

FECHA: 03/12/2011

1. Se quiere determinar si un tour es copado, según el gusto de una persona dada, para ir de vacaciones. Las personas se dividen en aquellos que tienen predisposición hacia la cultura, otras cuyo gusto para por ver muchos paisajes lindos y otras a los que simplemente les gusta ir a la playa. De cada ciudad, se conocen sus museos, si tiene paisajes lindos y si tiene playa.

A una persona que le gusta la cultura le van a gustar los tours que tienen más museos que ciudades. Una persona que le gustan los paisajes requiere que todos los destinos tengan paisajes lindos. Una persona que le gusta la playa se contenta con que al menos una ciudad en el tour tenga playa.

Se cuenta con la siguiente implementación:

```
>>Tour (VI: ciudades)
```

```
leCopaA: unaPersona
```

```
  unaPersona gusto = #Cultura ifTrue: [ ^self esCultural ]
```

```
  unaPersona gusto = #Paisajes ifTrue: [ ^self tieneMuchosPaisajes ]
```

```
  ^self tienePlayas    "Asumimos que le gusta descansar en la playa"
```

```
tieneMuchosPaisajes
```

```
  | pai |
```

```
  pai := true.
```

```
  self ciudades do: [ :l | l tieneLindosPaisajes ifFalse: [ pai := false ] ].
```

```
  ^pai
```

```
esCultural
```

```
  | cantMuseos |
```

```
  cantMuseos := 0.
```

```
  self ciudades do: [ :ciudad | cantMuseos := cantMuseos + ciudad museos size ].
```

```
  ^cantMuseos > self ciudades size
```

```
tienePlaya
```

```
  ^self ciudades anySatisfy: [:c | c tienePlaya ]
```

- En los métodos **tieneMuchosPaisajes** y **esCultural**, 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.
- En la solución presentada el desarrollador dice que hay polimorfismo. Indique dónde se da este concepto, justificando la respuesta.
- Explique en no más de 3 líneas qué pasaría en la clase Tour si se agregaran personas que tienen más de un gusto y para que un tour les cope hay que satisfacerlos todos.
- ¿Cómo mejoraría la solución en base a lo expuesto en los dos puntos anteriores? Codifique teniendo en cuenta todas las mejoras que considere pertinentes. Explique

FECHA: 03/12/2011

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).
2. Se quiere conocer información de un conjunto de jugadores de basket. De cada jugador se sabe en qué equipo juega y su puntaje promedio por partido. Un equipo se considera grosso si al sumar los promedios de puntos de todos sus jugadores se superan los 100 puntos.

**PROLOG**

```
jugador(ginobili, spurs).
```

```
jugador(duncan, spurs).
```

```
promedioPuntos(ginobili, 24).
```

```
promedioPuntos(duncan, 33).
```

```
equipoGroso(Equipo):-
```

```
    findall(Pts, (jugador(Jug, Equipo), promedioPuntos(Jug, Pts)), ListaPts),
```

```
    sumlist(ListaPts, TotalPts),
```

```
    TotalPts >= 100.
```

**HASKELL**

```
spurs = ["ginobili", "duncan", ...]
```

```
promedioPuntos = [("ginobili", 24), ("duncan", 33)]
```

```
equipoGroso equipo = (sum. map (snd . buscaren promedioPuntos)) equipo >=100
```

```
buscaren tuplas key = (head . filter ((==key).fst)) tuplas
```

- a. Marque dónde se utiliza 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. ¿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.
- d. Analice la expresividad de cada uno de los dos programas y proponga las mejoras necesarias para corregir los problemas detectados.