

# 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 29-Out:

1ª parte - O compasso como sequência infinita e a flexibilidade dessa abordagem. Introdução à análise musical por amostragem. 'Sampling' e o seu papel em 'computer-aided musicology'. O operador 'sample' da biblioteca `lpm.hs`. Ilustração com vários exemplos. Caso de estudo: análise do tema das Variações Abegg (op.1) de R. Schumann (1810-1856).

2ª parte - **Apresentação do 1º trabalho prático:** alocação de trabalho na edição colaborativa na [Wiki::score](#) da ópera [Demetrio a Rodi](#) de Gaetano Pugnani (1731-1798).

**⚠ Importante:** correr sem mexer as células seguintes.

```
In [ ]:
:opt no-lint
:m Data.Char
:m Data.List
:m Data.List.Split
:m Data.Ratio
```

Módulos desenvolvidos para a disciplina:

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

Dados ("case studies"):

```
In [ ]: :l ../src/CS.hs
```

## De volta aos compassos

Voltemos a este tema começando por fazer o exercício que se segue:

**7.1** - Avaliar as células seguintes - qual foi a diferença?

```
In [ ]: abcPlayM "C" "4/4" carnaval_serrano
```

```
In [ ]: abcPlay "C" "4/4" quatern carnaval_serrano
```

**7.2** - Antecipar o resultado da célula seguinte - qual é a diferença?

```
In [ ]: abcPlay "C" "4/4" (1%2:quatern) carnaval_serrano
```

👉 Sobre `abcplease`, `abcPlayM` e `abcPlay`:

Designação	Significado	Descrição detalhada
<code>abcplease</code>	mostrar partitura	<code>abcplease m</code> mostra a partitura de <code>m</code> sem armação de clave nem compasso, nem barras de compasso
<code>abcPlayM</code>	mostrar partitura	<code>abcPlay K C m</code> - mostra a partitura de <code>m</code> com armação de clave <code>K</code> , compasso <code>C</code> e barras de compasso ( <b>regular</b> ) deduzidas a partir de <code>C</code>
<code>abcPlay</code>	mostrar partitura	<code>abcPlay K C c m</code> - mostra a partitura de <code>m</code> com armação de clave <code>K</code> , compasso <code>C</code> e barras de compasso ( <b>possivelmente irregular</b> ) segundo <code>c</code>

**Compassos** - Os seguintes compassos estão pré definidos:

Designação	Significado	Descrição detalhada
una	unário	barras de compasso a cada 1 semínima ( $\frac{1}{4}$ )
bin	binário	barras de compasso a cada 2 semínimas ( $\frac{2}{4}$ )
tern	ternário	barras de compasso a cada 3 semínimas ( $\frac{3}{4}$ )
quatern	quaternário	barras de compasso a cada semibreve (1)

**7.3** - Indique que expressão das seguintes

(a) `abcPlay "C" "none" [1%8,1%4,1%2,3%4,5%4] carnaval_serrano`

(b) `abcPlay "C" "none" [1%4,2%4,3%4,3%8,1%16,5%8] carnaval_serrano`

(c) `abcPlay "C" "none" [1%2,2%4,3%4,4%4,9%8] carnaval_serrano`

produz que pentagrama da figura:

1 

2 

3 

Verifique as suas respostas nas células seguintes.

In 1:

In [ ]:

In [ ]:

**7.4** - A célula seguinte regista uma ocorrência do tema *Promenade* recorrente nos *Quadros de uma Exposição*, de Modeste Mussorgsky (1839-1881):

In [ ]:

```
promenade = [("F",1 % 4),("E",1 % 4),("A",1 % 4),("B",1 % 8),("e",1 % 8),("c",1 % 4),
             ("B",1 % 8),("e",1 % 8),("c",1 % 4),("A",1 % 4),("B",1 % 4),("F",1 % 4),
             ("E",1 % 4)]
```

Mostre-a em partitura sabendo que o primeiro compasso é  $\frac{5}{4}$  e o segundo é  $\frac{6}{4}$ :

In [ ]:

**7.5** - O que são afinal bin , tern , quatern ? Avaliar as expressões que constam das células seguintes e tirar conclusões: `

In [ ]:

```
take 10 bin
```

In [ ]:

```
take 10 tern
```

In [ ]:

```
take 1000 quatern
```

**7.6** - Avaliar as expressões que constam da célula seguinte: `

```
In [ ]: tern = 3%4 : tern
-----
take 20 tern
```

 Quer dizer: tern é a sequência que começa por 3%4 e a seguir é igual a si própria:

- tern = 3%4 : tern
- tern = 3%4 : (3%4 : tern)
- tern = 3%4 : (3%4 : (3%4 : tern))
- ...

É uma sequência infinita...

---

**7.7** - Definir bin e quatern de forma idêntica a tern :

```
In [ ]: bin = undefined
quatern = undefined
-----
take 20 bin
take 50 quatern
```

**7.8** - E se nos tivéssemos enganado e definido...

```
In [ ]: bin = 1%2 : bin
take 20 bin
```

Interpretar o resultado.

---

## Padrões repetitivos (Conclusão)

Já verificamos que saber gerar sequências (infinitas) que seguem um dado padrão é muito útil. Vamos praticar um pouco mais sobre a forma de as definir.

**7.9** - Queremos uma sequência repetitiva `s` que comece por `1`, depois `3`, depois `0` e assim sucessivamente. Completar a sua definição e observar os seus primeiros 10 elementos:

```
In [ ]:
s = undefined
take 1000 s
```

**7.10** - Agora mais complicado: `s` deverá começar por `1`, `3`, `0` e depois continuar como `r`; e `r` deverá começar por `10`, `24` e continuar como `s`, etc, etc:

```
In [ ]:
s = undefined
r = undefined
--
take 1000 s
```

**7.11** - Finalmente, antecipe o resultado da célula seguinte, antes de a correr: que sequência é `s` ?

```
In [ ]:
s = 1: map (2+) s
--
take 10 s
```

## Amostragem (*Sampling*)

- Pela remoção de notas repetidas produz obtem-se *abstração* musical sempre que o ritmo não é importante e a análise da base melódica for o objectivo.
- Mais ainda, frequentemente desejamos abstrair detalhes da própria melodia e focar na linha tonal removendo, por exemplo, notas de passagem, apogiaturas, etc.
- A amostragem (*sampling*) faz isso, cf:

Designação	Significado	Descrição detalhada
<code>sample</code>	amostragem	<code>sample s m</code> amostra a melodia <code>m</code> segundo a sequência de durações <code>s</code>

**7.12** - Voltando a

```
In [ ]: abcPlayM "F" "C" frerej
```

```
In [ ]: s = 1%2 : 1%4 : 1%4 : s  
-----  
abcPlayM "F" "C" (sample s frerej)
```

correr a seguinte célula para confirmar como `sample` funciona:

```
In [ ]: abcPlayM "F" "C" (sample bin frerej)
```

---

**7.13** - Antecipar o resultado das células seguintes, antes de as executar:

```
In [ ]: abcPlayM "F" "C" (sample tern frerej)
```

```
In [ ]: abcPlayM "F" "C" (sample quatern frerej)
```

```
In [ ]: abcPlayM "F" "C" (sample una frerej)
```

```
In [ ]: abcPlayM "F" "C" (sample una frerej)
```

---

**7.14** - Repetir o exercício anterior quanto à célula seguinte:

```
In [ ]: s = 1%4 : 1%2 : 1%4 : s  
-----  
abcPlayM "F" "C" (sample s frerej)
```

In [ ]: abcShow

7.15 - Definir a sequência de amostragem `s` que permite, na célula seguinte, obter o seguinte efeito:



In [ ]: `s = undefined`  
 ----  
`abcPlayM "F" "C" (sample s frerej)`

**+** **Caso de estudo** (sampling) - Tema do opus 1 de Robert Schumann (1810-1856) - as *Variações Abegg* sobre o nome (acha-se) da sua aluna Pauline von Abegg, cf:

In [ ]: `m = "ABegg"`  
 ----  
`m`  
`abcplease m`

No tema elabora-se esse motivo A-b-e-g-g :

In [ ]: `c = 1%4:tern`  
 ----  
`abcPlay "F" "3/4" c abegg`

7.16 - Experimentem a célula seguinte, mas com o vosso próprio nome:

```
In [ ]: n = name2music "Nuno Oliveira"
-----
abcplease (zip n una)
abcShow
```

---

**7.17** - Considerando o tema agora com o baixo,

```
In [ ]: m = abegg # abegg2
-----
abcPlay "F" "3/4" c m
```

fazer *sampling* de toda a melodia por forma a preservar apenas a primeira nota de cada compasso (mantendo a anacrusa):

```
In [ ]: a = undefined
-----
abcPlay "F" "3/4" c a
```

---

**7.18** - Fazer agora *sampling* da mesma melodia por forma a preservarmos apenas a primeira nota de cada dois compassos (mantendo a anacrusa) e mantendo o que tínhamos no baixo.

```
In [ ]: a = undefined
-----
abcPlay "F" "3/4" c a
```

```
In [ ]: abcShow
```

---

**7.19** - Antecipar agora o que vai ser produzido na célula seguinte, sem executar:

```
In [ ]: abcPlay "F" "3/4" c (sample tern abegg)
```

---

**7.20** - Queremos agora: (a) mostrar o tema segundo o padrão rítmico *semínima, mínima, semínima, mínima, ....*; (b) manter o baixo como acima. Fazer isso na célula que se segue:

In [ ]:

```
x = undefined
---
abcPlay "F" "3/4" undefined
```

---