

Ejercicio 1

Queremos hacer un programa para armar partituras. Sabemos que las notas se escriben con distintas figuras que representan la duración de la misma, por ejemplo una redonda dura 4 tiempos, una blanca 2 y una negra 1. A su vez, a una figura se le puede agregar **hasta dos puntillos** que incrementan la duración de la misma: una redonda con puntillo dura 6 tiempos (equivale a redonda+blanca) y una redonda con 2 puntillos dura 7 tiempos (equivale a redonda+blanca+negra).

Tenemos las siguientes implementaciones:

<pre> duracion(redonda,4). duracion(redondaConPuntillo,6). duracion(redondaCon2Puntillos,7). duracion(blanca,2). duracion(blancaConPuntillo,3). duracion(blancaCon2Puntillos,3.5) </pre>	<pre> #Fig (v.i. dur, cantPs) >> duracion cantPs = 0 ifTrue: [^ dur]. cantPs = 1 ifTrue: [^ dur * 1.5]. cantPs = 2 ifTrue: [^ dur * 1.75]. </pre>
--	---

- A. Comparar ambas soluciones en términos de declaratividad y expresividad.
- B. Mirando estas consultas:


```

?- duracion(blanca, 4).      ?- duracion(blanca, D).
?- duracion(F, 3).          ?- duracion(F, D).

```

 ¿Qué “consultas” debo hacer en Smalltalk para obtener el mismo funcionamiento? ¿A qué se debe que sea más sencillo / más difícil? (No se requiere implementar la solución)
- C. La forma de modelar las figuras con puntillo en la solución de Prolog no es conveniente. Generalizar la lógica de los puntillos para que la duración de esas figuras quede definida por comprensión y no por extensión para cada figura base. Indicar qué conceptos se usaron para poder hacerlo.

Ejercicio 2

1. `primerUtilAPartirDe n = (head . filter esUtil) [n..]`
2. `primerUtilAPartirDe n = head (filter esUtil [n..])`
3. `primerUtilAPartirDe n = head filter.esUtil [n..]`
4. `primerUtilAPartirDe n = head.filter (esUtil [n..])`

- A. ¿Cuáles de las anteriores definiciones son correctas? Justifique.

- B. Si se define la función:

```

esUtil 0 = True
esUtil 1 = True
esUtil _ = False

```

En caso de ser posible, mostrar con la función `primerUtilAPartirDe`:

- una invocación que devuelva una respuesta válida,
- otra invocación que, sin dar error, nunca termine de evaluarse y
- otra invocación que provoque un error.

Justificar conceptualmente **en todos los casos**.

- C. Se quiere hacer más genérica la función anterior, de manera que no se llame necesariamente a la función `esUtil`, sino que se pueda llamar en su lugar a otra función, según los parámetros con que se invoque a la función principal.
 1. Implementarla
 2. Justificar conceptualmente.
 3. Definir su tipo
 4. Mostrar un ejemplo de invocación y respuesta

Ejercicio 3

En un aula hay diferentes sillas y bancos de diferente tamaño, que ocupan espacio. Se quiere saber si el aula está llena.

<pre>#Aula (vi: sillas, bancos, espacioTotal) >>estaLlena ^ espacioOcupado = espacioTotal >>espacioOcupado ^(sillas sum: [:silla silla tamañoSilla]) + (bancos sum: [:banco banco tamañoBanco])</pre>	<pre>#Banco >>tamañoBanco "devuelve el tamaño del banco" #Silla >>tamañoSilla "devuelve el tamaño de la silla"</pre>
--	---

- A. Esta solución tiene un error que hace que no funcione. ¿Cuál es? Corregirlo.
- B. Esta solución puede ser mejorada aplicando conceptos fundamentales de la materia. Plantear dicha mejora y explicar sus beneficios con un caso concreto
- C. Ahora, en el aula hay también bancos dobles, cuyo tamaño se calcula como el doble del que tendrían si fueran bancos como los anteriores. Resolverlo de la mejor manera y justificar los conceptos que intervienen (No es necesario definir cómo se calcula el tamaño de un banco común).

Ejercicio 4

El mismo problema del punto anterior se plantea el Lógico. En la base de conocimiento se tienen hechos como los siguientes:

```
%aula(id, mueble, fila).
%espacioTotal(id, espacio).
%banco(peso, material, altura).
%silla(altura, material).
aula(618, banco( 2, madera, 1.50), 1).
aula(618, banco(3, metal, 1.30), 1).
aula(618, silla(1.20, madera), 1).
aula(618, silla(1.30, madera), 2).
aula(magna, banco(2, plomo, 2.1), 5).
espacioTotal(618, 500).
```

El tamaño que ocupa el banco se calcula como la altura por el peso al cuadrado, mientras que en las sillas equivale al doble de la altura.

- A. Desarrollar el predicado que resuelva el espacio disponible de un aula.
- B. Justificar las ventajas del polimorfismo, orden superior, aplicación parcial y herencia en la solución, en caso de haberlos aplicado, o explicar los motivos de su ausencia, en caso contrario.