


## Condiciones de aprobación

Para aprobar es necesario simultáneamente: <ul style="list-style-type: none"> <li>• completar el 60% del examen, y</li> <li>• obtener al menos la mitad de los puntos en cada paradigma.</li> </ul>	<b>En todas tus respuestas sé puntual</b> , no pierdas el foco de lo que se pregunta. Respuestas en exceso generales son tan malas como respuestas incompletas.	
---	---	---

## Parte A

Queremos estudiar el comportamiento de compra para un vendedor mayorista de productos generales. Contamos con:

<pre>data Producto = Prod {   nombre :: String,   precio :: Int }</pre>	<pre>queEmpieceConA = (== 'a') . head . nombre barato = (&lt; 50) . precio  restriccionesDeCompra = [queEmpieceConA, barato]</pre>
---	--

1. Definir los tipos de la función `restriccionesDeCompra`.
2. Modificar la función `queEmpieceConA` para que verifique que el nombre del producto comience con una letra que se pase como parámetro. Indicar qué cambiaría en la representación de `restriccionesDeCompra` y qué concepto está relacionado.
3. Codificar una función que permita establecer, dada una lista de productos y una lista de restricciones, cuáles productos verifican todas las restricciones. ¿Qué conceptos se destacan en tu solución y en dónde?

## Parte B

Tenemos un predicado `toma/2` que relaciona a una persona con aquella bebida que le gusta tomar. La bebida puede ser cerveza, que tiene una variedad, un amargor y un porcentaje de alcohol (0 si es cerveza sin alcohol), o vino, que tiene un tipo y una cantidad de años de añejamiento. El vino siempre tiene alcohol. Y también existen gaseosas varias.

<pre>toma(juan, coca). toma(juan, vino(malbec, 3)). toma(daiana, cerveza(golden, 18, 0)). toma(gisela, cerveza(ipa, 52, 7)). toma(edu, cerveza(stout, 28, 6)).</pre>	<pre>tieneProblemas(Persona):-   findall(C,(toma(Persona, cerveza(C,_A)), A&gt;0),Cs),   findall(V, toma(Persona, vino(V,_)), Vs),   findall(T, toma(Persona, T), Ts),   length(Cs, CCs),   length(Vs, CVs),   length(Ts, CTs),   CTs is CCs + CVs.</pre>
--	---

1. Justificar V o F
  - a. No se repite lógica, dado que la estructura de las bebidas alcohólicas son distintas.
  - b. La solución planteada para `tieneProblemas/1` es declarativa.
  - c. La solución planteada podría mejorarse con polimorfismo.
2. Explique y justifique cuál es el significado de lo que se estaría consultando con el siguiente código:
 

```
?- tieneProblemas(P).
```
3. Implemente una solución superadora de `tieneProblemas/1`.

## Parte C

Queremos hacer un programa que nos diga si una persona puede jugar a un determinado juego. Se sabe que para jugar a la pelota la persona tiene que tener más de un año de edad, para los juegos de mesa tiene que estar en el rango de edad que indica el juego y para los videojuegos tiene que tener una edad superior a la mínima determinada para el género del mismo y menor a 80. Además, los videojuegos sólo se pueden jugar si se tiene una consola que lo soporte, con lo cual sólo podrá jugarlo si tiene alguna de esas consolas.

Se tiene el siguiente código Wollok para resolver lo pedido:



```
class Persona {  
  var property edad  
  var property consolas = []  
  method puedeJugar(unJuego) {  
    if(unJuego.tipo() = "Juego de mesa")  
      return edad.between(unJuego.edadMinima(), unJuego.edadMaxima())  
    if(unJuego.tipo() = "Videojuego") {  
      if(edad.between(unJuego.genero().edadMinima(), 80)) {  
        consolas.forEach{  
          consola => unJuego.consolasPosibles().forEach{  
            otraConsola => if(consola == otraConsola) return true  
          }  
        }  
        return false  
      } else return false  
    }  
    return edad > 1 // si llega hasta acá, es porque es una pelota  
  }  
}
```

```
class Juego {  
  var property tipo  
  var property edadMinima  
  var property edadMaxima  
}
```

```
class Videojuego inherits Juego {  
  var property consolasPosibles  
  var property genero  
}
```

```
class genero {  
  var property titulo  
  var property edadMinima  
}
```

1. Critique la solución presentada en términos de **declaratividad**, **polimorfismo**, **delegación**, **encapsulamiento** y **herencia** marcando los diferentes lugares en los cuales encuentra problemas.
2. Codifique una nueva solución teniendo en cuenta los problemas detectados y fundamente los cambios realizados conceptualmente para cada cambio (no se considerarán respuestas genéricas).