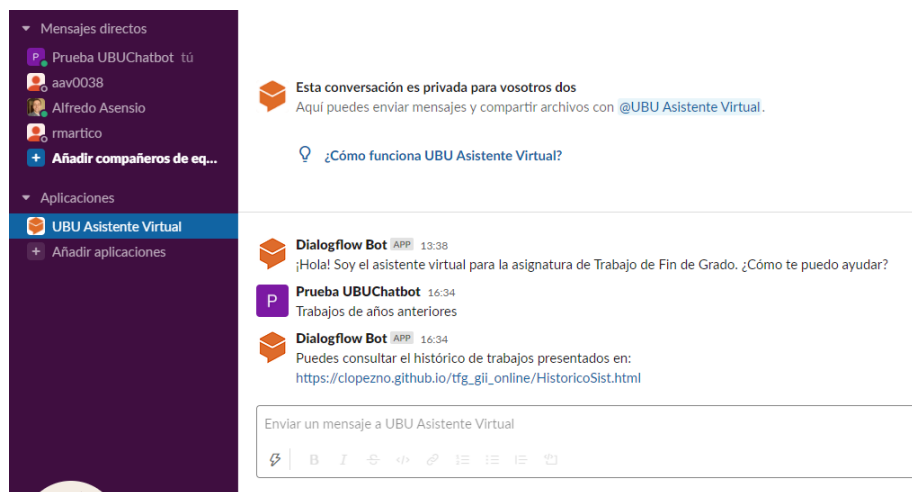


Figura E.19: Página del chatbot en Slack.

---

## Bibliografía

---

- [1] Adobe. Planes y precios de creative cloud, 2021. [Internet; descargado 29-junio-2021].
- [2] Proyectos Agiles. Qué es scrum, 2021. [Internet; descargado 24junio-2021].
- [3] Bankinter. Retención irpf 2021, 2021. [Internet; descargado 30-junio-2021].
- [4] Google Cloud. Dialogflow messenger, 2021. [Internet; descargado 02-julio-2021].
- [5] Gobierno de España. Bases y tipos de cotización 2021, 2021. [Internet; descargado 30-junio-2021].
- [6] MDN Web Docs. Using custom elements, 2021. [Internet; descargado 02-julio-2021].
- [7] Google. Dialogflow trial edition terms of service, 2020. [Internet; descargado 29-junio-2021].
- [8] Google. Dialogflow pricing, 2021. [Internet; descargado 29-junio-2021].
- [9] Heroku. Heroku pricing, 2021. [Internet; descargado 05-julio-2021].
- [10] Don HO. Notepad++ license, 2021. [Internet; descargado 29-junio-2021].
- [11] Idealista. Evolución del precio de la vivienda en alquiler en zaragoza, 2021. [Internet; descargado 29-junio-2021].

- [12] JSON.org. The json license, 2002. [Internet; descargado 29-junio-2021].
- [13] Microsoft. Licencia windows 10 home, 2021. [Internet; descargado 29-junio-2021].
- [14] Microsoft. Microsoft software license terms, 2021. [Internet; descargado 29-junio-2021].
- [15] Movistar. Tarifas de internet, 2021. [Internet; descargado 29-junio-2021].
- [16] Slack. Slack api terms of service, 2020. [Internet; descargado 29-junio-2021].
- [17] Slack. Create slack, 2021. [Internet; descargado 02-julio-2021].
- [18] Slack. Cuáles son los requisitos mínimos para usar slack, 2021. [Internet; descargado 20-junio-2021].