

Informática para a Musicologia (IPM) 2024/25

Jupyter Notebooks

Docente: [J.N. Oliveira](#)

Departamento de Informática da U. Minho, em colaboração com a  ENSICO

Aula de 8-Out: Haskell em Jupyter Notebook

Manipulação de sequências na linguagem Haskell.

Dos números e palavras até às frases musicais. A operação de colagem ("zipping") de duas sequências. Sequências de pares.

Sequências que representam música: representação de eventos sonoros por pares (altura,duração). Interação com notação Abc.

Caso de estudo: melodia pentatónica 'Carnaval serrano' (Peru-Bolivia), fonte: Flutes from the Andes, por Guillermo De La Roca. Epm Music B000027YVO.

Encadeamento de operações (composicionalidade).

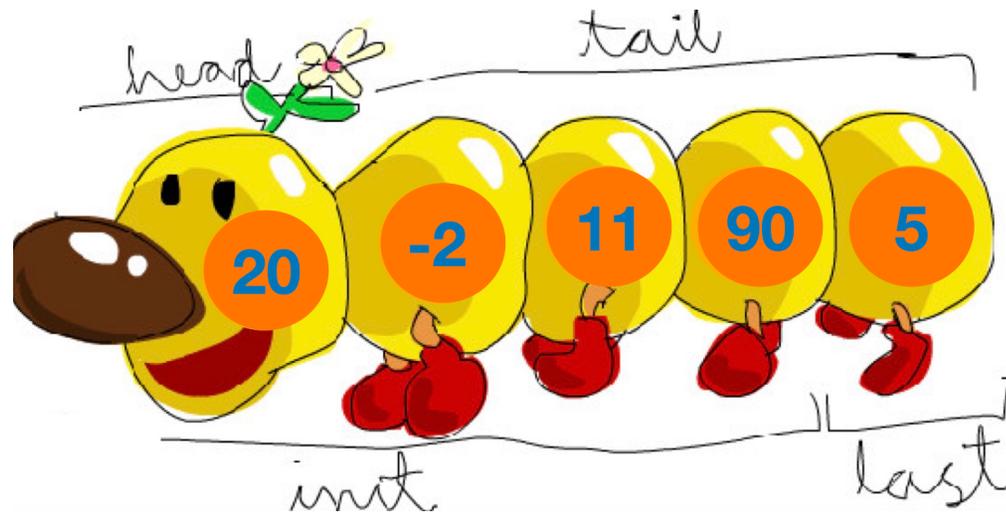
⚠ Importante: correr sem mexer as células seguintes.

In []:

```
:opt no-lint
:m Data.Ratio
:m Data.Char
:m Data.List
```

```
In [ ]: :l ../src/Cp.hs
        :l ../src/Reducer.hs
        :l ../src/Ipm.hs
        :l ../src/Abc.hs
        :l ../src/CS.hs
```

Sequências em geral



➡ As funções da "centopeia"

3.1 - Depois de observar a figura, avaliar as expressões seguintes e tirar conclusões:

```
In [ ]: centopeia = [20,-2,11,90,5]
        ---
        centopeia
```

```
In [ ]: head centopeia
```

```
In [ ]: tail centopeia
```

```
In [ ]: last centopeia
```

```
In [ ]: init centopeia
```

3.2 - Continuar, avaliando as expressões seguintes e tirando conclusões:

```
In [ ]: length centopeia
```

```
In [ ]: sum centopeia
```

```
In [ ]: 9 : centopeia
```

```
In [ ]: centopeia ++ centopeia
```

```
In [ ]: rotr centopeia
```

```
In [ ]: rotl centopeia
```

```
In [ ]: reverse centopeia
```

```
In [ ]: sort centopeia
```

3.3 - Acima, a sequência `centopeia` ficou ordenada.

▮ O que sem dificuldade faríamos sem precisar de um computador...

Mas experimentem ordenar "à mão" a lista seguinte:

```
In [ ]: lista_longa = [21,421,28,166,83,371,334,265,110,257,198,438,59,16,329,450,393,209,
112,61,45,242,8,432,296,122,360,48,92,116,30,434,103,389,500,446,131,
165,316,322,324,137,160,210,289,238,295,174,285,499,71,384,376,171,478,
338,230,52,352,186,36,146,120,256,424,108,388,487,188,156,344,261,395,
224,232,75,211,119,113,58,76,50,350,14,63,439,47,169,363,191,313,459,9,
114,409,27,65,117,443,437,283,109,217,321,204,341,299,355,323,288,278,185,
202,404,170,239,178,291,126,234,417,189,335,12,153,330,220,33,337,82,496,
227,412,88,498,426,78,128,148,177,213,358,408,343,365,364,357,318,455,430
]
```

E comparem com fazer o mesmo por computador:

```
In [ ]: sort lista_longa
```

● Computação e programação

As operações acima (reverse , sort , sum etc) foram **programadas** para funcionarem num **computador** para quaisquer listas, de qualquer tamanho.

É para isto que os **computadores** servem:

Um **computador** é uma máquina que "faz" coisas muito **trabalhosas** por nós...

... **depressa** ...

... sem se **aborrecer** ...

... nem se **enganar** !



👉 Vamos tirar partido disso em **musicologia assistida por computador**.

3.4 - É dada a seguinte linha melódica:

```
In [ ]: mel = ["A", "^G", "A", "B", "c", "B"]
```

Vamos visualizá-la:

```
In [ ]: abcplease mel
```

Repare-se que é a célula melódica básica do tema [La Folia](#) de Arcangelo [Corelli](#) (1653-1713), aqui transposta de ré menor para lá menor.

3.5 - Mostrar na célula abaixo que `mel` só tem 4 notas (diferentes).

Sugestão: basta eliminar as repetidas e contar as restantes...

```
In [ ]:
```

Pares

Podemos agrupar números em **pares**, **triplos**, etc. A célula seguinte contém dois pares de números - corram-na.

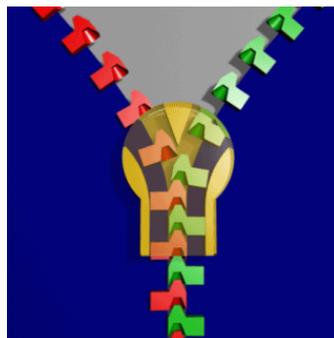


```
In [ ]: p = (5,2)
        q = (4,0)
        --
        p
        q
```

3.6 - O que fazem as operações `fst` e `snd` da célula seguinte? Corram-na e respondam conforme o que observam como resultado.

```
In [ ]: fst p
        snd q
```

 "Zipping"



3.7 - Avaliar as expressões seguintes e tirar conclusões quanto ao significado de `zip` :

```
In [ ]: zip [1..5] centopeia
```

```
In [ ]: zip [] centopeia
```

```
In [ ]: zip centopeia [12,100]
```

```
In [ ]: zip ["Maria","Manuel"] [1994,2002]
```

```
In [ ]: zip ["Maria","Manuel"] [1994,2002,2020,1955]
```

3.8 - Avaliar a célula seguinte:

```
In [ ]: r = [3%4,3%4,3%4,3%4,3%4,3%4,1%4]
m = zip me l r
-----
m
```

Tirar conclusões após a "visualização" de m :

```
In [ ]: abcplease m
```

3.9 - Repetir o que se fez acima na célula seguinte por forma a que cada nota tenha a duração de uma mínima:

```
In [ ]:
```

3.10 - O que fez abcplease ?

```
In [ ]: abcShow
```

👉 Para ouvir: (a) copiar para o *clip-board* o código ABC que abcShow gerou acima ; (b) entrar no [editor ABC on-line](#), colar e pôr a tocar.

3.11 - Avaliar a célula seguinte e tirar conclusões quanto ao significado de unzip :

In []:

unzip m

3.12 - De volta a

Designação	Significado	Descrição detalhada
fst	o primeiro	dá o primeiro elemento do par (a, b) , isto é a
snd	o segundo	dá o segundo elemento do par (a, b) , isto é b

seja dada a nota:

In []:

n = ("^G", 3%4)

Escrever nas células seguintes expressões que calculem:

- o número de caracteres do primeiro elemento da nota n
- o segundo elemento dessa nota
- o dobro deste último.

In []:

In []:

In []:

 **Caso de estudo**

Em `carnaval_serrano` está registada uma pequena melodia andina (Peru-Bolivia). [Fonte: *Flutes from the Andes*, por Guillermo De La Roca. *Epm Music B000027YVO*].

Sem inspecionar `carnaval_serrano` , escrever expressões que permitam saber:

3.13 - Quantas notas e quantas notas diferentes tem essa música?

In []:

```
.....
```

In []:

```
...
```

3.14 - Qual é a primeira nota?

In []:

```
...
```

3.15 - Qual é a duração da última nota?

In []:

```
...
```

3.16 - E a respectiva altura de som (*pitch*)?

In []:

```
...
```

3.17 - Vamos verificar visualizando (e ouvindo) `carnaval_serrano` :

In []:

```
abcplease carnaval_serrano
```

In []:

```
abcShow
```

3.18 - Vamos agora usar `unzip` para decompor `carnaval_serrano` na sua parte melódica e rítmica:

```
In [ ]: (mel, ritm) = unzip carnaval_serrano
```

3.19 - Qual é a duração total de `carnaval_serrano` ?

```
In [ ]: ...
```

3.20 - Quantas notas diferentes tem `carnaval_serrano` ignorando a parte rítmica?

```
In [ ]: ...
```

3.21 - Que notas são essas? (Substituir `undefined` .)

```
In [ ]: ...
```

Visualize-as:

```
In [ ]: abcp\ease ....
```

Conclua que a melodia está numa [escala pentatónica](#). De que tipo?

Encadeamento de operações

Designação	Significado	Descrição detalhada
$f \cdot g$	depois de	aplicar f depois de g , i.e.: $(f \cdot g) a = f (g a)$
id	não fazer nada	$id x = x$ qualquer que seja x

(Composição funcional). Por exemplo, em vez de

```
| abcplease (reverse (rotl notas))
```

escrever

```
| (abcplease . reverse . rotl) notas
```

É mais prático.

3.22 - Para apreciarmos o encadeamento de operações começemos por executar a célula seguinte, que **resume** tudo o que acima fizemos até à visualização em partitura da escala subjacente a `carnaval_serrano` :

```
In [ 1]: (abcplease . reverse . rotl . nub . fst . unzip) carnaval_serrano
```

3.23 - Vamos agora correr agora todos os passos dessa expressão um de cada vez e comentar os resultados:

```
In [ ]: unzip carnaval_serrano
```

```
In [ ]: (fst . unzip) carnaval_serrano
```

```
In [ ]: (nub . fst . unzip) carnaval_serrano
```

```
In [ ]: (rotl . nub . fst . unzip) carnaval_serrano
```

```
In [ 1]: (reverse . rotl . nub . fst . unzip) carnaval_serrano
```

```
In [ ]: (abcplease . reverse . rotl . nub . fst . unzip) carnaval_serrano
```

