

# Paradigmas de Programación – UTN – FRBA

1er cuatrimestre de 2009

## Paradigma Lógico – Trabajo Práctico

Un conjunto de amigos fanáticos de los panqueques desarrollaron un sistema para organizar sus idas a comer panqueques. Luego de unas breves deliberaciones dedujeron que lo más adecuado era hacerlo en Prolog, para aprovechar las amplias ventajas que brinda el paradigma lógico.

Entonces comenzaron por representar la carta del local del cual son *habitués*, utilizando el predicado `panqueque/3` para relacionar el nombre de un panqueque de los que se venden en el local, con los ingredientes y su precio.

```
%panqueque(nombre del panqueque, [ingrediente], precio).
panqueque(maradonna, [queso, cebolla, aceitunas], 10).
panqueque(gardel, [cebolla, morron, panceta], 12).
panqueque(madonna, [panceta,caviar], 60).
```

Adicionalmente el local permite armar tu propio panqueque; en ese caso se cobra en función de cada ingrediente que se incluya. El valor de cada ingrediente se refleja en el predicado `ingrediente/2`.

```
%ingrediente(nombre, precio)
ingrediente(alpiste,0).
ingrediente(cebolla,1).
ingrediente(queso,1).
ingrediente(aceitunas,2).
ingrediente(morron,2).
ingrediente(panceta,2).
ingrediente(caviar,100).
ingrediente(salmonRosadoDelPacifico,120).
ingrediente(quesoAzulDanes,85).
```

Luego agregaron la información de lo que pidieron en una de sus visitas al local, para a partir de ahí obtener las estadísticas que requiere cualquier salida entre amigos. Para ello, el predicado `pidio/2` relaciona a cada persona con un functor que representa su pedido. Para los que eligieron un panqueque de la carta, eso se refleja como un functor `prearmado/1` que tiene como dato el nombre del panqueque elegido. En cambio para los que eligieron armar su propio panqueque, se representa como un functor `custom/1` que contiene la lista de todos los ingredientes elegidos.

```
%pidio(pedro, prearmado(nombre del panqueque)).
%pidio(pedro, custom([ingrediente])).
pidio(pedro, prearmado(maradona)).
pidio(luis, prearmado(gardel)).
pidio(juan, custom([cebolla, panceta, alpiste])).
pidio(martin, custom([cebolla, queso, aceitunas])).
pidio(mauricio, custom([caviar,salmonRosadoDelPacifico,quesoAzulDanes,cebolla])).
```

### Ejercicio 1

Información básica de los panqueques.

- a) Calcular el precio de un panqueque.

```
?-precio(prearmado(maradona),X).  
X=10.
```

```
?-precio(custom([cebolla,panceta,alpiste]), X).  
X=3.
```

b) Obtener los ingredientes de un panqueque.

```
?-ingrediente(maradona, X).  
X=panceta;  
X=queso;  
X=cebolla;  
X=aceitunas;  
No.
```

Hacerlo inversible en sus dos argumentos.

```
?-ingrediente(X, queso).  
X=maradona.
```

## Ejercicio 2

Información del pedido realizado.

a) Realizar el predicado `todosPidieronPanquequesCon/1` que se verifique verdadero si todos los comensales pidieron un panqueque con un ingrediente dado. Debe ser inversible.

```
?-todosPidieronPanquequesCon(X)  
X=cebolla.
```

b) Realizar un predicado `precioTotal/1`, que devuelva el precio total del pedido, es decir, la suma de los precios de cada uno de los panqueques que se pidieron.

```
?- precioTotal(X).  
X=330 %Espero haber hecho bien la cuenta!
```

c) Definir el predicado `seZarpo/1`, que permita determinar si alguien pidió un panqueque que vale más del doble que cualquiera de los demás. Debe ser inversible.

```
?- seZarpo(X)  
X=mauricio.
```

## Ejercicio 3

Para la segunda versión del programa, los amigos decidieron que era muy limitante poder comer un solo panqueque, y decidieron modificar el sistema para adaptarlo a su glotonería. Para ello se pide rehacer el predicado `todosPidieronPanquequesCon/1`, que ahora será verdadero si cada una de las personas pidió ese ingrediente en *alguna* de los panqueques que consumió esa noche. Por ejemplo, si agregamos a la base de conocimiento los pedidos:

```
pidio(pedro, custom([panceta,alpiste])).  
pidio(mauricio, prearmado(madonna)).
```

Se da que todos han pedido algún panqueque con panceta:

```
?-todosPidieronPanquequesCon(X)  
X=cebolla.  
X=panceta.
```

## Ejercicio 4

Ahora queremos estudiar un poco más la carta, para optimizar nuestros pedidos...

a) Realizar un predicado `panquequeCon/2` que permita encontrar, dada una lista de ingredientes, todos los panqueques prearmados que contienen al menos un ingrediente de dicha lista.

```
?-panquequeCon([panceta,caviar],X).  
X=gardel;  
X=madonna.  
No.
```

- b) Se desea determinar si alguien pidió un panqueque *custom* que en realidad es igual a alguno de los prearmados existentes; para ello realizar el predicado `pidioPanquequeExistente/2`, que relaciona a una persona que pidió un panqueque *custom* y el nombre del panqueque prearmado equivalente.

```
?-pidioPanquequeExistente(Quien, Cual).  
Quien=martin, Cual=maradona
```

## Ejercicio 5

Luego de varios problemas a la hora de pagar, se decidió agregar a la base de conocimientos la cantidad de plata que tiene cada uno, utilizando para ello el predicado `tiene/2`, que relaciona a una persona con el dinero disponible para esta noche.

```
tiene(luis, 8).           % le faltan $4, necesita que le preste  
tiene(pedro, 15).        % le sobran $5, puede prestarle a luis  
tiene(juan, 3).          % tiene justo  
tiene(martin, 8).        % le sobran $3, no alcanza para prestarle  
tiene(mauricio, 1000).   % puede prestarle a cualquiera
```

A partir de esa información, se desea realizar el predicado `lePuedePrestar`, que relaciona a una persona que no tiene suficiente plata para pagar su pedido, con otra a la que le sobra suficiente plata como para prestarle lo que le falta. Debe ser totalmente inversible.

```
?-lePuedePrestar(X,Y).  
X=luis, Y=pedro;  
X=luis, Y=mauricio;  
No.
```

## Ejercicio 6

Realizar el predicado `masPedido/2`, que permita obtener el ingrediente más utilizado contabilizando todos los pedidos y la cantidad de veces que fue utilizado:

```
?-masPedido(Ing, Veces).  
Ing=cebolla, Veces=4.
```