

Punto 1

Se sabe que en una carpintería, el costo de producción de un mueble es siempre la suma del costo de materiales y de mano de obra. Algunos casos particulares:

- El costo de materiales de una mesa es igual a su volumen multiplicado por el costo de compra de la madera empleada; el costo de mano de obra es del costo de trabajo de la madera.
- Las banquetas siempre tienen un costo de \$10 de mano de obra, y un costo fijo de materiales que depende de cada una.

El siguiente código Prolog tiene por objetivo hallar el costo de producción de un mueble:

```
%% costoMadera(Madera, CostoCompra, CostoTrabajo)
costoMadera(cedro, 10, 2).
costoMadera(brasil, 12, 2).
costoMadera(pino, 8, 6).

%% etc...

costoProduccionMesa(Madera, Volumen, Costo) :-
    costoMadera(Madera, CostoCompraMadera, CostoTrabajoMadera),
    Costo is Volumen * CostoCompraMadera,
    Costo is Costo + CostoTrabajoMadera.

costoProduccionBanqueta(CostoMateriales, Costo) :-
    Costo is CostoMateriales + 10.
```

1. El código presentado, ¿funciona adecuadamente? Si lo hace, dar ejemplos de consulta de costos de producción de mesas y banquetas, indicando claramente la respuesta. Si no lo hace, justificar con qué concepto está relacionado el error y corríjalo (se pide re-codificar).
2. Los predicados de producción de una mesa y de una banqueta, ¿son polimórficos?
 - a) Justifique, y si no fueran, modifíquelos para lograrlo.
 - b) Relacione con el concepto de asignación de responsabilidades.
 - c) Relacione con el concepto de pattern-matching.
3. Se tiene un lote de muebles (una lista de mesas y banquetas) y se desea saber el costo total de producción de las mismas.
 - a) Implementar un predicado costoLote/2 que relacione un lote con su costo total de producción (la suma de los costos totales de cada mueble).
 - b) Indicar que cambios realizados en el punto 2) reportaron beneficios en la resolución de este requerimiento
4. Se desea implementar el requerimiento del punto 3 en Smalltalk.
 - a) Codifique el costo del lote asumiendo que ya existen objetos que representan a los distintos tipos de madera. ¿Cuál es la interfaz de estos objetos (los tipos de madera)?
 - b) Indicar si se usó orden superior en dicho requerimiento, tanto en la solución lógica como de objetos, y en tal caso, los beneficios del mismo.
5. Suponiendo que ahora se incorpora un nuevo tipo de mueble, la silla, que cuyos costos de mano de obra y materiales se calculan de forma diferente, explique y compare cómo agregaría este nuevo concepto en ambos paradigmas. Relacione con los conceptos de polimorfismo y tipo de dato.
6. Un programador desea implementar el mismo requerimiento de obtención del costo total de un lote empleando el lenguaje Haskell. Para ello, escribe la siguiente consulta:
(sum . map costoTotal) lote
Asumiendo que el lote puede estar conformado por mesas de la forma (Madera, Volumen) de tipo

¹ El conjunto de mensajes que entiende un objeto.

(String, Integer) y sillas de la forma (Madera, TipoRespaldo, Volumen), de tipo (String, String, Integer), indicar si existen impedimentos para resolver el problema con dicha consulta, según lo visto en la cursada.

Punto 2

Tengo el predicado p y la función f , definidos de esta forma:

$p(1, 3).$	$f\ 1 = 3$
$p(2, 6).$	$f\ 2 = 6$
$p(X, 9) :- \text{esNumero}(X).$	$f\ x = 9$

suponiendo que el predicado esNumero se satisface para todos los números naturales, o sea, que la consulta

$?- \text{esNumero}(X).$

da las siguientes respuestas:

$X = 1;$

$X = 2;$

$X = 3;$

... etc.

Hago estas consultas:

$?- p(1, X).$	$> f\ 1$
---------------	----------

El comportamiento de Prolog y de Haskell ante estas consultas va a ser distinto.

Se pide describir

- cuál es la diferencia, y con qué conceptos inherentes a los paradigmas lógico y funcional está relacionados.
- qué otros tipos de consulta puedo hacer en Prolog que no puedo hacer en Haskell, referidas a este ejemplo, y con qué cualidad que asociamos al paradigma lógico tiene que ver.

Punto 3

Se cuenta con estos ejemplos, uno en Haskell y otro en Smalltalk

```
(head . filter even) listaDeNumeros
```

```
(numeros select: [:numero | numero even]) first
```

Recordemos que `filter` recibe una función booleana F y una lista L , y devuelve la sublista de L de los elementos que verifican F . Puede codificarse así:

```
filter f [] = []
filter f (x:xs) | f x      = x:filter xs
                | otherwise = filter xs
```

En la evaluación de uno de estos dos ejemplos aparece el concepto de evaluación diferida, mientras que en el otro no. Se pide:

- indicar en cuál de los dos ejemplos aparece la evaluación diferida, relacionando con conceptos vistos en la materia.
- Indique qué consultas son posibles gracias a esta característica (en base al ejemplo presentado en el punto).