

Condiciones de aprobación

Para aprobar es necesario simultáneamente:

- completar el 60% del examen, y
- obtener al menos la mitad de los puntos en cada paradigma.

En todas tus respuestas sé puntual, no pierdas el foco de lo que se pregunta. Respuestas en exceso generales son tan malas como respuestas incompletas.



Parte A

Una empresa inmobiliaria tiene la siguiente base de conocimiento para sus avisos. Aparentemente aseguran que existe cierta relación entre las ubicaciones altas, el precio y el barrio.

Esto quiere decir que a partir de cierto precio en determinado barrio, todos los anuncios les pasa que:

- Todas las casas tienen más de 3 pisos
- Todos los PH tienen más de 2 pisos
- Todos los dptos están ubicados por encima del 5to piso

Para esto se tiene la siguiente solución en prolog

```
% precio(casa(ambientes, pisos), barrio,
precio).
% precio(dpto(ambientes, piso,
totalUnidades), barrio, precio).
% precio(ph(ambientes, pisos,
totalUnidades), barrio, precio).
```

```
precio(casa(4, 3), Flores, 14000000).
precio(dpto(4, 6, 28), Palermo, 18000000).
precio(ph(4, 2, 4), Flores, 12000000).
```

```
todosAltosConPrecioEn(Precio, Barrio):-
precio(casa(_, Pisos), Barrio, Precio1),
Precio1 >= Precio, Pisos > 3,
precio(dpto(_, Piso, _), Barrio, Precio2),
Precio2 >= Precio, Piso > 5,
precio(ph(_, Pisosph, _), Barrio, Precio3),
Precio3 >= Precio, Pisosph > 2,
```

Se pide:

1. Responder **verdadero** o **falso** y justificar en **todos** los casos:
 - a. La solución propuesta para todosAltosConPrecioEn/2 es correcta.
 - b. El uso de polimorfismo para verificar la altura es correcto.
 - c. Como se plantea la solución el predicado es totalmente inversible.
2. Armar una solución superador que mejore los problemas detectados en el punto anterior.

Parte B

Tenemos estas declaraciones:

edad:: Persona -> Float (no importa cómo se define)

f1 _ [] = False

f1 numero (x:xs) = 2 * x < numero || f1 numero xs

f2 e lista = length (filter (\x -> edad x < e) lista) >= 1

f3 tope palabras = any ((< tope).length) palabras

Las funciones dadas andan, pero queremos analizarlas más en profundidad:

1. Explicar qué es lo que hace cada una y proponer mejoras en términos de expresividad.
2. Comparar las funciones f1 y f2, indicando qué conceptos pueden verse en una y no en la otra. En caso de que una sea mejor a la otra en algún sentido, explicar cuál y por qué motivo.
3. Comparar las funciones f2 y f3, indicando qué conceptos se aprovechan en una y no en la otra. Explicar para qué se están usando.
4. Indicar para cada una si podría terminar de evaluarse ante una lista infinita como parámetro, justificando el motivo tanto si podrían terminar como si no.
5. Si bien el objetivo de las funciones f1, f2 y f3 es distinto y se resuelven de formas diferentes, hay una idea general que se repite en las 3.
Extraer toda la lógica común en una función nueva y volver a definir f1, f2 y f3 usándola adecuadamente.

Parte C

Necesitamos modelar un sistema de venta de helado para una heladería. Por ahora tienen dos productos: kg de helado y cucuruchos, estos últimos pueden venir con un recargo por cobertura de chocolate. Podemos suponer que los cucuruchos valen \$50 más un recargo según la cobertura de cada uno, y el kg de helado \$200. Se desea realizar una venta acumulando lo facturado y disminuyendo el stock de helado de acuerdo a lo vendido. El cucurucho lleva 200 g de helado.

```
class Heladeria {
  var facturado
  var stock
  method vender(prod){
    if(prod.tipo() == 1){
      facturado += 200
      stock = (stock-1).max(0)
    }
    if (prod.tipo() == 2){
      facturado += 50 + prod.recargoCobertura()
      stock = (stock-0.2).max(0)
    }
  }
}
```

```
class KgDeHelado{
  method tipo(){ return 1 }
}

class Cucurucho {
  var property recargoCobertura
  method tipo(){ return 2 }
}
```

1. Responder **verdadero o falso** y justificar en **todos** los casos:
 - a. El sistema tiene un problema: cuando se intenta hacer una venta sin stock suficiente, se factura igual. Debería no hacer nada si no hay stock.
 - b. Hay un buen uso de polimorfismo, ya que si quiero agregar un nuevo producto no debo modificar ningún producto existente.
 - c. Las responsabilidades están mal distribuidas.
 - d. Se rompe el encapsulamiento del cucurucho.
2. De acuerdo a lo justificado en el punto anterior, escriba una nueva solución incluyendo diagrama de clases, de manera que se corrijan los problemas mencionados.
3. Al requerimiento inicial se suma un pedido: queremos que se cobre 20% más barato el helado si el mismo es de gustos tradicionales (lo cual se indicará en cada caso). Ese descuento no se aplicaría sobre el recargo de cobertura que tengan los cucuruchos.
Agregar la lógica pedida a la nueva solución de modo que no haya repetición de lógica.