

Condiciones de aprobación:

Para aprobar es necesario simultáneamente:

- obtener 8 puntos de 14, y
- obtener al menos la mitad de los puntos en cada paradigma.

Y recordá: en todas tus respuestas sé puntual, no pierdas el foco de lo que se pregunta. Respuestas en exceso generales son tan malas como respuestas incompletas.



Parte A

Se tienen las siguientes funciones:

```
minBy :: Ord b => (a->b) -> a -> a -> a
```

```
minBy f x y
```

```
    | f x < f y = x
```

```
    | otherwise = y
```

```
minimumBy f (x:xs) = foldl (minBy f) x xs
```

```
foo f g lista = (minimumBy f . filter g) lista
```

Los estudiantes están modelados como tuplas de tres elementos con el nombre, la nota en el primer parcial y la nota en el segundo, por ejemplo:

```
listaPrueba = [("Julia",5,8), ("Andres",2,9), ("Beto",7,2), ("Nina",4,4)]
```

Se cuenta también con las siguientes funciones:

```
esNotaAprobada nota = nota >= 4
```

```
promedio a b = div (a + b) 2
```

1. Se quiere obtener, dada una lista de alumnos:
 - qué alumno de los que aprobaron el primer parcial tuvo la nota más baja en el segundo
 - quién tiene el menor promedio de los 2 parciales habiendo aprobado ambos.

Resolver los dos requerimientos como **consultas** usando las funciones dadas (y las predefinidas en haskell), pero **sin definir nuevas funciones**.

2. Justificar la utilidad del concepto orden superior basándose en los ejemplos anteriores.
3. Inferir el tipo de foo.
4. ¿Qué pasaría si foo se usa con una lista infinita como último parámetro? Justificá tu respuesta.

Parte B

Los siguientes fragmentos de código andan para consultas individuales, su lógica es correcta y no poseen errores de sintaxis **pero** todas las soluciones tienen cosas a mejorar.

Para cada punto:

- a. Analizar si los predicados son totalmente inversibles y hacerlos inversibles si no lo fueran.
- b. Realizar las mejoras que identifiques, más allá de la inversibilidad, explicando por qué es mejor.

```
todosSiguenA(Rey) :-
```

```
    not((personaje(Personaje), not(sigueA(Personaje, Rey)))).
```

```
%Se quiere averiguar qué personajes son seguidos por todos los personajes.
```

```
%Todos los personajes que existen están definidos con hechos del predicado personaje/1
```

```
%El predicado sigueA/2 relaciona dos personajes, donde el primero sigue al segundo.
```

```
comidaVegana(Comida) :-
```

```
    ingrediente( _, Comida),
```

```
    forall(ingrediente(Ingrediente, Comida),
```

```
        (not(contieneCarne(Ingrediente)),
```

```
not(contieneHuevo(Ingrediente)),
not(contieneLeche(Ingrediente))).
```

%El predicado ingrediente relaciona a un ingrediente con la comida de la que forma parte, por ejemplo ingrediente(mayonesa,ensaladaRusa).

```
hayAlgunTestigo(Delito) :-
```

```
    findall(Testigo, testigo(Testigo, Delito), Testigos),
```

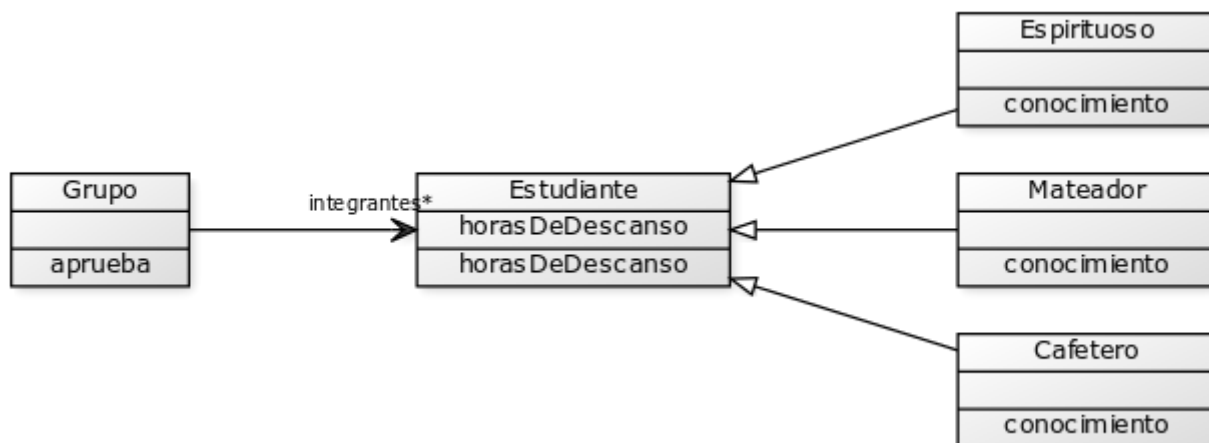
```
    length(Testigos, Cantidad),
```

```
    Cantidad > 0.
```

%El predicado testigo/2 relaciona a una persona con un delito si la misma fue testigo.

Parte C

Recibimos un llamado urgente de la empresa de un amigo que está necesitando una mano: se le fue el único desarrollador a cargo de un nuevo proyecto. Por suerte, nos dejaron un diagrama y comentarios sobre la solución propuesta que nos nos van a servir de ayuda.



Un grupo está formado por varios estudiantes. El mismo, aprueba si la sumatoria de las fuerzas intelectuales, que es el producto de las horas de descanso por el conocimiento de cada uno de sus integrantes, es mayor a 10. La manera en que se calcula el conocimiento de cada tipo de estudiante es diferente.

- ¿Donde definirías la lógica donde se haga el producto entre el descanso y el conocimiento: en Grupo, en Estudiante o en cada subclase de Estudiante? Justifícalo.
- Asumiendo que sólo conocimiento está definido, implementar lo necesario para saber si un grupo **aprueba**.
- Que pasaria si hubiera estudiantes que no son ni cafeteros, ni mateadores ni espirituosos? Incluir el concepto de clase abstracta en la explicación.
- No tenemos detalles, pero sabemos que el conocimiento de un mateador es un 35% más que el de un cafetero, y en general suelen comportarse parecido. Qué cambios introducirías en la solución para contemplarlo. Justificar e implementar esta parte de la solución.
- Se quiere agregar una nueva funcionalidad que es que el grupo pasa la noche haciendo un trabajo práctico. Como consecuencia de ello, puede ser que algún estudiante espirituoso abandone su preferencia por las bebidas alcohólicas y se vuelva cafetero, o que todos se hagan mateadores. ¿Qué agregados o cambios harías para contemplar esta posibilidad? Justificar y codificar las partes claves de la solución propuesta.
- Aparecen también la figura de un tutor, que no es estudiante, pero se lo agrega como un integrante más del grupo. ¿Qué sería imprescindible definir?