## FINAL DE PARADIGMAS DE PROGRAMACION

# Punto 1

Tengo el predicado p y la función f, definidos de esta forma:

p(1,3). f 1 = 3 p(2,6). f 2 = 6p(X,9):- esNumero(X). f x = 9

suponiendo que el predicado esNumero se satisface para todos los números naturales, o sea, que la consulta ?- esNumero(X).

da las siguientes respuestas:

X = 1; X = 2; X = 3; ... etc.

Hago estas consultas:

?- p(1,X). > f 1

El comportamiento de Prolog y de Haskell ante estas consultas va a ser distinto. Se pide describir

- a. cuál es la diferencia, y con qué conceptos inherentes a los paradigmas lógico y funcional está relacionada.
- b. qué otros tipos de consulta puedo hacer en Prolog que no puedo hacer en Haskell, referidas a este ejemplo, y con qué cualidad que asociamos al paradigma lógico tiene que ver.

# Punto 2

Considerar estas dos expresiones, una en Haskell y la otra en Smalltalk, que describen la misma operación, que se evalúa sobre una secuencia.

```
take 5 (filter (coprimoCon 30) lista)
(coleccion select: [:n | n esCoprimoCon: 30]) from: 1 to: 5
```

La secuencia se representa mediante una lista en el primer caso, y mediante una colección ordenada en el segundo.

Si la secuencia tiene 25000 elementos, tales que se alternan los que sí son coprimos con 30 y los que no (el primero sí, el segundo no, el tercero sí ...) la evaluación de una de las dos expresiones va a tardar mucho menos que la otra.

Indicar:

- a. Cuál es la que tarda menos, y con qué concepto está relacionada la diferencia.
- Si quiero que el caso que tarda mucho pase a tardar poco, ¿qué tengo que hacer?
   O sea ¿en qué sentido tengo que modificar la expresión?
   Relacionar con el concepto de declarativo vs. procedural.
- c. este concepto, además de permitir que tarde menos en el caso de una secuencia larga, permite que la expresión "rápida" funcione para ciertas secuencias que no pueden (en principio) tratarse en el otro caso. Proporcione distintos ejemplos (cuanto más distintos mejor) de listas que pueden tratarse en un caso sí y en el otro no (hay algunos casos obvios y otros no tanto).
- d. Además de lo que se desarrolla en los ítems anteriores, en la expresión en Haskell aparecen otros dos conceptos que relacionamos en la materia con el paradigma funcional.
   Indique cúales son estos conceptos y describa su aparición.

## Punto 3

Los juzgados federales manejan expedientes, una de las cosas que puede pasar es que se tenga que hacer una consulta a un experto.

Cada juzgado tiene un registro de peritos, cada perito puede opinar sobre algunos expedientes, mientras que sobre otros no (pensar en un contador, un ing. en sistemas y un veterinario). La forma de decidir si puede o no depende del tipo de perito.

A todos los peritos se les notifica que tienen un expediente para mirar de la misma forma (p.ej. enviándoles un despacho oficial y anotando la consulta en un libro que los peritos pueden ir a ver al juzgado cuando quieran).

Además, el juzgado puede hacer convenios con facultades, de forma tal que el juzgado puede hacer consultas enviando los expedientes a la facultad, que los maneja. La facultad también puede opinar sobre algunos expedientes sí y otros no. La forma de notificación es distinta para las facultades que para los peritos.

## FINAL DE PARADIGMAS DE PROGRAMACION

Se construye un modelo con estas clases:

(detect: devuelve un elemento cualquiera de una colección que cumpla con la condición).

# Se pide:

- a. Indicar en qué clase/s falta implementar el método puedeOpinarSobre: .
- b. Describir el uso de polimorfismo que se aprecia en este ejemplo, indicando en particular quién lo está aprovechando, o sea, qué código sería más complejo si no hubiera polimorfismo, y en qué consistiría esta complejidad.
- c. Nos dicen que para notificar a los ingenieros en sistemas, además de lo que se hace para cualquier perito, hay que enviarle un mail.
  - Indique cómo resolvería este agregado, y qué concepto/s asociados al paradigma de objetos está usando.
- d. Usando como ejemplo las clases incluidas en este ejemplo, indique qué criterio usaría para decidir, dado un conjunto de clases cuyas instancias deben ser todas polimórficas entre sí, en qué casos usar la herencia y en qué casos no.

# 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 terminarías la frase que empieza así

"en el paradigma lógico siempre hay transparencia referencial, entonces ..."?

Lo mismo para los otros dos paradigmas.

En el caso específico de objetos, dar un ejemplo que colabore con la explicación.