



Materia: programación lógica

Maestro: Aldo Fabian Chávez Villanueva

Proyecto final

Grado y grupo: 82SA

Clave de curso: SCD - 1019

Nombre: Jose Miguel Vazquez Méndez

## Contenido

|  |   |
|--|---|
| 1.Resultados más relevantes en el desarrollo del sistema experto).....                         | 3 |
| 2.Introducción.....  | 3 |
| 3.-Metodología.....  | 3 |
| 4.-Resultados.....   | 4 |
| 5.-Conclusiones .....  | 5 |
| 6.-Referencias (deben citarse en el texto y numerar conforme se fue citando en el texto) ..... | 5 |

## 1.Resultados más relevantes en el desarrollo del sistema experto

Logramos crear un algoritmo que regrese los valores de la escala que se está buscando, ya sea mayor, menor o alguna otra variante. Dicho algoritmo opera en una base de conocimientos lo suficiente grande y abarca casi todas las escalas más populares.

## 2.-Introducción:

El mundo de la música siempre se ha auxiliado de la tecnología presente en su época: desde los sintetizadores de frecuencias FM análogos hasta los programas más modernos para computador que emulan dichos sintetizadores.

Mi proyecto busca agregar valor al ya inmenso mundo de la música, auxiliando al musico con una herramienta más para encontrar las notas “adecuadas” para cada ocasión.

Una de las razones por las que embarco este proyecto es porque, aunque PROLOG tiene bastante uso en el ámbito de la inteligencia artificial, no hay ningún proyecto que intente resolver un problema musical. Busco hacer una contribución para quien necesite tal sistema.

## 3.-Metodología:

PROLOG puede crear estructuras de objetos complejos con varios componentes.[1]. Por ejemplo, una lista es una simple estructura de datos usada ampliamente en la programación.[2] Dicha lista es representada en PROLOG con dos corchetes, encerrando dentro de ellos los componentes de la lista.

[ann, pete, thomas, richard]

En PROLOG, si una lista esta vacía se escribe como []. Si no, se considera que tiene una cabeza (h) y cola(c). [h|c]. [3] El símbolo | significa que, a partir de esa línea, cualquier otro elemento puede estar dentro.[4]

Hay varias operaciones que se pueden lograr.

- Inspeccionar si un elemento está en la lista
- concatenación de 2 o más listas
- Añadir un nuevo objeto a la lista

Para inspeccionar si un elemento esta en la lista, utilizamos una simple formula que regresa verdadero si encuentra los parámetros, utilizando la función member(Item, Items). como un pivote de memoria que ira checando entre la base de conocimientos a encontrar el valor.[5] La función member en prolog es utilizada para regresar True si los elementos de una lista son idénticos a la primera.[7]

## 4.-Resultados

El primer algoritmo de búsqueda, “search”, hace bien el trabajo a la primera. Encuentra las notas atadas a la escala o acorde que elijas. Por ejemplo, si queremos buscar D menor el algoritmo arroja:

```
?- search(d, minor, Item).  
Item = c ;  
Item = d ;  
Item = e ;  
Item = f ;  
Item = g ;  
Item = a ;  
Item = b_b
```

Si queremos buscar E menor el algoritmo arroja:

```
?- search(e, minor, Item).  
Item = c ;  
Item = d ;  
Item = e ;  
Item = f_s ;  
Item = g ;  
Item = a ;  
Item = b.
```

```
?- search(c_s, dorian, Item).  
Item = c_s ;  
Item = d_s ;  
Item = e ;
```

Si queremos algo mas especifico, como c sostenido Dorian, el algoritmo también se adapta:

## 5.-Conclusiones

La música y tecnología van de mano en mano, una existiendo a partir de otra y una moldeándose a partir de la otra. Me es importante contribuir a ambos mundos con un aporte para quien lo requiera. Este es un sistema complejo, pero de fácil uso que de segura manera ayudara a muchos quienes necesiten un auxilio para aprender el contrapunto básico. Espero que este simple proyecto le sirva a quien lo necesite. Dejo el link a github.

## 6.-Referencias (deben citarse en el texto y numerar conforme se fue citando en el texto)

[1] Ivan Bratko. (2001). Data objects. En Prolog: programming for artificial intelligence(29). Ljubljana University: Addison Wesley.

[2], [3] Ivan Bratko. (2001). Data objects. En Prolog: programming for artificial intelligence(61). Ljubljana University: Addison Wesley.

[4] Ivan Bratko. (2001). Data objects. En Prolog: programming for artificial intelligence(63). Ljubljana University: Addison Wesley.

[6], [7] [https://www.swi-prolog.org/pldoc/doc\\_for?object=member/2](https://www.swi-prolog.org/pldoc/doc_for?object=member/2)