**Proyecto programado ll – Prolog.**

**José Andrés Navarro Acuña.**

**2016254241.**

**Instituto Tecnológico de Costa Rica.**

**Ingenieria en computación.**

**IC-4700 Lenguajes de programación.**

**José Enrique Araya Monge.**

**Grupo 1.**

**I semestre 2018.**

Índice

Comentarios finales 3

Corridas de ejemplo 3

Estructuras de datos usadas 3

**Introducción** 3

### **Introducción**

El presente segundo proyecto programado consistió en la resolución de dos acertijos lógicos, por medio del lenguaje de programación Prolog. Dichos acertijos consisten en algunos hechos acerca de un número pequeño de objetivos que tienen varios atributos. Además, se especifica un número mínimo de hechos acerca de los objetivos y sus respectivos atributos, con el motivo de que se obtenga una única manera de asociar atributos con objetos. A continuación, una breve explicación de cada caso.

* 1. **Caso 1: Smoothies.**

El primer caso se desarrolla en un comercio llamado “Super Smoothies”, donde una dependiente de dicho local tiene una larga fila de clientes esperando por sus órdenes. Cada orden consiste en un "super alimento" (semilla de lino, pasto de trigo, semillas de chia, gengibre, quinoa) y una fruta. Cada cliente pidió algo de un precio distinto ($5, $6, $7, $8 , $9 y $10). Usando solamente las claves que se incluyen a continuación, encuentre para cada cliente qué ordenó y a qué precio.

1. El cliente que pagó $6 no pidió arándanos.
2. El cliente que ordenó semilla de lino pagó más que la persona que ordenó pasto de trigo.
3. Isabel pidió semillas de chia.
4. El cliente que solicitó gengibre es Paulette o es la persona que pagó $10.
5. Paulette, el cliente que pidió arándanos y la persona que pidió naranjas, son tres personas distintas.
6. El cliente que pidió naranjas pagó 1 dólar más que la persona que pidió bananos.
7. Otis, o pagó $6 o pagó $10.
8. La persona que pidió quinoa pagó $3 más que Mercedes.
9. Sobre Paulette y la persona que ordenó frambuesas: una pidió pasto de trigo y la otra persona pagó $8.
10. Isabel pagó 3 dólares menos que Amelia.
11. ¿Quién pidió mandarina?
    1. **Caso 2: Cruceros.**

El segundo caso se desenvuelve en un crucero llamado “Emerald Crystal”, donde hay un grupo de siete viajeros de larga data sentados en la mesa del capitán, compartiendo historias de sus viajes anteriores, realizados en los años 1983-1989. Haga coincidir a cada persona con su destino y determine en qué año viajaron y en cuál barco crucero navegaron.

**1.** Eugene no viajó en el crucero Azure Seas.

**2.** La persona que fue a Trinidad zarpó 1 año antes que Lee.

**3.** La persona que se embarcó en el crucero Silver Shores es Francis o es quien viajó en 1984.

**4.** Los siete viajeros son: la persona que fue a Saint Lucia, Greg, la persona que se embarcó en el crucero Neptunia, la persona que viajó en 1987, la persona que tomó el crucero Trinity, la persona que se embarcó en el crucero Baroness y la persona que tomó un crucero en 1986.

**5.** Sobre los que tomaron el crucero Farralon y el crucero Caprica, uno es Greg y el otro fue a Martinique.

**6.** La persona que fue a Puerto Rico viajó 1 año después de la persona que tomó el crucero Silver Shores.

**7.** Kathy no viajó en el crucero Azure Seas.

**8.** Natasha viajó ya sea en el crucero Baroness o en el crucero de 1985.

**9.** La persona que fue a Martinique está entre Eugene y la persona que tomó el crucero Caprica.

**10.** La persona que tomó el crucero de 1987 no fue la misma que viajó en el crucero Caprica.

**11.** Sobre Francis y la persona que fue a Trinidad: uno estuvo en el crucero de 1983 y el otro tomó el crucero Neptunia.

**12.** Bradley, o fue a Jamaica o más bien tomó el crucero de 1987.

**13.** La persona que fue a Grenada viajó 2 años después que Kathy.

**14.** La persona que tomó el crucero Neptunia lo hizo 1 year año después de que quién tomó el crucero Silver Shores.

**15.** La persona que viajó en el crucero Trinity zarpó 1 año después de quien tomó el crucero Baroness.

**16.** Uno de los viajeros fue a Barbados.

## Estructuras de datos usadas

Seguidamente, se presentarán las estructuras utilizadas para la resolución de los acertijos planteados. Primeramente, se define una relación llamada “test\_puzzle/2” donde define un cuádruple llamado “puzzle” que consiste en las siguientes variables:

* “Structure”: define la estructura de los datos a determinar.
* “Clues”: corresponde a una lista de pistas que van instanciando los elementos de la estructura.
* “Queries”: es una lista de consultas a la estructura para responder a las preguntas.
* “Solution”: corresponde a una lista con las respuestas del acertijo.

Cabe mencionar que la relación “test\_puzzle” recibe una variable “Name” la cual tiene la función de llamar y diferenciar los acertijos al momento de ejecutar el programa.

Se define, además, una relación “solve\_puzzle/1” donde funciona como el mecanismo por el cual se resuelve el acertijo. Ya que, toma un acertijo con sus componentes y va instanciando la estructura al ir resolviendo las pistas y consultas.

Seguidamente, se establece la relación “solve/1” donde va tomando uno por uno los elementos de las listas, ya sea de pistas o de consultas y trata de solverlos.

Para mostrar los datos, se estableció la relación llamada “mostrar/1” donde imprime en diferentes líneas las entradas de una lista de un acertijo. En este caso, se utiliza para mostrar “Structure”.

Se declara la relación “resolver/3” donde recibe el nombre del acertijo y produce una solución. Además, se incluye “Structure” con el fin de visualizar los datos al resolver un acertijo y facilitar la depuración. “Solution” se incluye para almacenar la solución del acertijo.

La relación “structure/2” crea un acertijo dado una lista con los elementos a determinar. La forma es: “structure(NombreAcertijo, ListaElementosADeterminar)”.

La relación “clues/3” implementa las pistas dadas. En este punto, cada pista es un elemento de una lista, y cada una de ellas es una meta a resolver que va instanciando los elementos de la estructura.

En lo que respecta las preguntas de los acertijos, la relación “queries/3” hace consultas a la estructura común y luego prepara las respuestas según sea el enunciado que se propone.

## Corridas de ejemplo

## Caso 1: Smoothies.

* + 1. **Resultados al procesar los acertijos.**

?- resolver(smoothies, Struct, Sol).

orden(6,isabel,semillasDeChia,mandarina)

orden(7,mercedes,pastoDeTrigo,frambuesas)

orden(8,paulette,gengibre,bananos)

orden(9,amelia,semillaDeLino,naranjas)

orden(10,otis,quinoa,arandanos)

Struct = [orden(6, isabel, semillasDeChia, mandarina), orden(7, mercedes, pastoDeTrigo, frambuesas), orden(8, paulette, gengibre, bananos), orden(9, amelia, semillaDeLino, naranjas), orden(10, otis, quinoa, arandanos)],

Sol = [['La persona que pidio mandarina fue ', isabel]] ;

false.

* + 1. **Primeros 2 resultados con las 2 primeras pistas.**

?- resolver(smoothies, Struct, Sol).

orden(6,\_3950,pastoDeTrigo,mandarina)

orden(7,\_3966,semillaDeLino,arandanos)

orden(8,\_3982,\_3984,\_3986)

orden(9,\_3998,\_4000,\_4002)

orden(10,\_4014,\_4016,\_4018)

Struct = [orden(6, \_3950, pastoDeTrigo, mandarina), orden(7, \_3966, semillaDeLino, arandanos), orden(8, \_3982, \_3984, \_3986), orden(9, \_3998, \_4000, \_4002), orden(10, \_4014, \_4016, \_4018)],

Sol = [['La persona que pidio mandarina fue ', \_3950]] ;

orden(6,\_3950,pastoDeTrigo,\_3954)

orden(7,\_3966,semillaDeLino,arandanos)

orden(8,\_3982,\_3984,mandarina)

orden(9,\_3998,\_4000,\_4002)

orden(10,\_4014,\_4016,\_4018)

Struct = [orden(6, \_3950, pastoDeTrigo, \_3954), orden(7, \_3966, semillaDeLino, arandanos), orden(8, \_3982, \_3984, mandarina), orden(9, \_3998, \_4000, \_4002), orden(10, \_4014, \_4016, \_4018)],

Sol = [['La persona que pidio mandarina fue ', \_3982]] .

## Caso 2: Cruceros.

* + 1. **Resultados al procesar los acertijos.**

?- resolver(viajes, Struct, Sol).

viaje(greg,farralon,trinidad,1983)

viaje(lee,silverShores,saintLucia,1984)

viaje(francis,neptunia,puertoRico,1985)

viaje(kathy,caprica,martinique,1986)

viaje(bradley,azureSeas,barbados,1987)

viaje(natalia,baroness,grenada,1988)

viaje(eugene,trinity,jamaica,1989)

Struct = [viaje(greg, farralon, trinidad, 1983), viaje(lee, silverShores, saintLucia, 1984), viaje(francis, neptunia, puertoRico, 1985), viaje(kathy, caprica, martinique, 1986), viaje(bradley, azureSeas, barbados, 1987), viaje(natalia, baroness, grenada, 1988), viaje(eugene, trinity, jamaica, 1989)],

Sol = [['Caso sin preguntas ', greg]] ;

false.

* + 1. **Primeros 2 resultados con las 2 primeras pistas.**

?- resolver(viajes, Struct, Sol).

viaje(eugene,\_3948,trinidad,1983)

viaje(lee,azureSeas,\_3966,1984)

viaje(\_3978,\_3980,\_3982,1985)

viaje(\_3994,\_3996,\_3998,1986)

viaje(\_4010,\_4012,\_4014,1987)

viaje(\_4026,\_4028,\_4030,1988)

viaje(\_4042,\_4044,\_4046,1989)

Struct = [viaje(eugene, \_3948, trinidad, 1983), viaje(lee, azureSeas, \_3966, 1984), viaje(\_3978, \_3980, \_3982, 1985), viaje(\_3994, \_3996, \_3998, 1986), viaje(\_4010, \_4012, \_4014, 1987), viaje(\_4026, \_4028, \_4030, 1988), viaje(\_4042, \_4044, \_4046, 1989)],

Sol = [['Caso sin preguntas ', eugene]] ;

viaje(eugene,\_3948,\_3950,1983)

viaje(\_3962,azureSeas,trinidad,1984)

viaje(lee,\_3980,\_3982,1985)

viaje(\_3994,\_3996,\_3998,1986)

viaje(\_4010,\_4012,\_4014,1987)

viaje(\_4026,\_4028,\_4030,1988)

viaje(\_4042,\_4044,\_4046,1989)

Struct = [viaje(eugene, \_3948, \_3950, 1983), viaje(\_3962, azureSeas, trinidad, 1984), viaje(lee, \_3980, \_3982, 1985), viaje(\_3994, \_3996, \_3998, 1986), viaje(\_4010, \_4012, \_4014, 1987), viaje(\_4026, \_4028, \_4030, 1988), viaje(\_4042, \_4044, \_4046, 1989)],

Sol = [['Caso sin preguntas ', eugene]] .

## Comentarios finales

El estado final del presente proyecto es del 100%, es decir, todas las funcionalidades y especificaciones propuestas fueron implementadas en lo que respecta las pistas de ambos acertijos lógicos. Cada pista tanto del caso uno como el caso dos, fueron totalmente implementadas desde mi perspectiva y punto de vista, ya que, al ser acertijos lógicos la interpretación puede ser distinta en algunas personas, sin embargo, la finalidad de éstas sigue siendo la misma.

Finalmente, no se encontró ninguna limitación adicional o problema que impidiera el desarrollo del proyecto.