INTELIGENCIA ARTIFICIAL

E.T.S. de Ingenierías Informática y de Telecomunicación

Práctica 1
Curso 2020/21

Agentes Conversacionales

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL
UNIVERSIDAD DE GRANADA
Curso 2020-2021

1. Objetivo

El objetivo de esta práctica es familiarizarse con el diseño de agentes conversacionales, en particular con un lenguaje que ha sido especialmente diseñado para la descripción de bases de conocimiento en la construcción de agentes conversacionales, como es AIML.

Un agente conversacional es un agente software, diseñado con el objetivo de mejorar la comunicación entre los seres humanos y las máquinas, haciendo que estas últimas sean capaces de manejar conceptos del lenguaje natural (palabras, frases) como símbolos y reglas que actúan sobre estos símbolos.

Los agentes conversacionales tienen multitud de aplicaciones, siendo su principal labor en estos últimos años la de asistente para los seres humanos. Es muy conocido **SIRI**, incluido en los dispositivos desarrollados por la empresa ***Apple***. La relevancia de este tipo de agentes se pone de manifiesto en el creciente número de dispositivos y páginas de empresas que los incluyen, por ejemplo, **Google Assistant** una aplicación de características similares a SIRI desarrollada por ***Google*** para dispositivos Android, o **CORTANA** la aplicación desarrollada por ***Microsoft***, o los asistentes incluidos en las páginas web de empresas muy importantes como ***Ikea***. La propia ***Universidad de Granada*** tiene su asistente llamado **Elvira** que te ayuda a encontrar información relativa a la Universidad.

Muy recientemente, empresas como Google (con **Google Home**), Apple (con **HomePod**) o Amazon (con **Amazon Echo**) han sacado dispositivos empotrados en altavoces dedicados al control domótico mediante voz. Las versiones comerciales de estos dispositivos aún están en una fase muy inicial, y con ellos no se puede establecer una conversación como tal, son más bien modelos de pregunta y acción (como respuesta a la pregunta). Están empezando a poder establecer conversaciones cortas y ya se avanza en la línea de mantener conversaciones más parecidas a las que se esperaría entre seres humanos.

AIML no es el único lenguaje definido para el desarrollo de agentes conversacionales, ya que podemos encontrar otros como ***API.AI*** usado para el desarrollo de **Google Assistant** (una aplicación Android que incorpora una gran flexibilidad debida a su capacidad para aprender). Pero AIML es un lenguaje mejor desde el punto de vista docente y del que es más fácil encontrar información.

En el primer seminario de la asignatura se ha impartido un tutorial sobre el uso de este lenguaje, indicando cuáles son las sentencias más habituales e ilustrando su uso con algunos ejemplos y ejercicios. El **objetivo de esta primera práctica** consiste en definir bases de conocimiento en AIML para establecer algunas conversaciones sobre temas concretos, usando ***program-ab***, un intérprete para el lenguaje ***AIML 2.0*** de código abierto y que se puede descargar desde https://prado.ugr.es/.

2. Detalles de la práctica

Como se ha indicado anteriormente, la práctica está orientada al uso del lenguaje AIML y a mostrar sus posibilidades como medio para desarrollar agentes conversacionales. En concreto, se pide construir un agente conversacional que sea capaz de seguir una conversación sobre dos posibles aplicaciones: (a) **interpretar de forma flexible** preguntas y responder **a un conjunto muy reducido** de preguntas básicas sobre

fechas, y (b) desarrollar una aplicación que juegue el papel de un asistente conversacional para la atención al cliente en la gestión de citas en una empresa. A continuación, detallamos más cada una ellas.

2.1. Preguntas básicas sobre información de fechas.

Para facilitar el uso de las operaciones sobre gestión de fechas, en el material de la práctica (dentro del directorio aiml del bot) se proporciona el fichero AIML ***dates_es.aiml***. El contenido de este fichero no está documentado, pero es fácil de interpretar una vez conocido el seminario y tutorial de AIML de las sesiones de prácticas. El objetivo en esta parte es aproximarse a cómo se representa el conocimiento en AIML, mediante la escritura de las reglas necesarias, basándose en las reglas escritas en este fichero, para responder a las siguientes 5 preguntas básicas:

1. **P: En qué estación estamos.**
    a. R: estamos en invierno/primavera/otoño/verano
2. **P: En qué fase del día estamos.**
    a. R:Ahora estamos en la <manyana/tarde/noche>
3. **P: Qué día de la semana es <hoy/manyana/pasado manyana>**. Observar que el intérprete de AIML no acepta la "ñ", adoptamos el convenio de sustituirla por "ny" en toda la práctica.
    a. R: <hoy/manyana/pasado manyana> es [Lunes … Domingo]
4. **P: Qué fecha será dentro de una semana.**
    a. R: Dentro de una semana será <fecha de mañana relativa al día actual>
5. **P: Qué fecha será el próximo [Lunes Martes … Domingo]**
    a. R: El próximo [Lunes Martes … Domingo] será <dia> de [Enero … Diciembre] del <año>.

Las tareas en este nivel consisten en escribir las reglas en el fichero Nivel0.aiml considerando las siguientes obligaciones y restricciones:

- Debe existir una única regla para representar la respuesta y, por cada posible pregunta alternativa con la misma respuesta, se deben permitir varias reglas para poder añadir flexibilidad al conjunto de frases aceptadas e interpretadas por el bot.
- Utilizar al menos comodines, variables y la estructura <srai> para que el bot pueda admitir y responder ante la entrada de la forma más flexible posible, capturando distintas posibilidades para cada pregunta. También pueden usarse otros componentes del lenguaje

que se han explicado en clase como sets o maps.

- Restricciones: Aparte de usar en el <pattern> de cada regla comodines, se pide además el uso de funcionalidades extendidas de AIML para expresiones regulares más comunes, descritas en el Anexo 2 (documento aparte), punto 2.3. Esto incluye utilizar [], () y prefijos/sufijos.

Por ejemplo, considerando las dos reglas de más abajo, observar que la primera regla tiene representada la respuesta a la pregunta "Hablame un poco de ti". Además, la segunda regla tiene un patrón que acepta variaciones sintácticas de esa frase y con <srai> redirige la respuesta de esa regla a la primera regla (para entender este ejemplo consultar la presentación sobre AIML de la clase de prácticas).

```
<category>
    <pattern>HABLAME UN POCO DE TI</pattern>
    <template> Respuesta de Pregunta 1</template>
</category>

<category>
    <pattern> ^ [hablame cuentame] ^ ti ^ </pattern>
    <template> <srai>HABLAME UN POCO DE TI</srai></template>
 </category>
```

No sólo se pide que sepa responder a la pregunta en su formulación original, sino que sea capaz de responder correctamente usando variantes de la misma pregunta. Por ejemplo, si la pregunta original es "¿En qué estación estamos?" debe también saber responder a otras dos variantes al menos, como por ejemplo "¿Cuál es la estación actual?" o "¿En qué época del año estamos?".

2.2. Agente conversacional para la atención al cliente en la gestión de citas.

Uno de los problemas con los que actualmente se enfrentan empresas con una gran cantidad de clientes, es la provisión de un servicio de atención al cliente de calidad que no produzca rechazo en el usuario a la hora de interactuar. El principal inconveniente es que la mayor parte de los sistemas automáticos de atención al cliente incorporan una interfaz hablada basada fundamentalmente en opciones de menú y los usuarios agradeceríamos que la interacción fuera más próxima a la que realizamos con un humano.

Por tanto, en esta parte de la práctica nos planteamos desarrollar un bot conversacional que sirva como asistente inteligente para interactuar con humanos en la gestión de citas de una clínica dental (por ejemplo). El desarrollo lo haremos en 3 etapas: en una primera escribiremos las reglas necesarias para dar soporte a operaciones de consulta de las citas y para entablar una conversación básica con un usuario que desea agendar una cita en una fecha concreta (por ejemplo "12 de Febrero"), en una segunda fase el bot podrá aceptar descripciones de fechas más elaboradas (por ejemplo "mañana por la tarde", "el próximo lunes por la mañana") y en una tercera se extenderá el bot desarrollado para que responda de la forma más amigable posible sobre la disponibilidad de horas para poder asignarlas.

Antes de hacer una descripción más detallada de estas etapas, vamos a fijar el marco de trabajo en el que se va a desarrollar esta parte de la práctica.

2.2.1. Planteamiento del problema

Imaginemos un usuario que contacta por teléfono con una clínica dental y al otro lado contesta un asistente digital. El usuario desea agendar una cita para una operación dental, no importa cuál en esta práctica, y el asistente tratará de ayudarle a determinar qué día y hora puede cerrar la cita, informándole de la forma más amigable posible (es decir, lo más fácilmente comprensible por parte del humano) sobre qué días tiene disponibles y, entre esos días, qué horas. El usuario puede tener muy claro qué día quiere la cita o puede que no lo tenga tan claro, en cualquier caso, el bot actuará en consecuencia dependiendo del estado del usuario.

Vamos a asumir que el bot parte de una representación de una agenda de eventos como un diccionario (map) AIML llamado ***eventos.txt*** en el que se guarda la información sobre la fecha, hora y nombre del evento (en general aparecerá como "NOLIBRE" o "LIBRE"). Por ejemplo, la siguiente tabla muestra un evento en el que el intervalo de las 17:00 a las 18:00 y el intervalo de las 18:00 a las 19:00 del día 20 de febrero de 2021 están "Ocupados".

eventos.txt
20_02_2021:LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE LIBRE NOLIBRE NOLIBRE LIBRE LIBRE LIBRE LIBRE LIBRE

En la tabla se puede apreciar que el diccionario eventos.txt utiliza como clave la fecha en que se produce el evento y luego tiene una lista de 24 posiciones que indican los intervalos de 1 hora de cada día. Esta lista comienza en la hora 00:00, siguiendo por, 01:00, 02:00, etc. Por tanto, **la hora 17:00 corresponde a la posición 18** de esa lista. Las 00:00 corresponde a la posición 1 (la primera) y las 23:00 corresponde a la posición 24 (la última).

Es importante tener en cuenta que cuando se desarrollen las reglas oportunas para el agente conversacional, se debe asumir que el usuario con el que interactúa habla castellano y tiene una representación de fechas

habitual en España que sigue el formato *"dd **de** MMMMMMMMM **del** yy"*. Finalmente observar que la **representación de fechas almacenadas como claves en el diccionario es de formato "dd_mm_aaaa", y diferente de la usada en la conversación con el humano.**

Para facilitar el uso de las operaciones sobre gestión de fechas, en el material de la práctica (dentro del directorio aiml del bot) se proporciona el fichero AIML ***dates_es.aiml*** el contenido de este fichero no está documentado, por lo que se recomienda entender el contenido de las operaciones implementadas en él mediante experimentación. En resumen, el alumno probará el funcionamiento de las funciones en el terminal.

Además, se proporcionan dos ficheros de utilidades:

- utilidades.aiml: que consiste en una librería de utilidades que pueden facilitar la implementación de las prácticas. Las operaciones incluidas en este fichero están recogidas en el documento Anexo3_UtilidadesAIML.pdf proporcionado en el material de esta práctica.
- utilidades_2021.aiml: que consiste en una librería de utilidades específicas para esta práctica y destinadas a simplificar el trabajo del alumno. Las operaciones incluidas en este fichero están recogidas en el documento Anexo4_UtilidadesAIML_2021.pdf proporcionado en el material de esta práctica.

A partir de esta información se van a desarrollar las tres fases comentadas anteriormente a las que llamaremos Nivel 1, Nivel 2 y Nivel 3.

2.2.2. Nivel 1: consultas sobre calendario de citas, con especificación de fecha sencilla y conversación básica

Las tareas a realizar en este nivel son:

1. Escribir las reglas necesarias para poder realizar las siguientes consultas **dado** un día representado como "dd **de** MMMMMMMMM **del** yy"
    a. informar sobre si el día está libre o no (un día está libre si tiene al menos una franja horaria no ocupada, es decir, con la palabra "LIBRE" entre las 08:00 y las 20:00, en otro caso está ocupado)
    b. devolver la lista de franjas horarias libres en un día, es decir, una secuencia de horas en el formato hh:mm representando las horas libres de un día (entre las 08:00 y las 20:00). Por ejemplo: 08:00 09:00 10:00 11:00 14:00 15:00 18:00. Observar que entre cada hora hay un espacio. Tener en cuenta que, como se ha descrito, se proporcionan utilidades que permiten

calcular listas de números representando la posición de las horas de un día. Se recomienda usar estas funciones de utilidades. Observar también que una lista de posiciones horarias, no se corresponde exactamente con una lista de horas. Por ejemplo, la lista de posiciones 11 12 15 16 19 representaría que las 10:00 11:00 14:00 15:00 y 18:00.

c. devolver una lista de franjas libres solo por la mañana (las horas de la mañana son de 08:00 a 12:00 ambas inclusive)

d. devolver una lista de franjas libres solo por la tarde (las horas de la tarde son desde las 13:00 hasta las 20:00 ambas inclusive).

2. Escribir las reglas necesarias para que el bot pueda entablar una conversación con un usuario que desea agendar una cita en una fecha concreta, especificada en un formato sencillo, en un contexto en el que el cliente tiene claro qué día quiere la cita, aunque le puede dar igual la hora.

Funcionalidad mínima. Para poder superar este nivel, al menos debe implementarse una regla para cada una de las siguientes funciones.

Informar del estado de un día
Entrada: un día representado como dd **de** MMMMMMMMM del yy
Salida: "SI" si el día está libre, "NO" si no lo está.

Ejemplo:
Comando en la terminal del bot LIBRE 07 de JULIO del 21
Salida en la terminal: SI (o NO)

Informar de las franjas libres en un día
Entrada: un dia representado como dd de MMMMMMMMM del yy
Salida: una lista de horas en formato HH:MM separadas por espacios. O la cadena "***EMPTYLIST***" en caso de que la lista esté vacía

Ejemplo:
Comando: HORASLIBRES 07 de JULIO del 21
Salida: 09:00 14:00 15:00 18:00.

Informar de las franjas libres por la mañana en un día
Entrada: un dia representado como dd **de** MMMMMMMMM **del** yy
Salida: una lista de horas en formato HH:MM separadas por espacios. O la cadena "***EMPTYLIST***" en caso de que la lista esté vacía

Ejemplo:
Comando: HLMANYANA 07 de JULIO del 21
Salida: 08:00 11:00

Informar de las franjas libres por la tarde en un día
Entrada: un dia representado como dd **de** MMMMMMMMM del yy
Salida: una lista de horas en formato HH:MM separadas por espacios. O un valor apropiado en caso de que la lista esté vacía

Ejemplo:
Comando: HLTARDE 07 **de** JULIO del 21
Salida: 14:00 15:00 18:00.

Un ejemplo de conversación para superar el nivel 1 sería el siguiente (H = Humano, R = Bot)

H: Quisiera una cita para **el 20 de febrero del 21**

R: Muy bien voy a comprobarlo, espere un momentico….
Puede que ese día esté ocupado en cuyo caso el bot contestaría

Lo siento no puedo está ocupado, ¿desea otro día?

H: Sí (o No)

R: *<el bot contesta dependiendo de la respuesta del cliente, preguntando por un nuevo día o terminando la conversación>*

[...]

Puede que ese día queden algunos huecos libres, en cuyo caso el bot contestaría, por ejemplo

Si tiene huecos libres por la mañana y por la tarde

R1: Pues **por la mañana** tengo los siguientes huecos libres 10:00 y 11:00 , y **por la tarde** 14:00 15:00 y 18:00

Si tiene huecos solo por la mañana

R2: Pues por la mañana tengo los siguientes huecos libres 10:00 y 11:00 y toda la tarde ocupada

Si tiene huecos solo por la tarde

R3: Pues la mañana la tengo ocupada y por la tarde los siguientes huecos libres 14:00 15:00 y 18:00

Si tiene la mañana libre

R4: Pues la mañana está libre y por la tarde …

Si tiene la tarde libre

R5: <respuesta análoga>

¿Quiere alguno en concreto?

En este contexto el usuario puede responder de distinta forma, bien determina la hora o le da igual.

H:Sí a las 09:00

<el bot contesta adecuadamente, si la hora no es correcta debe informar al usuario de que la hora es errónea y volver a preguntar si quiere alguna hora concreta,

R: la respuesta debe ser amigable, por ejemplo, "La hora es incorrecta, recuerda que <informar sobre huecos libres>"

< en otro caso incluye la cita en la agenda y finaliza la conversación o pide si el usuario quiere otra>

H: Me da igual

R: *<en este caso el bot rellena el siguiente hueco libre e informa al humano de la decisión>*

2.2.3. Nivel 2: Consultas sobre calendario con especificación de fechas elaborada

Las tareas a realizar en este nivel son:

1. Escribir las reglas necesarias para que el bot pueda entablar una conversación con un usuario que desea agendar una cita en una fecha concreta, **especificada mediante la gramática descrita más abajo**, en un contexto en el que el cliente tiene claro qué día quiere la cita.

Un **aspecto fundamental en el nivel 2** es la flexibilidad del bot para aceptar una descripción de fecha, por ejemplo ***pasado mañana, el próximo lunes por la tarde, mañana por la mañana, etc.*** Para especificar todas las posibles entradas que representan una descripción de una fecha se tendrán en cuenta las siguientes reglas gramaticales descritas en notación BNF.

```
<DESCRIPCION FECHA>::= <ESPECIFICA DIA> <COMPLEMENTO>
<ESPECIFICA DIA>::= <FECHA CONCRETA> | HOY | MANYANA| PASADO MANYANA|
PROXIMO <DIA SEMANA> | SIGUIENTE <DIA SEMANA>
<FECHA CONCRETA>::= 13 de Febrero del 21
<DIA SEMANA>::= LUNES | MARTES | MIERCOLES | JUEVES | VIERNES
<COMPLEMENTO>::= POR LA <FIN COMPLEMENTO> |
                 <FIN COMPLEMENTO> | A PARTIR DE LAS <HORA>
<FIN COMPLEMENTO>::= MANYANA | TARDE
<HORA>::= HH:00
```

Para superar este nivel es necesario al menos implementar todas las funcionalidades que se describen a continuación, la nota máxima de este nivel dependerá de la implementación que realice el alumno sobre todas las posibilidades que ofrece esta gramática. La conversación para este nivel puede ser similar a la mostrada para el Nivel 1.

Funcionalidad mínima. Para poder superar este nivel, al menos debe implementarse una regla para cada una de las siguientes funciones.

Informar del estado de un día especificado con <ESPECIFICA DIA>
Entrada: un día representado como ***<ESPECIFICA DIA>***
Salida: "SI" si el día esta libre, "NO" si no lo está.

Ejemplo:
Comando en la terminal del bot LIBRE PROXIMO LUNES
Salida en la terminal: SI (o NO)

Informar de las franjas libres en un día especificado con <ESPECIFICA DIA>
Entrada: un día representado como ***<ESPECIFICA DIA>***
Salida: una lista de horas en formato HH:MM separadas por espacios. O un valor apropiado en caso de que la lista esté vacía

Ejemplo:
Comando: HORASLIBRES PASADO MANYANA
Salida: 09:00 14:00 15:00 18:00.

Informar de las franjas libres por la mañana en un día especificado con <ESPECIFICA DIA>
Entrada: un día representado como <ESPECIFICA DIA> POR LA MANYANA
Salida: una lista de horas en formato HH:MM separadas por espacios. O un valor apropiado en caso de que la lista esté vacía

Ejemplo:
Comando: HORASLIBRES MANYANA POR LA MANYANA
Salida: 08:00 11:00

Informar de las franjas libres por la tarde en un día especificado con <ESPECIFICA DIA>
Entrada: un día representado como <ESPECIFICA DIA> POR LA TARDE
Salida: una lista de horas en formato HH:MM separadas por espacios. O un valor apropiado en caso de que la lista esté vacía

Ejemplo:
Comando: HORASLIBRES HOY POR LA TARDE
Salida: 13:00 18:00

Informar de las franjas libres a partir de una hora dada en un día especificado con <ESPECIFICA DIA>
Entrada: un día representado como <ESPECIFICA DIA> A PARTIR DE LAS <HH:MM>
Salida: una lista de horas en formato HH:MM separadas por espacios. O un valor apropiado en caso de que la lista esté vacía

Ejemplo:
Comando: HORASLIBRES ELPROXIMO JUEVES A PARTIR DE LAS 14:00
Salida: 13:00 18:00

2.2.4. Nivel 3: Mejora de las respuestas proporcionadas por el asistente conversacional.

En esta última fase, debemos adaptar la respuesta al usuario de una forma amigable, extendiendo y mejorando las respuestas que se han proporcionado en los niveles anteriores. Dependiendo de las horas libres para un día y de su distribución en intervalos de horas consecutivas, el bot ofrecerá una respuesta amigable para el humano.

Por ejemplo, si consideramos la siguiente lista de horas

09:00 10:00 11:00 13:00 15:00 16:00 17:00 19:00 20:00

Observamos que tiene un intervalo de horas consecutivas de [09:00 a 11:00], un intervalo de un único valor a las [13:00], un intervalo de [15:00 a 17:00] y otro de [19:00 a 20:00]).

El bot debería responder POR LA MANYANA TENGO LIBRE DESDE LAS ***NUEVE*** HASTA LAS ***ONCE***, Y POR LA TARDE TENGO LIBRE A LA ***UNA***, DE LAS ***TRES*** A LAS ***CINCO*** Y DE LAS ***SIETE*** A LAS ***OCHO***.

Observar que en este caso es necesario determinar qué intervalos de horas consecutivas hay en la lista y "traducir" cada hora a su correspondiente adjetivo numeral cardinal.

Por parte del humano, debemos contemplar la posibilidad de que cuando nos indique la hora pueda responder:

1. A las 09:00
2. A las NUEVE DE LA MANYANA
3. A las OCHO DE LA TARDE

La conversación que se pide en este nivel es similar a las de los puntos anteriores, pero con las respuestas mejoradas tal y como se ha indicado.

La idea es crear nuevas reglas, basadas en la información devuelta por las reglas definidas en el nivel anterior, para poder responder adecuadamente al humano.

3. Evaluación

3.1 Definición de los niveles de dificultad

Hemos diseñado un modelo de evaluación para que el alumno decida con qué intensidad y a qué nivel desea implicarse en su elaboración. Obviamente, a mayor nivel de implicación, el alumno opta a una mayor

calificación en la práctica. Por tanto, hemos definido 4 niveles en la entrega de esta primera práctica que son, de menor a mayor implicación, los siguientes:

(a) **Nivel 0**: Entrega del conocimiento necesario para responder a las preguntas básicas sobre fechas. En este caso, el alumno puede optar hasta **TRES** puntos sobre diez.

(b) **Nivel 1**: Incorporación del conocimiento necesario para que el agente pueda cumplir con lo descrito anteriormente para el Nivel 1 en el apartado 2.2.2. Como mínimo debe incorporar las funcionalidades descritas en el apartado 2.2.2 y entablar una conversación básica para decidir qué día agendar la cita. En este caso, el alumno puede optar hasta una calificación de **SEIS** puntos sobre diez.

(c) **Nivel 2** Incorporación del conocimiento necesario para que el agente pueda cumplir con lo descrito anteriormente para el Nivel 2 en el apartado 2.2.3. Como mínimo debe incorporar todas las funcionalidades descritas en el apartado 2.2.3 (esto permitiría alcanzar la calificación de **SIETE)** y la calificación hasta **OCHO** dependerá de **las posibilidades que el alumno haya implementado, descritas** por la gramática del punto 2.2.3. Para esta opción, el alumno puede obtener hasta una calificación de **OCHO** sobre diez.

(d) **Nivel 3**: Incorporación del conocimiento necesario para que el agente pueda cumplir con lo descrito anteriormente para el Nivel 3 en el apartado 2.2.4. El alumno puede obtener hasta un **DIEZ** sobre diez si completa adecuadamente este nivel**.**

No se considerarán aquellas entregas que no puedan ubicarse en uno de estos niveles; es decir, estas son las únicas posibilidades de entrega. Para que un nivel pueda ser valorado, es necesario que se hayan superado todos los niveles anteriores.

Un ejemplo de entrega inválida, es aquella donde el alumno entrega el nivel 1 y no realiza el nivel 0, ya que esta última es obligatoria en todos los niveles.

3.2 ¿Cómo se evaluará cada uno de los problemas?

Al final de este guion se establece una fecha límite para la entrega de la práctica. Antes de dicha fecha, el alumno debe subir el material que más adelante se indica conteniendo el conocimiento necesario para resolver los problemas hasta el nivel que ha decidido. En una segunda fecha posterior, que también se indica al final de este guion se entregará un cuestionario de autoevaluación de la práctica.

Hemos definido una forma de evaluación para cada uno de los niveles definidos en la entrega. A continuación, se indica cómo se evaluará cada nivel:

Nivel 0: Será evaluado por el profesor. De no superar este nivel la práctica estará suspensa y no se considerará.

Nivel 1: Para llegar a este nivel en la evaluación, debe haberse superado el nivel anterior. En este caso, la evaluación es simple: junto al **cuestionario de autoevaluación**, se **proporcionará un fichero de eventos predefinido** que servirá como base de datos inicial para las operaciones en este nivel y en los siguientes. Se pedirá al estudiante que haga uso del bot para comprobar que se realizan correctamente las operaciones de consulta y modificación de fechas que se han implementado, así como exponer al bot en una conversación básica en los términos descritos para el Nivel 1.

El estudiante sumará **TRES** puntos más a su calificación si realiza bien todas las tareas encomendadas (teniendo en cuenta tanto el cuestionario como las operaciones de evaluación que realice el profesor), y su calificación será de **SEIS** puntos.

En caso de una resolución no satisfactoria del nivel 1, la evaluación termina y la calificación del alumno será de **TRES** puntos.

El profesor comprobará si el contenido del cuestionario de autoevaluación coincide con las respuestas reales del bot, caso de no ser así la práctica se considerará directamente suspensa.

Nivel 2: Al igual que antes, debe haberse superado el nivel 1 con una puntuación de **SEIS** puntos para pasar a evaluar este nivel, y la forma de evaluarlo estará basada en las respuestas del alumno en el cuestionario de autoevaluación y en las operaciones con el bot del alumno que el profesor estime oportunas para comprobar el grado de completitud de implementación de la gramática. En todo caso, durante la conversación con el bot, el agente deberá responder con coherencia. Si el agente mantiene la conversación y la secuencia de respuestas son correctas, el alumno superará este nivel, y añadirá

- **UN** punto si ha implementado todas las funcionalidades descritas para el Nivel 2.
- **Una puntuación entre 0 y 1** dependiendo del grado de completitud de la implementación de la gramática.

Por tanto se pueden obtener **DOS** puntos como máximo y su calificación máxima será de **OCHO**. Si no es así, si ha obtenido **UN** punto por las funcionalidades, podrá continuar con la evaluación. En caso contrario, la evaluación habrá terminado y la calificación del alumno será de **SEIS** puntos.

Nivel 3: Para acceder a este nivel, el agente presentado para su evaluación ha debido superar el nivel 2 con una puntuación de **SIETE** puntos. El proceso de evaluación será similar al del nivel anterior Si el alumno supera las tareas descritas para el Nivel 3, su calificación final será de hasta **DOS** puntos adicionales a la obtenida en el Nivel 2, si no es así la evaluación termina y su calificación será la suma de la obtenida en el Nivel 2 a la calificación que determine el profesor para el Nivel 3. Se optará a la calificación máxima de **DIEZ** si se superan las tareas requeridas en el Nivel 3 y se ha obtenido un OCHO en el Nivel 2..

3.3. ¿Qué hay que entregar?

Antes de que termine el plazo de entrega fijado en este guion, el alumno deberá subir a la plataforma de la asignatura en https://prado.ugr.es, un archivo comprimido con zip, llamado "practica1E.zip". Este archivo debe tener la misma estructura en carpetas que cuelga de la carpeta "mybot" donde **obviamente las carpetas "aiml", "aimlf", "sets" y "maps" contienen los ficheros necesarios** para resolver la práctica al nivel que ha decidido el alumno. **Si el alumno ha creado sets o maps adicionales a los proporcionados debe incluirlos** también en la entrega. Es obligatorio entregar, dentro del archivo comprimido, **un fichero aiml por cada uno de los niveles resueltos**, denominados de la siguiente forma:

1. Nivel0.aiml: para las reglas del Nivel 0.
2. Nivel1.aiml: para las de Nivel 1.
3. Nivel2.aiml: para las de Nivel 2.
4. Nivel3.aiml: para las de Nivel 3.

Nota: es muy importante repetir que se entrega el contenido COMPLETO del directorio "mybot", pero en ningún caso el directorio del "program-ab" completo, si una entrega no se ajusta a este requisito se penalizará correspondientemente.

3.4. Observaciones Finales

Esta **práctica es INDIVIDUAL** y trata de establecer la capacidad del alumno para desarrollar una base de conocimiento usando el lenguaje AIML. El profesorado para asegurar la originalidad de cada una de las entregas, someterá a estas a un procedimiento de detección de copias.

En el caso de detectar prácticas copiadas, los involucrados (**tanto el que se copió como el que se ha dejado copiar**) tendrán suspensa la asignatura. Por esta razón, recomendamos que en ningún caso se intercambie código entre los alumnos. No servirá como justificación del parecido entre dos prácticas el argumento *"es que la hemos hecho juntos y por eso son tan parecidas"*, o "como estudiamos juntos, pues...", ya que como se ha dicho antes, **las prácticas son INDIVIDUALES**.

Como se ha comentado previamente, el objetivo de la defensa de prácticas es evaluar la capacidad del alumno para enfrentarse a este problema. Por consiguiente, se asume que todo el código que aparece en su práctica ha sido introducido por él por alguna razón y que dicho alumno domina perfectamente el código que entrega. Así, si durante cualquier momento del proceso de defensa el alumno no justifica adecuadamente algo de lo que aparece en su código, la práctica se considerará copiada y tendrá suspensa la asignatura. Por esta razón, aconsejamos que el alumno no incluya nada en su código que no sea capaz de explicar qué misión cumple dentro de su práctica y que revise el código con anterioridad a la defensa de prácticas.

Por último, las prácticas presentadas en tiempo y forma, pero no defendidas por el alumno, se considerarán como no entregadas y el alumno obtendrá la calificación de 0. El supuesto anterior se aplica a aquellas prácticas

no involucradas en un proceso de copia. En este último caso, el alumno tendrá suspensa la asignatura.

4. Entrega de la práctica

La fecha tope para la entrega será **el LUNES 5 DE ABRIL DE 2021 antes de las 23:00 horas y el 6 DE ABRIL estará disponible la entrega para el cuestionario autoevaluación hasta EL 8 DE ABRIL a las 23:00.**