

**Punto 1**

A continuación se muestran 2 versiones de la codificación del método `#puedeLlevar: fanatico` perteneciente a la clase `Colectivo`. Un colectivo puede llevar a un fanático cuando tiene un asiento libre y además el fanático entiende el idioma que habla el chofer del colectivo. El fanático sabe hablar varios idiomas y el chofer sólo uno.

**Version A**

```
puedeLlevar: fanatico
  ^ self pasajeros size < self cantAsientos &
    (fanatico idiomas includes: chofer idioma)
```

**Version B**

```
puedeLlevar: fanatico
  ^ self quedaUnEspacio & (fanatico entiendeA: chofer)
```

- ¿Qué ventajas tiene una versión sobre la otra? Justifique los conceptos en los que se fundamenta esta ventaja.
- Desarrolle un ejemplo de modificación del sistema (describa el cambio en el dominio y codifique los métodos necesarios) donde esta ventaja se haga evidente.
- Un nuevo requerimiento indica que hay algunos fanáticos que ocupan más de un asiento, ello depende de si trabajan para el comité organizador del evento (por ejemplo si es empleado 2 asientos, si es jefe 3, etc.). ¿Cómo modificaría el código (la versión que prefiera) para adaptarse a este cambio? Relacione este cambio con los conceptos mencionados anteriormente.

**Punto 2**

Se tienen los siguientes códigos, y una invocación en cada caso.

<b>Smalltalk</b> Clase Serie <pre>&gt;&gt; enesimo: n   ^ (self generarSerie: 1) at: n  &gt;&gt;generarSerie: n   ^ (OrderedCollection with: (2 ** n))     union: (self generarSerie: (n+1) )</pre>	<b>Workspace</b> <pre>^ Serie new enesimo: 1</pre>
<b>Prolog</b> <pre>enesimo(N,Enesimo):-serie(S),nth1(N,S,Enesimo).  serie(S):-serieDesde(1,S). serieDesde(Base,[X S]):-X is 2 ** Base, B1 is Base + 1, serieDesde(B1, S).</pre>	<b>Workspace</b> <pre>?- enesimo(1, X).</pre>
<b>Haskell</b> <pre>enesimo n = (generarSerie 1) !! (n-1) generarSerie n = (2 ^ n): generarSerie (n+1)</pre>	<b>Workspace</b> <pre>Main&gt; enesimo 1</pre>

Para cada caso, se asegura que la sintaxis es válida (\*\* y ^ son el operador de potencia; with: es un método de clase que genera una nueva colección con el elemento enviado como parámetro; union: devuelve una colección con los elementos que resultan de la unión de dos colecciones; y !!, at: y nth1/3 indica la posición de un elemento en una lista/colección).

- a- ¿Qué hace el código? Indique en cada caso qué ocurre cuando se ejecutan los workspaces respectivos.
- b- Justifique cada resultado que se obtiene, explique qué conceptos entran en juego.

### **Punto 3**

*“La asignación de turnos en un centro médico se realiza de acuerdo a diferentes reglas. En el centro médico se atienden diferentes especialidades, cada una de ellas tiene un equipo médico con un jefe, una cantidad determinada de especialistas, y enfermeros asignados. El centro atiende de lunes a viernes de 8 a 19 hs. El manejo de la agenda es mensual, a finales de cada mes se le solicita a cada médico que indique qué días va a concurrir y en qué horario. Al médico jefe no se pregunta disponibilidad, se considera que va todos los días de 8 a 15 hs.*

*Los turnos son de 30 minutos. Cuando una persona solicita un turno para una especialidad, se le muestran los siguientes 5 turnos libres de acuerdo a la disponibilidad horaria y los turnos ya ocupados.”*

En base al relevamiento anterior, indique qué ventajas o facilidades y qué desventajas o dificultades presenta resolver el problema utilizando

- a- Paradigma de objetos
- b- Paradigma lógico.

### **Punto 4**

En la materia decimos que en los paradigmas lógico y funcional se respeta la transparencia referencial, mientras que en el de objetos no necesariamente.

Indique qué quiere decir esto en términos de cada paradigma. En otras palabras, ¿Cómo terminaría la frase que empieza así: *“En el paradigma lógico siempre hay transparencia referencial, entonces...”?*

Lo mismo para los otros dos paradigmas.

Dar un ejemplo de cada paradigma que colabore con la explicación