



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Inteligencia Artificial:

Práctica 4. Proyecto individual de Prolog.

Cheuk Kelly Ng Pante
(alu0101364544@ull.edu.es)



Índice:

1. Introducción.	2
2. Proyecto.	2
2.1. Base de conocimientos.	2
2.2. Programa.	4
3. Funcionamiento.	6
4. Conclusión.	7
5. Referencias.	7



1. Introducción.

Este proyecto está realizado en el lenguaje de programación Prolog. **Prolog**, proveniente del francés **PRO**grammation en **LOG**ique, es un lenguaje de programación lógico e interpretado usado habitualmente en el campo de la inteligencia artificial. Este lenguaje no es un lenguaje de programación para usos generales, sino que está orientado a resolver problemas usando el cálculo de predicados.

2. Proyecto.

Este proyecto trata de recomendar canciones según el estado de ánimo que tiene el usuario en ese momento y de su género favorito. Lo que hace el programa es realizar una serie de preguntas y dependiendo de las respuestas obtenidas recomendará una lista de canciones disponibles.

Las principales características de Prolog usadas en este proyecto son:

- Listas.
- Estructuras para almacenar la información de las canciones.
- Entrada/Salida complejas.

2.1. Base de conocimientos.

Para esta aplicación se han recopilado las “*Claves Musicales*” y cada una de ellas están relacionadas con una emoción. Las que se han recopilado con las siguientes:

```
% Key del tipo de songes
key(a_mayor, alegre).
key(a_minor, melancolico).
key(b_mayor, fuerte).
key(b_minor, solitario).
key(c_mayor, feliz).
key(c_minor, triste).
key(d_mayor, triunfante).
key(d_minor, serio).
key(e_mayor, bullicioso).
key(e_minor, inquieto).
key(f_mayor, furioso).
key(f_minor, depresivo).
```



```
key(g_mayor, calmado).  
key(g_minor, descontento).
```

Esto se ha hecho para que a la hora de introducir un estado de ánimo esta te recomiende canciones de la “Clave Musical” relacionada con el estado de ánimo que ha introducido el usuario.

Luego, se ha definido los géneros musicales que van a estar disponibles que son los siguientes:

```
% Generos de musica  
genre(pop).  
genre(rock).  
genre(dance).  
genre(techno).  
genre(rap).  
genre(hip_hop).  
genre(rnb).  
genre(soul).  
genre(disco).  
genre(classical).  
genre(blues).  
genre(jazz).  
genre(reggae).  
genre(reggaeton).  
genre(metal).  
genre(trap).  
genre(electronic).
```

Y por último, se ha hecho una base de datos de canciones sobre los géneros musicales que ha definido anteriormente en la que la regla es la siguiente:

```
song(key, mood, cancion, artista, genero, año)
```

La base de datos tiene aproximadamente unas 340 canciones.



2.2. Programa.

Este programa está diferenciado por varias partes, la primera parte que es la función principal que es la `main_menu/0` que esta es que contiene la gran mayoría del programa sin contar la base del conocimiento. Al entrar en la función principal nos vamos a encontrar dos bloques que están divididos por el predicado de control `repeat/0` este predicado funciona como un `while` en C++ lo que hace es repetir todo lo que hay a continuación del predicado hasta que la condición que se haya programado es verdadera.

En el primer bloque preguntamos al usuario cómo se siente hoy y a continuación con el predicado `forall/2` imprime todos los estados de ánimo que hay disponibles en el programa. El funcionamiento del segundo bloque es igual pero preguntando por el género musical que le gusta más al usuario de los que hay disponibles.

A continuación, se busca las canciones que coincidan con el estado de ánimo y el género en la base del conocimiento con el predicado `findall/3` de la siguiente forma:

```
% Busca las canciones que coincidan con el estado de animo y el genero en la
% base de conocimiento y las guarda en una lista que es List
findall([Song, Artist], song(_, AnswerMood, Song, Artist, AnswerGenre, _),
List),
```

lo que hace es crear una lista y almacena con los parámetros que se quiere buscar, en este caso "Song", nombre de la canción y "Artist" artista de la canción, sobre las reglas `song/6`.

Luego, nos encontramos con una condicional para comprobar si no hay canciones que coincidan con el estado de ánimo y el género:

```
% Si no hay canciones que coincidan con el estado de animo y el genero salta a
answer_more
(List = [] -> write('\nNo hay canciones que coincidan con el estado de animo y
el genero.\n\n'), answer_more; true),
```

Si esta condición se cumple salta a una función que es `answer_more` que lo que hace es preguntar al usuario si quiere más recomendaciones de canciones. Aquí está la implementación:

```
answer_more :-
    % Preguntar si quiere mas recomendaciones
```



```
repeat,  
write('\n¿Quieres mas recomendaciones? (si/no)\n'),  
read(Answer),  
% Si se introduce una opcion no valida, si/no, repite el ciclo  
(Answer = 'si' -> true; Answer = 'no' -> true; write(Answer), write(':  
No es una opcion valida.\n\n'), fail),  
(Answer = 'si' -> true, again; nl, nl,  
    tab(20), write('=====\n'),  
    tab(20), write(';Gracias por usar La aplicacion!\n'),  
    tab(20), write('=====\n\n'),  
    write('\n'), halt).
```

Si se quiere más recomendaciones empieza de nuevo el programa.

En el caso de que se muestre las canciones que coincidan con el estado de ánimo y el género musical introducido por el usuario este va a mostrar la clave música que está relacionada con el estado de ánimo del usuario y a continuación pregunta si quiere que le recomiende canciones de la clave musical.

```
% Recomienda la "key" que coincide con el estado de animo  
write('\n\nTe recomiendo buscar canciones con esta \"key\": \n'),  
findall(Key, key(Key, AnswerMood), KeyList),  
write_down_list(KeyList),  
  
% Preguntar si quiere mas recomendaciones de la key  
repeat,  
write('\n¿Quieres mas recomendaciones de la key? (si/no)\n'),  
read(AnswerKey),  
% Si se introduce una opcion no valida, si/no, repite el ciclo  
(AnswerKey = 'si' -> true, key_song_recommender(AnswerMood); AnswerKey = 'no'  
-> true, answer_more; write(AnswerKey), write(': No es una opcion  
valida.\n\n'), fail).
```

Como se ve en la implementación y es que si el usuario quiere más recomendaciones sobre la clave musical este salta a una función `key_song_recommender/1` su funcionamiento es igual que a los otros predicados para mostrar las canciones.



3. Funcionamiento.

Antes de comenzar con el funcionamiento del programa hay que estar seguros de que está instalado el intérprete de Prolog. Una vez está instalado el interprete su ejecución es la siguiente:

```
$ swipl song-recommender.pl
```

Al ejecutarse se verá de la siguiente manera:

```
=====
¡Bienvenido a la aplicacion de recomendacion de canciones!
=====

¿Cómo te sientes hoy?
alegre
melancolico
fuerte
solitario
feliz
triste
triunfante
serio
bullicioso
inquieto
furioso
depresivo
calmado
descontento

Elige un estado de animo.
|: feliz.

¿Qué género musical te gusta más?
pop
rock
dance
techno
rap
hip_hop
rnb
soul
disco
classical
blues
jazz
reggae
reggaeton
metal
trap
electronic

|: trap.

Estas son las canciones que te recomiendo del genero "trap" y estado de animo "feliz":
[Kod,J. Cole]
[One Dance,Drake]
[Black Beatles,Rae Sremmurd, Gucci Mane]
[Dos Mil 16,Bad Bunny]

Te recomiendo buscar canciones con esta "key":
c_major

¿Quieres mas recomendaciones de la key? (si/no)
|: no.

¿Quieres mas recomendaciones? (si/no)
|: no.

=====
¡Gracias por usar la aplicacion!
=====
```



4. Conclusión.

En conclusión, la realización de este proyecto me ha hecho dar cuenta de la potencia que tiene este lenguaje de programación para realizar programas. Además, de ver un nuevo paradigma de programación que me ha empezado a gustar. Luego, a la hora de crear la base del conocimiento ha sido lo más pesado ya que es grande y había que buscar las canciones de los diferentes géneros musicales y luego buscar la clave de valor que tiene cada canción. Por lo demás me ha parecido muy divertido el desarrollo de la aplicación.

5. Referencias.

1. http://fcqi.tij.uabc.mx/usuarios/ardiaz/material/manual_lab_prolog.pdf
2. <https://es.wikipedia.org/wiki/Prolog>
3. musicstax.com
4. <https://ledgernote.com/blog/interesting/musical-key-characteristics-emotions/>
5. <https://open.spotify.com>