

# *Programação Reativa para Educação*

**Francisco Sant'Anna**

`francisco@ime.uerj.br`

`http://github.com/fsantanna-no/101`

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await KEY_PRESS;  
emit GRAPHICS_DRAW_PIXEL(26,26);
```

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await KEY_PRESS;  
emit GRAPHICS_DRAW_PIXEL(26,26);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(26,26);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(27,27);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(28,28);
```

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await KEY_PRESS;  
emit GRAPHICS_DRAW_PIXEL(26,26);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(26,26);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(27,27);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(28,28);
```

Como desenhar uma linha  
cruzando a tela?

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await KEY_PRESS;  
emit GRAPHICS_DRAW_PIXEL(26,26);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(26,26);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(27,27);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(28,28);
```

Como desenhar uma linha  
cruzando a tela?

# “Olá Mundo!”

```
emit GRAPHICS_DRAW_PIXEL(25,25);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await KEY_PRESS;  
emit GRAPHICS_DRAW_PIXEL(26,26);
```

```
emit GRAPHICS_DRAW_PIXEL(25,25);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(26,26);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(27,27);  
await 1s;  
emit GRAPHICS_DRAW_PIXEL(28,28);
```

Como desenhar uma linha  
cruzando a tela?



# Programas “reais”

- Fugir de problemas matemáticos e abstratos
- Buscar o palpável, concreto e visual

# **Repetição**

**(O “burro de carga” da computação)**

# **Repetição**

**(O “burro de carga” da computação)**

# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;  
loop do  
    emit GRAPHICS_DRAW_PIXEL(posicao,posicao);  
    await 100ms;  
    posicao ← posicao + 1;  
end
```

# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;
loop do
    emit GRAPHICS_DRAW_PIXEL(posicao,posicao);
    await 100ms;
    posicao ← posicao + 1;
end
```

```
var integer posicao ← 0;
loop do
    emit GRAPHICS_DRAW_PIXEL(posicao,posicao);
    await 100ms;
    posicao ← posicao + 1;
    if posicao = 50 then
        break;
    end
end
```

# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;  
loop do  
    emit GRAPHICS_DRAW_PIXEL(posicao,posicao);  
    await 100ms;  
    posicao ← posicao + 1;  
end
```

Palavras em **negrito**?

```
var integer posicao ← 0;  
loop do  
    emit GRAPHICS_DRAW_PIXEL(posicao,posicao);  
    await 100ms;  
    posicao ← posicao + 1;  
    if posicao = 50 then  
        break;  
    end  
end
```

# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;  
loop do  
    emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
    await 100ms;  
    posicao ← posicao + 1;  
end
```

Palavras em **negrito**?

```
var integer posicao ← 0;  
loop do  
    emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
    await 100ms;  
    posicao ← posicao + 1;  
    if posicao = 50 then  
        break;  
    end  
end
```

# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;  
loop do  
  emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
  await 100ms;  
  posicao ← posicao + 1;  
end
```

Palavras em **negrito**?

```
var integer posicao ← 0;  
loop do  
  emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
  await 100ms;  
  posicao ← posicao + 1;  
  if posicao = 50 then  
    break;  
  end  
end
```



# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;  
loop do  
  emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
  await 100ms;  
  posicao ← posicao + 1;  
end
```

Palavras em **negrito**?

```
var integer posicao ← 0;  
loop do  
  emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
  await 100ms;  
  posicao ← posicao + 1;  
  if posicao = 50 then  
    break;  
  end  
end
```

# Repetição

## (O “burro de carga” da computação)

```
var integer posicao ← 0;  
loop do  
  emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
  await 100ms;  
  posicao ← posicao + 1;  
end
```

Palavras em **negrito**?

```
var integer posicao ← 0;  
loop do  
  emit GRAPHICS_DRAW_PIXEL(posicao, posicao);  
  await 100ms;  
  posicao ← posicao + 1;  
  if posicao = 50 then  
    break;  
  end  
end
```

Indentação  
(reco no código)

# Dando tempo ao tempo

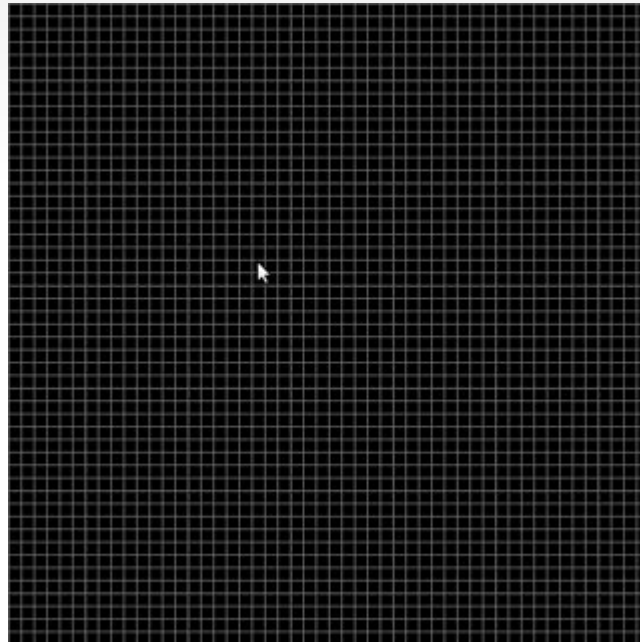
- Sensação maior de interação e imersão
- Possibilidade de entender o programa em etapas e em tempo real

# Exercício 1.1

- Desenhar um **X** na tela:

# Exercício 1.1

- Desenhar um **X** na tela:



# Concorrência e Paralelismo

- Noção rudimentar
- Abordagem antecipada ao tema
- O que significa “ao mesmo tempo”?
  - para o usuário, para o computador
  - para a especificação/requisitos, para a implementação
  - concorrência, paralelismo lógico, paralelismo real
- Exercício 1.1 e 1.4

# Exercício 1.2

# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:



# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:
  - usar uma cor diferente

# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:
  - usar uma cor diferente
  - desenhar outra figura geométrica

# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:
  - usar uma cor diferente
  - desenhar outra figura geométrica
    - linha, retângulo, etc.

# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:
  - usar uma cor diferente
  - desenhar outra figura geométrica
    - linha, retângulo, etc.
  - escrever um texto

# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:
  - usar uma cor diferente
  - desenhar outra figura geométrica
    - linha, retângulo, etc.
  - escrever um texto
  - ...

# Exercício 1.2

- Adicionar uma funcionalidade qualquer, ex.:
  - usar uma cor diferente
  - desenhar outra figura geométrica
    - linha, retângulo, etc.
  - escrever um texto
  - ...
- <https://fsantanna.github.io/pico-ceu/out/manual/v0.30/graphics/>

# Cultura “Hacker/Maker”

- Experimentar adições e modificações
- Incentivar a criatividade
- Exercitar o estudo autodidata
- Evitar exercícios com “a resposta certa”

# “Hello world!”

```
#include <stdio.h>
int main(void)
{
    int count;

    for(count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```

ANAND 10-3

