IMPORTANTE: No olvide firmar y numerar todas las hojas (hoja 2 de 7, hoja 3 de 7, etc.)

1) Se parte del problema de encontrar el valor absoluto de un número. Explique
   a) las formas de consultar por el valor absoluto de -2 en Smalltalk y Haskell (codifique en caso de duda o detalle bien la forma de resolver el requerimiento)

   b) ¿Qué debemos modificar para obtener el valor absoluto de 3.14 en cada solución? ¿Encuentra similitudes entre las soluciones propuestas? Justifique.

   OJO - Cualquier concepto o idea que se mencione en este punto debe relacionarse concretamente con el problema planteado, si se indica-define-menciona sin relación con el problema, resta en vez de sumar.

2) Se quiere conocer los países que tienen en su población más de 50 millones de habitantes. Se tienen dos soluciones:

En Smalltalk:

```
#Mapamundi
paisesPoblados
   | aux |
   aux := Set new.
   paises do:
      [ :pais | pais poblacion > 50
                  ifTrue: [ aux add: pais ] ].
   ^aux
```

En Prolog:

```
pais(argentina, 50).
pais(bolivia, 30).
pais(nepal, 120).

paisesPoblados([], []).
paisesPoblados([Pais|Paises], [Pais|Resto]):-
   pais(Pais, Poblacion), Poblacion > 50,
   paisesPoblados(Paises, Resto).
paisesPoblados([_|Paises], Resto):-
   paisesPoblados(Paises, Resto).
```

Se pide:

a) Cuando pedimos conocer los paisesPoblados en Smalltalk nos devuelve la lista de países que son poblados, mientras que en Prolog nos devuelve una explosión combinatoria de los posibles países poblados. ¿A qué se debe esta diferencia? Justifique relacionando con conceptos vistos en la cursada.

b) Modifique ambas soluciones poniendo énfasis en los conceptos de **delegación / declaratividad / expresividad (se pide codificar)**[1]. Justifique sus decisiones.

c) Compare la solución final de Objetos con la solución final de Lógico. Explique qué conceptos equivalentes aparecen (qué grado de paralelismo encuentra entre ambos paradigmas).

3) Se quiere obtener el primer número capicúa a partir de un número dado. Las soluciones propuestas son:

   - Resolverlo en el paradigma funcional, definiendo una función en Haskell:

[1] En la solución de Prolog se valorará reemplazar el algoritmo recursivo por alguna otra técnica

```
primerCapicua n = (head . filter capicua . secuenciaDesde) n

secuenciaDesde n = n: secuenciaDesde (n+1)
```

- Resolverlo en el paradigma imperativo, definiendo una función en un lenguaje imperativo que tiene evaluación ansiosa:

```
int primerCapicua(int n) {
   return (head(filter(&capicua, secuenciaDesde(n)));
}

vectorDeInt secuenciaDesde(int n) {
   return(concatenar(n, secuenciaDesde(n + 1));
}
```

Se asume que están hechas en el lenguaje C las siguientes funciones con el mismo resultado que en Haskell:

- **filter**: recibe un puntero a una función booleana y un vector de enteros y devuelve otro vector con los enteros que verifiquen la función
- **head:** recibe un vector y devuelve su primer elemento
- **concatenar**: recibe un vector y un entero y devuelve un nuevo vector con ese entero al principio
- **capicua**: recibe un número y dice si es capicúa

Realizar un análisis comparativo respecto de los siguientes conceptos:

a) Definición explícita/implícita de tipos
b) ¿Qué ventajas presenta la evaluación diferida en la solución de Haskell?

Justifique en todos los puntos haciendo referencia a los conceptos que correspondan.

4) Para calcular el recargo por morosidad de sus clientes, una empresa tiene la siguiente definición:

- a los grandes clientes no se les cobra recargo
- a los clientes medianos les aplica un recargo de 2% sobre el saldo.
- a los clientes comunes (menos de 10.000 $ mensuales) le cobra un monto fijo (a definir para cada cliente) en conceptos de gastos administrativos.

Un cliente mediano nunca puede convertirse en común ni en otro y viceversa.

Explique qué diferencias habría en la resolución de dicho requerimiento en Objetos y un lenguaje imperativo (C / Pascal / o similar) y qué conceptos intervendrían cada alternativa. Aquí no hace falta que codifique.

Nota: debe adaptar los conceptos teóricos al contexto del enunciado (no vale teorizar solamente).