

## Ejercicio 1

Para una obra, se requiere conocer el costo de los artículos necesarios para construir una casa.

|  |   |
|--|---|
| <pre>% precio de cada material por unidad de medida  precio(ladrillo, 10). precio(aluminio(Espesor), Precio):-     Precio is Espesor * 100. precio(madera(pino), 200). precio(madera(roble), 250). precio(maderaPintada(Arbol,AdicionalColor), Precio):-     precio(madera(Arbol), PrecioComun),     Precio is PrecioComun + AdicionalColor.</pre> | <pre>% cantidad necesaria, en la unidad de medida correspondiente. Cada articulo se hace con una cierta cantidad de un único material.  articulo(puerta, 2, madera(roble)). articulo(ventana, 5, aluminio(1)). articulo(pared, 2000, ladrillo). articulo(mesa, 4, maderaPintada(roble,20)).  % Requerimiento principal costo(Articulo, Costo):-     Costo is Cantidad * Precio,     articulo(Articulo, Cantidad, Material),     precio(Material, Precio).</pre> |
|--|---|

1. En la implementación anterior hay un error. Identifique el mismo y corríjalo. Indique el concepto relacionado.
2. Indique, justificando, si existe o no polimorfismo en la solución dada (¿dónde? ¿cómo?).
3. Implemente el requerimiento del ejercicio 1 en el paradigma de objetos. Realice diagrama y codificación.
4. ¿Existe algún concepto en común que haya sido aprovechado tanto en el paradigma lógico como en el paradigma de objetos? Justifique por qué aparece en ambas soluciones.
5. Justifique la utilización de la herencia ó la no utilización de dicho concepto en la solución de objetos.
6. ¿Qué otros conceptos aparecen en una de las soluciones y no en la otra? Justificar para qué aparecen en cada caso.

## Ejercicio 2

```
funcionLoca a lista = map (\b -> b a) lista
```

1. ¿Qué hace la función? Mostrar un ejemplo de invocación y respuesta de la función `funcionLoca`. Mejorar la expresividad de esta función en base a este análisis.
2. Definir el tipo de la función.
3. ¿Cómo implementaría lo mismo en Prolog y en Smalltalk? (si no se puede, justificar por qué, si se puede, explicar claramente cómo se lo haría o directamente desarrollar el código principal)
4. ¿Cuál es el concepto fundamental para que la función pueda funcionar? ¿Por qué? ¿Qué ventajas aporta?
5. Si se pasa **como primer argumento** a `funcionLoca` la lista infinita `[1..]` ¿hay alguna forma de evaluar la función de modo que el procesamiento termine correctamente que no sea con una lista vacía como segundo argumento? De ser posible, mostrar algún ejemplo que lo permita. De no serlo, explicar por qué no se puede. ¿Qué concepto está asociado a esta problemática?
6. Un programador afirma que “la función `funcionLoca` es una función `map` parcialmente aplicada”. ¿Estás de acuerdo? ¿Por qué?