

El objetivo de la práctica es crear un chatbot con la librería de python program-y para que lea hojas de especificaciones AIML:

1. Crea una cuenta de pandorabots para crear un chatbot que permita jugar al juego piedra, papel o tijeras:

- 1.1. Genera una categoría para que, cuando el usuario introduzca el patrón QUIERO JUGAR, se genere un topic llamado “piedrapapeltijeras”, con dos contadores, uno de victorias y otro de derrotas:

```
<category>
  <pattern>QUIERO JUGAR</pattern>
  <template>
    Ok, vamos a jugar al piedra, papel o tijeras. Empieza tú.
    <think>
      <set name="topic">piedrapapeltijeras</set>
      <set name="victorias">0</set>
      <set name="derrotas">0</set>
    </think>
  </template>
</category>
```

- 1.2. Crea el topic piedrapapeltijeras de manera que, cuando el usuario introduzca una elección, se conteste con un valor aleatorio y se incremente el número de victorias o derrotas acorde a la elección del usuario y el valor aleatorio. Puedes utilizar la etiqueta “random” para generar una acción aleatoria, por ejemplo:

```
<condition name="eleccion">
  <li value="piedra">
    <random>
      <li>Papel<srai>INCREMENTAR DERROTAS</srai></li>
      <li>Tijeras<srai>INCREMENTAR VICTORIAS</srai></li>
      <li>Piedra</li>
    </random>
  </li>
  ...
</condition>
```

- 1.3. Crea una categoría para que, cuando el usuario introduzca el patrón “TERMINAR”, se borre el topic creado en el punto anterior, y se dé por terminada la partida visualizando el número de victorias y derrotas.

- 1.4. Crea una categoría para que, cuando el usuario introduzca el patrón “DIME LA PUNTUACIÓN”, se muestre por pantalla el número de victorias y derrotas del usuario.

2. Ahora, vamos a poner el chatbot creado en el punto 1 en un entorno de backend. Para ello, instala un entorno virtual con conda compatible con la librería program-y (python 3.7) y actívalo:

```
conda create -n "aiml" python=3.7
conda activate aiml
```

3. Instala la librería program-y y la librería Flask 2.1.0:

```
pip install programy  
pip install Flask==2.1.0
```

4. De los ficheros adjuntos a la práctica 2, copia el archivo piedra_papel_tijeras.py, que contiene a una carpeta y crea la carpeta aiml-source. Dentro de la carpeta aiml-source crea el fichero piedra-papel-tijeras.aiml y añade el código AIML necesario para jugar al juego Piedra Papel o Tijeras.
5. De los ficheros adjuntos a la práctica 2, abre el fichero ppt_server.py (servidor de piedra papel o tijera) y examina su contenido. ¿Qué piensas que hace el código python que incluye este fichero? Prueba el comando netcat <https://nmap.org/ncat/> para comunicarte con el servidor. Por ejemplo

```
Netcat localhost 8000
```

Puedes encontrar más información aquí:

<https://github.com/AIML-Foundation/AIML-2.0-Spec/blob/master/aiml.md>

<http://www.aiml.foundation/doc.html>

EVALUACIÓN DE LA PRÁCTICA:

Se evaluarán las respuestas con una nota alfanumérica:

A= sobresaliente [10-9]

B= notable [8-7]

C= suficiente [6-5]

D= insuficiente [<5]

E= no entregado