



UNIVERSIDADE  
FEDERAL DO CEARÁ

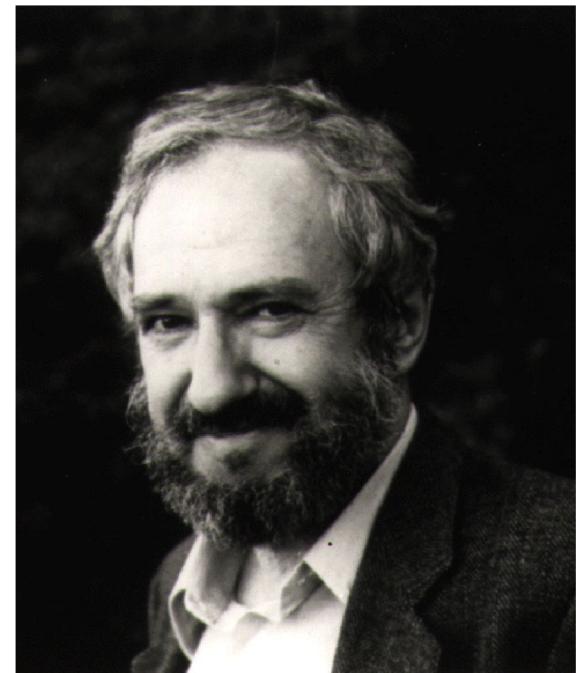


## **CK0176/177 - Informática Aplicada ao Ensino de Ciências: Construcionismo (Papert)**

Prof. Maurício Moreira Neto

# Quem foi Papert?

- Seymour Papert
  - Matemático
  - Considerado um dos pais do campo da Inteligência Artificial
  - Também ficou conhecido como um dos principais pensadores de como utilizar a tecnologia no âmbito da educação



# Piaget e Papert

- Para Piaget, uma das principais chaves do desenvolvimento é a ação do indivíduo sobre o mundo e o modo pelo qual isto se converte em um processo de construção interno
- Neste contexto, o professor é aquele que enriquece o ambiente e provoca situações em que o aprendiz possa desenvolver de forma ativa
  - Gerando suas próprias descobertas e não assimilando conhecimentos “prontos”

# Piaget e Papert

- Logo, o individuo aprende mesmo sem ser “ensinado”, uma vez que está em constante interação com o ambiente
  - Elaborando e reelaborando hipóteses
- O individuo é visto como construtor de sua própria estrutura intelectual (cognitiva)

# Piaget e Papert

- A partir de sua estrutura mental:
  - formulam hipóteses para resolver os problemas e situações propostas
- Quando não conseguem resolver os problemas, ocorre os conflitos cognitivos que as levam a busca de reformulações das hipóteses iniciais
  - Ampliando o sistema de compreensão, buscando o equilíbrio de suas estruturas cognitivas
- O aprendizado é uma resultante da interação do individuo com o objeto do conhecimento, incluindo o ambiente em que esta inserido

# Construcionismo

- É uma reconstrução teórica feita a partir do construtivismo Piagetiano feito por Papert (1994)
- Papert concorda com Piaget de que a criança é um “ser pensante” e construtora de suas próprias estruturas cognitivas, mesmo sem ser ensinada
- Porém, levantou a seguinte interrogação:
  - Como criar condições para que mais conhecimento possa ser adquirido por este indivíduo?

# Construcionismo

- A busca do construcionismo é alcançar meios de aprendizagem fortes que valorizem a construção mental do indivíduo, apoiado suas próprias construções no mundo
- Porém, dizer que estruturas intelectuais são construídas pelo aluno, ao invés de ensinadas por um professor não significa que elas sejam construídas do nada
- Como qualquer construtor, a criança se apropria, para seu próprio uso, de materiais que ela encontra e de modelos e metáforas sugeridos pela cultura que a rodeia (Papert, 1986)

# Construcionismo

- Papert enfatiza que as etapas de interação com o ambiente são determinadas pelos materiais disponíveis para a exploração
  - Intensificado a medida que o conhecimento se torna útil
- Papert também critica a passagem rápida do individuo do pensamento operatório concreto para o abstrato (processo formal)

# Construcionismo

- É importante a conexão entre as entidades mentais existentes para o progresso e criação de novas entidades mentais (conhecimento prévio)!
  - Criança e adulto
- Papel do professor (facilitador):
  - Proporcionar um ambiente capaz de fornecer conexões individuais e coletivos
  - Desenvolver projetos vinculados com a realidade dos alunos (em diferentes áreas do conhecimento)

# Linguagem LOGO

- Papert

- Informática como possibilidade de criar condições para mudanças significativas no desenvolvimento intelectuais do individuo
- Desenvolveu a linguagem de programação **LOGO**
  - Fácil compreensão para crianças e pessoas leigas na área de computação e matemática
  - Também tem poder das linguagens de programação profissionais
- Metologia LOGO
  - Metodologia de ensino-aprendizagem no ambiente computacional

# Linguagem LOGO

- **Logo** – Linguagem de programação simples e estruturada voltada a educação
  - **Objetivo:** permitir que pessoas se familiarizem com conceitos lógicos e matemáticos através da exploração de atividades espaciais que auxiliam o usuário a formalizar seus raciocínios cognitivos
  - Desenvolvida pelo MIT (*Massachusetts Institute of Technology*) em meados dos anos 60 ~ 70
  - Popularizou-se nas escolas para desenvolvimento de projetos de TIC (Tecnologia de Informação e Comunicação) – anos 80

# Linguagem LOGO

- Segundo Papert (1994):

“Minha meta tornou-se lutar para criar um ambiente no qual todas as crianças – seja qual for sua cultura, gênero ou personalidade – poderiam aprender Álgebra, Geometria, Ortografia e História de maneira mais semelhante à aprendizagem informal da criança pequena, pré-escolar, ou da criança excepcional, do que ao processo educacional seguido nas escolas.”

# Linguagem LOGO

- A linguagem de programação LOGO parte da exploração de atividades espaciais
  - Desenvolver inúmeros conceitos numéricos e geométricos

“A criança comanda uma tartaruga que se locomove, de modo direto ou através de programação, criando gráficos e animações. Como exemplo de comandos de locomoção e de rotação utiliza: para frente 10, para trás 10, direita 90, esquerda 90... Nesse processo, os movimentos pelos quais os indivíduos “ensinam” à tartaruga, favorece que estes externalizem suas hipóteses e conceitos.”

**“Tal fato possibilita que se pense e fale sobre eles, podendo fazer e refazer, descobrindo novos caminhos, criar novas soluções, trazendo outra perspectiva para um questão do erro”**

# Linguagem LOGO

- Ambiente em torno do LOGO
  - Pedagogia de projetos
  - Diversas áreas do conhecimento (integradas)
  - Diferentes resoluções de problemas (cooperativa ou facilitada)
- Vantagens:
  - Acompanhamento gradual do raciocínio lógico do aluno
  - Desperta a curiosidade para as diversas possibilidades de uso
  - Integralização do conhecimento
  - Diferentes soluções para um problema
  - ...

# Linguagem LOGO

- Foram criados diversas linguagens com base na linguagem LOGO
  - Scratch – Lisp – SmallTalk – Pascal
- Estas linguagens foram criadas com caráter didático para facilitar o ensino de lógica computacional para os iniciantes em computação

SCRATCH



LOGO | Educational  
Programming  
Language

PASCAL

# Linguagem LOGO

- Atualmente existem diversas ambientes online para a prática destas linguagens de programação
- A seguir, tem-se os links de algumas:
  - LOGO - <https://www.calormen.com/jslogo/>
  - Scratch - <https://scratch.mit.edu/>
  - Pascal:  
[http://www.compileonline.com/compile\\_pascal\\_online.php](http://www.compileonline.com/compile_pascal_online.php)
  - ...

# Vamos praticar um pouco com o LOGO...



# Linguagem LOGO

# Logo Interpreter

[Tests](#) | [Source](#) | [Collaborate](#) |



Type your code here...

Run  Clear

[Reference](#) – the Logo language  
[Library](#) – your procedures  
[History](#) – everything you've done here  
[Examples](#) – fun things to try out  
[Extras](#) – helpful utilities  
[Links](#) – other Logo resources

```
TO star
  repeat 5 [ fd 100 rt 144 ]
END
clearscreen
star

TO square :length
  repeat 4 [ fd :length rt 90 ]
END
clearscreen
repeat 36 [ randomcolor square random 200 rt 10
]

clearscreen window hideturtle
repeat 144 [
  setlabelheight repcount
  penup
  fd repcount * repcount / 30
  label "Logo"
  bk repcount * repcount / 30
  pendown
  rt 10
  wait 5
]
showturtle

TO tree :size
  if :size < 5 [forward :size back :size stop]
  forward :size/3
  left 30 tree :size*2/3 right 30
  forward :size/6
  right 25 tree :size/2 left 25
  forward :size/3
  right 25 tree :size/2 left 25
]
```

# Linguagem LOGO

- Exemplo de códigos:

Foward 100



comando parâmetro

## Procedimentos

```
To square
repeat 5 [forward 60 right 100]
end
square
```

# Linguagem LOGO

- Exemplo de códigos:

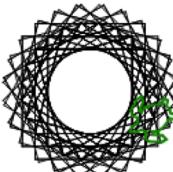
```
To square
repeat 5 [forward 80 right 90]
end
square
```

```
to flower
repeat 40 [right 15 square]
end
```

```
flower
```

# Linguagem LOGO

**Logo Interpreter** [Tests](#) | [Source](#) | [Collaborate](#) |



```

1 TO square
2 repeat 5 [forward 80 right 90]
3 END
4
5 TO flower
6 repeat 40 [right 15 square]
7 END
8
  
```

Run
Clear

[Reference](#) – the Logo language  
[Library](#) – your procedures  
[History](#) – everything you've done here  
[Examples](#) – fun things to try out  
[Extras](#) – helpful utilities  
[Links](#) – other Logo resources

```

TO star
repeat 5 [ fd 100 rt 144 ]
END
clearscreen
star

TO square :length
repeat 4 [ fd :length rt 90 ]
END
TO randomcolor
setcolor pick [ red orange yellow green blue violet ]
END
clearscreen
repeat 36 [ randomcolor square random 200 rt 10
]

clearscreen window hideturtle
repeat 144 [
  setlabelheight repcount
  penup
  fd repcount * repcount / 30
  label "Logo"
  bk repcount * repcount / 30
  pendown
  rt 10
  wait 5
]
showturtle

TO tree :size
if :size < 5 [forward :size back :size stop]
forward :size/3
left 30 tree :size*2/3 right 30
forward :size/6
right 25 tree :size/2 left 25
forward :size/3
  
```

# Dúvidas?

