

UNIDADE I Introdução a Lógica de Programação

O que é lógica?

É organizar passos para resolver um problema.

Trecho do código:

```
``swift
while continuar {
    print("=== Organizador de Estudos ===")
    print("1. Adicionar atividade")
    print("2. Listar atividades")
    print("3. Sair")
    let escolha = readLine()
```

Explicando:

- ``while continuar``: repete o menu enquanto a variável ``continuar`` for ``true``
- ``print(...)``: mostra opções na tela
- ``let escolha = readLine()``: lê a escolha digitada pelo usuário

Esse é o **fluxo principal** do sistema: mostrar menu esperar escolha agir.

UNIDADE II Variáveis e Operadores



O que são variáveis?

São caixinhas onde guardamos dados.

Trecho do código:

```
var atividades: [[String]] = []
```

Explicando:

- ``var``: cria uma variável que pode mudar
- ``atividades``: é o nome da variável
- ``[[String]]``: diz que vamos guardar uma **lista de listas** com textos
- ``= []``: começa vazia

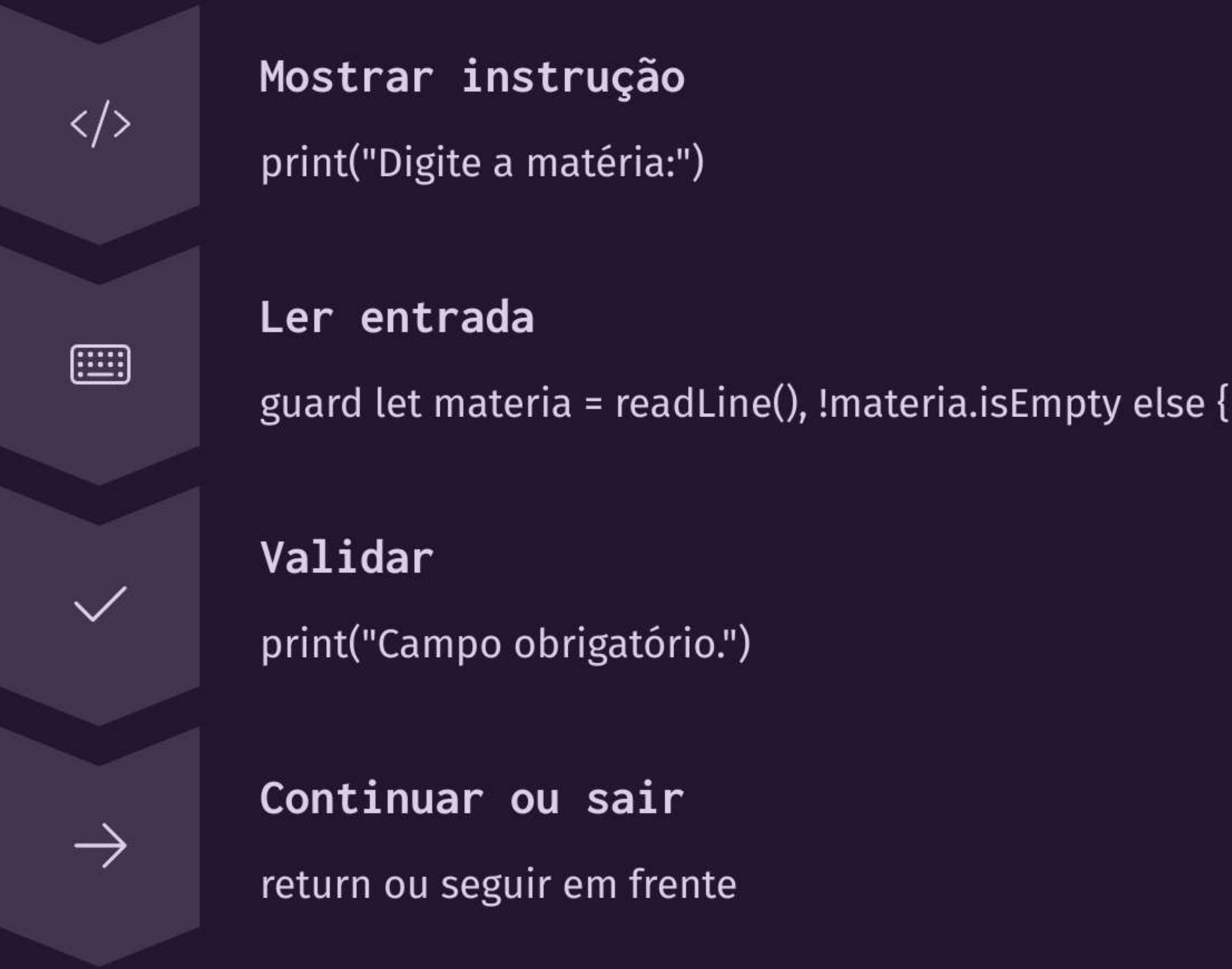
Outro trecho:

```
atividades.append([materia, descricao, prazo])
```

Explicando:

- ``append(...)``: adiciona uma nova atividade à lista
- Cada atividade é uma lista com 3 itens (matéria, descrição, prazo)

UNIDADE III Sequência de instruções



A execução é linha por linha.

Trecho:

```
swift
print("Digite a materia:")
guard let materia = readLine(), !materia.isEmpty else {
    print("Campo obrigatorio.")
}
```

```
return





}
```

Explicando:

- ``print(...)``: mostra um texto na tela
- ``readLine()``: lê o que o usuário digitar
- ``guard let ...``: verifica se a entrada é válida
- ``!materia.isEmpty``: verifica se não está vazio
- ``return``: sai da função se estiver errado

Isso acontece em sequência: perguntar validar seguir em frente

UNIDADE IV Decisão

	Switch Testa várias opções
	Case "1" Adicionar atividade
	Case "2" Listar atividades
	Case "3" Encerrar programa

Tomar decisões com `if`, `guard` ou `switch`.

Trecho:

```
swift

switch escolha {

case "1":

    adicionarAtividade()

case "2":
```

listarAtividades()

```
case "3":

    continuar = false

default:
```

print("Opcao invalida.")

}

Explicando:

- `switch`: testa várias opções
- `case "1"`: se for "1", chama a função de adicionar
- `case "2"`: se for "2", chama listar
- `case "3"`: altera `continuar` para `false` encerra menu
- `default`: se não for nenhuma das anteriores, mostra erro

UNIDADE V Repetição



Repetir tarefas com ``while`` ou ``for``.

Trecho:

```
for (index, atividade) in atividades.enumerated() {  
  
    print("(index + 1). (atividade[0]) - (atividade[1]) (Entrega: (atividade[2]))")  
  
}
```

Explicando:

- ``for (index, atividade) in ...``: percorre todas as atividades com índice
- ``atividade[0]``: mostra a matéria
- ``atividade[1]``: mostra a descrição
- ``atividade[2]``: mostra a data
- ``index + 1``: numeramos a lista de 1 em diante

UNIDADE VI Dados compostos

Índice	Matéria	Descrição	Prazo
0	Matemática	Estudar prova	30/05
1	História	Trabalho	01/06

Guardar listas de dados relacionados.

Trecho:

```
var novaAtividade = [materia, descricao, prazo]

atividades.append(novaAtividade)
```

Explicando:

- `novaAtividade`: é uma lista com os dados da nova tarefa
- `append`: adiciona essa nova linha na matriz `atividades`

A estrutura da matriz fica assim:

[

```
["Matematica", "Estudar prova", "30/05"],
["Historia", "Trabalho", "01/06"]
```

]