

# Universidad de Alcalá

## Escuela Politécnica Superior

Grado en Ingeniería de Computadores

### Trabajo Fin de Grado

Desarrollo y testeo de un chatbot informativo de apoyo a alumnos de nuevo ingreso en la EPS de la Universidad de Alcalá.

**Autor:** María Chantal Cogolludo

**Tutor:** Francisco Javier Bueno Guillén

2023

UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior

**Grado en Ingeniería de Computadores**

Trabajo Fin de Grado  
Desarrollo y testeo de un chatbot informativo a alumnos de  
nuevo ingreso en la EPS de la Universidad de Alcalá

**Autor:** María Chantal Cogolludo

**Tutor:** Francisco Javier Bueno Guillén

**TRIBUNAL:**

**Presidente:** Iván González Diego

**Vocal 1º:** Manuel de Buenaga Rodríguez

**Vocal 2º:** Francisco Javier Bueno Guillén

**FECHA:** 19 de septiembre de 2023

# Índice

<b>ÍNDICE</b>	<b>1</b>
<b>ÍNDICE DE FIGURAS</b>	<b>3</b>
<b>RESUMEN</b>	<b>7</b>
<b>ABSTRACT</b>	<b>7</b>
<b>PALABRAS CLAVE</b>	<b>7</b>
<b>INTRODUCCIÓN</b>	<b>8</b>
<b>CAPÍTULO 1</b>	<b>11</b>
<b>DESARROLLO Y CREACIÓN DEL CHATBOT</b>	<b>11</b>
REGISTRO WEB	11
PROGRAMACIÓN DEL CHATBOT	12
<i>INTENTS</i>	12
<i>SKILLS</i>	12
FUNCIONAMIENTO DEL CHATBOT	15
LISTA APARTADOS DEL CHATBOT	16
<b>CAPÍTULO 2</b>	<b>20</b>
<b>CREACIÓN WEB Y ENLACE CON EL CHATBOT</b>	<b>20</b>
<b>CAPÍTULO 3</b>	<b>24</b>
<b>CONFIGURACIÓN HERRAMIENTAS DE PRUEBA</b>	<b>24</b>
VARIABLES DE ENTORNO	24
DEPENDENCIAS DE SELENIUM WEB DRIVER Y DE CUCUMBER	26
CREACIÓN DE LAS PRUEBAS AUTOMATIZADAS	27
CASOS DE PRUEBA	32
ESCENARIOS PARA LOS CASOS DE PRUEBA	33
CÓMO SABER SI LOS ESCENARIOS HAN SIDO EXITOSOS	38
<b>CAPÍTULO 4</b>	<b>39</b>
<b>CAMBIOS DE DISEÑO</b>	<b>39</b>
CAMBIOS DE DISEÑO EN EL CHATBOT	39

CAMBIOS DE DISEÑO EN LA WEB	40
<b>CAPÍTULO 5</b>	<b>49</b>
CHATGPT Y LAS INTELIGENCIAS ARTIFICIALES	49
CHATBASE	49
<b>CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO</b>	<b>55</b>
<b>BIBLIOGRAFÍA</b>	<b>57</b>
<b>ANEXOS</b>	<b>58</b>
ANEXO 1: APARTADOS RECOGIDOS EN LAS PRUEBAS DE INTERACCIÓN ESCRITA: PREGUNTAS FRECUENTES	58
ANEXO 2: APARTADOS RECOGIDOS EN LAS PRUEBAS DE INTERACCIÓN ESCRITA: TRÁMITES ADMINISTRATIVOS	62
ANEXO 3: APARTADOS RECOGIDOS EN LAS PRUEBAS DE INTERACCIÓN ESCRITA: CAMBIOS DE GRUPO	64
ANEXO 4: APARTADOS RECOGIDOS EN LAS PRUEBAS DE INTERACCIÓN ESCRITA: WIFI Y VPN	65
ANEXO 5: APARTADOS RECOGIDOS EN LAS PRUEBAS DE INTERACCIÓN ESCRITA: LOCALIZACIONES	66

## Índice de figuras

<b>Figura 1: Registro SAP Conversational AI</b>	<b>11</b>
<b>Figura 2: Política de privacidad del chatbot</b>	<b>11</b>
<b>Figura 3: Tipo de chatbot, título y descripción</b>	<b>11</b>
<b>Figura 4: Creación intent</b>	<b>12</b>
<b>Figura 5: Primeras frases del intent</b>	<b>12</b>
<b>Figura 6: Creación skill</b>	<b>12</b>
<b>Figura 7: Enlace de la skill con un trigger</b>	<b>13</b>
<b>Figura 8: Elección tipo de respuesta de la skill</b>	<b>13</b>
<b>Figura 9: Elección del tipo de botón a insertar en la skill</b>	<b>13</b>
<b>Figura 10: Skill con redirección a otras skills</b>	<b>14</b>
<b>Figura 11: Skill con redirección a enlaces externos</b>	<b>14</b>
<b>Figura 12: Tipos de mensajes de respuesta para una skill</b>	<b>14</b>
<b>Figura 13: Chatbot sin entrenar</b>	<b>15</b>
<b>Figura 14: Chatbot entrenando</b>	<b>15</b>
<b>Figura 15: Chatbot entrenado</b>	<b>15</b>
<b>Figura 16: Vista previa del chatbot en la web SAP Conversational AI</b>	<b>16</b>
<b>Figura 17: Analizador de expresiones del chatbot</b>	<b>16</b>
<b>Figura 18: Mensaje de bienvenida del chatbot</b>	<b>18</b>
<b>Figura 19: Vista previa en SAP Conversational AI con mensaje de bienvenida</b>	<b>19</b>
<b>Figura 20: Creación proyecto Maven</b>	<b>20</b>
<b>Figura 21: Creación archivo index.html</b>	<b>20</b>
<b>Figura 22: Apartado de conexión en SAP Conversational AI</b>	<b>21</b>
<b>Figura 23: Script del chatbot para su incorporación en la web</b>	<b>21</b>
<b>Figura 24: Configuración inicial del chatbot para la web</b>	<b>21</b>
<b>Figura 25: Programación inicial para alojar el chatbot</b>	<b>22</b>
<b>Figura 26: Vista previa chatbot abierto en web inicial</b>	<b>22</b>
<b>Figura 27: Vista previa chatbot cerrado en web inicial</b>	<b>23</b>
<b>Figura 28: Saludo al chatbot</b>	<b>23</b>
<b>Figura 29: Pregunta al chatbot</b>	<b>23</b>

<b>Figura 30: Propiedades del sistema Windows 10</b>	<b>24</b>
<b>Figura 31: Creación variable de usuario para java</b>	<b>24</b>
<b>Figura 32: Vista final variables de entorno</b>	<b>24</b>
<b>Figura 33: Comprobación versión de java</b>	<b>25</b>
<b>Figura 34: Vista de descarga de Maven</b>	<b>25</b>
<b>Figura 35: Creación variable de usuario para Maven</b>	<b>25</b>
<b>Figura 36: Comprobación versión de Maven</b>	<b>26</b>
<b>Figura 37: Instalación de dependencias en pom.xml</b>	<b>26</b>
<b>Figura 38: Plugin Cucumber</b>	<b>27</b>
<b>Figura 39: Plugin Gherkin</b>	<b>27</b>
<b>Figura 40: Estructura inicial del proyecto</b>	<b>27</b>
<b>Figura 41: Escenario Cucumber: first greetings</b>	<b>28</b>
<b>Figura 42: Escenario Cucumber: first greetings y Background</b>	<b>28</b>
<b>Figura 43: Inspeccionar elemento en el navegador Google Chrome</b>	<b>29</b>
<b>Figura 44: Definición variables</b>	<b>29</b>
<b>Figura 45: Asignación del identificador a la variable</b>	<b>30</b>
<b>Figura 46: Creación métodos getter</b>	<b>30</b>
<b>Figura 47: Definición métodos en Selenium del escenario de Cucumber</b>	<b>31</b>
<b>Figura 48: Escenarios Cucumber: selección escrita</b>	<b>34</b>
<b>Figura 49: Método Selenium para el escenario selección escrita</b>	<b>35</b>
<b>Figura 50: Escenarios Cucumber: trámites administrativos y convalidaciones</b>	<b>35</b>
<b>Figura 51: Escenarios Cucumber: preguntas frecuentes</b>	<b>36</b>
<b>Figura 52: Escenarios Cucumber: redes, cómo moverme y localizaciones</b>	<b>36</b>
<b>Figura 53: Métodos para interacción con botones</b>	<b>37</b>
<b>Figura 54: Evidencia pruebas exitosas</b>	<b>38</b>
<b>Figura 55: Cambio en el método para provocar fallos</b>	<b>38</b>
<b>Figura 56: Evidencia escenario fallido</b>	<b>38</b>
<b>Figura 57: Apartado de conexiones en SAP Conversational AI</b>	<b>39</b>
<b>Figura 58: Elección colores para el chatbot</b>	<b>39</b>
<b>Figura 59: Elección logo y título para el chatbot</b>	<b>39</b>
<b>Figura 60: Elección iconos y mensaje de bienvenida</b>	<b>40</b>

<b>Figura 61: Vista final del chatbot en web con saludo</b>	<b>40</b>
<b>Figura 62: Edición de index.html: importación de main.css e iconos</b>	<b>41</b>
<b>Figura 63: Edición cuerpo de index.html</b>	<b>41</b>
<b>Figura 64: Creación sap.html</b>	<b>42</b>
<b>Figura 65: Cuerpo sap.html</b>	<b>42</b>
<b>Figura 66: Creación y cuerpo de pruebas.html</b>	<b>43</b>
<b>Figura 67: Pruebas.html – definición Cucumber</b>	<b>43</b>
<b>Figura 68: Pruebas.html – definición Selenium</b>	<b>44</b>
<b>Figura 69: Pruebas.html – finalización definición Selenium</b>	<b>44</b>
<b>Figura 70: Pruebas.html – introducción vídeos de las pruebas</b>	<b>44</b>
<b>Figura 71: Pruebas.html – carrusel para vídeos de las pruebas</b>	<b>45</b>
<b>Figura 72: Pruebas.html – vídeo interacción escrita</b>	<b>45</b>
<b>Figura 73: Pruebas.html – vídeo preguntas frecuentes</b>	<b>45</b>
<b>Figura 74: Pruebas.html – vídeo trámites administrativos</b>	<b>45</b>
<b>Figura 75: Pruebas.html – vídeo convalidaciones</b>	<b>46</b>
<b>Figura 76: Pruebas.html – vídeo localizaciones</b>	<b>46</b>
<b>Figura 77: Pruebas.html – vídeo redes</b>	<b>46</b>
<b>Figura 78: Vista en la web del carrusel de vídeos</b>	<b>46</b>
<b>Figura 79: Creación memoria.html</b>	<b>47</b>
<b>Figura 80: Clases main.css usadas en los html parte 1</b>	<b>47</b>
<b>Figura 81: Clases main.css usadas en los html parte 2</b>	<b>47</b>
<b>Figura 82: Clases main.css usadas en los html parte 3</b>	<b>48</b>
<b>Figura 83: Clases main.css usadas en los html parte 4</b>	<b>48</b>
<b>Figura 84: Clases main.css usadas en los html parte 5</b>	<b>48</b>
<b>Figura 85: Página principal Chatbase</b>	<b>49</b>
<b>Figura 86: Creación chatbot a partir de enlace en Chatbase</b>	<b>49</b>
<b>Figura 87: Enlaces incluidos en el chatbot de Chatbase parte 1</b>	<b>50</b>
<b>Figura 88: Enlaces incluidos en el chatbot de Chatbase parte 2</b>	<b>51</b>
<b>Figura 89: Primera vista al chatbot creado con Chatbase</b>	<b>51</b>
<b>Figura 90: Preguntas con chatbot de Chatbase: sitio web</b>	<b>52</b>
<b>Figura 91: Preguntas con chatbot de Chatbase: carreras</b>	<b>52</b>

<b>Figura 92: Preguntas con chatbot de Chatbase: carreras reformulada</b>	<b>53</b>
<b>Figura 93: Grados de la Escuela Politécnica Superior</b>	<b>53</b>
<b>Figura 94: Preguntas con chatbot de Chatbase: información tfg</b>	<b>54</b>
<b>Figura 95: Preguntas con chatbot de Chatbase: fechas de exámenes</b>	<b>54</b>



## Resumen

Se va a crear y a desarrollar un chatbot de forma manual enfocado a alumnos de nuevo ingreso de la Escuela Politécnica Superior mediante la tecnología encontrada en la web *SAP Conversational AI*. Para poder implementar este chatbot, se creará una página web enfocada al trabajo, donde se podrá abrir este bot así como consultar información de interés relacionada. Por último, se trabajará con las herramientas Selenium y Cucumber para ejecutar pruebas automatizadas en el chatbot, con el fin de comprobar que el desarrollo del mismo cumple con su versión implementada en la página web.

## Abstract

A chatbot will be created and developed manually focused on new students of the “Escuela Politécnica Superior” using the technology found on the SAP Conversational AI website. In order to implement this chatbot, a web page focused on work will be created, where this bot can be opened as well as consult related information of interest. Finally, we will work with the Selenium and Cucumber tools to run automated tests on the chatbot, in order to verify that its development complies with its version implemented on the website.

## Palabras clave

- Chatbot
- Selenium
- Cucumber
- Asistente
- Pruebas

## Introducción

Esta propuesta de Trabajo Fin de Grado combina la formación recibida en el Grado de Ingeniería de Computadores con la experiencia adquirida en las prácticas en empresa realizadas en la compañía UST. En dichas prácticas he desarrollado un conjunto de chatbots para su posterior fase de pruebas automatizadas. Aplicando, por un lado, conocimientos y habilidades adquiridas en asignaturas tales como Ingeniería del Software, Desarrollo del Software, Fundamentos de Programación, y, en menor medida y en situaciones concretas, Compiladores, y adquiriendo, por otro, nuevos conocimientos propios de las tareas asignadas en las prácticas.

Los alumnos de nuevo ingreso a menudo tienen múltiples problemas y dudas en cuanto al funcionamiento de la Universidad en general y de la Escuela Politécnica en particular, las gestiones que deben realizar, los plazos, la organización del centro, así como la forma de actuar ante determinadas situaciones.

El objetivo de este TFG es el de desarrollar un chatbot que oriente a los nuevos alumnos y puedan realizar su ingreso en la Universidad de la forma más informada posible. Se propone que el chatbot sea capaz de responder a las dudas más frecuentes que se encuentren los alumnos y de hacer un recorrido simple y conciso por la web de la Escuela Politécnica Superior de la Universidad de Alcalá. Como parte del proyecto, se integrará en una página web para, posteriormente, poder realizar pruebas automatizadas en él y comprobar su correcto funcionamiento.

Gracias a la herramienta *SAP Conversational AI* [1], el chatbot se entrenará con preguntas que el usuario pueda realizar al asistente, para que así este proporcione respuestas que puedan servir de ayuda a cualquier persona que quiera navegar por la página web. Una vez desarrollado y entrenado el Chatbot, se añadirá a una página web que permitirá al usuario interactuar con él. Esta página web se desarrollará desde cero como punto de apoyo para poder visualizar el chatbot y para poder realizar las pruebas mediante drivers que apunten al navegador Google Chrome. Debido a ciertos problemas con los servidores de la Universidad, así como la gran cantidad de alumnos y profesores que necesitan estos servicios, he decidido crear un dominio por cuenta propia mediante la web Nominalia [2], bajo la url de mariachantaltfg.es [3].

Cabe mencionar que los chatbots desarrollados con esta herramienta pueden integrarse en cualquier página web de la que se tenga acceso a su código fuente para poder modificarlo, simplemente añadiendo un script. De querer, por ejemplo, integrarse este chatbot en la web de la Escuela Politécnica Superior, se necesitaría el código original con el que se haya programado la web para poder modificarlo y añadir el script necesario.

Una vez integrado en la web, el chatbot estará totalmente operativo para cualquier usuario que acceda a la web y quiera mantener una conversación. Para amenizar la página web y que su utilidad no sea únicamente alojar el chatbot, se le añadirán diferentes submenús con información relativa al TFG, como una breve descripción de las herramientas utilizadas y esta misma memoria para poder acceder a la información necesaria de manera inmediata.

Cuando se encuentre integrado en la página web y en una versión estable, el chatbot se someterá a pruebas de software web automatizadas. Estas pruebas serán realizadas con *Selenium Web Driver* [4] y con *Cucumber* [5]. Estas pruebas tienen el objetivo de comprobar que

las preguntas que pueda introducir un usuario, corresponden con la programación que se le ha dado en toda su fase de desarrollo.

Para organizar este trabajo, he decidido enfocarlo en tres fases diferenciadas y dos apartados de mejoras e investigación posterior al acabado del trabajo.

1. Fase 1: desarrollo y creación del chatbot: se empleará la herramienta *SAP Conversational AI* para desarrollar el chatbot. Para esto, se diferenciarán dos procesos: las preguntas que pueda hacer el usuario, y las respuestas que proporcionará el chatbot. Habrá que definir las preguntas y las respuestas para cada apartado que se quiera incluir en el chatbot (la lista de apartados que cubrirá el chatbot se puede encontrar explicada en el capítulo 1). Este chatbot tendrá la capacidad de mantener conversaciones con el usuario sobre las preguntas más frecuentes, información sobre trámites administrativos, temas relacionados con asignaturas y horarios, ajustes de WiFi y VPN y localizaciones dentro de la escuela entre otros.

Toda esta información será extraída de la página oficial de la Escuela Politécnica Superior [6]. A pesar de que la estructura de la lista de apartados cubiertos por el chatbot pueda parecer similar a la propuesta en la web de la EPS, esta lista está diseñada para agrupar los temas según he considerado que tienen relación entre sí. Es decir, no sigue exactamente los menús y submenús que se pueden encontrar en la página oficial. De igual forma, no cubre todo el contenido de la página oficial. He seleccionado los temas con más relevancia, o que más pueden llamar la atención a alumnos tanto novatos como veteranos. En el caso de que la EPS decidiera incorporar este tipo de asistente a su página web, sería necesario recoger la información faltante para incorporarla al chatbot.

El listado total de frases que el chatbot puede reconocer y responder se adjuntará un documento adicional en formato Excel. El listado de frases posibles para las pruebas de interacción escrita se encuentra al final del documento como anexo.

2. Fase 2: creación web y enlace con el chatbot: la página web se creará desde cero en formato *.html* con las extensiones necesarias para el posterior cambio de la interfaz de la misma (explicado en el apartado de cambios de diseño) y bajo el dominio *mariachantaltfg.es*. Esta web se desarrollará a través del IDE IntelliJ, creando así el proyecto completo que incluirá el código de todas las fases del trabajo. En una primera instancia, únicamente se añadirá el chatbot a la web en blanco para poder proceder con las pruebas automatizadas.

Una vez el chatbot esté integrado en la web, el usuario podrá interactuar con él. Dado que la programación de este asistente ha sido manual, hay cierto número de preguntas que va a poder entender, por lo que cabe la posibilidad de que haya que reformular ciertas frases para poder llegar a la respuesta deseada.

3. Fase 3: configuración herramientas de prueba: una vez que el chatbot se encuentre en una versión estable ya incorporado a la página web, comenzará la fase de las pruebas automatizadas. Para ello, en primer lugar se deberán instalar las herramientas necesarias, siendo estas Java 17 [7], *Maven* [8], *Selenium Web Driver* y *Cucumber*. La utilidad y funcionamiento de cada una de estas herramientas se encuentra explicado en el capítulo 3. Una vez el proyecto esté configurado y listo para proceder con las pruebas, el siguiente paso será formar el conjunto de casos de prueba que se van a llevar a cabo.

Estos casos de prueba cubrirán, al menos, el 70% de la lista de apartados que engloba el chatbot. No se ha realizado el 100% de los casos debido a una selección personal de los que se han considerado los casos más relevantes y los que más interés despiertan en los alumnos de forma general.

La herramienta *Cucumber* nos permitirá dividir por escenarios estos casos de prueba, separándolos por agrupaciones de preguntas similares, esto será el esqueleto de las pruebas. *Cucumber* permite utilizar el lenguaje natural para poder configurar estos escenarios. Este esqueleto tendrá dos variantes: interacción escrita, donde el usuario escribirá directamente la pregunta al chatbot, e interacción mediante botones, donde el usuario podrá desplazarse por la conversación pulsando los botones que crea necesarios hasta dar con su pregunta específica.

Una vez formado este esqueleto, se programará el funcionamiento de estas pruebas con la herramienta *Selenium Web Driver*. Seleccionando cada paso de cada escenario previamente descrito, se utilizará el lenguaje de programación Java para recoger los identificadores que especifican la posición de cada mensaje del chatbot, de esta forma, se podrá comprobar si dada cierta pregunta (o cierta interacción de botones), proporciona la respuesta que se espera del chatbot.

4. Cambios de diseño: este apartado se va a dedicar a la parte estética del trabajo. En primer lugar, se personalizará la imagen del chatbot dentro de la misma web de *SAP Conversational AI*, eligiendo los colores, los iconos y los mensajes predeterminados que tendrá la versión final del chatbot en la página web.

El siguiente paso será mejorar la interfaz de la página web. Dejando de lado la web blanca básica que se empleará para poder comprobar que las pruebas funcionan de forma adecuada, se trabajará en esta web para que sea más visual. En la página principal se podrá encontrar un pequeño mensaje de bienvenida que indicará al usuario los pasos más básicos a seguir para poder navegar por la web. En la esquina inferior derecha, se alojará el chatbot. También se incluirán tres menús en la parte superior de la web, en ellos, se podrá acceder a información sobre la herramienta de desarrollo *SAP Conversational AI*, a una pequeña introducción a las herramientas para las pruebas automatizadas, o a la memoria explicativa de este trabajo.

Se añadirán de igual forma una serie de vídeos que muestran el funcionamiento de las pruebas automatizadas, donde se podrá ver su ejecución completa así como la evidencia de que funcionan correctamente. Este apartado se encuentra explicado al final de la sección *Pruebas.html*

5. ChatGPT y las inteligencias artificiales: desde el comienzo de este trabajo hasta su (casi) finalización, el aumento de popularidad de las inteligencias artificiales en forma de asistente, como ChatGPT, ha sido exponencial. Dado que es un tema que afecta de forma directa a este trabajo, he decidido investigar sobre una variante de *ChatGPT* que permite crear un asistente basándose en una página web asociada. Aquí se hará una pequeña comparativa de la fuerza de las inteligencias artificiales contra el control de poder programar el asistente de forma manual.

# CAPÍTULO 1

## Desarrollo y creación del chatbot

### Registro web

El primer paso para poder proceder con la creación del Chatbot es registrarnos en la web de *SAP Conversational AI*. Aquí nos tendremos que registrar, y posteriormente elegir las características principales que va a tener el chatbot. Vamos a seleccionar que se quiere que el bot ejecute las acciones programadas en un formato de “*small talk*”; este término se refiere a una conversación informal, ligera, podría traducirse como conversación trivial. Después se le añade un nombre y una pequeña descripción, seguido de la configuración de tipo de dato que se va a usar, en este caso no personal. Y finalmente, el bot será privado.

The screenshot shows the 'Registro' (Registration) page. It includes fields for 'Datos personales' (Personal data) with inputs for Name (María), Surname (Chantal Cogolludo), and Email (maria.chantal@edu.uah.es). There is a section for 'Establecer contraseña' (Set password) with a password field and a confirmation field. Below that are 'Términos y condiciones' (Terms and conditions) with a checkbox and a link to the privacy declaration. A 'Registrarse' (Register) button is at the bottom right. At the bottom left, there is a 'SAP ID Service' section with a link to 'Get SAP Universal ID'.

Figura 1: Registro SAP Conversational AI

The screenshot shows the 'Data Policy' and 'Bot visibility' settings. Under 'Data Policy', there are radio buttons for 'Non-personal' (selected), 'Personal', 'Sensitive Personal', and 'Health'. Below this is a 'Data retention' field set to 'unlimited duration'. Under 'Store conversation data', there are radio buttons for 'Store (recommended)' (selected), 'Do not store', and 'Do not store'. Below this is a 'Bot visibility' section with radio buttons for 'Public' and 'Private' (selected). A 'Create' button is at the bottom.

Figura 2: Política de privacidad del chatbo

The screenshot shows the 'What do you want your chatbot to do?' step. It has two options: 'Perform Actions using conversational flows' (selected) and 'Retrieve Answers from FAQ documents'. Below this is the 'Select predefined skills for your bot' step with five options: 'Greetings', 'Small Talk' (selected), 'Weather', 'Customer Satisfaction', and 'Set Alarm'. Below that is the 'Create your bot' step with a name field (maria.chantal / Ifg), a description field (Chatbot informativo de apoyo a alumnos de nuevo ingreso en la EPS de la Universidad de Alcalá), a topics field (communications), and a default language dropdown (Spanish).

Figura 3: Tipo de chatbot, título y descripción

## Programación del chatbot

Después de esto procedemos a la creación del chatbot. Para crear el bot vamos a separar dos conceptos principales: *intents* y *skills*.

**Intents:** Con esto vamos a definir las posibles frases que el usuario va a introducir al chatbot. Se van a separar por los bloques especificados por el esquema de preguntas que se va a programar. Por ejemplo, para el apartado de “webs”, un usuario podría preguntar cosas como: “cuál es la web de la UAH”, “dime cual es la página web”, o “por favor, llévame a la web de la uah”. Pensando y añadiendo diversas preguntas relacionadas al tema que se quiere responder, podremos entrenar al bot para que sepa reconocer a lo que el usuario le pregunta.

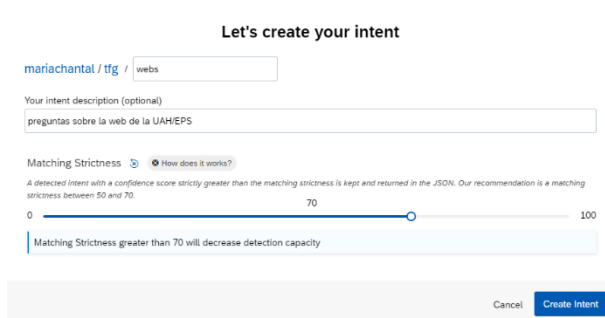


Figura 4: Creación intent

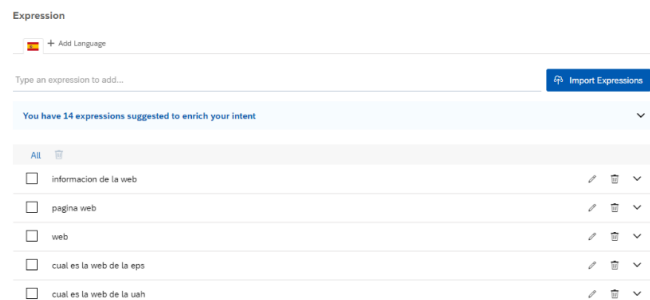


Figura 5: Primeras frases del intent

Cuantas más frases se introduzcan en cada *intent*, más inteligente se volverá el chatbot, y de esta forma podrá tener más opciones de reconocer la entrada de texto del usuario.

**Skills:** Dentro del apartado “Build”, se van a poder empezar a definir las “skills”. Una skill va a ser la encargada de reconocer las frases recogidas en los intent y proporcionar una respuesta. Para lograr esto, se deben seguir diferentes pasos, seguiremos centrándonos en el apartado “web”.

1) Creamos la *skill*. Aquí habrá que ponerle un nombre identificativo y elegir el tipo de *skill* que se quiere probar. Para todo este proyecto, se va a trabajar con el tipo “*Floating*”, que es la más indicada para las pequeñas conversaciones que vamos a querer tener con el chatbot.

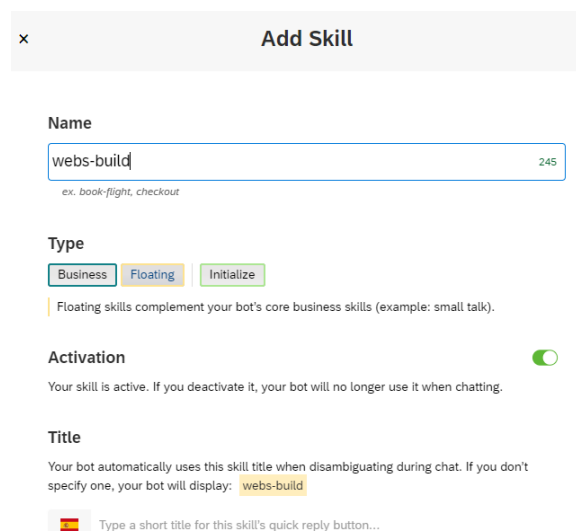


Figura 6: Creación skill

2) Definimos la sección “*trigger*”. Un *trigger* actúa como un capturador, el cual será el encargado de analizar el *intent* que se ha definido previamente y relacionarlo con la *skill* que estamos definiendo. Por ejemplo, al haber creado antes el *intent* “webs”, tendremos que decir que esta *skill* se lanzará si ese *intent* está presente, como se puede observar en la siguiente figura 7.

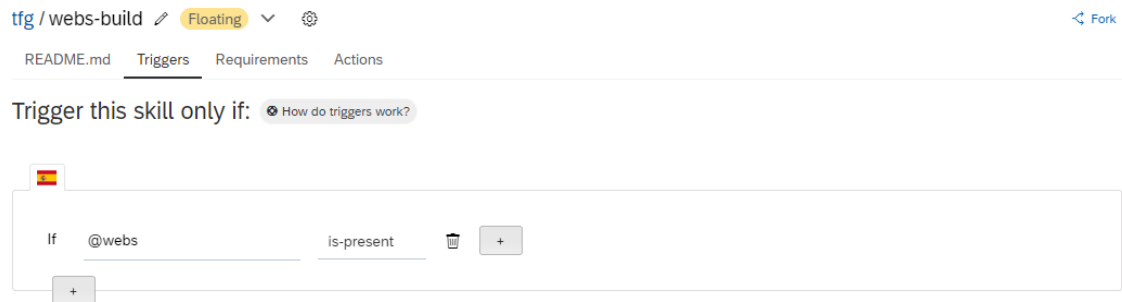


Figura 7: Enlace de la skill con un trigger

3) Definimos la sección “*Actions*”. En esta sección vamos a poder configurar cómo va a ser el mensaje que vea el usuario una vez escriba su pregunta.

En primer lugar, se selecciona la opción “*Choose message Type*”. Dentro de esta opción, podremos elegir la forma de interacción. Este chatbot va a estar configurado mediante botones, para que aparte de proporcionar al usuario la información que busca, pueda acceder a enlaces que complementen esa información. Todas las opciones de respuesta se pueden consultar en la Figura 8.

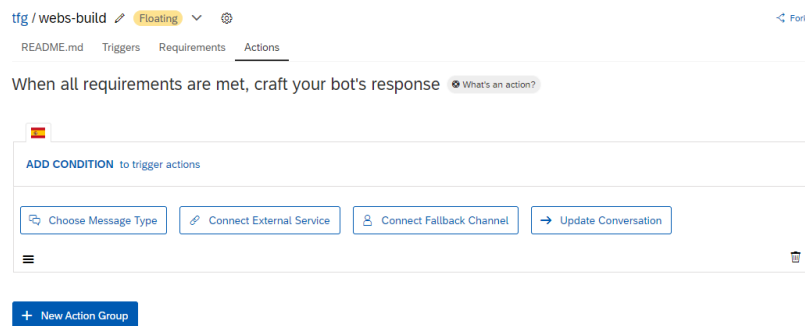


Figura 8: Elección tipo de respuesta de la skill

Cuando seleccionamos la opción “*Buttons*”, nos aparecerá un cuadro para escribir el mensaje que se quiere mostrar por pantalla. Aquí escribiremos la información relacionada a la pregunta hecha por el usuario. En la opción “*Add a button*”, existen cuatro alternativas para añadir un

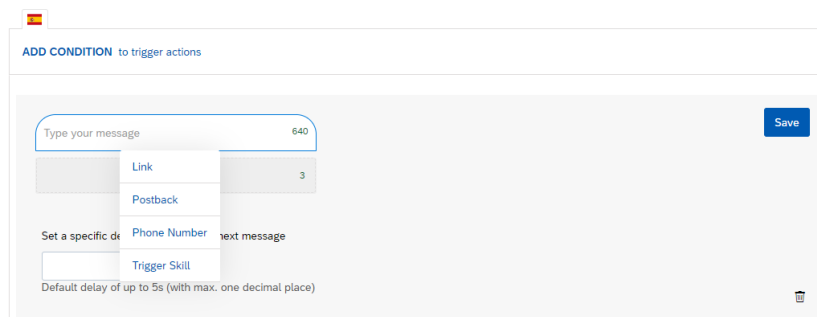


Figura 9: Elección del tipo de botón a insertar en la skill

botón: añadir un enlace a una nueva pestaña, un *postback* (este término se emplea cuando un usuario hace una llamada a cierto servicio y este tarda un tiempo en emitir la respuesta), un número de teléfono, o un enlace directo a otra *skill* relacionada. En la figura 11 se puede observar una *skill* rellena con dos botones a enlaces externos, los cuales redirigirán al usuario a las páginas deseadas.

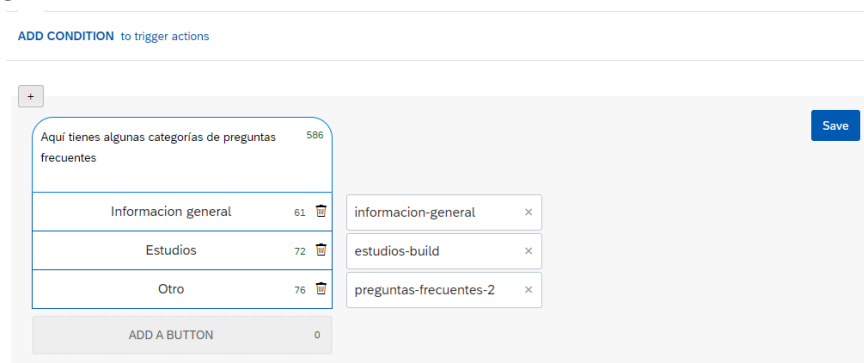


Figura 10: Skill con redirección a otras skills

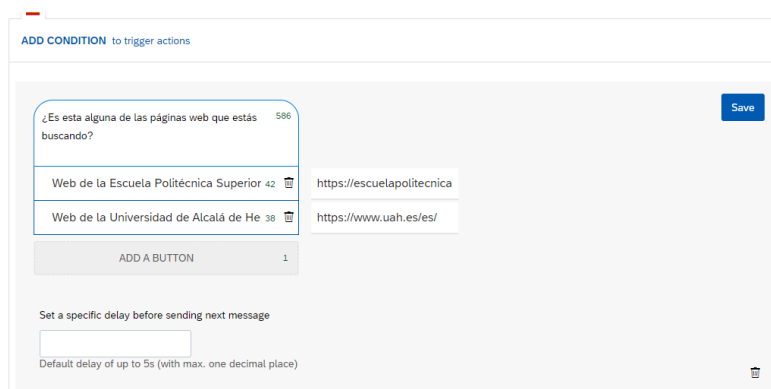


Figura 11: Skill con redirección a enlaces externos

Para este proyecto, se van a utilizar los botones de enlace a una nueva pestaña y de enlace a otra *skill* relacionada. Un ejemplo de enlace a otra *skill* relacionada sería el de la figura 12, donde podemos ver una *skill* de preguntas frecuentes que recoge ciertos apartados que, dependiendo del botón que seleccione el usuario, lo redirigirá a esa *skill* en concreto. Una vez la navegación entre *skills* finalice, el usuario encontrará la respuesta a su pregunta o algún enlace para complementar la información ofrecida

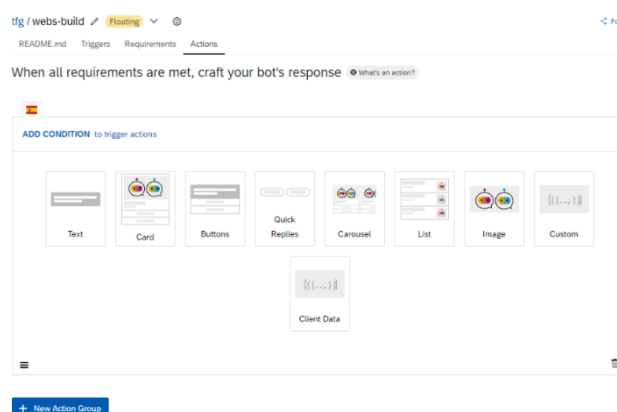


Figura 12: Tipos de mensajes de respuesta para una skill



Ahora, hay que probar que el *intent* de las webs funciona correctamente. Para esto, hay que entrenar al chatbot. Para ello, en la parte superior de la página, encontraremos un botón “Train” (figura 13) con un círculo naranja al lado. El color naranja significa que hay cambios nuevos en el chatbot, y que necesita entrenar. Al pulsar el botón de *train*, primero veremos una rueda que indica que el entrenamiento está en proceso (figura 14), y finalmente aparecerá un círculo verde al lado, lo cual significa que el chatbot ha sido entrenado y está listo para conversar (figura 15).

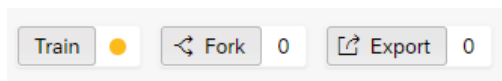


Figura 13: Chatbot sin entrenar

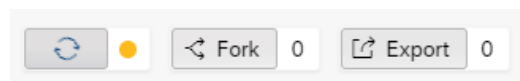


Figura 14: Chatbot entrenando

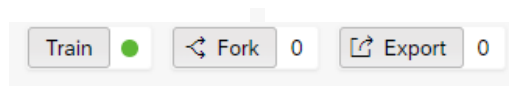


Figura 15: Chatbot entrenado

## Funcionamiento del chatbot

Cabe recalcar que la interacción con este chatbot tendrá dos variantes:

- 1) **Interacción escrita:** con esto, el chatbot responderá directamente las preguntas que el usuario escriba. Esto está íntegramente ligado con la configuración de los *triggers* vista en el paso 2 de la configuración de *skills*.
- 2) **Interacción con botones:** si el usuario no tiene muy claro qué va a preguntar, el chatbot tendrá una configuración de botones mediante la cual el usuario podrá navegar entre los temas seleccionados sin necesidad de escribir nada, siguiendo el sistema redirigir a un enlace de otra *skill*. Una vez llegue a la opción que considere, podrá consultar la información de igual forma que si escribiera directamente una frase. Para poder iniciar esta interacción con botones, el usuario deberá saludar al chatbot primero.

Para comprobar que la programación de los *intents* y de las *skills* funciona correctamente, la web de *SAP Conversational AI* proporciona una vista previa de cómo quedaría el chatbot, así como un analizador de expresiones para comprobar que la respuesta del chatbot corresponde con lo establecido a la entrada de texto que le pueda dar el usuario.

La vista previa se muestra como un chat normal, donde el usuario procederá a escribir y el chatbot le responderá con la *skill* programada (figura 16).

En el analizador de expresiones (figura 17), se introducirá la misma expresión que se ha puesto en la vista previa del chat. Como se puede ver en la imagen, al introducir la misma frase, la configuración del chatbot devuelve que el usuario se está refiriendo al *intent* concreto que acabamos de programar.

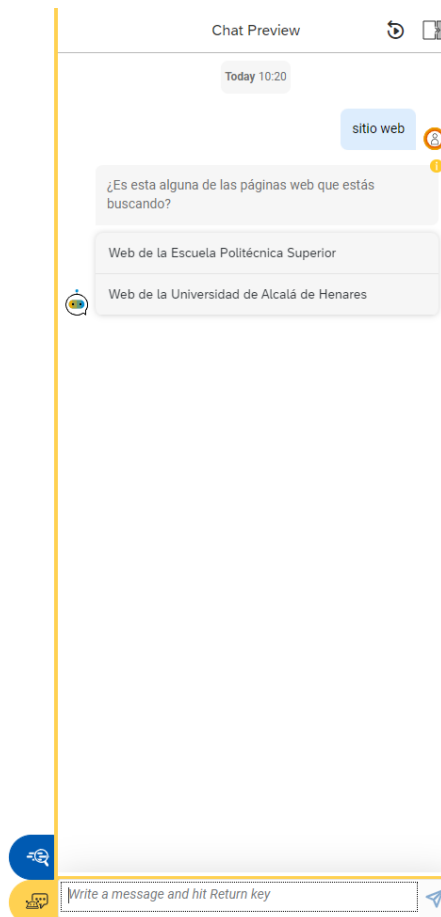


Figura 16: Vista previa del chatbot en la web  
SAP Conversational AI

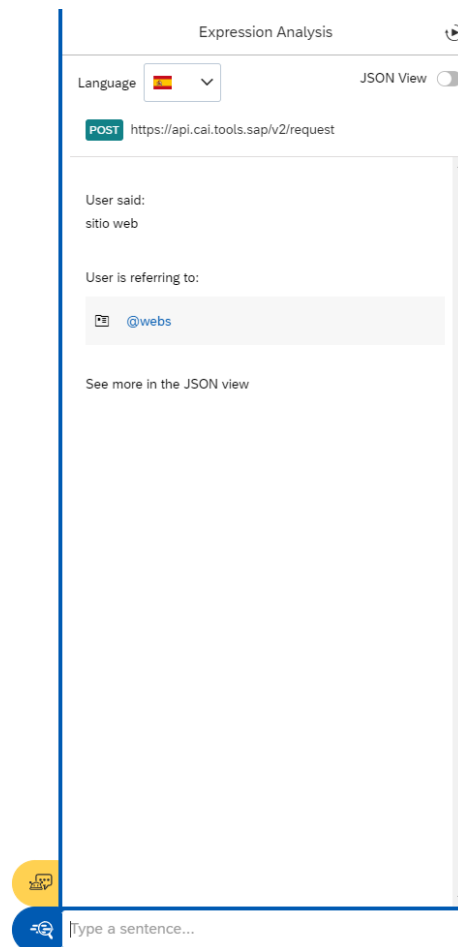


Figura 17: Analizador de expresiones del  
chatbot

La pregunta ahora es, ¿qué casos se van a recoger para entrenar al chatbot? En la siguiente lista se van a enumerar las situaciones para las cuales se entrenará el chatbot

### Lista apartados del Chatbot

- Preguntas frecuentes
  - Información general
    - Webs
    - Qué ofrece
    - Ramas conocimiento
    - Normativas
    - Coordinadores
  - Estudios
    - Qué grados hay
    - Qué masters hay
    - Doctorados
    - Otra educación
  - Estudiantes novatos
    - Primeros pasos
      - TUI

- Correo electrónico
  - Cuenta de usuario
  - Mi portal
  - Cursos cero
    - Cursos cero informática/químicas/industriales
    - Cursos cero informática y ADE
    - Cursos cero sistemas de la información
  - Panorama universitario
  - Departamentos
  - Asignaturas grado y master
  - Horarios
  - Calendario de actividades
  - Estudiantes veteranos
    - Fechas exámenes
    - Prácticas – enlaza con Prácticas en trámites administrativos
    - Erasmus
    - TFG/TFM
    - Becas
    - Delegación
    - Grupos investigación
  - Servicios
    - Cafetería
    - Biblioteca
    - Reprografía
- Trámites administrativos
  - Matrícula
    - Estudiantes nuevos/veteranos
  - Calendario académico
  - Fechas exámenes
  - Solicitudes
    - Solicitudes dirección
    - Solicitudes pdi/pas/delegaciones
    - Solicitudes e impresos
  - Becas y ayudas
  - Pruebas de acceso
    - Evau
    - > 25
    - > 40
    - > 45
  - Prácticas
- Cambios de grupo
  - Cambio de grupo grande
    - Solicitud de compensación
    - Solicitud genérica al director
    - Solicitud evaluación final
- Convalidaciones
  - Reconocimiento créditos
  - Acreditación nivel idioma

- Traslado/reapertura expediente
- Solicitud de certificado y títulos
- Otras solicitudes
- Wifi y VPN
  - Red privada virtual
  - Correo electrónico
  - Cuentas y contraseñas de usuario
  - Acceso wifi
- Localizaciones
  - Dónde está
  - Cómo moverme
  - Departamentos
  - Delegación
  - Laboratorios

Para poder comenzar la conversación, la web de *SAP Conversational AI* proporciona una herramienta que permite poner un mensaje de bienvenida en el chatbot a la hora de enlazarlo a una página web, tal y como se puede observar en la figura 18. Este mensaje solo se podrá ver en la página web donde se va a alojar, no en la vista previa del chatbot.

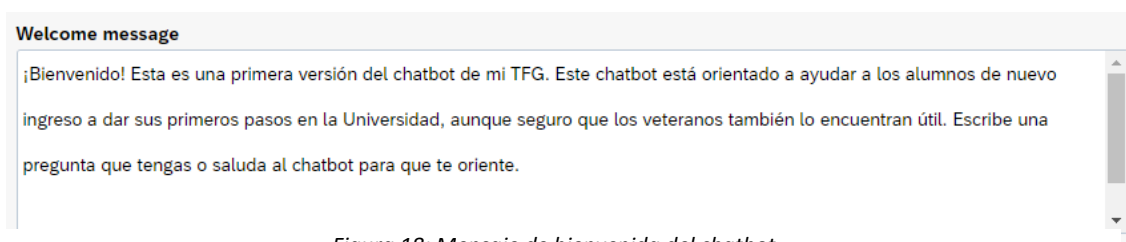


Figura 18: Mensaje de bienvenida del chatbot

A pesar de esta característica, el usuario siempre va a tener que empezar la conversación con el chatbot. Podrá elegir si preguntar cualquier cosa relacionada con la lista anterior, lo cual iniciará la conversación directamente. Pero si se quiere seguir al dinámica de botones, el usuario tendrá que saludar al chatbot con cualquier mensaje de bienvenida (hola, que tal, quien eres...).

Si el usuario decide seguir la dinámica de botones y saludar al chatbot, este le responderá con un segundo mensaje de bienvenida, donde se enlazarán con los diversos botones a todas las áreas mencionadas antes.

Dado que el chatbot en formato web solo puede soportar 3 botones por cada interacción, se tiene que dividir la respuesta en varias secciones. En las figuras 19 se puede ver el funcionamiento de esta dinámica mediante la vista previa que proporciona el chatbot.

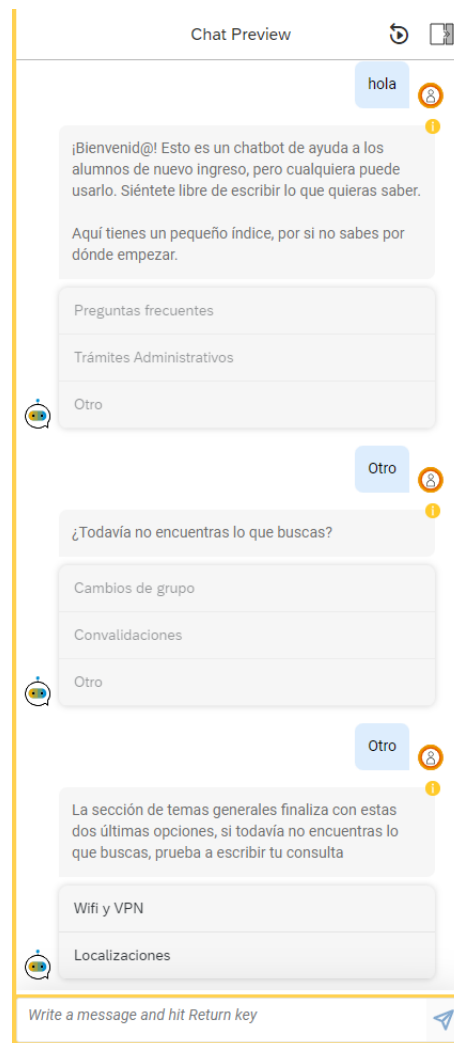


Figura 19: Vista previa en SAP Conversational AI con mensaje de bienvenida

Dentro de cada apartado del índice (de cada botón), el chatbot nos redirigirá a los subíndices que cada apartado tenga asignados. De esta forma, si se quiere llegar a la sección “Fechas de exámenes”, el usuario tendrá que pulsar en preguntas frecuentes, buscar el apartado de estudiantes veteranos y posteriormente el apartado de fechas de exámenes.

De nuevo, el árbol de apartados se puede encontrar en la lista de apartados del chatbot.

# CAPÍTULO 2

## Creación web y enlace con el chatbot

Para esta fase, se necesitará tener Java 17 y *Maven* 3.9.2 instalado en el ordenador. El proyecto se va a crear en el IDE *IntelliJ* en su versión *Community*.

Cabe mencionar que para este trabajo, se va a crear una página web desde cero para poder adaptarla a las necesidades del TFG. El chatbot se puede añadir a cualquier página web, siempre y cuando se tengan los archivos de desarrollo necesarios para la modificación de la misma.

Procedemos con la creación del proyecto *Maven* con el *jdk* de Java 17.

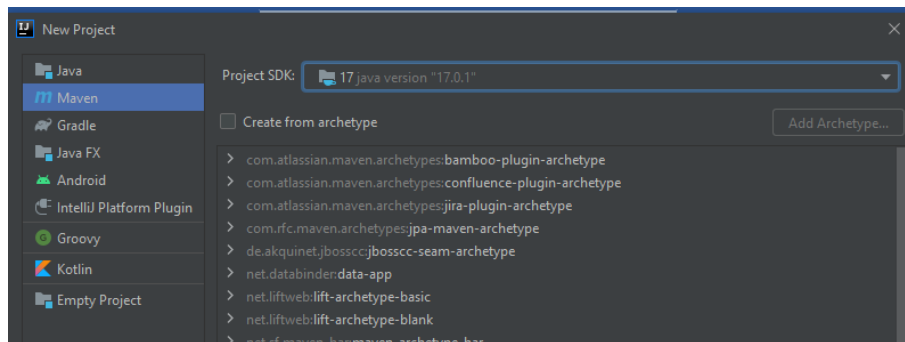


Figura 20: Creación proyecto Maven

La web se va a programar con *.html*. Para ello, el primer paso es crear en la carpeta principal de java una vez iniciado el proyecto, un nuevo archivo llamado *"index.html"*. En este fichero se programará la interfaz de la página web y se realizará la configuración del chatbot.

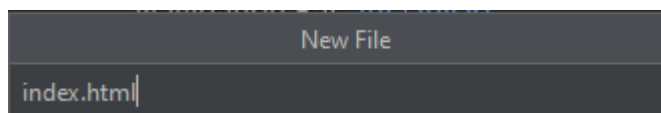


Figura 21: Creación archivo index.html

Para proceder con dicha configuración, y de regreso en la web de *SAP Conversational AI* donde se ha programado el chatbot de la fase 1, nos dirigimos al apartado *"Connect"*, que se puede observar en la figura 22.

Para una primera conexión, se va a dejar el diseño por defecto que nos proporciona la herramienta, únicamente cambiando el nombre del chatbot, el mensaje de bienvenida y los mensajes emergentes que saldrán del bot

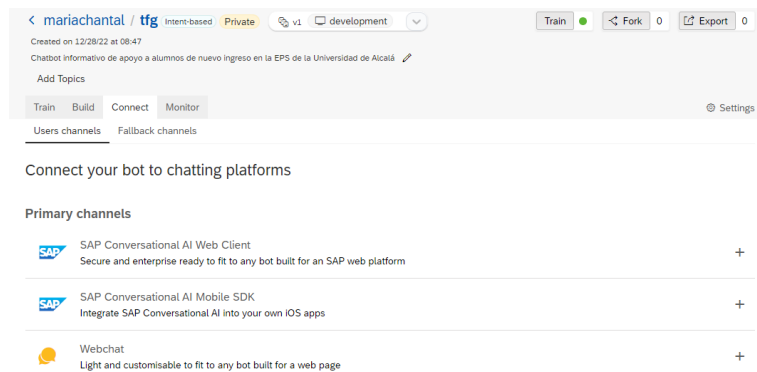


Figura 22: Apartado de conexión en SAP Conversational AI

Esta configuración comenzará en la sección “WebChat”. En la figura 23 se puede observar esta interacción.

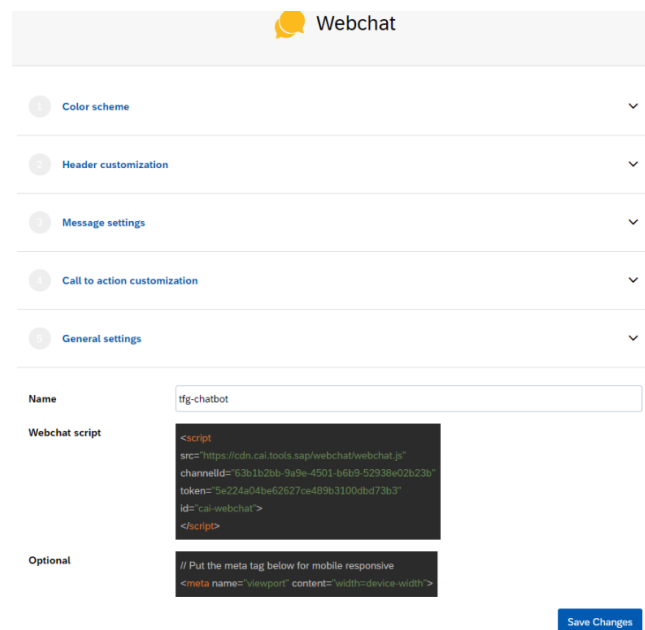


Figura 23: Script del chatbot para su incorporación en la web

Aquí se nos ofrecen diversas opciones de personalización, desde los colores hasta las imágenes de perfil tanto del chatbot como del usuario. De la sección “Message settings” (configuración de mensajes), simplemente se va a modificar, como se ha mencionado antes, el

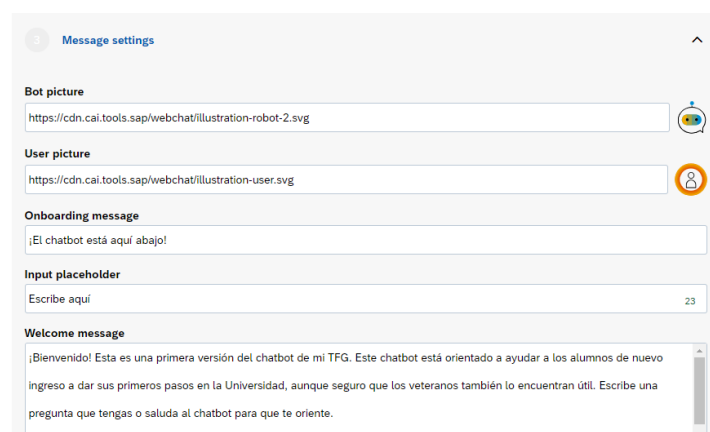


Figura 24: Configuración inicial del chatbot para la web

mensaje de bienvenida genérico (figura 24). Más adelante se volverá a este apartado para perfeccionar el diseño del chatbot

De nuevo viendo la figura 23, se puede observar el *script* que será necesario introducir en el fichero *html* previamente declarado. En la figura 25 se observa una programación básica de lo que va a mostrar la página web, con el título, un pequeño encabezado y, de las líneas 32 a la 37, el script del chatbot. Finalmente, el resultado de la programación inicial de la web con el chatbot incorporado sería el de las figuras 26 y 27.

```
1 <html>
2 <head>
3   <title> Maria Chantal TFG </title>
4   <style type="text/css">
5     * {
6       padding:0px;
7       margin:0px;
8     }
9     #header {
10      margin:auto;
11      width:500px;
12    }
13    ul, ol {
14      list-style:none;
15    }
16    .nav li a {
17      background-color:rgb(132, 15, 228);
18      color: #fff;
19      text-decoration:none;
20      padding: 10px 15 px
21    }
22    .nav > li {
23      float:left
24    }
25  </style>
26 </head>
27 <body>
28   <h1 align="center"> Web de prueba en blanco el chatbot del TFG </h1>
29   <hr>
30   </div>
31   <script
32     src="https://cdn.cai.tools.sap/webchat/webchat.js"
33     channelId="63b1b2bb-9a9e-4501-b6b9-52938e02b23b"
34     token="5e224a04bec2627ce489b3100dbd73b3"
35     id="cai-webchat">
36   </script>
37   <hr>
38   </body>
39 </html>
```

Figura 25: Programación inicial para alojar el chatbot

## Web de prueba en blanco el chatbot del TFG

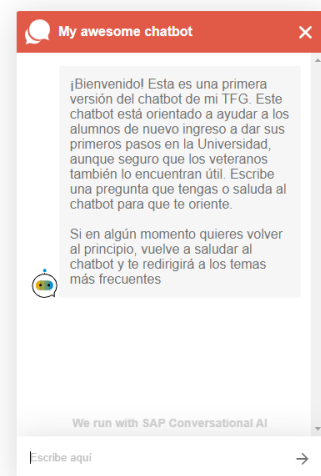


Figura 26: Vista previa chatbot abierto en web inicial



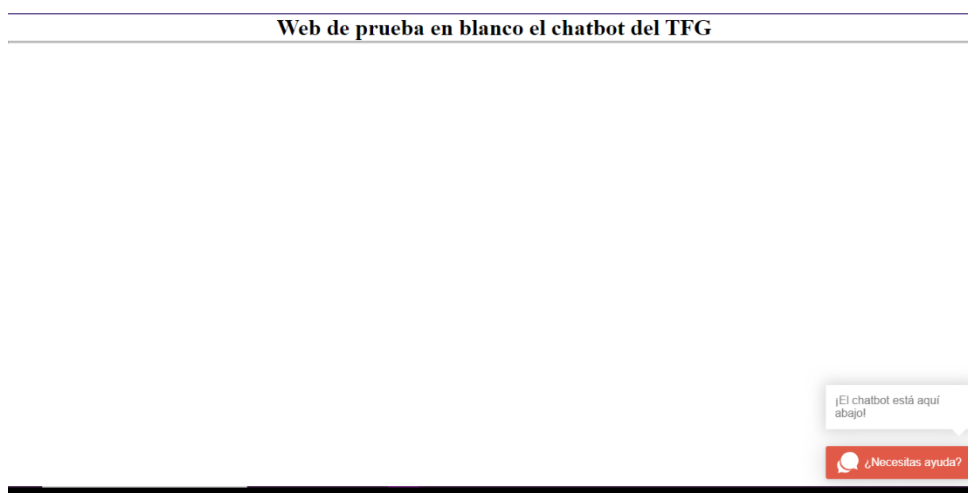


Figura 27: Vista previa chatbot cerrad en web inicial

Para comprobar el funcionamiento del chatbot, en la figura 28 se va a inciar una conversación con un saludo para comprobar la dinámica que se seguiría para una conversación con botones, y en la figura 29 se le va a lanzar la misma pregunta que en el capítulo 1.

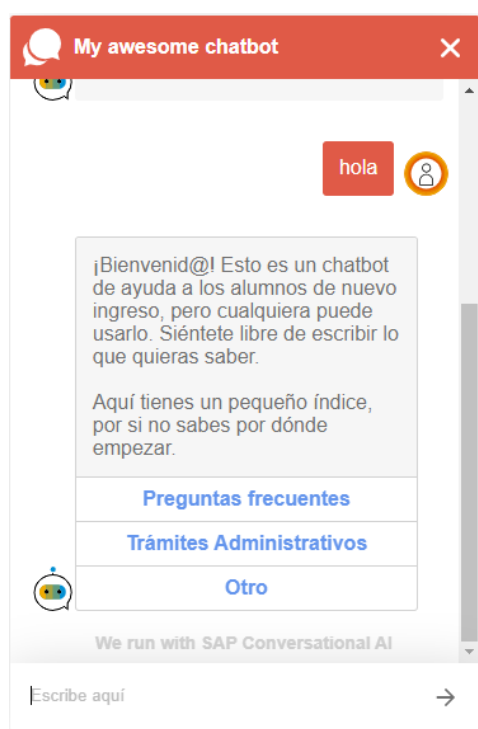


Figura 28: Saludo al chatbot

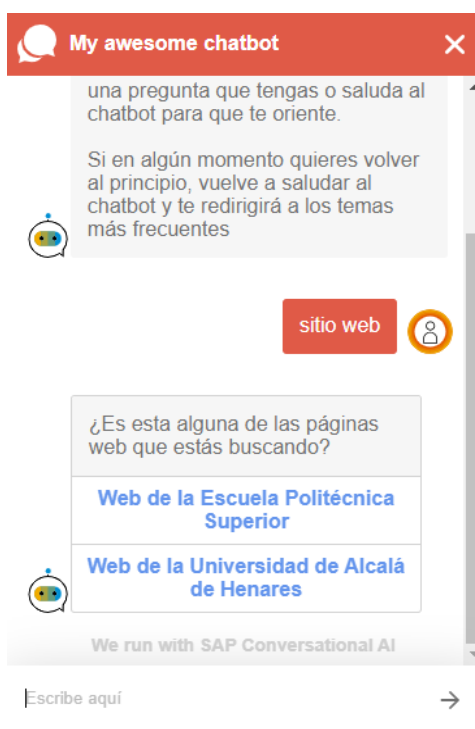


Figura 29: Pregunta al chatbot

# CAPÍTULO 3

## Configuración herramientas de prueba

### Variables de entorno

Para comenzar con la configuración de todas las herramientas necesarias para poder trabajar con *Selenium Web Driver* y con *Cucumber*, el primer paso es configurar las variables de entorno de Java 17 y de Maven 3.9.2

Para este paso es necesario descargar Java 17 [7]. El siguiente paso será acceder a las variables de entorno desde el panel de control.

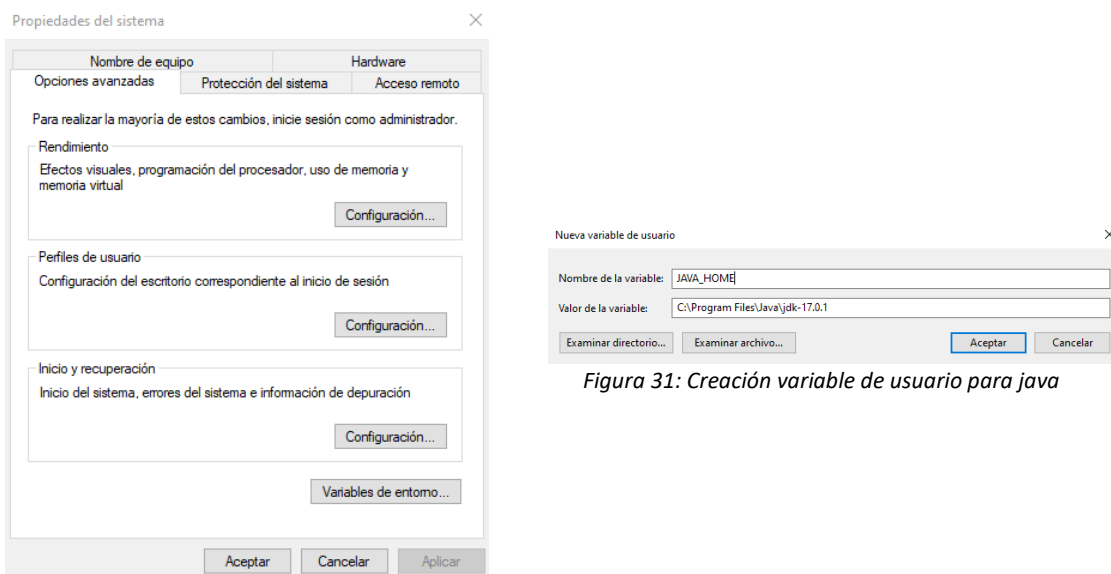


Figura 31: Creación variable de usuario para java

Figura 30: Propiedades del sistema Windows 10

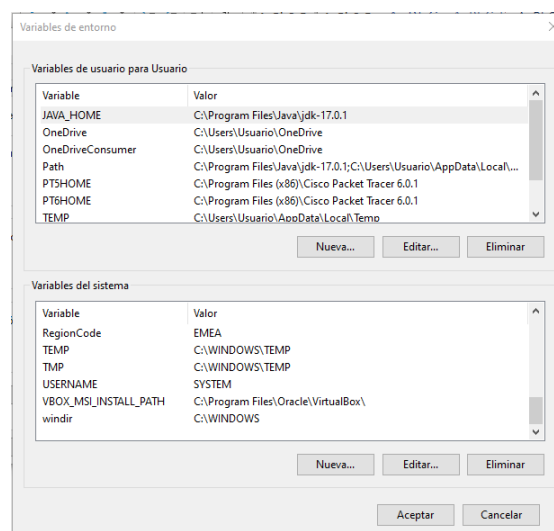
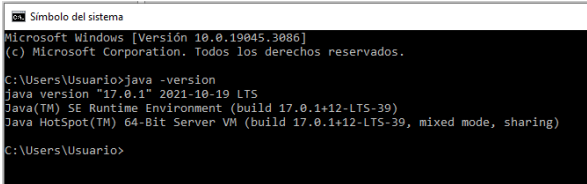


Figura 32: Vista final variables de entorno

Al hacer clic en variables de entorno, aparecerán las variables de usuario y de sistema. Para configurar Java 17, será necesario crear una nueva variable de usuario.

Para comprobar que esta instalación se ha hecho de forma correcta, desde la terminal de Windows, al escribir el comando “*java -version*”, se puede verificar la versión de java actual instalada en el sistema.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.3086]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)

C:\Users\Usuario>
```

Figura 33: Comprobación versión de java

Para proceder, será necesario descargar *Maven* [8]

La web nos mostrará distintas opciones de descarga. Para este proyecto enfocado en Windows, habrá que descargarse el archivo “*apache-maven-3.9.2-bin.zip*”

**Files**

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	<a href="#">apache-maven-3.9.2-bin.tar.gz</a>	<a href="#">apache-maven-3.9.2-bin.tar.gz.sha512</a>	<a href="#">apache-maven-3.9.2-bin.tar.gz.asc</a>
Binary zip archive	<a href="#">apache-maven-3.9.2-bin.zip</a>	<a href="#">apache-maven-3.9.2-bin.zip.sha512</a>	<a href="#">apache-maven-3.9.2-bin.zip.asc</a>
Source tar.gz archive	<a href="#">apache-maven-3.9.2-src.tar.gz</a>	<a href="#">apache-maven-3.9.2-src.tar.gz.sha512</a>	<a href="#">apache-maven-3.9.2-src.tar.gz.asc</a>
Source zip archive	<a href="#">apache-maven-3.9.2-src.zip</a>	<a href="#">apache-maven-3.9.2-src.zip.sha512</a>	<a href="#">apache-maven-3.9.2-src.zip.asc</a>

Figura 34: Vista de descarga de Maven

Se descomprime el archivo en el disco C del ordenador y se añade la ruta a las variables del sistema, al igual que se hizo con Java.

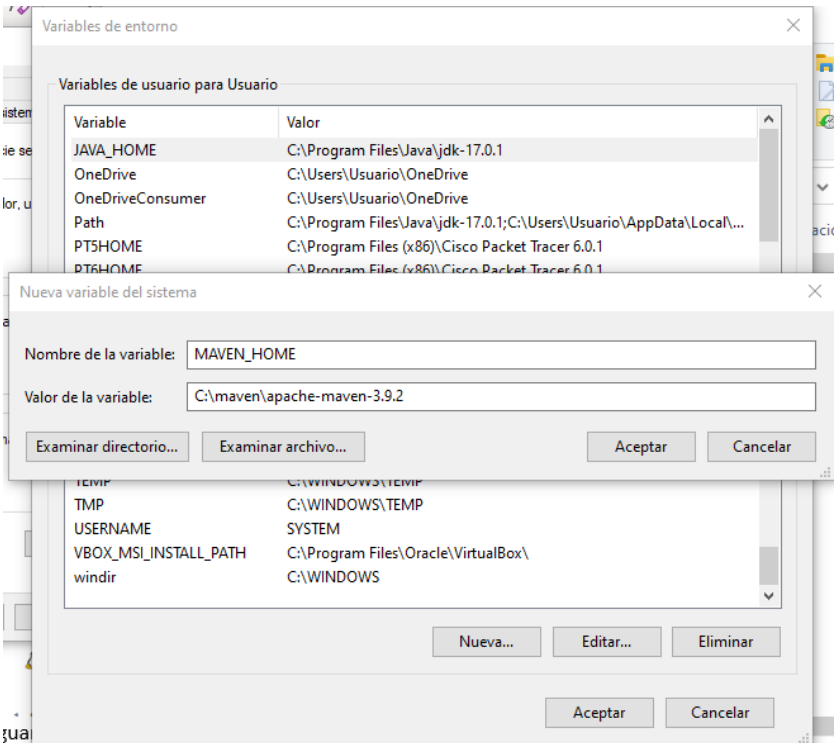
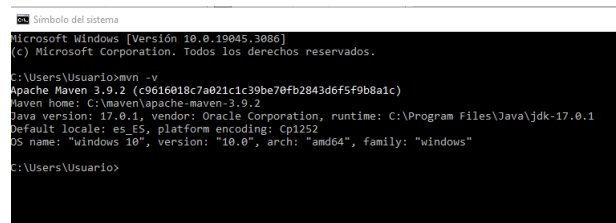


Figura 35: Creación variable de usuario para Maven

Y de nuevo, se comprueba que la instalación ha sido exitosa con el comando “mvn -v” en la terminal.



```
Microsoft Windows [Versión 10.0.19045.3886]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>mvn -v
Apache Maven 3.9.2 (c9616018c7a021c1c39be70fb2843d6f5f9b8a1c)
Maven home: C:\maven\apache-maven-3.9.2
Java version: 17.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17.0.1
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

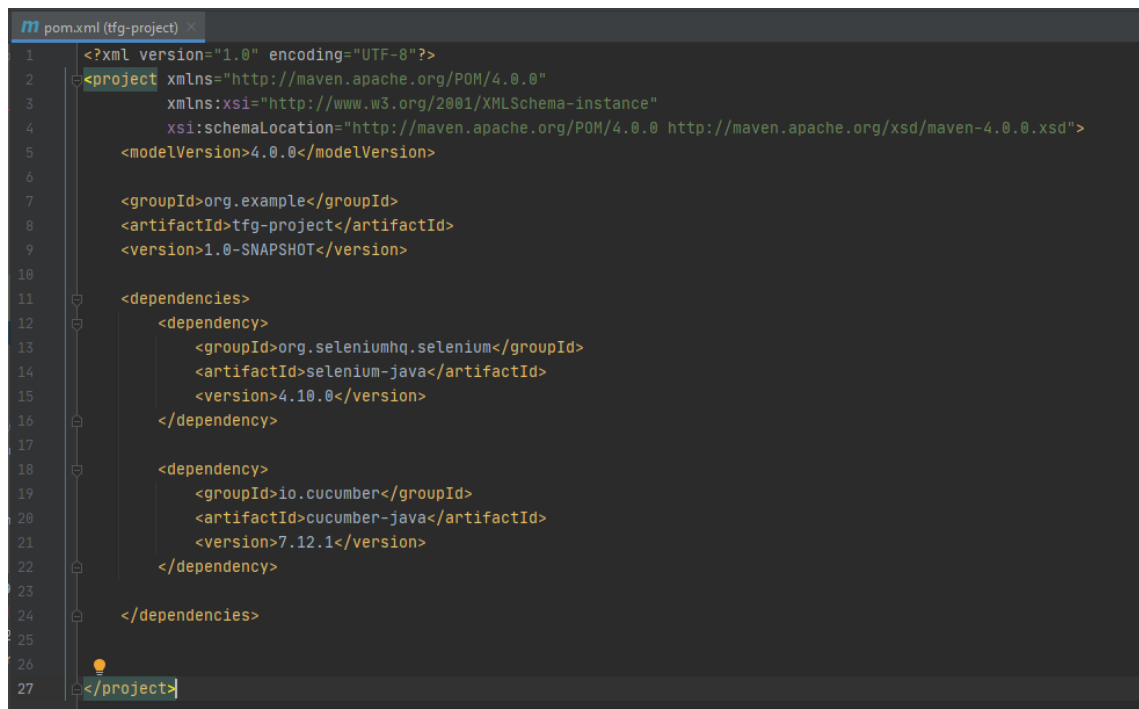
C:\Users\Usuario>
```

Figura 36: Comprobación versión de Maven

## Dependencias de Selenium Web Driver y de Cucumber

*Selenium Web Driver* y *Cucumber* no tiene que ser instaladas directamente en el sistema. Al contrario que Java y Maven, estas dos herramientas se instalarán directamente en el proyecto de IntelliJ que se ha creado anteriormente mediante dependencias.

Las dependencias de *Selenium* [9] y de *Cucumber* [10] se instalarán en la clase “pom.xml”, y es tan sencillo como se muestra en la figura 37.



```
m pom.xml (tfg-project)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>org.example</groupId>
8     <artifactId>tfg-project</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <dependencies>
12         <dependency>
13             <groupId>org.seleniumhq.selenium</groupId>
14             <artifactId>selenium-java</artifactId>
15             <version>4.10.0</version>
16         </dependency>
17
18         <dependency>
19             <groupId>io.cucumber</groupId>
20             <artifactId>cucumber-java</artifactId>
21             <version>7.12.1</version>
22         </dependency>
23     </dependencies>
24
25
26 </project>
```

Figura 37: Instalación de dependencias en pom.xml

*Cucumber* tiene una interfaz un tanto característica, por lo que para facilitar la interacción del usuario con la herramienta, es recomendable instalar los plugin de *Gherkin* y de *Cucumber for Java* en IntelliJ.

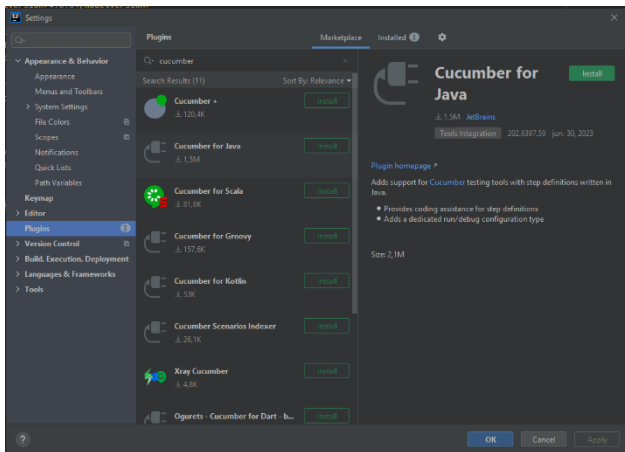


Figura 38: Plugin Cucumber

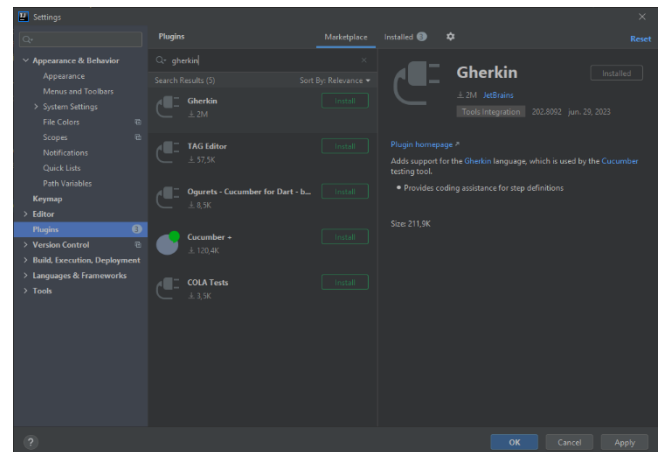


Figura 39: Plugin Gherkin

*Gherkin* es el lenguaje que se va a usar para poder comprender *Cucumber*. Con él, se podrán definir los escenarios y las funcionalidades de forma muy comprensible para el usuario.

*Cucumber* es una de las herramientas que se pueden usar para automatizar las pruebas.

## Creación de las pruebas automatizadas

Antes de comenzar, es necesario aclarar qué son las pruebas automatizadas y cómo vamos a trabajar con ellas.

A pesar de que su nombre pueda hacer pensar que todo es un proceso automático, estas pruebas necesitan de la interacción humana para garantizar su funcionamiento. El programador deberá saber cuál es el resultado deseado de la web, aplicación o, en este caso, del chatbot. Si el programador sabe que al hacer clic en cierto botón se debe redirigir al usuario a cierta web, esto se va a plasmar en las pruebas. Para ello, va a ser necesario acceder a las herramientas de desarrollador de la web o de la aplicación en la que estemos trabajando, el proceso se explicará más adelante.

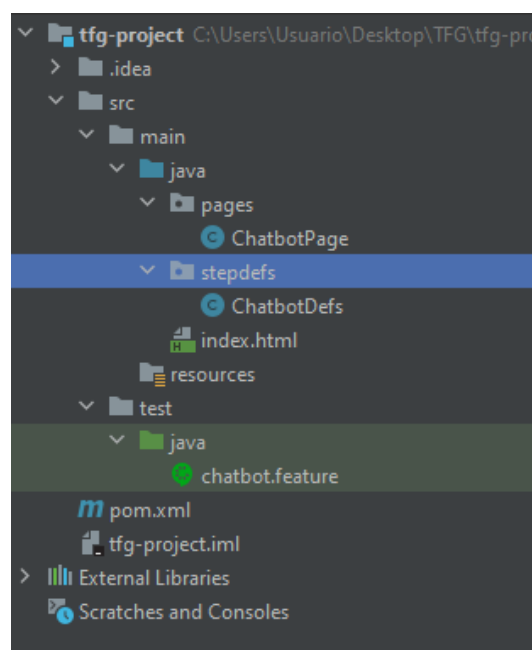


Figura 40: Estructura inicial del proyecto

Una vez hecha la configuración de las herramientas, se deben crear tres clases para poder programar el funcionamiento de las pruebas, siguiendo la estructura de la figura 40:

- La clase “ChatbotPage” servirá como clase para poder almacenar los identificadores de los elementos de la web para poder hacer las llamadas en las pruebas automatizadas.
- La clase “ChatbotDefs” será en la que se programe la funcionalidad de las pruebas.

Finalmente, la clase “chatbot.feature” será la clase *Cucumber* que se encargará de definir los escenarios de cada uno de los casos de prueba que se quieran llevar a cabo. Para ver un ejemplo práctico de cómo funcionan los escenarios en *Cucumber*, podemos observar la figura 41.

```
Feature: Maria Chantal TFG
  Scenario: First greetings
    Given I click on the bot
    When I write 'hola'
    Then the bot gives me a welcome greeting
```

Figura 41: Escenario Cucumber: first greetings

Una clase en *Cucumber* siempre va a llevar la extensión *.feature* como distintivo. Estas clases funcionan mediante la configuración de distintos escenarios. Cada escenario es un conjunto de instrucciones que seguirán cierta lógica para poder ejecutar las pruebas. Cada escenario, va a tener las palabras reservadas *Given*, *When*, *Then* y *And*. Cada una de ellas se utilizará para saber en qué orden se va a ejecutar cada paso del escenario. Se puede traducir a español directamente para aclarar su funcionamiento. Con base en la imagen de arriba, las sentencias serían “dado que hago clic en el bot, cuando escriba hola, entonces el bot me va a dar un mensaje de bienvenida”. Estas palabras reservadas se pueden intercalar y son únicamente un recurso para que el usuario que programa las pruebas tenga una mejor perspectiva de lo que está haciendo.

```
chatbot.feature
1  Feature: Maria Chantal TFG
2
3  Background:
4    Given I am in the main page
5    Then I click on the bot
6
7  Scenario: First greetings
8    When I write 'hola'
9    Then the bot gives me a welcome greeting
```

Figura 42: Escenario Cucumber: first greetings y Background

Se pueden programar tantos escenarios como se deseen. Por ello, también una opción llamada “Background”, cuya función es ejecutar las sentencias que estén en su interior siempre antes de cada escenario. Todos los pasos que estén dentro del *background*, se ejecutarán con todos los escenarios, independientemente de si se lanza uno de forma individual o todos los que agrupe la clase *.feature* en la que esté declarado.

Estos escenarios por su cuenta no son capaces de ejecutar ningún tipo de prueba. Para completar su funcionalidad, es necesario comenzar con la parte de *Selenium*.

*Selenium Web Driver* es la herramienta que nos va a permitir acceder a los elementos de la web que servirán como guía para saber si el funcionamiento del Chatbot es el esperado. Estas pruebas, al ser configuradas de forma manual, tienen el precedente de que el programador debe analizar cuál es el resultado deseado. Aquí es donde entran en juego las herramientas de desarrollador de, en este caso, Google Chrome.

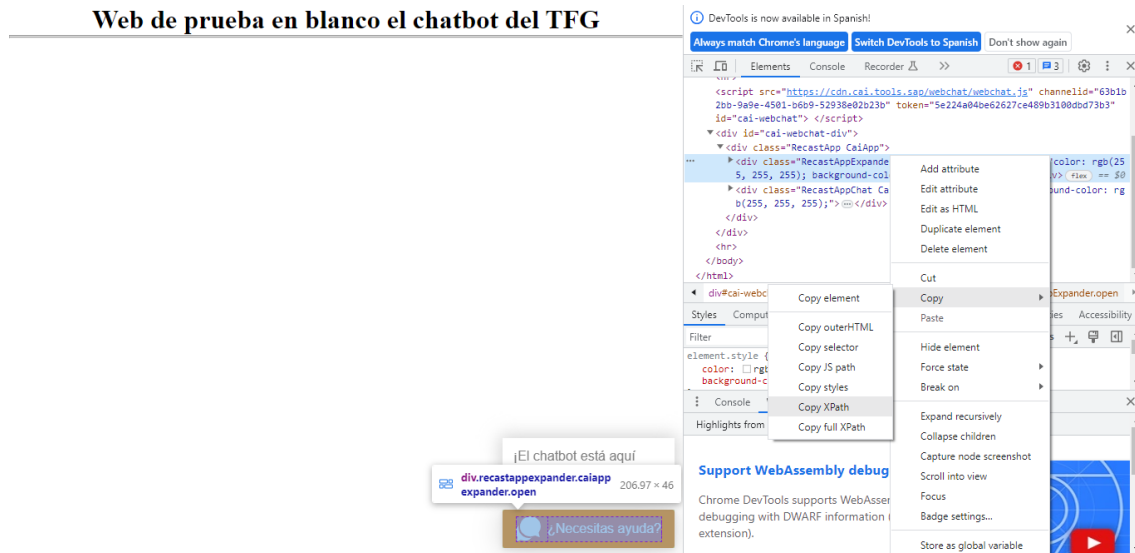


Figura 43: Inspeccionar elemento en el navegador Google Chrome

En la web que se ha creado para almacenar el chatbot, haremos clic derecho sobre el botón que abre el bot. Después de hacer clic en la sección “inspeccionar”, nos aparecerán las herramientas de desarrollador tal y como se aprecia en la figura 43. Se marcará en azul el elemento que hemos seleccionado, y de nuevo con el clic derecho, iremos hasta el desplegable “copy” para poder copiar la dirección *xpath*.

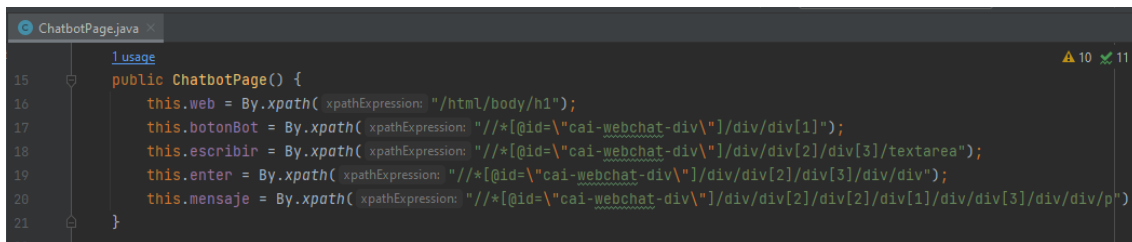
Es importante recalcar que para la configuración de pruebas del chatbot, se van a usar en su mayoría identificadores *xpath* para poder jugar con las secuencias numéricas.

Será necesario hacer esto con todos los elementos que se deseen incluir en las pruebas. En este caso, para el escenario de prueba que se ha visto previamente en la figura 42, será necesario el botón de abrir el chatbot, hacer clic en el panel de texto, escribir la pregunta deseada, hacer clic en el botón de enviar, y comprobar que el mensaje que recibimos es el indicado. Cada uno de estos pasos tiene su propio elemento asociado. Cogiendo el *xpath* de cada uno de ellos, se comienza a programar la clase *ChatbotPage*.

```
ChatbotPage.java
7 public class ChatbotPage{
8     private ChromeDriver driver;
9     private final By web ;
10    private final By botonBot ;
11    private final By escribir ;
12    private final By enter ;
13    private final By mensaje ;
```

Figura 44: Definición variables

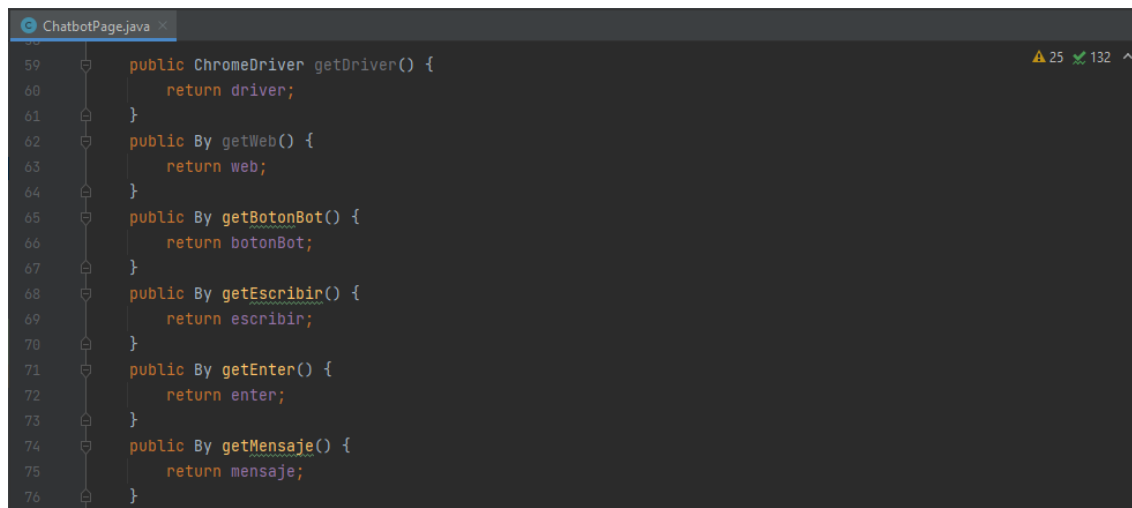
En primer lugar, se declara el nombre de las variables que se van a utilizar. Serán de tipo “By”, dado que *Selenium* utiliza este tipo de datos para recoger los elementos de identificación de la web. También es necesario declarar el driver del navegador que se vaya a emplear para realizar las pruebas, como se ha mencionado anteriormente, en este caso se usará Google Chrome.



```
15 public ChatbotPage() {
16     this.web = By.xpath(xpathExpression: "/html/body/h1");
17     this.botonBot = By.xpath(xpathExpression: "//*[@id='cai-webchat-div']/div/div[1]");
18     this.escribir = By.xpath(xpathExpression: "//*[@id='cai-webchat-div']/div/div[2]/div[3]/textarea");
19     this.enter = By.xpath(xpathExpression: "//*[@id='cai-webchat-div']/div/div[2]/div[3]/div/div");
20     this.mensaje = By.xpath(xpathExpression: "//*[@id='cai-webchat-div']/div/div[2]/div[2]/div[1]/div/div[3]/div/div/p");
21 }
```

Figura 45: Asignación del identificador a la variable

A continuación, dentro del constructor, se van a inicializar estas variables a los valores *xpath* que se han recogido anteriormente del inspector de elementos de la web.



```
59 public WebDriver getDriver() {
60     return driver;
61 }
62 public By getWeb() {
63     return web;
64 }
65 public By getBotonBot() {
66     return botonBot;
67 }
68 public By getEscribir() {
69     return escribir;
70 }
71 public By getEnter() {
72     return enter;
73 }
74 public By getMensaje() {
75     return mensaje;
76 }
```

Figura 46: Creación métodos getter

Finalmente, para poder utilizar estas variables fuera de esta clase, se crearán métodos *get* para poder recuperar los valores.

Una vez recogidos los identificadores de cada elemento, se pueden programar los escenarios en la clase *ChatbotDefs.java*.

Para poder definir cada paso de cada escenario previamente declarados en la clase *.feature*, se tienen que crear métodos que llamen a estos pasos de forma individual. La forma de hacer esto es mediante etiquetas. Las etiquetas de *Cucumber* permiten especificar el tipo de palabra reservada que se ha empleado para cada paso (*Given*, *When*, *Then* y *And*). Junto con esta etiqueta, debe ir el nombre exacto que se le haya dado al paso específico dentro, con la estructura: `@When("^I say hola to the chatbot$")`. De esta forma, se realizará la llamada al paso con ese nombre.



```

@Given("^I am in the main page$")
public void mainpage() {
    System.setProperty("webdriver.chrome.driver",
        "C:\\Users\\Usuario\\Desktop\\TF6\\tfg-project\\src\\main\\resources\\drivers\\chromedriver.exe");
    this.driver = new ChromeDriver();
    driver.get("http://www.mariachantaltfg.es/");
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait(time: 10, TimeUnit.SECONDS);
    this.chp = new ChatbotPage();
}

@Then("^I click on the bot$")
public void clickonbot() throws InterruptedException {
    driver.manage().timeouts().implicitlyWait(time: 2, TimeUnit.SECONDS);
    driver.findElement(chp.getBotonBot()).click();
}

@When("^I say hola to the chatbot$")
public void iSayHi(){
    driver.findElement(chp.getEscribir()).click();
    driver.findElement(chp.getEscribir()).sendKeys(...keysToSend: "Hola");
    driver.findElement(chp.getEnter()).click();
}

@Then("^the bot gives me a welcome greeting$")
public void welcomeGreeting(){
    driver.findElement(chp.getMensaje()).toString().substring(0, 31)
        .equals(("¡Bienvenid@! Esto es un chatbot"));
}

```

Figura 47: Definición métodos en Selenium del escenario de Cucumber

Estos son los cuatro pasos que forman el “background” y el escenario de prueba diseñado para ejemplificar el funcionamiento de *Cucumber* y *Selenium*. La estructura general sería crear un método con la etiqueta del nombre del paso correspondiente.

En el primer método *mainpage()*, se puede observar la construcción del driver que va a ejecutar las pruebas. Se le debe pasar la ruta donde está guardado el driver del navegador deseado, e inicializarlo. Se le dice al driver que se ponga a pantalla completa y se le asigna una espera implícita para garantizar que la página ha cargado de forma satisfactoria.

En el siguiente método, se hará clic en el botón del chatbot para abrirlo. Llamando al driver, se hará clic en el botón declarado mediante el *xpath* en la clase *ChatbotPage*. Este método y el anterior se ejecutarán siempre, dado que han sido configurados como *background*.

El siguiente “When” ya es específico del escenario que estamos construyendo. En primer lugar, será necesario hacer clic en el recuadro de escritura del chatbot, para posteriormente escribir el mensaje deseado y luego enviarlo, pulsando nuevamente un botón.

Para finalizar, comprobaremos que el mensaje que nos devuelve el chatbot es el que hemos configurado previamente. Para ello (y para no meter todo el mensaje), se selecciona una cantidad corta de caracteres y se comprueba que esos caracteres son iguales que lo que está recibiendo el driver como respuesta.

Con esto, se ha realizado un primer escenario básico de una prueba automatizada del chatbot de ayuda a los estudiantes de nuevo ingreso.

## Casos de prueba

Como se explicó previamente en la Fase 1, en el apartado de “Funcionamiento del chatbot”, la interacción con el chatbot se dividirá en dos secciones, una interacción escrita y una interacción con botones. Se puede ver en la siguiente lista los casos de prueba seleccionados para cada tipo de interacción. Subrayado en morado, los casos de interacción escrita, y subrayado en amarillo, los casos de interacción con botones.

- Preguntas frecuentes
  - Información general
    - Webs
    - Qué ofrece
    - Ramas conocimiento – enlazado con estudios
    - Normativas
    - Coordinadores
  - Estudios
    - Qué grados hay
    - Qué masters hay
    - Doctorados
    - Otra educación
  - Estudiantes novatos
    - Primeros pasos
      - TUI
      - Correo electrónico – enlazado con Wifi y VPN
      - Cuenta de usuario
      - Mi portal
    - Cursos cero
      - Cursos cero informática/químicas/industriales
      - Cursos cero informática y ADE
      - Cursos cero sistemas de la información
    - Panorama universitario
    - Departamentos
    - Asignaturas grado y master
    - Horarios
    - Calendario de actividades
  - Estudiantes veteranos
    - Fechas exámenes – enlazado con trámites administrativos
    - Prácticas – enlaza con Prácticas en trámites administrativos
    - Erasmus
    - TFG/TFM
    - Becas – enlazado con trámites administrativos
    - Delegación – enlazado con localizaciones
    - Grupos investigación
  - Servicios
    - Cafetería
    - Biblioteca
    - Reprografía
- Trámites administrativos

- Matrícula
  - Estudiantes nuevos/veteranos
- Calendario académico
- Fechas exámenes
- Solicitudes
  - Solicitudes dirección
  - Solicitudes pdi/pas/delegaciones
  - Solicitudes e impresos
- Becas y ayudas
- Pruebas de acceso
  - Evau
  - > 25
  - > 40
  - > 45
- Prácticas
- Cambios de grupo
  - Cambio de grupo grande – enlazado con solicitudes a dirección
  - Solicitud de compensación – enlazado con solicitudes a dirección
  - Solicitud genérica al director – enlazado con solicitudes a dirección
  - Solicitud evaluación final – enlazado con solicitudes a dirección
- Convalidaciones
  - Reconocimiento créditos
  - Acreditación nivel idioma
  - Traslado/reapertura expediente
  - Solicitud de certificado y títulos
  - Otras solicitudes
- Wifi y VPN
  - Red privada virtual
  - Correo electrónico – enlaza con preguntas frecuentes
  - Cuentas y contraseñas de usuario
  - Acceso wifi
- Localizaciones
  - Dónde está
  - Cómo moverme
  - Departamentos
  - Delegación
  - Laboratorios

## Escenarios para los casos de prueba

Anteriormente se ha explicado el funcionamiento de un escenario básico de *Cucumber* y su configuración. Para poder optimizar la creación de escenarios para los casos de prueba, *Cucumber* proporciona otro tipo de escenarios, llamados “Scenarios Outline”. Este tipo de escenarios permitirá hacer iteraciones entre diversos ejemplos que se le proporcionen.

Dejando el “Background” que se ha configurado para el ejemplo, procedemos a crear el “Scenario Outline” para la interacción escrita con el chatbot

```
Scenario Outline: seleccion escrita
  Then I write <seleccion>
  And I receive the <seleccion> response
  Examples:
  |seleccion|
  |que ofrece|
  |masters|
  |doctorados|
  |TUI|
  |cursos cero|
  |departamentos|
  |erasmus|
  |delegacion|
  |cafeteria|
  |reprografia|
  |matricula|
  |becas|
  |fechas examenes|
  |cambio de grupo grande|
  |wifi|
  |donde esta|
  |laboratorios|
```

Figura 48: Escenario Cucumber: selección escrita

La principal diferencia que se observa en relación a un escenario simple, es la existencia de ejemplos. Estos ejemplos funcionan como un bucle *for*: en las sentencias de los pasos a realizar (*Then* y *And* en este caso) se puede observar que se está llamando a una variable <selección>. Esta variable irá llamando a cada ejemplo de forma escalonada, de forma que en primer lugar se ejecutará todo el escenario para el ejemplo “que ofrece”, después, para el ejemplo “masters”, y así hasta llegar al ejemplo “laboratorios”.

La programación de estos ejemplos se ve de forma muy intuitiva si se emplea un sistema *switch-case* a la hora de programar cada paso, pues así se podrá ir iterando entre cada ejemplo de forma muy visual.

Para poder llamar a estas variables desde la programación de los pasos, habrá que hacer referencia a la variable <selección>. Esta llamada se puede realizar escribiendo los caracteres reservados ([^\\"]\*) en el lugar donde iría esta variable en los pasos de *Cucumber*, tal y como se muestra en la figura 48. Este método hará que el chatbot escriba la frase que se ve reflejada en la figura anterior y que la envíe haciendo clic en el botón *enter*

```

@Then("^I write ([^\\"]*)$")
public void queOfrece(String sel){
    waitMethod();
    driver.findElement(chp.getEscribir()).click();
    switch (sel) {
        case "que ofrece":
            driver.findElement(chp.getEscribir()).sendKeys( ...keysToSend: "quiero saber que ofrece la eps");
            driver.findElement(chp.getEnter()).click();
            break;
        case "masters":
            driver.findElement(chp.getEscribir()).sendKeys( ...keysToSend: "por favor dime el listado de masters");
            driver.findElement(chp.getEnter()).click();
            break;
        case "doctorados":
            driver.findElement(chp.getEscribir()).sendKeys( ...keysToSend: "quiero cursar un doctorado");
            driver.findElement(chp.getEnter()).click();
            break;
        case "cursos cero":
            driver.findElement(chp.getEscribir()).sendKeys( ...keysToSend: "Hay algun curso de introduccion?");
            driver.findElement(chp.getEnter()).click();
            break;
        case "departamentos":
            driver.findElement(chp.getEscribir()).sendKeys( ...keysToSend: "departamento de ciencias de la computacion");
            driver.findElement(chp.getEnter()).click();
            break;
    }
}

```

Figura 49: Método Selenium para el escenario selección escrita

En la figura 49 se puede observar un funcionamiento similar, poniendo los caracteres reservados ([^\\"]\*) para hacer referencia a la variable selección que se quiere ir sustituyendo. La diferencia con este método es que este será el encargado de recuperar la respuesta del bot y de asegurarse de que la contestación prevista concuerda con lo que se está recibiendo en realidad

Siguiendo con la interacción mediante botones, aquí se podrán encontrar dos “Scenario Outline” y dos escenarios normales.

```

Scenario Outline: tramites administrativos
When I say hola to the chatbot
Then I click on the tramites administrativos button
And I search until I find my destiny for tramites in <seleccion>
Then I go to the <seleccion> web for tramites
Examples:
|seleccion|
|calendario academico|
|solicitudes|
|evau|
|mas25|
|mas40|

Scenario Outline: convalidaciones
When I say hola to the chatbot
Then I click on the convalidaciones button
And I search until I find my destiny for convalidaciones in <seleccion>
Then I go to the <seleccion> web for convalidaciones
Examples:
|seleccion|
|reconocimiento creditos|
|nivel idioma|
|traslado expediente|

```

Figura 50: Escenarios Cucumber: trámites administrativos y convalidaciones

Se van a dividir por secciones dentro de los casos de prueba. En la figura 50 se puede observar que, como en los casos de prueba seleccionados para interacción con botones en la sección de “trámites administrativos” hay tres opciones que se van a llevar a cabo mediante botones,

conviene más configurar un *Scenario Outline* para que la iteración sea más dinámica. Lo mismo ocurre con la sección de convalidaciones.

Esta dinámica es muy útil sobre todo para el apartado de preguntas frecuentes, el cual cuenta con numerosas interacciones con botones programadas.

```
Scenario Outline: preguntas frecuentes
  When I say hola to the chatbot
  Then I click on the preguntas frecuentes button
  And I search until I find my destiny in <seleccion>
  Then I go to the <seleccion> web
  Examples:
    |seleccion|
    |informacion general|
    |normativa|
    |correo electronico|
    |cuenta usuario|
    |curso cero ADE|
    |panorama universitario|
    |asignaturas|
    |horario|
    |practicas|
    |tfg|
    |grupos investigacion|
    |biblioteca|
```

Figura 51: Escenarios Cucumber: preguntas frecuentes

```
Scenario Outline: redes
  When I say hola to the chatbot
  Then I click on the redes button
  And I search until I find my destiny for redes in <seleccion>
  Then I go to the <seleccion> web for redes
  Examples:
    |seleccion|
    |vpn|
    |contrasenas|

Scenario: como moverme
  When I say hola to the chatbot
  Then I click on the localizaciones button
  And I search until I find my destiny for como moverme
  Then I receive the como moverme response message

Scenario: delegacion
  When I say hola to the chatbot
  Then I click on the localizaciones button
  And I search until I find my destiny for delegacion
  Then I receive the delegacion response message
```

Figura 52: Escenarios Cucumber: redes, cómo moverme y localizaciones

Por otro lado, para las secciones de VPN y de Localizaciones, únicamente se ha seleccionado un caso de prueba, por lo que configurar un escenario simple es la opción más conveniente en estos casos. La dinámica de programación será igual que como se ha visto en el escenario de ejemplo de explicación.

Para el apartado de VPN y WiFi se ha creado otro *scenario outline* con los casos de prueba de VPN y contraseñas de usuario. Finalmente, para el apartado de localizaciones, se han hecho dos escenarios individuales debido a que el escenario “como moverme” no cuenta con un hipervínculo a otra página web, y causaba errores si se incluía en un *scenario outline*.

La interacción con botones tiene una característica peculiar que no tenía la interacción escrita. Como es posible que el usuario tenga que pulsar varios botones para llegar hasta la consulta deseada, los enlaces de los *xpath* van a variar, de forma que dependiendo del número de mensaje, de la cantidad de botones y de la cantidad de links, estos identificadores van a cambiar.

Como crear una variable para cada uno de estos casos sería poco óptimo para el código, se van a seleccionar los distintos casos que se pueden dar y se irán modificando mediante variables de método.

```
public WebElement getMessage(int i) {
    WebElement message = driver.findElement(By.xpath( xpathExpression: "//*[@id=\"cai-webchat-div\"]/div/div[2]/div[2]/div[1]/div/div["+i+"]/div/div"));
    return message;
}

public WebElement getSingleMessage(int i) {
    WebElement message = driver.findElement(By.xpath( xpathExpression: "//*[@id=\"cai-webchat-div\"]/div/div[2]/div[2]/div[1]/div/div["+i+"]/div/div/div/div"));
    return message;
}

public WebElement getButton(int i, int j){
    WebElement button = driver.findElement(By.xpath( xpathExpression: "//*[@id=\"cai-webchat-div\"]/div/div[2]/div[2]/div[1]/div/div["+i+"]/div/div/div/div["+j+"]"));
    return button;
}

public WebElement getLinkMessage(int i, int j) {
    WebElement message = driver.findElement(By.xpath( xpathExpression: "//*[@id=\"cai-webchat-div\"]/div/div[2]/div[2]/div[1]/div/div["+i+"]/div/div/div/a["+j+"]"));
    return message;
}

public WebElement getSingleLinkMessage(int i) {
    WebElement message = driver.findElement(By.xpath( xpathExpression: "//*[@id=\"cai-webchat-div\"]/div/div[2]/div[2]/div[1]/div/div["+i+"]/div/div/div/a"));
    return message;
}
```

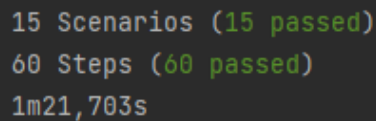
Figura 53: Métodos para interacción con botones

- `getMessage()`: este método recuperará el identificador *xpath* para un mensaje. Es decir, este será el mensaje que o bien escribe el usuario o bien el que responde el chatbot, sin meterse en los botones que puede llevar asociados esa respuesta. La variable *i* se referirá al número de mensaje en el que se encuentra este identificador.
- `getSingleMessage()`: este método se usará para hacer *clic* en un botón de respuesta. La característica de este tipo de botón es que es específico para el único botón que no redirija a una web externa. Por ejemplo, si hay dos respuestas que redirigen al usuario a una web externa y uno que redirige a otra *skill* para seguir navegando por botones, se usará este método para esta última opción. La variable *i* se referirá al número de mensaje en el que se encuentra este identificador.
- `getButton()`: este método se encarga de pulsar cualquier botón que redirija a otra *skill*. La variable *i* se referirá al número de mensaje en el que se encuentra este identificador, mientras que la variable *j* se referirá al número de botón que se quiere pulsar.
- `getLinkMessage()`: este método se encargará de redirigir al usuario a una web externa. Esta redirección estará siempre dentro de uno de los botones disponibles. La variable *i* se referirá al número de mensaje en el que se encuentra este identificador, mientras que la variable *j* se referirá al número de botón que se quiere pulsar.

- `getSingleLinkMessage()`: este método, al igual que `getSingleMessage()`, tiene la peculiaridad de que se usará en el caso de que el botón a pulsar sea el único que tiene un link a una web externa. Por ejemplo, si hay dos botones que redirigen a otra *skill*, y un botón que te lleva a una web externa, para este último caso se usará este método. La variable *i* se referirá al número de mensaje en el que se encuentra este identificador.

## Cómo saber si los escenarios han sido exitosos

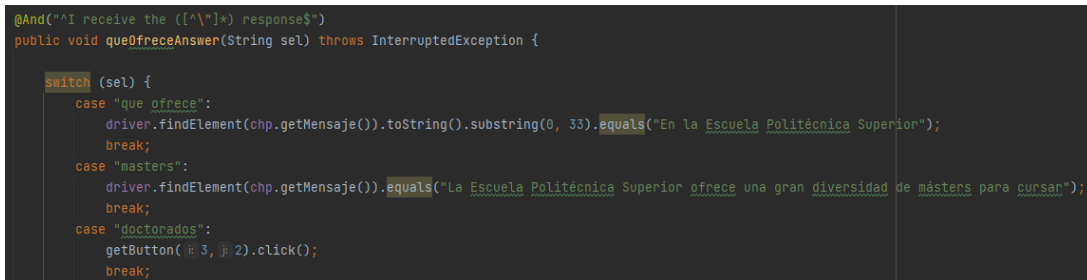
Una vez con los casos de prueba montados y listos para ejecutarse, surge la duda de cómo se puede comprobar si efectivamente estas pruebas están siendo exitosas o no. En este caso, *Selenium* no ofrece un reporte donde se pueda observar el porcentaje de éxito o de fallo, así que hay que fijarse en los *logs* (los mensajes que va imprimiendo la terminal de IntelliJ mientras se ejecutan los test) para ver el resultado final. Si se ejecuta, por ejemplo, el escenario de selección escrita, en la figura 54 se puede observar el resultado satisfactorio.



```
15 Scenarios (15 passed)
60 Steps (60 passed)
1m21,703s
```

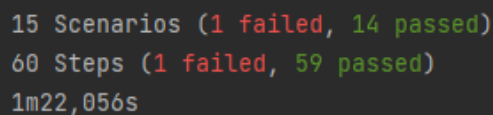
Figura 54: Evidencia pruebas exitosas

Pongámonos en el caso de que las pruebas no han sido satisfactorias. Se va a simular un caso en el que, dentro de selección escrita, se compruebe que la respuesta del chatbot no es la esperada. En la figura 55, se puede observar cómo en el caso de “doctorados”, se ha introducido una interacción con un botón que no existe.



```
@And("^I receive the ([^\\"]+) response$")
public void queOfreceAnswer(String sel) throws InterruptedException {
    switch (sel) {
        case "que ofrece":
            driver.findElement(chp.getMensaje()).toString().substring(0, 33).equals("En la Escuela Politécnica Superior");
            break;
        case "masters":
            driver.findElement(chp.getMensaje()).equals("La Escuela Politécnica Superior ofrece una gran diversidad de másters para cursar");
            break;
        case "doctorados":
            getButton(3, 2).click();
            break;
    }
}
```

Figura 55: Cambio en el método para provocar fallos



```
15 Scenarios (1 failed, 14 passed)
60 Steps (1 failed, 59 passed)
1m22,056s
```

Figura 56: Evidencia escenario fallido

Como se puede observar en la figura 56, aparece un escenario y un paso fallidos debido a esta interacción inexistente. Este sería el reporte para las pruebas que fallan. Puede fallar más de una, o incluso saltarse escenarios si no encuentra ciertas características de la web.



# CAPÍTULO 4

## Cambios de diseño

### Cambios de diseño en el chatbot

Para personalizar un poco el chatbot que se ve en la web, *SAP Conversational AI* ofrece la posibilidad de editar la apariencia del bot. Para ello, hay que dirigirse al apartado “Connect” y posteriormente hacer clic sobre “Webchat”. Aquí estará almacenada la configuración inicial que se le dio al chatbot

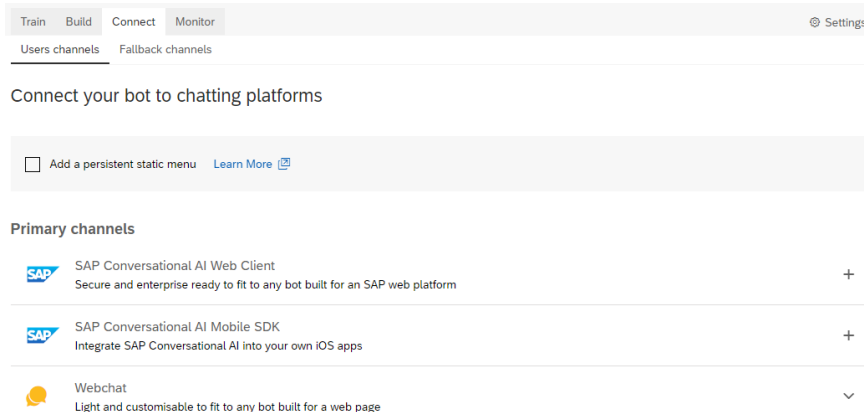


Figura 57: Apartado de conexiones en SAP Conversational AI

Aquí se abrirá un desplegable. En la sección “color scheme” (esquema de colores) se podrán elegir los colores que se mostrarán en el chatbot

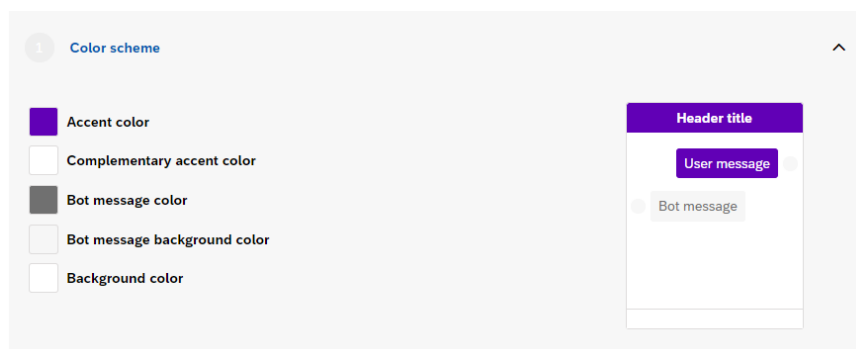


Figura 58: Elección colores para el chatbot

Para la cabecera, simplemente se le pondrá el nombre de “Chatbot TFG” como algo identificativo”, como puede verse en la figura 59



Figura 59: Elección logo y título para el chatbot

También se pueden personalizar las imágenes de icono que se ven en el bot. Para este caso, se va a elegir un icono general para el usuario y el icono de la Universidad de Alcalá para mostrar como icono del bot.

Figura 60: Elección iconos y mensaje de bienvenida

El resultado final sería el mostrado en la figura 61.

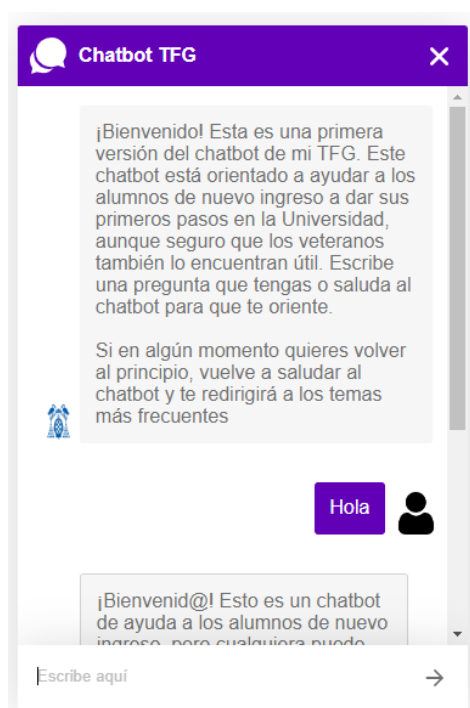


Figura 61: Vista final del chatbot en web con saludo

## Cambios de diseño en la web

Después de haber trabajado con un diseño muy básico en la web hasta ahora, es momento de retocarla para que sea más visible y tenga más funcionalidades. Vamos a introducir aquí, aparte de HTML, el lenguaje CSS para dar estilo a la web.

Partiendo de la clase ya existente index.html, en la cabecera se le tendrán que insertar las hojas de estilo. Esto se compone desde la clase “main.css”, que será la clase donde definamos las clases que definirán el diseño y la distribución de la página, hasta el tipo de letra que se va a emplear.

```
<head>
  <meta charset="UTF-8">
  <title>Maria Chantal TF6</title>

  <link rel="stylesheet" href="main.css" type="text/css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css">
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-4bw+aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInix6AoxmnLMuBnhbgrkm"
    crossorigin="anonymous"></script>
</head>
```

Figura 62: Edición de index.html: importación de main.css e iconos

La forma de trabajar con .css es mediante clases. Las clases se añaden en el .html a través de los sectores de división (<div>), como se puede ver en la figura 63, que correspondería al cuerpo de la clase index.html. Los <div> son divisiones en “cajas” que permitirán dividir la página en secciones para poder tener cierto control sobre ellas. También haremos divisiones por párrafos (<p>).

```
<body>

  <div class="pagina">
    <div class="menu">
      <div class="logoMenu">
        <a href="index.html"><i class="bi bi-c-circle"></i>Página principal</a>
      </div>
      <div class="accionesMenu">
        <div class="cajaAccionesMenu"><a href="SAP.html">SAP Conversational AI</a></div>
        <div class="cajaAccionesMenu"><a href="Pruebas.html">Pruebas automatizadas</a></div>
        <div class="cajaAccionesMenu"><a href="Memoria.html">Memoria explicativa</a></div>
      </div>
      <div class="derechaMenu"></div>
    </div>
    <div class="rowCaja">
      <div class="cajaFlexTexto">
        <div class="cajaTexto">
          <H2>Bienvenido a la web de prueba del chatbot</H2>
        </div>
        <div class="cajaTexto">
          <p>Esta es la página desde las que se harán las pruebas automatizadas con Cucumber y Selenium para el chatbot.
          Puedes encontrar el chatbot en un desplegable abajo a la derecha. Haciendo click sobre él,
          podrás empezar una conversación.
          Para más información sobre cómo se ha creado el chatbot, puedes consultar los menús que
          encontrarás en la parte superior de la página</p>
        </div>
      </div>
    </div>
  </div>
```

Figura 63: Edición cuerpo de index.html

Este nuevo diseño de la web va a tener un menú en la zona superior y un botón para regresar a la página principal. En el menú, se podrá acceder a diferentes .html codificados con la información relativa:

- SAP.html: aquí habrá una breve explicación sobre la herramienta que se ha usado para desarrollar el chatbot así como un enlace para acceder.

- Pruebas.html: aquí se podrá encontrar información sobre las pruebas automatizadas que se han usado para el proyecto.
- Memoria.html: aquí aparecerá el .pdf de la memoria de este TFG, para poder acceder a ella de forma dinámica en la web en cualquier momento.

En todas estas páginas, aparecerá en la zona inferior derecha el chatbot en todo momento y la configuración del menú será la misma.

### SAP.html

En esta clase, a parte de las divisiones <div> y los párrafos <p> previamente mencionadas, se va a trabajar también con puntos de lista. Estos se van a diferenciar por estar en la división <ul> e indicados con <li>.

```
<div class="rowCaja">
  <div class="cajaFlexTexto">
    <div class="cajaTextoTitulo">
      <H2>¿Qué es SAP Conversational AI?</H2>
    </div>
    <div class="cajaTexto">
      <p>
        SAP Conversational AI es una herramienta que ofrece una interfaz para poder desarrollar,
        entrenar,
        probar y conectar chatbots creados por el usuario.
        Permite simplificar información convirtiéndola en una sencilla charla entre el chatbot y el
        usuario.
      </p>
      <p>
        Esta herramienta incluye todas las características necesarias para poder construir tu propio
        chatbot
        desde cero.
      </p>
    </div>
  </div>
```

Figura 64: Creación sap.html

```
<div class="cajaTextoLista">
  <ul>
    <li>En primer lugar, te permite, de forma muy intuitiva, crear tu propio chatbot. Desde las
      expresiones que el usuario introducirá como preguntas hasta las respuestas que se esperan
      del
      bot.</li>
    <li>Un paso fundamental a la hora de construir este chatbot es su entrenamiento. Gracias al
      entrenamiento, el chatbot podrá asimilar todas las preguntas y respuestas que se le han
      asignado
      para poder mantener una conversación por sí mismo.</li>
    <li>En este caso hemos conectado el chatbot a una web, pero SAP Conversational AI también
      permite
      conectarlo a canales de mensajería, como Slack o Telegram.</li>
  </ul>
  <p>En la esquina inferior derecha, podrás observar el desplegable del chatbot. Haciendo click
    sobre
    él, podrás iniciar una conversación. Es muy importante recordar que, para activarlo, es muy
    importante saludarle o hacerle directamente una pregunta.</p>
</div>
<div class="cajaBoton">
  <button type="button" class="btn btn-outline-primary btn-lg" onclick="">Visita la página de SAP
    Conversational AI</button>
</div>
```

Figura 65: Cuerpo sap.html

## Pruebas.html

Aquí se van a encontrar muchas divisiones de párrafos ya que es una clase puramente explicativa. Habrá dos subtítulos para explicar *Cucumber* y *Selenium*, y un botón con enlace a cada una de las webs oficiales.

```
<div class="contenido">
  <div class="cajaFlexTexto">
    <div class="cajaTextoTitulo">
      <H2>¿Qué son las pruebas automatizadas?</H2>
    </div>
    <div class="cajaTexto">
      <p>
        Las pruebas automatizadas consisten en emplear una o varias herramientas para probar algún tipo de software.
      </p>
      <p>
        El objetivo de las pruebas automatizadas es comprobar que la ejecución del software está funcionando de manera esperada.
        El programador debe tener conocimiento de cómo debería funcionar su software, por lo que deberá plasmar en las pruebas automatizadas las órdenes necesarias para garantizar que el código funciona correctamente
      </p>
      <p>
        Existen numerosas herramientas para realizar pruebas de este estilo, pero para este TFe se ha decidido emplear Cucumber y Selenium
      </p>
    </div>
  </div>
```

Figura 66: Creación y cuerpo de pruebas.html

```
<div class="cajaTitulo">
  <H3> ¿Qué es Cucumber?</H3>
</div>
<div class="cajaTitulo">
  <p>
    A pesar de que en el proyecto se está hablando de Cucumber, el lenguaje de programación utilizado es Gherkin, el cual se utiliza para desarrollar pruebas automatizadas. Cucumber no es más que una de las herramientas que proporciona Gherkin para poder desarrollar estas pruebas. Cucumber funciona a base de escenarios. Cuando se crea un escenario bajo la extensión ".feature", se deben tener en cuenta los siguientes pasos:
  </p>
</div>
<div class="cajaTextolista">
  <ul>
    <li style="...">Identificar el escenario de la prueba: nos referimos a escenario como el conjunto de acciones que se van a llevar a cabo para que la prueba siga las indicaciones que le da el programador. Cada escenario debe llevar un nombre y diferentes pasos a seguir. Existen dos tipos de escenarios:</li>
    <ul style="...">
      <li>
        Scenario: es un escenario clásico. Se construye con los indicadores de los pasos
      </li>
      <li>
        Scenario Outline: funciona igual que un escenario clásico, salvo el detalle de que permite incluir variables en los pasos a ejecutar. Este tipo de escenarios se utiliza en ocasiones donde se quiere repetir un paso con diferentes valores.
      </li>
    </ul>
    <li style="...">Pasos a seguir: para que un escenario funcione, cada "paso" está definido por un indicador. De esta forma, podemos asemejar el lenguaje humano a lo que va a hacer cada escenario. Para esto, el escenario seguirá un orden ejecución basado en una lista de acciones: "Cuando tengo esta situación dado este ejemplo, entonces ejecutaré cierto comando y comprobaré el resultado. Los indicadores se marcan con las siguientes palabras reservadas
    <ul style="...">
      <li>Given</li>
      <li>When</li>
      <li>Then</li>
      <li>And</li>
    </ul>
  </li>
  </ul>
</div>
</div>
```

Figura 67: Pruebas.html – definición Cucumber

```

<div class="cajaFlexTexto">
  <div class="cajaTitulo">
    <H3> ¿Qué es Selenium Web Driver?</H3>
  </div>
  <div class="cajaTitulo">
    <p>
      Selenium Web Driver es una interfaz que simula las interacciones del usuario con cualquier
navegador. En el caso de este proyecto, servirá de intermediario para recoger las órdenes
escenificadas con Cucumber y posteriormente interaccionar con el chatbot alojado en la web.
    </p>
    <p>
      La conexión que tienen Cucumber y Selenium es muy sencilla. Gracias a los escenarios
definidos con Cucumber, Selenium cogerá el nombre de cada uno de los pasos definidos con los
identificadores y se le asignarán órdenes para proceder con las pruebas.
      Por ejemplo, si la primera orden de un escenario en Cucumber es "Given I am on the main
      page", la programación de ese paso con Selenium permitirá crear la conexión con la página
principal, abriendo el driver y mostrando la dirección web establecida.
    </p>
  </div>
  <div class="cajaTitulo">
    <H5> ¿Cómo funciona la interacción de Selenium con el chatbot?</H5>
  </div>
  <div class="cajaTitulo">
    <p>

```

Figura 68: Pruebas.html – definición Selenium

```

    del bot, para posteriormente ir leyendo los xpath de cada mensaje para comprobar que los
    textos, botones e interacciones son los esperados
  </p>
</div>
</div>
<div class="rowBoton">
  <div class="cajaBoton">
    <a class="btn btn-outline-primary btn-lg" href="">Visita la página oficial de
      Selenium Web Drive</a>
  </div>
</div>

```

Figura 69: Pruebas.html – finalización definición Selenium

Como añadido a esta sección, y para poder dejar plasmado el funcionamiento de las pruebas, se van a incluir en este submenú de la web una serie de vídeos donde se podrá comprobar la ejecución de cada uno de los escenarios de prueba.

```

<div class="cajaFlexTexto">
  <div class="cajaTexto">
    <H2>¿Cómo funcionan las pruebas?</H2>
  </div>
  <div class="cajaTexto">
    <p>
      En los siguientes videos se van a poder observar las ejecucciones de las pruebas automatizadas.
      Estos seis videos mostrarán de principio a fin el proceso de ejecución de las pruebas, así como la
      evidencia de que han sido exitosas
    </p>
    <p>
      Para poder ver todos los videos, desplázate con las felchas que encontrarás a ambos lados de los videos.
    </p>
  </div>
</div>

```

Figura 70: Pruebas.html – introducción vídeos de las pruebas

Los vídeos han subido a YouTube en una lista de reproducción oculta [11] para que solo puedan visualizarse accediendo desde la página web. Se ha programado un formato en carrusel para la visualización de los mismos, añadiendo un pequeño pie de vídeo para diferenciar qué escenario se está ejecutando. En la figura 71 se puede ver el código empleado.

```

<div class="rowCarousel">
  <div id="carouselExampleCaptions" class="carousel slide">
    <div class="carousel-indicators">
      <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active"
        aria-current="true" aria-label="Interacción escrita"></button>
      <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1"
        aria-label="Preguntas frecuentes"></button>
      <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2"
        aria-label="Trámites administrativos"></button>
      <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="3"
        aria-label="Convalidaciones"></button>
      <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="4"
        aria-label="Redes"></button>
      <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="4"
        aria-label="Localizaciones"></button>
    </div>
  </div>

```

Figura 71: Pruebas.html – carrusel para videos de las pruebas

Para cada escenario, es necesario crear diferentes clases, donde se insertará el video de YouTube y se especificará el pie de foto correspondiente.

```

<div class="carousel-inner">
  <div class="carousel-item active">
    <iframe width="1120" height="630" src="https://www.youtube.com/embed/xoUG5Us9NkQ"
      title="YouTube video player" frameborder="0"
      allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
      allowfullscreen>
    </iframe>
    <div class="carousel-caption d-none d-md-block">
      <h5 style="..">Interacción escrita </h5>
    </div>
  </div>

```

Figura 72: Pruebas.html – video interacción escrita

```

<div class="carousel-item">
  <iframe width="1120" height="630" src="https://www.youtube.com/embed/8F2tIqmp6mM"
    title="YouTube video player" frameborder="0"
    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
    allowfullscreen>
  </iframe>
  <div class="carousel-caption d-none d-md-block">
    <h5 style="font-weight: bolder;">Preguntas frecuentes</h5>
  </div>
</div>

```

Figura 73: Pruebas.html – video preguntas frecuentes

```

<div class="carousel-item">
  <iframe width="1120" height="630" src="https://www.youtube.com/embed/7tplzvJT64"
    title="YouTube video player" frameborder="0"
    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
    allowfullscreen>
  </iframe>
  <div class="carousel-caption d-none d-md-block">
    <h5 style="font-weight: bolder;">Trámites administrativos</h5>
  </div>
</div>

```

Figura 74: Pruebas.html – video trámites administrativos

```

<div class="carousel-item">
  <iframe width="1120" height="630" src="https://www.youtube.com/embed/Hl3mJVCv4Lo"
    title="YouTube video player" frameborder="0"
    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
    allowfullscreen>
  </iframe>
  <div class="carousel-caption d-none d-md-block">
    <h5 style="font-weight: bolder;">Convalidaciones</h5>
  </div>
</div>

```

Figura 75: Pruebas.html – vídeo convalidaciones

```

<div class="carousel-item">
  <iframe width="1120" height="630" src="https://www.youtube.com/embed/d91u3c6QkeI"
    title="YouTube video player" frameborder="0"
    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
    allowfullscreen>
  </iframe>
  <div class="carousel-caption d-none d-md-block">
    <h5 style="font-weight: bolder;">Localizaciones</h5>
  </div>
</div>

```

Figura 76: Pruebas.html – vídeo localizaciones

```

<div class="carousel-item">
  <iframe width="1120" height="630" src="https://www.youtube.com/embed/Ku7Nh5jN5zA"
    title="YouTube video player" frameborder="0"
    allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
    allowfullscreen>
  </iframe>
  <div class="carousel-caption d-none d-md-block">
    <h5 style="font-weight: bolder;">Redes</h5>
  </div>
</div>

```

Figura 77: Pruebas.html – vídeo redes

Finalmente, en la figura 78 se puede observar cómo se visualizarían los vídeos en la web.

## ¿Cómo funcionan las pruebas?

En los siguientes vídeos se van a poder observar las ejecuciones de las pruebas automatizadas. Estos seis vídeos mostrarán de principio a fin el proceso de ejecución de las pruebas, así como la evidencia de que han sido exitosas

Para poder ver todos los vídeos, desplázate con las flechas que encontrarás a ambos lados de los vídeos.

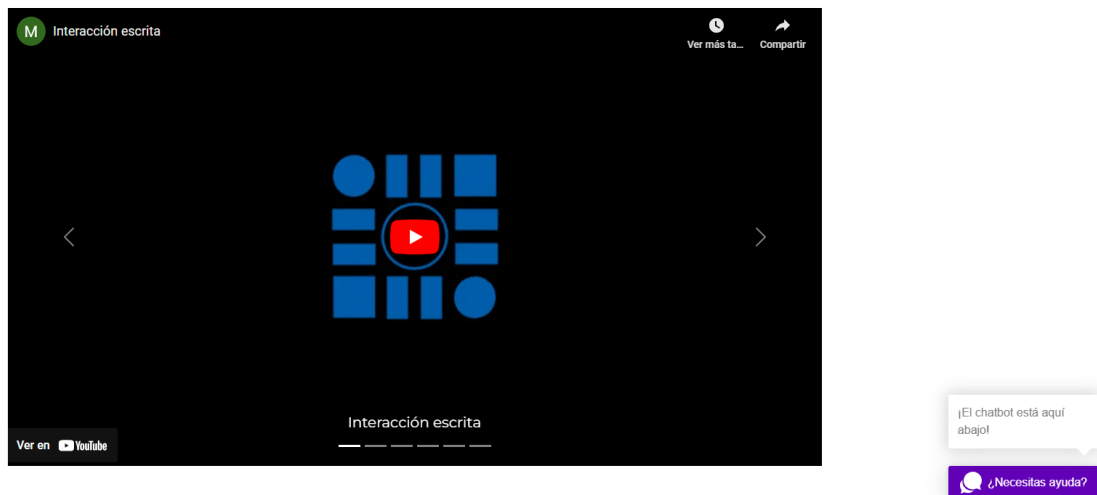


Figura 78: Vista en la web del carrusel de vídeos



## Memoria.html

En esta clase se va a usar la división <embed>, la cual se va a utilizar para adjuntar el PDF de la memoria de este TFG.

```
<div class="rowCaja">
  <div class="cajaFlexTexto">
    <div class="cajaTextoTitulo">
      <H2>Memoria Explicativa del TFG</H2>
    </div>
    <div class="cajaVisorPdf">
      <embed src="docs/TFG.pdf" type="application/pdf" width="100%" height="100%"/>
    </div>
  </div>
</div>
/body>
```

Figura 79: Creación memoria.html

Esto permitirá la visualización de la memoria de este trabajo en formato .pdf directamente desde la web.

En las siguientes figuras se mostrarán todas las clases creadas en main.css. En la mayoría de figuras se van a observar codificaciones bastante similares entre sí. El sentido de esto es que cada “caja” dividida en el .html va a tener una flexibilidad y un comportamiento diferente. Por ejemplo, el menú que va a aparecer en la parte superior, necesita unos márgenes distintos a los que tendrán los títulos de texto de cada página. De esta forma, se puede dinamizar el movimiento de las cajas y de los textos.

```
.pagina {
  background-image: repeating-radial-gradient
(circle at 0 0, transparent 0, #ffffff 56px),
repeating-linear-gradient(#faf6fd, #faf6fd);
background-color: #ffffff;

  width: 100%;
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: start;
  font-family: "Montserrat";
}
.contenido{
  width: 100%;
  height: wrap-content;
  display: flex;
  flex-direction: column;
  font-size: 1.5vw;
  justify-content: start;
  align-items: center;
}
.menu {
  width: 100%;
  height: wrap-content;
  display: flex;
  flex-direction: row;
  font-size: 1.5vw;
  justify-content: space-between;
  align-items: center;
}
```

Figura 80 : Clases main.css usadas en los html parte 1

```
.logoMenu {
  width: 20%;
  height: wrap-content;
  margin: 1%;
}

.accionesMenu {
  width: 50%;
  height: wrap-content;
  margin: 1%;
  display: flex;
  flex-direction: row;
  justify-content: center;
}

.logoMenu a {
  text-decoration: none;
  color: black;
}

.accionesMenu a {
  text-decoration: none;
  color: black;
}

.derechaMenu {
  width: 20%;
  height: 100%;
  margin: 1%;
}
```

Figura 81: cClases main.css usadas en los html parte 2

```

.accionesMenu a {
    text-decoration: none;
    color: black;
}

.derechaMenu {
    width: 20%;
    height: 100%;
    margin: 1%;
}

.cajaAccionesMenu {
    width: fit-content;
    height: 100%;
    margin: 1%;
    margin-left: 3%;
    margin-right: 3%;
    font-size: 1.1vw;
    text-align: center;
}

.rowCaja {
    width: 100%;
    height: wrap-content;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: start;
}

.cajaTextoTitulo{
    width: wrap-content;
    height: wrap-content;
    margin: 1%;
    margin-top: 10%;
    margin-left: 17%;
    margin-right: 17%;
    text-align: center;
}

```

Figura 82: Clases main.css usadas en los html parte 3

```

.cajaTexto{
    width: wrap-content;
    height: wrap-content;
    margin: 1%;
    margin-left: 17%;
    margin-right: 17%;
    text-align: center;
}

.cajaTextoLista{
    width: wrap-content;
    height: wrap-content;
    margin: 1%;
    margin-left: 17%;
    margin-right: 17%;
}

.cajaFlexTexto {
    width: 100%;
    height: wrap-content;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    align-items: center;
    font-size: 1.2vw;
}

.cajaFlexTexto2 {
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    align-items: center;
    font-size: 1.2vw;
    margin-right: 17%;
    margin-left: 17%;
}

```

Figura 83: Clases main.css usadas en los html parte 4

```

.cajaTitulo {
    width: wrap-content;
    height: wrap-content;
    margin: 1%;
    margin-left: 17%;
    margin-right: 17%;
    text-align: center;
}

.rowBoton {
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: row;
    font-size: 1vw;
    font-weight: 50;
    justify-content: center;
    align-items: center;
}

.cajaBoton {
    width: wrap-content;
    height: wrap-content;
    margin: 1%;
    margin-bottom: 3%;
}

.cajaFlexTexto h2 {
    font-size: 1.9vw;
    font-weight: 1000;
}

.cajaVisorPdf{
    width: 80%;
    height: 800px;
    margin: 2%;
}

```

Figura 84: Clases main.css usadas en los html parte 5

# CAPÍTULO 5

## ChatGPT y las inteligencias artificiales

No podemos negar el impacto gigantesco que han tenido las Inteligencias Artificiales en el ámbito de los chatbot en el último año. Desde que empecé a trabajar en este proyecto, *ChatGPT* ha crecido de forma exponencial hasta ser conocido por prácticamente todo el mundo.

*ChatGPT* es una aplicación desarrollada en 2022 por *OpenAI* [12] programado con *Python*. La característica de este chatbot es que utiliza técnicas de aprendizaje reforzado y supervisado para aprender sobre los temas que se le preguntan. Gracias a este aprendizaje, es capaz de dar respuestas muy concretas y con bastante veracidad, basándose en todo lo que ha ido aprendiendo de los usuarios que lo han usado.

*ChatGPT* ofrece la posibilidad de crear tu propio chatbot, una aplicación llamada *Chatbase*. Para hacer una comparación de lo que puede hacer un chatbot creado con inteligencia artificial contra el chatbot que se ha creado en este proyecto desde cero de forma manual, voy a crear un prototipo de chatbot con *Chatbase* para hablar de las principales diferencias.

### Chatbase

En primer lugar, es obligatorio registrarse en el sitio web [13] de *Chatbase* para poder configurar el chatbot.

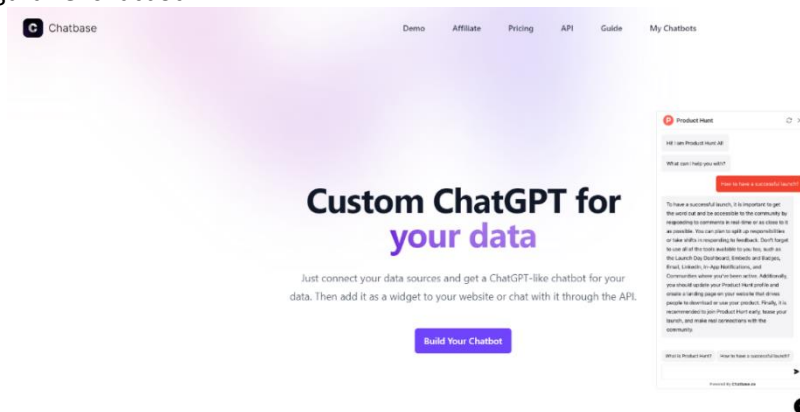
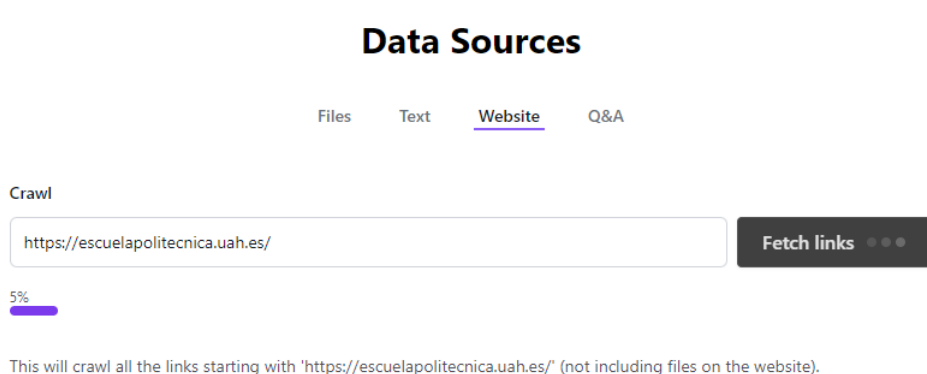


Figura 85: Página principal Chatbase



Nada más registrarse, al ir a crear el chatbot, he optado por configurarlo vía web. Con esto, se introduce la url de la Escuela Politécnica Superior y de esta forma el chatbot recogerá todos los enlaces relacionados con esa web, como se puede apreciar en la figura 86

El problema de esto es que, para la versión gratuita, únicamente permite reconocer un total de 400.000 caracteres entre todos los enlaces que recoja del enlace principal que se le ofrece. La web de la Escuela Politécnica Superior suma más de un millón de caracteres en total.

Para poder proseguir con esta prueba, he quitado los enlaces que, aunque son necesarios, veía algo menos relevantes para los alumnos de nuevo ingreso, como información sobre la cafetería o las cátedras. También he borrado enlaces a los dobles grados, así como el grado de Matemáticas y Computación (por ser un nuevo grado), así como las asignaturas de os dobles grados, del grado de Telemática y del grado de Electrónica de comunicaciones.

La selección final de los enlaces seleccionados es la que se muestra en las figuras 86 y 87

Included Links		Delete all
<a href="https://escuelapolitecnica.uah.es/">https://escuelapolitecnica.uah.es/</a>	10880	
<a href="https://escuelapolitecnica.uah.es/estudiantes/practicas-externas.asp">https://escuelapolitecnica.uah.es/estudiantes/practicas-externas.asp</a>	10502	
<a href="https://escuelapolitecnica.uah.es/estudiantes/trabajo-fin-grado.asp">https://escuelapolitecnica.uah.es/estudiantes/trabajo-fin-grado.asp</a>	11127	
<a href="https://escuelapolitecnica.uah.es/estudiantes/nuevos-alumnos.asp">https://escuelapolitecnica.uah.es/estudiantes/nuevos-alumnos.asp</a>	9173	
<a href="https://escuelapolitecnica.uah.es/estudios/grados.asp">https://escuelapolitecnica.uah.es/estudios/grados.asp</a>	9642	
<a href="https://escuelapolitecnica.uah.es/estudiantes/horarios.asp">https://escuelapolitecnica.uah.es/estudiantes/horarios.asp</a>	9528	
<a href="https://escuelapolitecnica.uah.es/estudiantes/fechas-examenes.asp">https://escuelapolitecnica.uah.es/estudiantes/fechas-examenes.asp</a>	9293	
<a href="https://escuelapolitecnica.uah.es/inicio.asp">https://escuelapolitecnica.uah.es/inicio.asp</a>	4256	
<a href="https://escuelapolitecnica.uah.es/estudiantes/asignaturas.asp">https://escuelapolitecnica.uah.es/estudiantes/asignaturas.asp</a>	8644	
<a href="https://escuelapolitecnica.uah.es/escuela/normativa.asp">https://escuelapolitecnica.uah.es/escuela/normativa.asp</a>	8150	
<a href="https://escuelapolitecnica.uah.es/tramites/formularios.asp">https://escuelapolitecnica.uah.es/tramites/formularios.asp</a>	12955	
<a href="https://escuelapolitecnica.uah.es/estudiantes/programas-internacionales.asp">https://escuelapolitecnica.uah.es/estudiantes/programas-internacionales.asp</a>	11612	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G781">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G781</a>	16093	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G390">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G390</a>	14031	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G581">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G581</a>	17859	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G370">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G370</a>	13448	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G350">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G350</a>	14364	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G380">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G380</a>	13550	

Figura 87: Enlaces incluidos en el chatbot de Chatbase parte 1

<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G610">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G610</a>	12745	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G60">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G60</a>	13604	
<a href="https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G591">https://escuelapolitecnica.uah.es/estudios/grado-int.asp?plan=G591</a>	19221	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G781">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G781</a>	20149	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G610">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G610</a>	18131	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G60">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G60</a>	19509	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G581">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G581</a>	18842	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G591">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G591</a>	20447	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G350">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G350</a>	25620	
<a href="https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G390">https://escuelapolitecnica.uah.es/estudios/1stAsignaturas-v3.asp?CodPlan=G390</a>	22767	

396,142 detected characters

Add

Included sources:

28 Links (396,142 detected chars)

Total detected characters: **396,142** / 400,000 limit

Create Chatbot

Figura 88: Enlaces incluidos en el chatbot de Chatbase parte 2

El siguiente paso sería directamente probar el chatbot. Según la interfaz principal, parece que solo se permiten 30 preguntas diarias al chatbot.

escuelapolitecnica.uah.es

Chatbot
Settings
Dashboard
Sources
Integrations
New
Embed on site
Share
Delete

Hi! What can I help you with?

30 message credits left

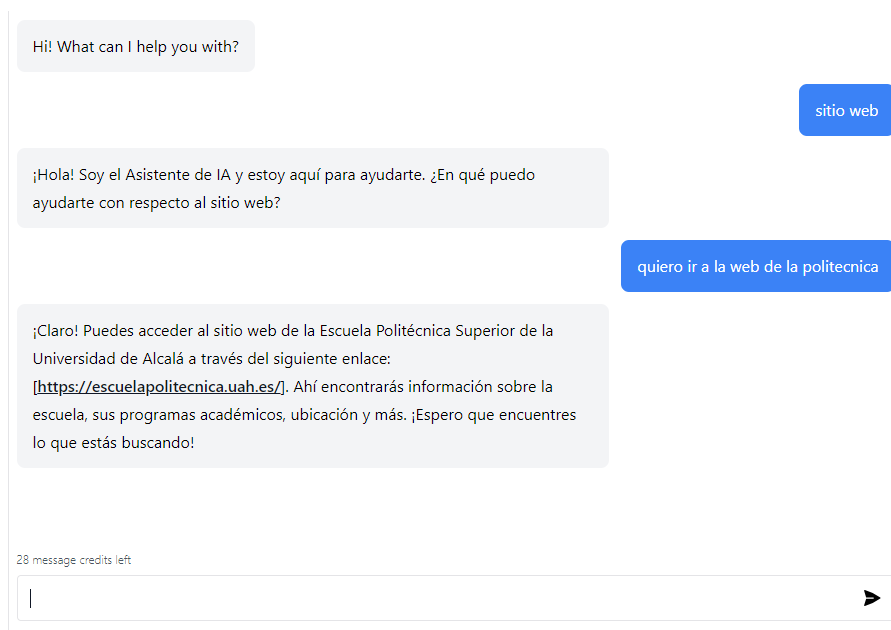
Still training your chatbot on the provided links  
This will take a few minutes

Hi! I am ChatbaseAI, ask me anything about Chatbase!

By the way, you can create a chatbot like me for your website!

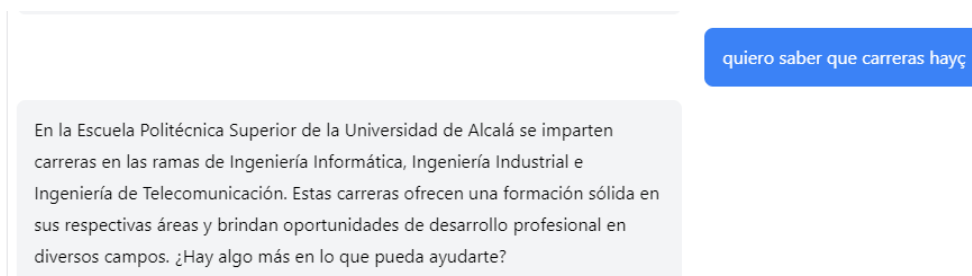
Figura 89: Primera vista al chatbot creado con Chatbase

Para probar este chatbot, se van a introducir algunas frases definidas manualmente en el chatbot del proyecto.



*Figura 90: Preguntas con chatbot de Chatbase: sitio web*

En la figura 89 se le pregunta al chatbot de *Chatbase* por el sitio web general. Mientras que en el chatbot configurado en el proyecto esta pregunta redirige directamente a la web principal de la Escuela Politécnica Superior, aquí interpreta la pregunta de forma más general, y hay que especificarle qué web se quiere buscar. Con la segunda pregunta, proporciona la respuesta adecuada.



*Figura 91: Preguntas con chatbot de Chatbase: carreras*

Con esta pregunta, mi intención era que el chatbot me devolviera una lista con todos los grados disponibles en la EPS. Sin embargo, parece que únicamente está reconociendo Ingeniería Informática, Ingeniería Industrial e Ingeniería de Telecomunicación, que no son grados, sino las ramas principales que engloban a todos los grados que tiene la escuela. Para profundizar un poco más, he reformulado la pregunta:

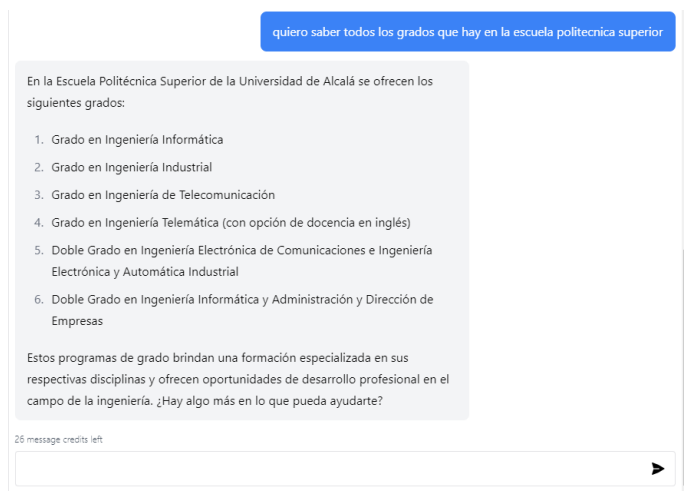


Figura 92: Preguntas con chatbot de Chatbase: carreras reformulada

Con esto se reafirma lo anterior. Está seleccionando los posibles campos de estudio, pero no todos los grados disponibles. Sin embargo, sí que asocia el grado de Ingeniería Telemática y los Dobles Grados como carreras aparte de los campos de estudio. Se puede comprobar en la figura 92 (sacada de la web de la EPS) que la lista mostrada en la figura 91 no es idéntica.




#### Estudios de Informática

- Grado en Ingeniería de Computadores Opción docencia inglés
- Grado en Ingeniería en Sistemas de Información
- Grado en Ingeniería Informática Opción docencia inglés
- Grado en Matemáticas y Computación **Guadalajara. ¡Nuevo!**

#### Estudios de Industriales

- Grado en Ingeniería en Electrónica y Automática Industrial Opción docencia inglés 
- Grado en Ingeniería en Tecnologías Industriales

#### Estudios de Telecomunicación

- Grado en Ingeniería Electrónica de Comunicaciones Opción docencia en inglés 
- Grado en Ingeniería en Sistemas de Telecomunicación Opción docencia en inglés 
- Grado en Ingeniería en Tecnologías de Telecomunicación Opción docencia en inglés
- Grado en Ingeniería Telemática Opción docencia en inglés 

#### Dobles Grados


- Doble Grado en Ingeniería Electrónica de Comunicaciones e Ingeniería Electrónica y Automática Industrial 
- Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas

Figura 93: Grados de la Escuela Politécnica Superior

También se le ha preguntado sobre los trabajos de fin de grado, aunque su respuesta ha sido más bien una descripción general que el enlace donde se puede encontrar toda la información concreta necesaria para poder afrontar el trabajo. La respuesta deriva al usuario a las guías docentes de cada asignatura, pero no muestra ningún enlace. Cabe decir que es posible que no tenga acceso a los enlaces por la limitación de caracteres, donde se tuvieron que eliminar diversos enlaces de acceso.

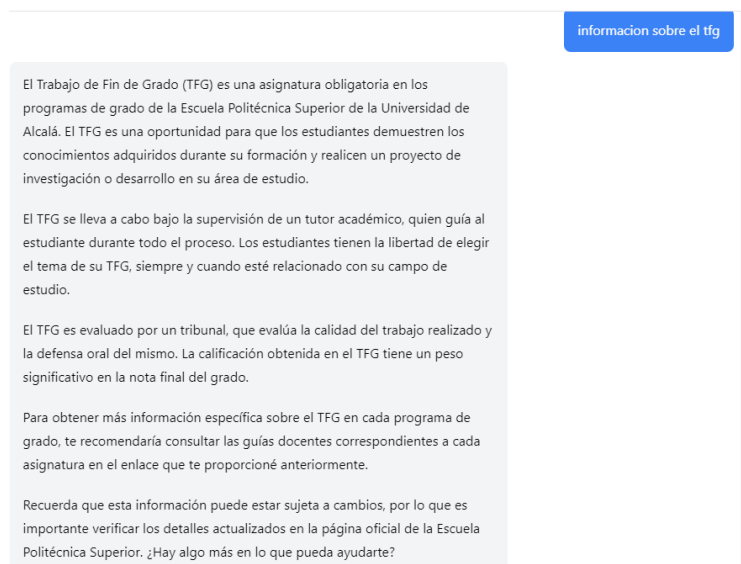


Figura 94: Preguntas con chatbot de Chatbase: información tfg

Como último ejemplo, se le ha preguntado por las fechas de exámenes de un grado en concreto. El chatbot, como se puede observar en la figura 95, dice que no tiene acceso a los datos en tiempo real ni a las fechas de exámenes. Esto puede ser debido a que los ficheros que se suben a la web indicando fechas de exámenes son dinámicos y cambian cada año. La respuesta proporciona las opciones de buscar en los documentos oficiales de la Universidad o contactar con el departamento académico.

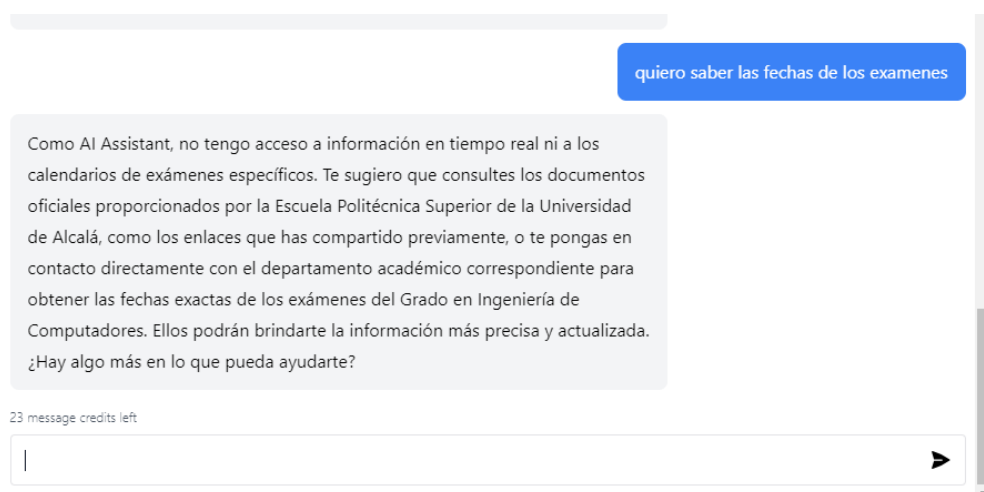


Figura 95: Preguntas con chatbot de Chatbase: fechas de exámenes



## Conclusiones y líneas futuras de trabajo

Como resumen final del trabajo, se ha conseguido, en primer lugar, desarrollar un chatbot funcional que cumple el propósito establecido. Este chatbot es capaz de reconocer entre una amplia gama de preguntas y de asociarlas con su correspondiente respuesta. Además, proporciona un sistema de navegación mediante botones que cubre de nuevo todos los apartados propuestos y que, de ser necesario, se puede intercalar con preguntas. De esta forma, tanto alumnos de nuevo ingreso como alumnos veteranos podrán mantener una conversación con el asistente para que responda las dudas que les puedan surgir. Gracias a las respuestas de texto, los enlaces y las fotografías, cualquier usuario podrá orientarse por la Universidad y saber cómo funciona por dentro.

El chatbot se ha alojado en una página web hecha exclusivamente para este trabajo, desde la cual se puede hablar con el asistente o consultar la documentación relativa al trabajo. De esta forma, cualquier persona puede saber con qué herramientas se ha desarrollado e incluso consultar la memoria a modo de guía para saber el trasfondo del proyecto. Como añadido, se puede observar la ejecución de las pruebas de principio a fin en el apartado de “Pruebas automatizadas”. Esta página web se puede visitar desde cualquier navegador a través del enlace <http://www.mariachantaltfg.es/>.

Las pruebas automatizadas se han ejecutado con éxito en cada uno de los escenarios desarrollados, tanto para la interacción escrita como para el sistema de navegación con botones. La ejecución de los casos de pruebas ha sido posible gracias a la conexión que facilita Selenium Web Driver al driver de Google Chrome, el cual ha sido el navegador seleccionado para ejecutar estas pruebas. Gracias al éxito de estas ejecuciones, existe la garantía de que las respuestas del chatbot son las que se esperan en comparación a lo programado, englobando así todo el trabajo realizado. En el caso de que en algún momento fallen, las pruebas también indicarán el escenario fallido, para así poder volver a él y modificar el código o el mismo chatbot por dentro.

Aunque, después de analizar la última parte del trabajo, se puede intuir que nunca va a ser igual la programación de este tipo de bots de forma manual que el contraste que nos encontramos al contar con una inteligencia artificial.

A lo largo de este trabajo, se ha explicado lo extenso que puede llegar a ser contemplar todas las opciones de preguntas que puede hacer un usuario, quedando aun así muchas por cubrir, así como la asociación de esas preguntas a una respuesta específica. Un Chatbot programado a mano puede cubrir cierto campo que se proponga, como en este caso, intentar guiar al alumno por la web de la Escuela Politécnica Superior con unas cuantas palabras o botones. En contraste, una inteligencia artificial puede abarcar cientos y cientos de páginas web y proporcionar respuestas de cualquier ámbito, analizando la entrada de texto del usuario sin importar faltas de ortografía, si se emplea un lenguaje más coloquial o si no se le hacen preguntas con signos de interrogación.

De esta forma, si se hubiera empleado inteligencia artificial, no se podrían hacer pruebas automatizadas. La IA cada vez puede dar una respuesta diferente, y aunque tenga una tasa de éxito generalmente alta, no se puede garantizar que la respuesta que ofrece sea totalmente verídica. Pero hablando de este trabajo, y en este caso en particular, la programación manual del chatbot ofrece la garantía de que la respuesta que el chatbot le dé al usuario va a ser la que

el programador desea. Un trato manual con este tipo de asistentes proporciona una estabilidad objetiva a la hora de responder.

A pesar de que no se ha podido explorar mucho en la creación del chatbot a través de Chatbase, una futura línea de investigación podría ser el ampliar a la versión de pago de esta herramienta. De esta forma se podría comprobar si realmente existe una diferencia de respuestas tan marcada como la que se ha visto en el trabajo. Esta versión de pago incluiría todos los enlaces necesarios asociados a la web de la Escuela Politécnica Superior, de esta forma no se limitarían las respuestas y el chatbot podría navegar por todos los enlaces para intentar encontrar la respuesta más apropiada.

Si por el contrario se quisiera expandir el chatbot creado para este proyecto, la mejora principal consistiría en cubrir el 100% de los casos de prueba. En este caso se ha cubierto el 70% del total, teniendo en cuenta que muchos de los apartados desarrollados acaban entrelazándose entre ellos, por lo que muchas pruebas habrían llevado al mismo sitio. Pero a pesar de esto, cubrir completamente los casos de prueba nos garantiza una fiabilidad y una seguridad total, sabiendo así que el usuario podrá interactuar con el chatbot sin ningún tipo de problema.

Dado que el chatbot se ha enfocado a alumnos de nuevo ingreso, existen diversos apartados que no se han incluido en la programación de este asistente, por lo que otra futura línea de trabajo podría ser el completar el contenido del chatbot con las secciones y apartados que se han omitido en este proyecto. De esta forma, el chatbot estaría preparado para ser integrado en la web de la Escuela Politécnica Superior. A gran escala, este proyecto se podría replicar para cada facultad y para cada campus de la Universidad, bien por separado, o bien en un único chatbot conjunto que recoja información sobre todas las áreas de conocimiento que se imparten en la Universidad de Alcalá de Henares.

## Bibliografía

- [1] “SAP Conversational AI | Low Code Chatbot Building Platform”. SAP Conversational AI. Accedido el 27 de julio de 2023. [En línea]. Disponible: <https://cai.tools.sap/>
- [2] “Nominalia: Dominios, Hosting, WordPress, Email, Servidores y mucho más” Accedido el 23 de junio de 2023. [En línea]. Disponible: <https://www.nominalia.com/>
- [3] “Maria Chantal TFG”. Maria Chantal TFG. Accedido el 24 de junio de 2023. [En línea]. Disponible: <http://www.mariachantaltfg.es/>
- [4] “WebDriver”. Selenium. Accedido el 10 de octubre de 2022. [En línea]. Disponible: <https://www.selenium.dev/documentation/webdriver/>
- [5] “BDD Testing & Collaboration Tools for Teams | Cucumber”. BDD Testing & Collaboration Tools for Teams | Cucumber. Accedido el 10 de octubre de 2022. [En línea]. Disponible: <https://cucumber.io/>
- [6] “Escuela Politécnica Superior – UAH”. Escuela Politécnica Superior – UAH. Accedido el 6 de septiembre de 2023. [En línea]. Disponible: <https://escuelapolitecnica.uah.es/>
- [7] “Java Archive Downloads – Java SE 17”. Oracle | Cloud Applications and Cloud Platform. Accedido el 25 de julio de 2023. [En línea]. Disponible: <https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>
- [8] “Maven – Download Apache Maven”. Maven – Welcome to Apache Maven. Accedido el 24 de junio de 2023. [En línea]. Disponible: <https://maven.apache.org/download.cgi>
- [9] “Maven Repository”. Selenium Java. Accedido el 4 de julio de 2023. [En línea]. Disponible: <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java>
- [10] “Maven Repository”. Cucumber JVM: Java. Accedido el 4 de julio de 2023. [En línea]. Disponible: <https://mvnrepository.com/artifact/io.cucumber/cucumber-java>
- [11] “María Chantal Cogolludo. *Interacción escrita*”. Accedido el 10 de septiembre de 2023. [Vídeo en línea]. Disponible: <https://youtu.be/z9h2Em3uThw>
- [12] “ChatGPT”. OpenAI. Accedido el 1 de agosto de 2023. [En línea]. Disponible: <https://openai.com/chatgpt>
- [13] “Chatbase | Custom ChatGPT for your data”. Accedido el 1 de Agosto de 2023. [En línea]. Disponible: <https://www.chatbase.co/>

## Anexos

### Anexo 1: Apartados recogidos en las pruebas de interacción escrita: Preguntas frecuentes

Qué ofrece	Qué masters hay	Doctorados	Cursos cero	Departamentos
que ofrece la eps	que masters hay	quiero cursar un doctorado	curso cero	departamentos
ventajas eps	que masters puedo cursar	quiero ir al curso de doctorado	iniciacion	que departamentos hay
Ventajas	quiero saber que oferta de masters hay	quiero estudiar para un doctorado	curso de inicio	como se organizan las asignaturas
que puedo encontrar en la eps	por favor dime el listado de masters	Quiero completar mi investigación de doctorado	onboarding	departamento de automática
que hay en la eps	masters	doctorado	curso de iniciación	departamento de automatica
Descripción	master	quiero hacer estudios de doctorado	cursos para empezar	departamento de ciencias de la computación
¿Qué puedo encontrar en la eps?	quiero saber que master hay	quiero hacer un doctorado	que es un curso cero	departamento de economía y direccion de empresas
que ofrece la escuela politecnica superior	quiero matricularme en un master	quiero saber que doctorados hay	hay algun curso de iniciacion	departamento de electronica
por que la escuela politecnica superior	quiero matricularme en un máster	doctorados	me podrias decir si existe algun curso de iniciacion	departamento de electrónica
por que la eps	¿Qué maestrias puedo tomar?		puedo estudiar algo antes de empezar?	departamento de física y matemáticas
por que elegir la eps	máster		como puedo hacer cursos para empezar	departamento de fisica y matematicas
por que elegir la escuela politecnica superior			¿Hay cursos para principiantes?	departamento de quimica
me puedes decir por que elegir la eps			¿Tienen cursos para principiantes?	departamento de quimica organica
me puedes decir por que elegir la escuela politecnica superior			¿Hay cursos de iniciación?	departamento de quimica inorganica

quiero saber que ofrece la eps			¿Hay algún curso de introducción?	departameteo telecomunicaciones
quiero saber las ventajas de la escuela politecnica superior			Por favor, hágamelo saber si hay un curso de introducción.	departamento de teoria de la señal y telecomunicaciones
¿que ofrece la eps?			¿Hay un curso para principiantes?	departamento quimica analitica
¿que ofrece la escuela politecnica superior?			¿Hay algo que pueda aprender antes de empezar?	departamento quimica fisica
			¿Puedo estudiar un poco antes de empezar?	departamento ingenieria quimica
			puedo ver algo antes de empezar?	me puedes decir los departamentos
				que es un departamento
				por favor explicame los departamentos que hay
				quiero saber sobre los departamentos
				quiero saber como se organizan las asignaturas

TUI	Erasmus	Delegación	Cafetería	Reprografía
que es la tui	quiero irme de erasmus	delegacion	donde esta la cafeteria	reprografia
tarjeta inteligente universitaria	quiero saber que necesito para irme de erasmus	delegación	que puedo comprar en la cafeteria	donde puedo imprimir
para que necesito la tui	cual es la normativa del erasmus	delegacion de estudiantes	aceptan tarjeta en la cafeteria	imprimir documentos
por que la tui	que necesito para hacer un erasmus	delegación de estudiantes	como puedo pagar en la cafeteria	imprimir trabajos
TUI	quiero hacer un intercambio	donde me puedo quejar	que menus hay en la cafeteria	fotocopias
por qué necesito la tui	programas de intercambio	quienes representan a los estudiantes	comer en la universidad	fotocopiadora

¿Qué es la tui?	programas internacionales de intercambio	quien representa a los estudiantes	donde puedo comer	donde hay una fotocopidora
¿Qué es la tui?	quiero irme a estudiar a otro país	representante de los estudiantes	hay sitios para comer	donde puedo imprimir
necesito una tarjeta para entrar a la universidad	quiero hacer un intercambio en otro país	delegados de clase	se aceptan tarjetas en la cafetería	impresión
¿necesito alguna acreditación?	quiero hacer un erasmus en europa	delegado	comer en el campus	reprografía
como acredito que soy estudiante	quiero hacer el erasmus en america	delegada de clase	comida en la universidad	¿dónde está la fotocopidora?
hay carnet de estudiante	erasmus	delegada	¿Qué puedo comprar en la cafetería?	donde esta la fotocopidora
quiero mi carnet de estudiante	quiero ir de erasmus	delegación estudiantil	donde esta el comedor	donde imprimir
carnet de estudiante	Me gustaría saber que necesito para el programa Erasmus	representante de los estudiantes	¿Cómo puedo pagar en la cafetería?	impresión de documentos
carnet	Quiero saber qué necesito para el programa Erasmus.	delegado de clase	¿Cuál es el menú del comedor?	maquina de copiar
quiero mi carné de estudiante	Quiero hacer el programa Erasmus en Europa	delegación	¿Cuál es el menú en la cafetería?	puedo imprimir
quisiera una identificación de estudiante	Quiero hacer Erasmus en Europa	delegacion de alumnos	cafeteria	donde puedo imprimir
Necesito mi identificación de estudiante	Quiero hacer Erasmus en USA.	delegación de alumnos	donde esta la cafeteria	donde puedo fotocopiar
tarjeta de estudiante	Quiero hacer Erasmus en América	delegación de alumnos	que puedo comer en la cafeteria	donde puedo hacer una fotocopia
Identificación del Estudiante	que nota necesito para hacer un erasmus	que es delegacion	quiero comer	quiero sacar una copia
¿Cómo puedo acreditar que soy estudiante?	me puedes decir el listado de países de erasmus	quiero saber quien representa a los estudiantes	puedo tomar cafe	donde esta reprografia
¿Cómo puedo acreditar que soy estudiante?	lista de países erasmus	quiero ser delegado	donde puedo tomarme un cafe	donde esta la copisteria
¿Cómo puedo acreditar que soy estudiante?	viajar	por favor dime quien representa a los estudiantes	donde puedo comer en la universidad	

¿Necesito acreditación?		dime como ser delegado	quiero ir a la cafeteria	
¿Necesito acreditación?		¿Qué es la delegación?	hay microondas	
necesito una tarjeta para llegar a la universidad		que es delegacion	hay microondas en la cafeteria?	
necesito una tarjeta para ingresar a la universidad		Quisiera saber quien representa al estudiante	hay un lugar para comer	
Se requiere una tarjeta para la admisión a la universidad.		Quiero saber a quién representan los estudiantes.	hay lugares para comer	
¿Qué es una tarjeta inteligente universitaria?		Quiero saber quien representa al estudiante	hay capacidad para comer	
que es tui		quiero saber quien ayuda a los estudiantes	Dónde está la cafetería	
		quiero representar	¿Dónde puedes tomar un café?	
		Di a quién representa el estudiante	donde puedo conseguir cafe	
		dime quien representa a los estudiantes	¿Dónde se puede comer en el campus?	
		por favor dígame quién representa al estudiante	donde puedo tomar cafe	
		por favor dime quien ayuda a los estudiantes	donde comer en el campus	
		¿Cómo puedo convertirme en representante?	quiero ir a un cafe	
			¿Tienes un microondas?	
			hay un microondas	
			¿La cafetería tiene microondas?	
			¿Hay un microondas en la cafetería?	

## Anexo 2: Apartados recogidos en las pruebas de interacción escrita: Trámites administrativos

Becas	Fechas exámenes	Matrícula
solicitar becas	fechas de exámenes	matricula
becas	calendario exámenes	matrícula
que becas hay	cuando son los exámenes	Matricula
ayuda al estudio	cuando me tengo que examinar	Matrícula
ayuda de dinero	fecha del examen	como matricularme
becas mec	fecha de examen	como hacer la matrícula
beca comunidad madrid	calendario de examen	como hacer la matricula
becas de la comunidad de madrid	cuando es el examen	como inscribirme en la universidad
beca	¿Cuándo debo examinarme?	como puedo hacer la matricula
solicitar beca	como se que días hay examen	no se hacer la matricula
qué tipo de becas están disponibles	cuando son los exámenes	ayuda para hacer la matrícula
que becas hay disponibles	como van los exámenes	informacion sobre la matrícula
Beca Comunidad de Madrid	quiero saber las fechas de exámenes	informacion sobre la matricula
ayuda de aprendizaje	hablame de las fechas de los exámenes	información sobre como matricularse
becas de la universidad de madrid	como se cuando tengo que hacer un examen	informacion sobre como matricularme
quiero que me den una beca	quiero saber el calendario de exámenes	informacion sobre como matricularme
como tengo que solicitar la beca	dame el calendario de exámenes	información matrícula
como puedo hacer para tener la beca	enseñame el calendario de exámenes	por favor, me dices como hacer la matricula
quiero una beca	¿Cómo puedo saber la fecha de mi examen?	me podrias decir como se hace la matricula
por favor dime como hacer la beca	¿Cómo saber qué días son los exámenes?	¿Como hacer la matricula?
aplicación de beca	¿Cómo puedo saber en qué días es el examen?	No se hacer la matricula
solicitar una beca	¿Cómo sé cuándo hacer la prueba?	ayuda para hacer la matricula
me gustaria recibir una beca	¿Cómo sabré cuándo debo tomar el examen?	quiero hacer la matricula
¿Cómo obtengo una beca?	¿Cómo sé cuándo tomar el examen?	me ayudas a hacer la matricula?
¿Cómo puedo obtener una beca?	Mostrar calendario de exámenes	
¿Cómo solicito la beca?	mostrar horario de examen	
¿Cómo puedo solicitar una beca?	quisiera saber la fecha del examen	
¿Cómo solicitar una beca?	Mostrar el calendario de exámenes	



como tengo que realizar la beca	quiero saber la fecha del examen	
como tengo que solicitar la vida	quiero saber las fechas de examen	
¿Cómo puedo obtener la beca?	¿Podrías decirme la fecha del examen?	
Por favor, dígame cómo recibir la beca.	por favor dime la fecha del examen	
Por favor, dime cómo obtener una beca.	dime la fecha del examen	
por favor dime como obtener la beca	quisiera saber el horario de examen	
	quiero saber el horario de examen	

### Anexo 3: Apartados recogidos en las pruebas de interacción escrita: Cambios de grupo

Solicitudes al director	
cambio de grupo	Solicitud de evaluación final
como me cambio de grupo	Cómo hacer una evaluación final
quiero cambiarme de grupo	Solo quiero una evaluación final.
cambio de grupo grande	evaluación final
cambio de grupo de teoria	hare la evaluacion final
cambio de tarde a mañana	solicitud de compensación
evaluacion final	solicitud de comensacion
cambiar de grupo	como puedo solicitar la evaluacion final
solicitud evaluacion final	como puedo hacer el cambio de grupo
evaluación final	quiero cambiarme de horario
quiero hacer la evaluación final	quiero cambiarme de grupo
como hacer la evaluación final	quiero hacer la evaluacion final
solo quiero hacer la evaluacion final	quiero escribir al director
intercambio de grupo	como puedo cambiarme de grupo
Cómo cambiar de grupo	¿Cómo solicito una calificación final?
como puedo cambiar de grupo	¿Cómo solicito una evaluación final?
como cambiar de grupo	¿Cómo puedo solicitar una evaluación final?
quiero cambiar de grupo	como puedo realizar la evaluacion final
cambios en el grupo principal	¿Cómo puedo cambiar de grupo?
cambio de grupo teórico	me gustaría cambiar mi horario
de la tarde a la mañana	quiero cambiar mi horario
Me gustaría escribir una carta al director.	quiero escribirle al director
cambia de la tarde a la mañana	

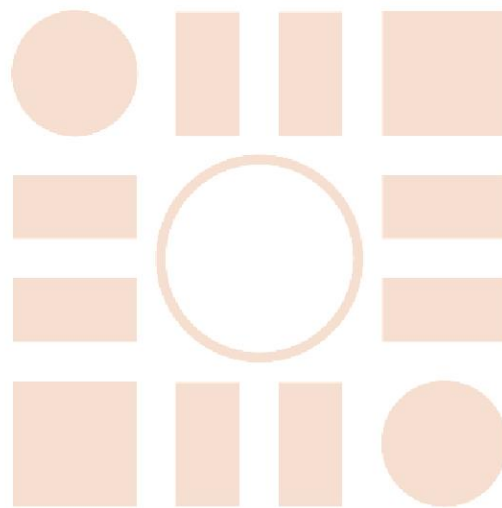
#### Anexo 4: Apartados recogidos en las pruebas de interacción escrita: Wifi y VPN

Acceso wifi	
como conectarme al wifi	hay wifi en la zona
que es eduroam	¿Qué es el wifi universitario?
eduroam	¿Cuánto cuesta el wifi en la universidad?
wifi en la universidad	que es eduroam
hay wifi en la universidad	no puedo conectarme al wifi
wifi	EDUROAM
cual es el wifi de la universidad	no se conectarme al wifi
Cómo conectarse a Wi-Fi	cual es el wifi de la universidad
como conectarse a la red wifi	por favor dime como conectarme al wifi
como conectar wifi	quiero usar el wifi
red inalámbrica en el campus	No puedo conectarme a la red Wi-Fi
WiFi en el campus	Por favor, dime cómo conectarme a wifi.
¿La universidad tiene wifi?	por favor dime como conectarme a wifi
¿Hay wifi en el campus?	no puedo conectarme a wifi
¿Tienes wifi en tu universidad?	¿Qué es WiFi universitario?

## Anexo 5: Apartados recogidos en las pruebas de interacción escrita: Localizaciones

Dónde está		Laboratorios
donde esta la eps	quiero saber como ir a la universidad	que es un laboratorio
donde esta la escuela politecnica superior	quiero saber como entrar a la universidad	que es el laboratorio
escuela politécnica superior	no se como ir a la universidad	clases de laboratorio
Localización	no se como entrar a la universidad	donde estan los laboratorios
como llegar a la eps	puedo ir en autobus	laboratorio
como llegar a la escuela politécnica superior	puedo ir a la universidad en coche	laboratorios
qué transporte público hay para llegar a la universidad	¿Puedo conducir a la universidad?	donde estan las clases de laboratorio
como llegar a la universidad	¿Hay estacionamiento en el campus?	donde se encuentra el laboratorio
donde esta la universidad	Hay estacionamiento disponible en la universidad.	clases de laboratorio
donde esta la facultad	¿La universidad tiene estacionamiento?	horas de laboratorio
¿Dónde está ubicada la universidad?	hay parking en la zona	¿dónde están los laboratorios?
¿Dónde está la facultad de tecnología?	que autobus va a la universidad	donde esta el laboratorio
como llegar a eps	como llegas a eps	¿Dónde están las clases de laboratorio?
¿Cuál es el transporte público a la universidad?	que bus llega a la universidad	por que tengo laboratorios
qué transporte público hay para acceder a la universidad		que son los laboratorios
¿Qué tipo de transporte público puedes usar para llegar a la universidad?		por que tengo que ir a laboratorios ademas de a clase
como ir a la universidad		por que hay clases practicas
donde esta ubicada la universidad		que son las clases practicas
donde está la Universidad		
quiero saber como llevar a la universidad		
no se como llegar a la universidad		
puedo ir en coche		
puedo ir en bus		
puedo ir en tren		
puedo ir en coche a la universidad		
hay parking en la universidad		

Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá