

**PUNTO 1**

Se quiere determinar si un menú es aceptable para una persona, de forma que las personas glotonas buscan un menú que tenga más de 2.000 calorías, los celíacos que todas las comidas sean aptas para ellos y los que no comen postre no deben tenerlo en el menú. Se asume que un menú tiene tres comidas: plato principal, bebida y postre.

Se cuenta con la siguiente implementación:

```
>>Menu (VI: platoPrincipal bebida postre)
esCopadoPara: unaPersona
  unaPersona esCeliaca ifTrue: [ ^self esCopadoCeliaco ]
  unaPersona esGlotona ifTrue: [ ^self esCopadoGlotona ]
  ^self esCopadoNoPostre

alimentos
  | col |
  col := OrderedCollection new.
  col add: platoPrincipal.
  col add: bebida.
  col add: postre.
  ^col

esCopadoCeliaco
  | hayAlguno |
  hayAlguno := false.
  self alimentos do: [ :alimento |
    alimento aptaParaCeliacos iffFalse: [ hayAlguno := true ] ].
  ^hayAlguno

esCopadoGlotona
  | totalCalorias |
  totalCalorias := 0.
  self alimentos do: [ :alimento |
    total := total + alimento calorías ].
  ^totalCalorias > 2000

esCopadoNoPostre
  ^postre isNil
```

>>Persona esGlotona ^false	>>Glotona esGlotona ^true	>>Persona esCeliaca ^false	>>Persona esCeliaca ^true
----------------------------------	---------------------------------	----------------------------------	---------------------------------

- En los métodos `esCopadoCeliaco`, `esCopadoGlotona` y `alimentos`, hay asignación destructiva. Explique dónde y para qué se utiliza.
- Codifique ambos métodos eliminando la asignación destructiva.
- ¿Hay efecto (colateral/de lado) en dichos métodos? Justifique. **Nota:** se pide justificar en base al ejemplo, no dar definiciones generales.

- d) En la solución presentada el desarrollador dice que hay polimorfismo. Indique dónde se da este concepto, justificando la respuesta.
- e) Explique en no más de 3 líneas qué pasaría en la clase Menu si ahora agregamos un menú para personas empinadas, que quieren una bebida cuya graduación alcohólica supere el 13%.
- f) ¿Cómo mejoraría la solución en base a lo expuesto en el punto anterior? Codifique teniendo en cuenta todas las mejoras que considere pertinentes. Explique brevemente cada mejora.
- g) Justifique *brevemente* qué ventajas tiene su solución respecto a la original (por favor sea conciso, 5 a 7 líneas deberían ser suficientes).

## PUNTO 2

Se quiere conocer información de un conjunto de jugadores de polo (nombre del jugador, hándicap que es un número de 1 a 10 y equipo en el que juega). Un equipo groso es el que tiene al menos 3 jugadores con hándicap 10:

PROLOG	Haskell
<pre>jugador(heguy, 10, dolфина). jugador(fulanito, 8, dolфина).  equipoGroso(Equipo):-   findall(Jugador,     jugador(Jugador, 10, Equipo),     Jugadores),   length(Jugadores, Cant),   Cant &gt; 2.</pre>	<pre>jugadores = [("heguy", 10, "dolфина"),              ("fulanito", 8, "dolфина"), ...             ]  equipoGroso equipo =   (&gt; 2) . length . filter (esJugadorGroso equipo)) jugadores  esJugadorGroso unEquipo (_, handicap, equipo)   = (handicap == 10) &amp;&amp; (equipo == unEquipo)</pre>

- a) Marque dónde se da la aplicación parcial en la solución de Haskell y explique *brevemente* qué ventajas tiene dentro de la solución planteada (qué pasaría si no existiera ese concepto). Nota: se pide hablar en base al ejemplo, no dar definiciones generales.
- b) Marque dónde se da el concepto orden superior en la solución de Haskell y la de Prolog y explique brevemente qué ventajas tiene dentro de cada una de las dos soluciones planteadas (qué pasaría si no existiera ese concepto). Nota: se pide hablar en base al ejemplo, no dar definiciones generales.
- c) Se puede mejorar la solución en Prolog. Haga las correcciones en código que considere necesarias.
- d) Justifique cuál es la motivación de la corrección hecha en c), ¿con qué concepto está relacionado? (Sea breve, con una o dos líneas por cada corrección debería ser suficiente.)
- e) ¿Qué tipo de consulta se puede hacer en Prolog que no es posible en Haskell? Explique a través de un ejemplo concreto en cada lenguaje.
- f) Relacione el concepto expuesto en e) con el concepto de tipo, siempre hablando en base al ejemplo. (Sea breve, 4 a 5 líneas deberían ser suficientes.)