

nudger (-0.00125)

```
d1 $ seqPLoop [
  (0, 16, 2, 1) $ seqPLoop [
    (|+| up (slow 7 $ "<0 -2>" (1/24) $ s "juno" # n "<7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -6.5 -7>"
    # legato 0.5
    # gain 1.15
    , striate' "<16 12>" (1/24) $ s "juno" # n "<7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -6.5 -7>"
    # legato 0.5
    # gain 1.15
  ] ),
  (16, 2, 1) $ seqPLoop [
    (|+| up (slow 7 $ "<0 -2>" (1/24) $ s "juno" # n "<7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -6.5 -7>"
    # legato 0.5
    # gain 1.15
    , striate' "<16 12>" (1/24) $ s "juno" # n "<7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -6.5 -7>"
    # legato 0.5
    # gain 1.15
  ] )
]
```

LIVE CODING

curso + algorave

código & música -> em rede

TidalCycles +

Troop +

d3 \$ s "peo-12a-e"

hush

d4 \$ (0.125 <- nudger (-0.00125) s "[midi](5,8)" # n "[3 0!3]" # legato (slow 32 (sine+0.1)) # gain 1 # orbit 3

d1 \$

```
degradeBy 0.2 $
stut' 2 (1/3) ( # coarse "[64 16 32 8 2 4]/1.75" ) $
slow 2 $
stut' 2 (1/3) ( # coarse "[64 16 32 8 2 4]/1.75" ) $
(|+| up (slow 7 $ "<0 -2>" (1/24) $ s "juno" # n "<7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -6.5 -7>"
# legato 0.5
# gain 1.15
, striate' "<16 12>" (1/24) $ s "juno" # n "<7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -6.5 -7>"
# legato 0.5
# gain 1.15
]
```

d2 \$

```
slow 16 $
swingBy (1/3) 4 $
striate' 2 (1/16) $
s "einv*8" # n "[3 0!3]" # legato (slow 32 (sine+0.1)) # gain 1 # orbit 3
# up "4"
```

d3 \$ loopAt "< 1 2 3 4 5 6>" \$ chop "[48 24 12 56]" \$ slow 48 \$ s "binv12*12" # n (run 12)

bônus:

Tidal + SuperCollider hacks

ou

Tidal + Processing

LIVECODING

curso + algorave

música feita com código

Nesse curso os participantes aprenderão a fazer música ao vivo usando códigos a partir de algoritmos simples, que se transformam e ganham complexidade enquanto são escritos. Seus sons podem ser ouvidos desde o primeiro instante, prática conhecida como **livecoding**.

O **livecoding** rompe com o paradigma das interfaces gráficas e softwares proprietários como únicos intermediários e guias entre a pessoa que cria a música e seu instrumento, o computador, proporcionando assim um modo livre de expressividade.

Utilizaremos a mais compacta e sucinta linguagem para tal: TidalCycles ou apenas Tidal, dos seus princípios básicos às combinações que podem gerar complexas composições. Tocaremos juntos colaborativamente em tempo real, possibilitando criações conjuntas e sonoridades inesperadas.

Ao final do curso os participantes poderão escolher entre uma iniciação ao livecoding de imagens, utilizando Tidal em conjunto com Processing, ou se aprofundarem no aspecto sonoro tendo uma iniciação ao SuperCollider e seu uso em conjunto com o Tidal.

Encerraremos com um evento aberto ao público conhecido como Algorave.

As Algoraves são eventos que acontecem em todo o mundo em que a(s) pessoa(s) no palco escrevem programas ao vivo que geram música e/ou imagens, mostrados ao público através de projeções. Os participantes terão aí a oportunidade de tocar lado a lado com livecoders experientes e assim poderao trocar conhecimentos adquiridos durante o curso.

Os objetivos gerais da oficina são:

- Propiciar sessões de experimentação e criação colaborativa e compartilhamento do conhecimento;
- Familiarizar os participantes com o conceito e a prática do livecoding, assim como com a sintaxe do Tidal;
- Desenvolver estratégias tanto para composição quanto para performance através de algoritmos;
- Fornecer ferramentas para que os participantes possam continuar seu aprendizado e desenvolvimento após o curso.

```
41 $ seqLoop {
  (0, 10, stack {
    ((+) up (slow 2 $ (0 1 2 3 4 5 6 7 8 9 10 11) $ (1+) up (slow 0 $ (0 1 2 3 4 5 6 7 8 9 10 11) $ s "blip" # n "[5 0 1]" # legato 0.05 # begin (slow 32 (sine*0.5))
    # orbit 2
    . s "blip"
    # n (fast
    . s "assay"
    # legato
    # gain 1.1
    . strlate" 2 (1/3) 4 s
    # legato 0
    # gain 0.8
  } )
}
```

1º encontro:

- Conceitos básicos de livecoding e uma visão geral das linguagens existentes.
- Instalação dos programas a serem utilizados, todos de código aberto ou software livre.
- Uma visão geral do Tidal e suas possibilidades.
- Seção de experimentação individual.
- Aprender a aprender: recursos e comunidades online para compartilhar conhecimento.

```
120. 2º encontro:
((+) up (slow 2 $ (0 1 2 3 4 5 6 7 8 9 10 11) $ (1+) up (slow 0 $ (0 1 2 3 4 5 6 7 8 9 10 11) $ s "blip" # n "[5 0 1]" # legato 0.05 # begin (slow 32 (sine*0.5))
# orbit 2
. s "blip"
# n (fast
. s "assay"
# legato
# gain 1.1
. strlate" 2 (1/3) 4 s
# legato 0
# gain 0.8
. s "blip"
# gain 0.8
}
```

2º encontro:

- Como criar e organizar a própria galeria de samples
- Do micro ao macro - de um único sample à uma composição: as funções do Tidal pensadas em suas dimensões dentro do processo de criação dos algoritmos.
- Estratégias para composição vs. estratégias para performance ao vivo.
- Seção de experimentação coletiva.
- Aprender a aprender: vídeos de performances, ler, experimentar e recombina códigos de outrem.

```
3º encontro:
((+) up (slow 2 $ (0 1 2 3 4 5 6 7 8 9 10 11) $ (1+) up (slow 0 $ (0 1 2 3 4 5 6 7 8 9 10 11) $ s "blip" # n "[5 0 1]" # legato 0.05 # begin (slow 32 (sine*0.5))
# orbit 2
. s "blip"
# n (fast
. s "assay"
# legato
# gain 1.1
. strlate" 2 (1/3) 4 s
# legato 0
# gain 0.8
. s "blip"
# gain 0.8
}
```

3º encontro:

- Os participantes poderão aqui escolher entre duas possibilidades:
- Opção 1:**
 - Introdução ao SuperCollider e como produzir seus próprios synths.
 - Expandir o repertório de possibilidades sonoras utilizando o SuperCollider em conjunto com o Tidal.
- Opção 2:**
 - Como utilizar o Tidal em conjunto com o Processing para fazer ao mesmo tempo som e imagem.

Algorave!

Os participantes do curso tocarão junto com artistas convidados em um evento aberto ao público, documentado e transmitido via streaming, acompanhados por artistas visuais que interpretarão suas músicas em projeções.

Participantes do curso:

- O número máximo é de 15 pessoas
- Não precisam ter quaisquer conhecimentos prévios sobre programação.
- Devem trazer laptops (Linux, Windows ou Mac) e fones de ouvido.

Participantes da Algorave:

Artistas convidados:

Livecoders: Gil Fuser, Gustavo Brunoro (Beise) e Lucas Rodrigues (LV).

Artistas visuais: Cabelo - Bijari, Sandro Micolli.

Streaming:

Operação: Luiza Xavier.

Programação visual: Gil Fuser e Luiza Xavier.

Documentação:

vídeo e foto: Georgia Branco + assistente.

Assistente de produção:

Gustavo Brunoro (a confirmar).

```
42 $
slow 10 $
swingly (1/3) 4 $
strlate" 2 (1/10) $
s "blip" # n (run 0)
# up "1-14"

43 $ looppt "x 1 2 3 4 5 6" $ chop "[1 40 20 12 50 1]" $ slow 40 $ s "blip" # n (run 12)
```


Tech rider:

Curso:

Mesas e cadeiras para os participantes.

Projektor, tela e cabo HDMI.

Mesa de som ou mixer.

Cabo P2 stereo -> 2 P10 (ou P10 stereo)™

Caixas acústicas pré-amplificadas ou passivas + amplificador.

Algarave:

Bancada para os participantes.

Mesa e cadeira próximas a bancada para a operação do streaming.

Mínimo luz: 3 Focos coloridos (El ipsos ou PC) com os equipamentos disponíveis no local.

Não há operação de luz durante a apresentação.

3 Caixas de retorno 250 Watts amplificadas

3 Réguas ce Ac 110 com 5 Tomadas

Mesa de som com um mínimo de oito canais.

PA compatível com o tamanho do local

Todo cabeamento necessário.

Internet banda larga via ethernet.

Orçamento:

Curso:

Valor hora: R\$ 250 x 3 encontros x 3 horas = R\$ 2.250

Desenvolvimento de material didático para consulta online e em aula: R\$ 1.000

Algarave:

Valor: R\$ 12.000

Duracao do evento: 3 horas

inclui: cachê para artistas convidados, assistência de produção, produção geral, executiva e técnica, montagem, transmissão via streaming, documentação em vídeo e fotos, produção de material gráfico online e divulgação nas redes sociais.

Total: R\$ 15.250