

Programar

É divertido!

Vitor Vaz da Silva

Qual é a coisa qual é ela ?!

- que se traduz para todas as línguas
- quem a dá não a perde
- quem a recebe fica mais sábio

I DEI A

Mouse de Chocolate

- 4 colheres de sopa de açúcar
- 100g de chocolate em barra
- 70g de manteiga
- 4 ovos
- Bater as gemas com o açúcar
- Derreter a manteiga e juntar o chocolate partido aos bocadinhos.
- Depois de tudo bem derretido, juntar as gemas.
- Bater as claras em neve e envolver no creme.
- Para ficar mais fofo pode juntar-se mais uma ou duas claras.
- Coloca-se no frigorífico

- **Título** Mouse de Chocolate

- **Ingredientes**

Açúcar – 4 colheres de sopa

Chocolate – 100g em barra

Manteiga – 70g

Ovos – 4

- **Procedimento**

Bater gemas e Açúcar

Derreter Manteiga

Adicionar Chocolate aos bocadinhos à Manteiga

Derreter tudo muito bem

Juntar a Manteiga e o Chocolate derretidos às Gemas

Bater claras em neve

Envolver as claras no creme

(Se quiser mais fofo então junte uma ou duas claras)

Colocar no frigorífico

- **Título:** Mouse de Chocolate
- **Ingredientes:**
 - Açúcar – 4 colheres de sopa
 - Chocolate – 100g em barra
 - Manteiga – 70g
 - Ovos – 4 (e mais uma ou duas claras)
- **Material:**
 - Batedeira
 - Fogão
 - Frigorífico
 - Bacia
 - Tacho
 - Terrina

- **Procedimento**

Colocar na Bacia as Gemas e o Açúcar

Bater o conteúdo da Bacia

Colocar Manteiga no Tacho

Colocar Tacho no Fogão

Aquecer o Tacho

Adicionar Chocolate ao conteúdo do Tacho

Derreter muito bem o conteúdo do Tacho

Juntar o conteúdo do Tacho ao da Bacia

Colocar claras na Terrina

Se quiser mais fofo então:

Colocar uma ou duas claras na Terrina

Bater o conteúdo da Terrina

Juntar o conteúdo da Terrina ao da Bacia

Colocar no frigorífico

Constantes

- #define acucar 4
- #define chocolate 100
- #define manteiga 70
- #define ovos 4
- #define claras_fofo 1



Variáveis



recipiente tacho, bacia, forma, terrina;
talheres colher_sopa, colher_pau;



Procedimentos

Batedeira();

Fogao();

Frigorifico();

PartirOvos();

Envolver();



Programa principal

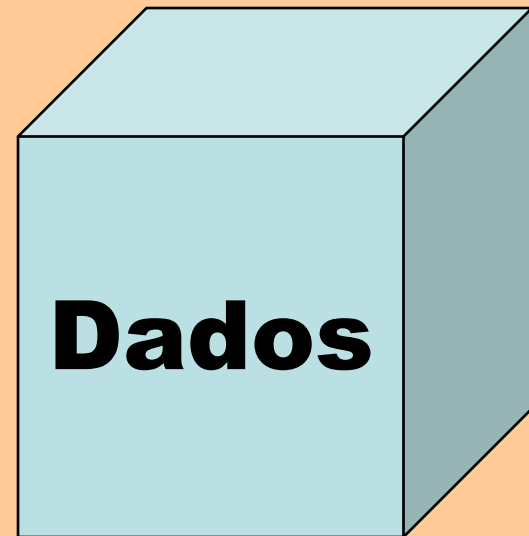
```
main()
{
  PartirOvos(ovos, &bacia, &terrina);
  bacia += acucar;
  if(mais_fofo) terrina += claras_fofo;
  tacho = manteiga + chocolate;
  Fogao(aquecer, &tacho);
  Batedeira(bater, &bacia);
  bacia += tacho;
  bacia = Envolver(bacia, terrina);
  Frigorifico(&bacia);
}
```



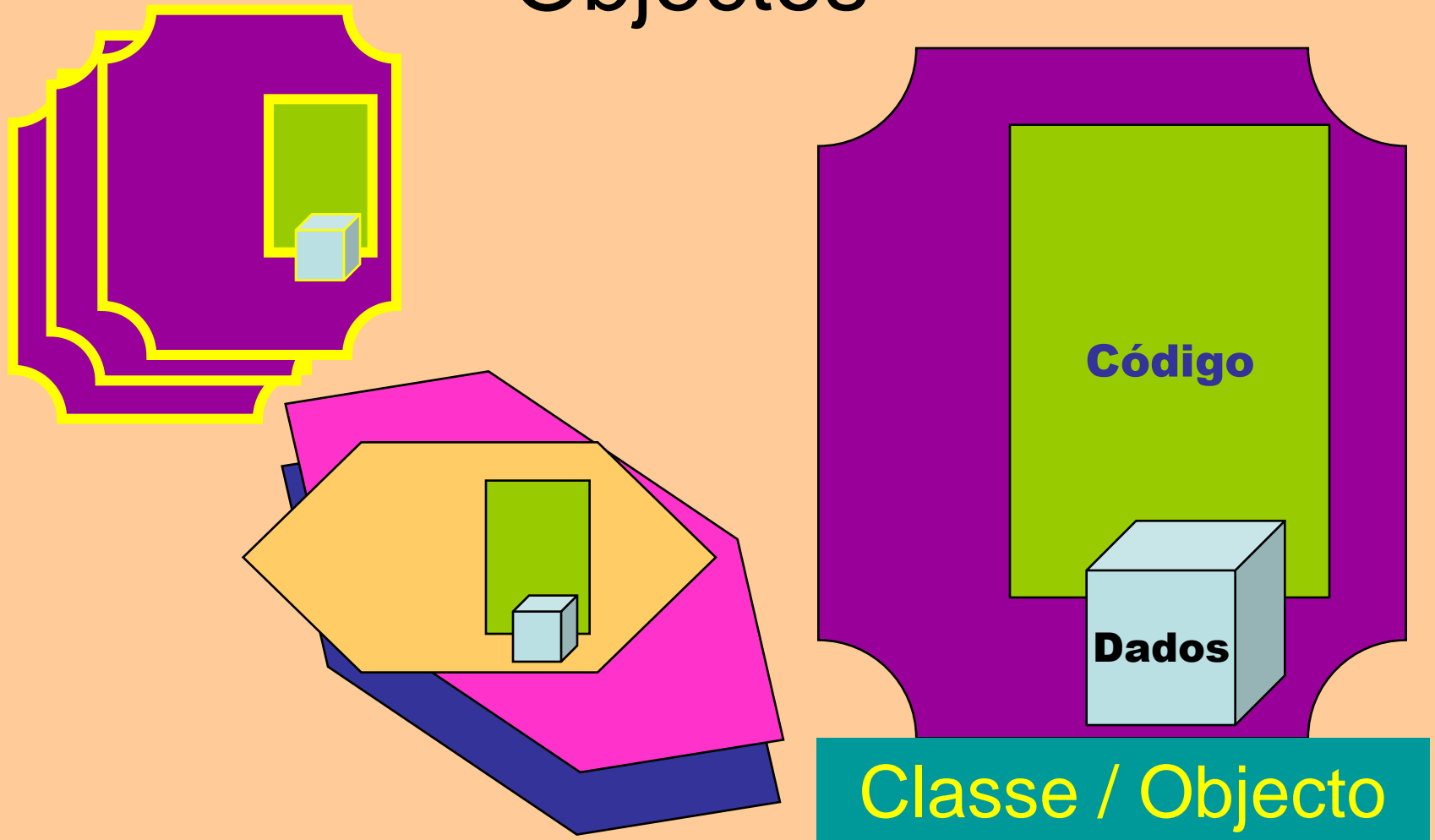
Tipos de Programação

- Clássica
 - Linear
 - Faz-se tudo
- Orientada por Objectos
 - Qualquer coisa é um objecto
 - Utiliza-se o que outros já fizeram

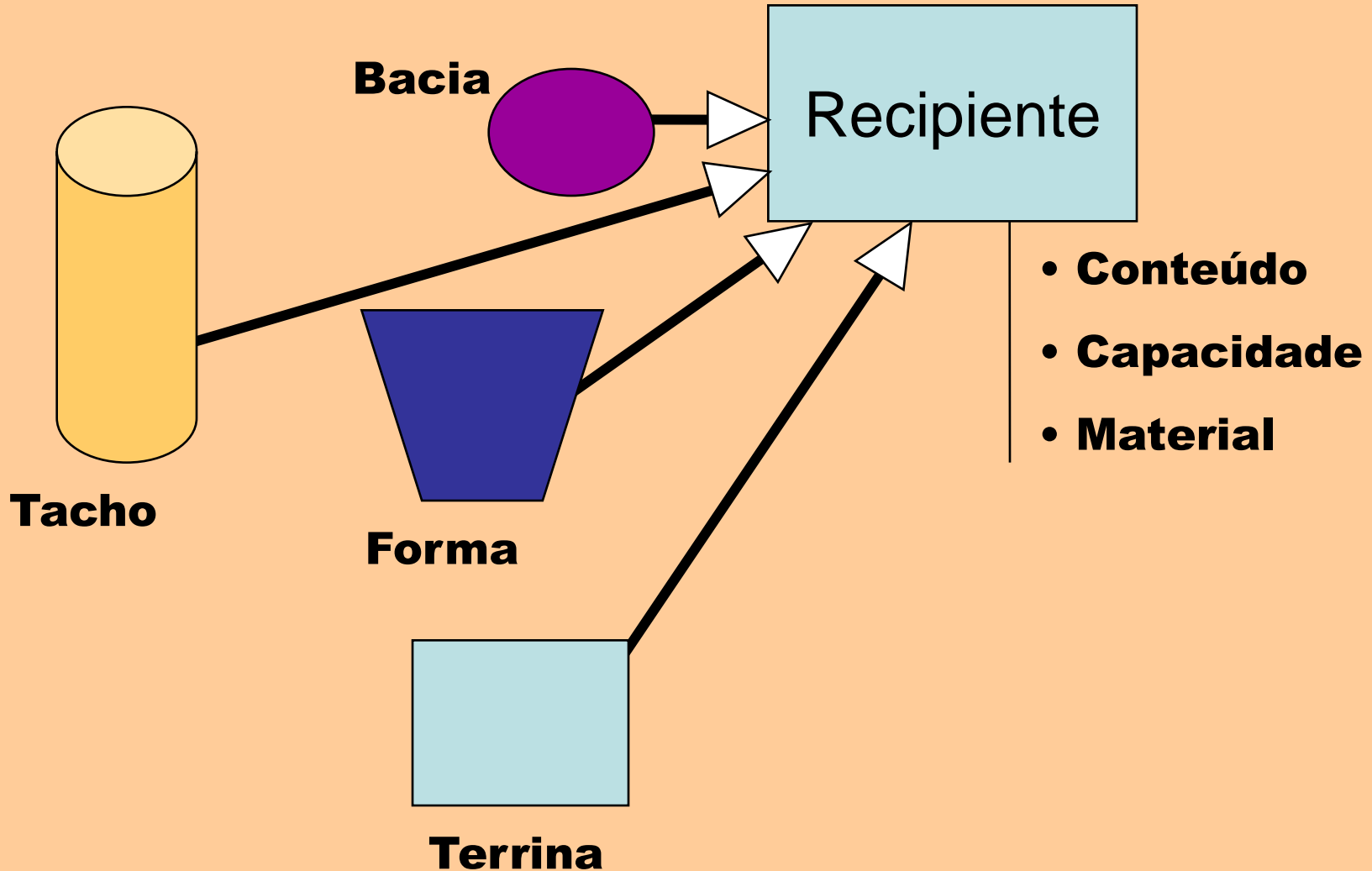
Programação Clássica



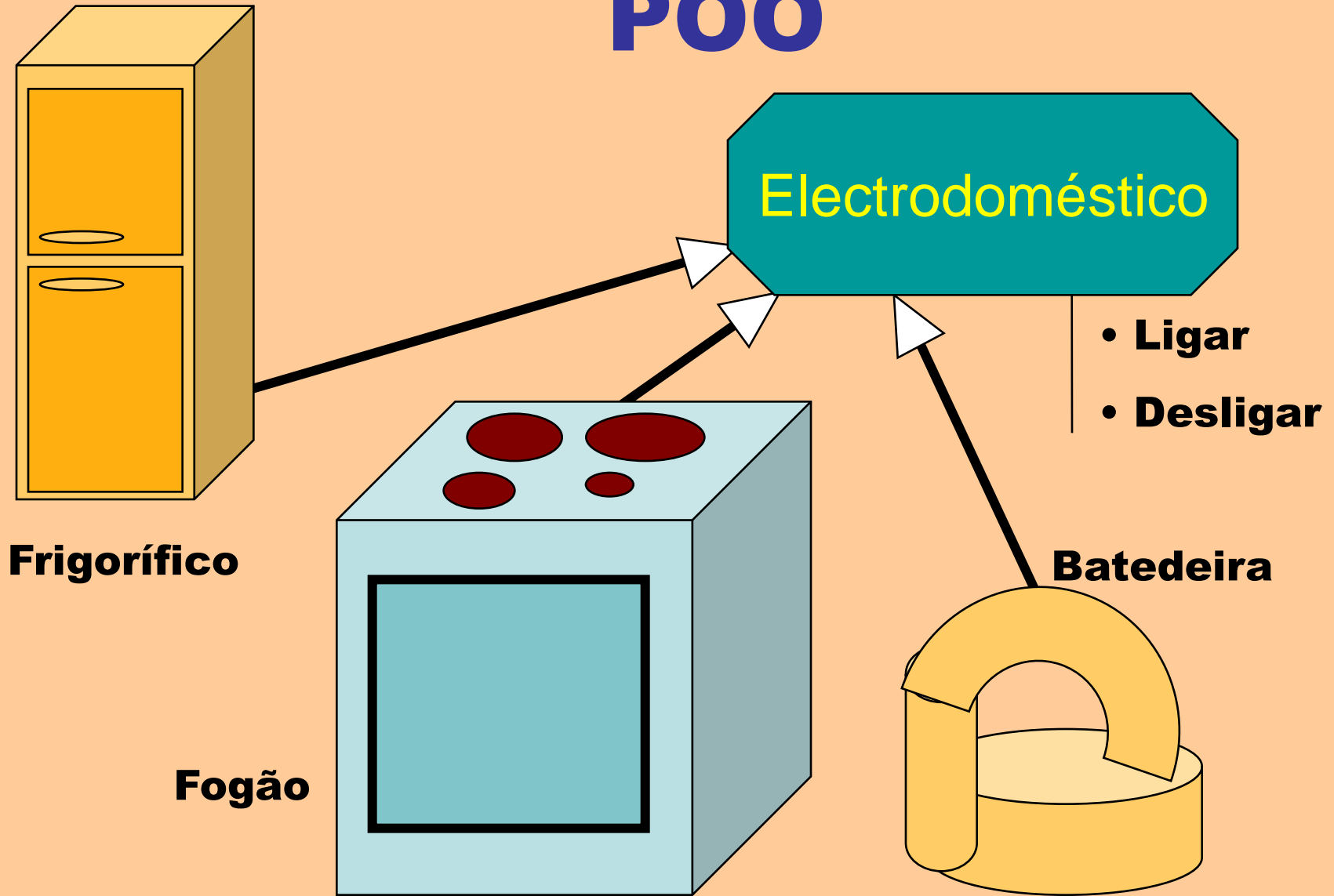
Programação Orientada por Objectos



POO

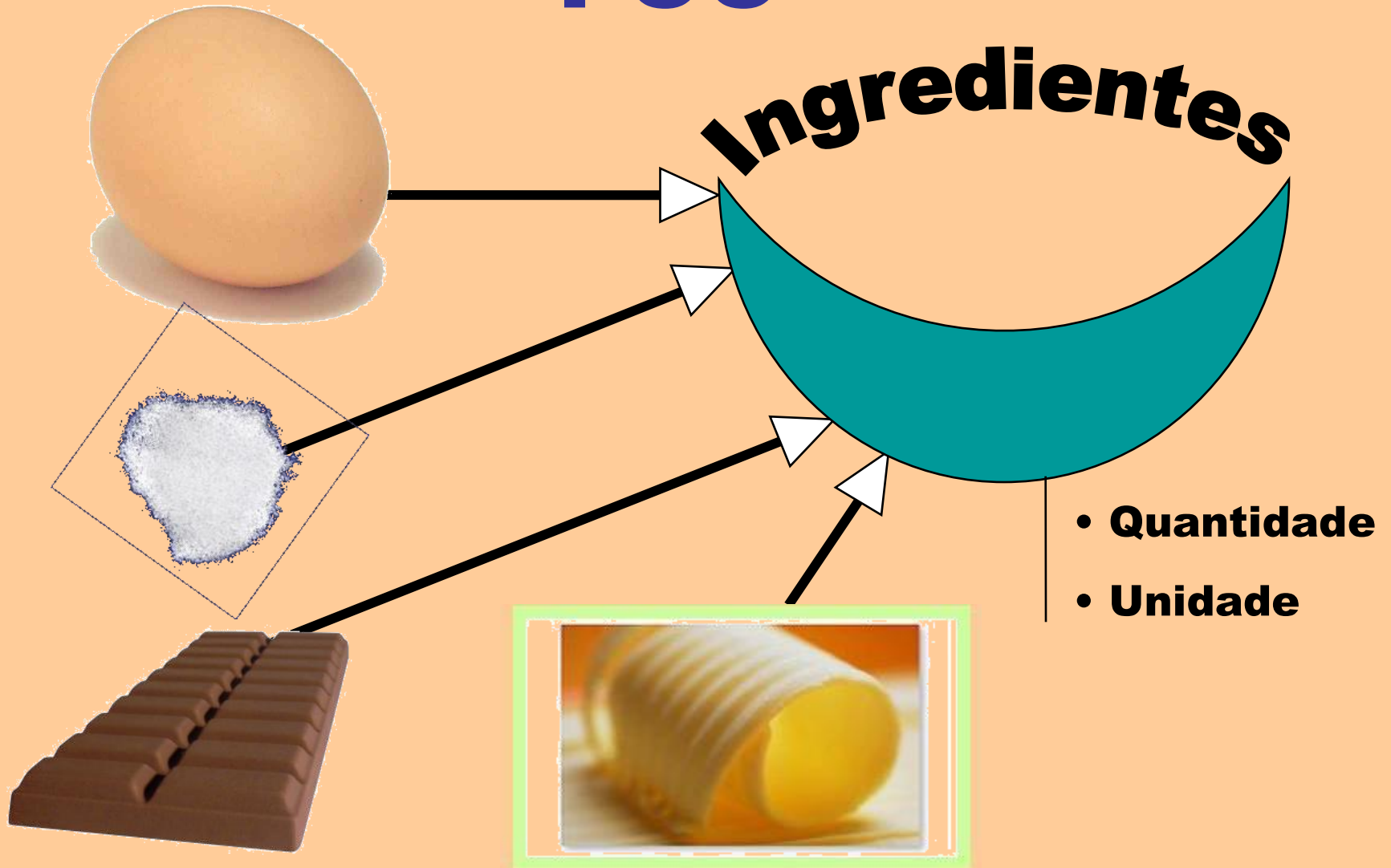


POO



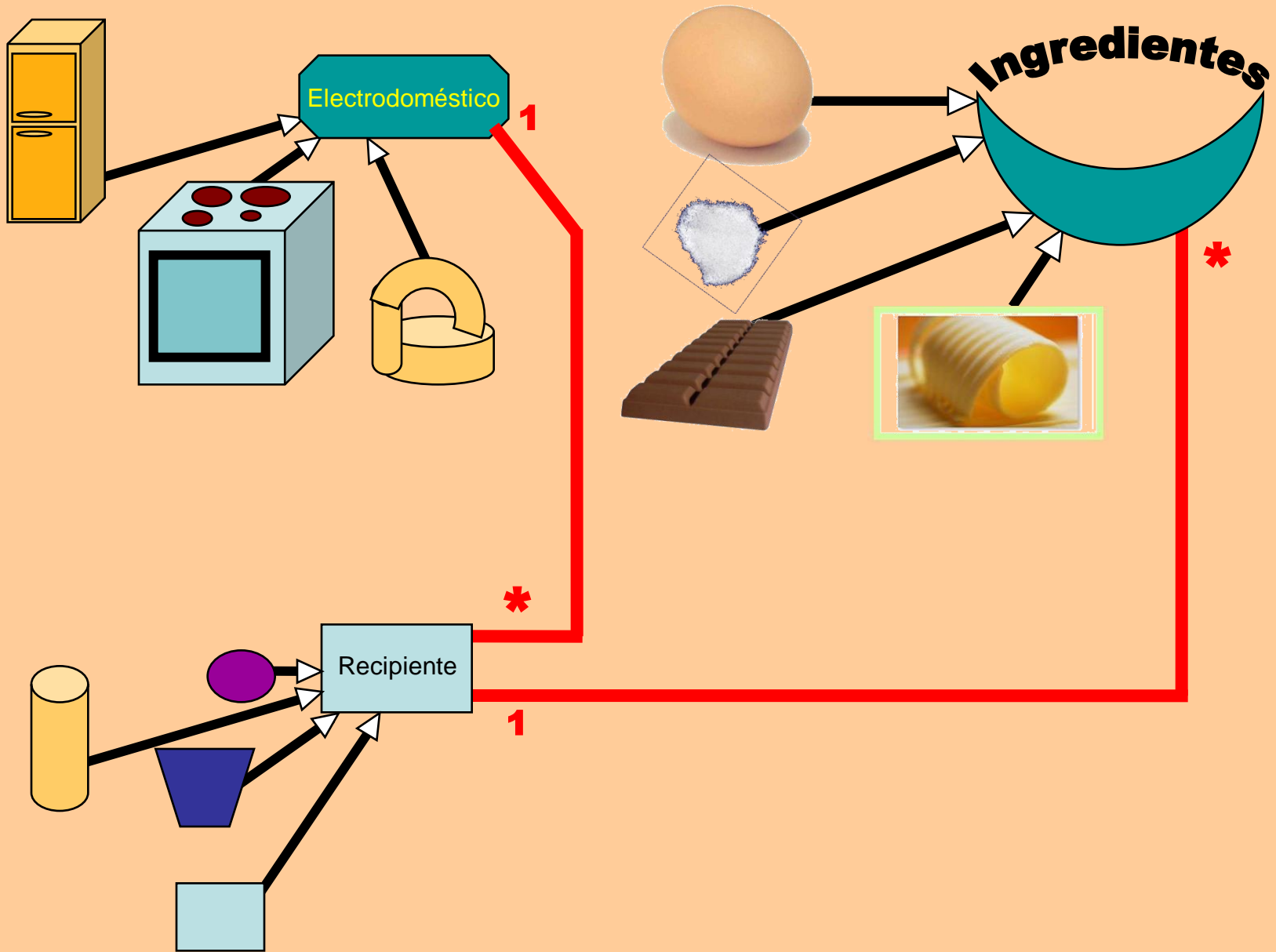
POO

Ingredientes

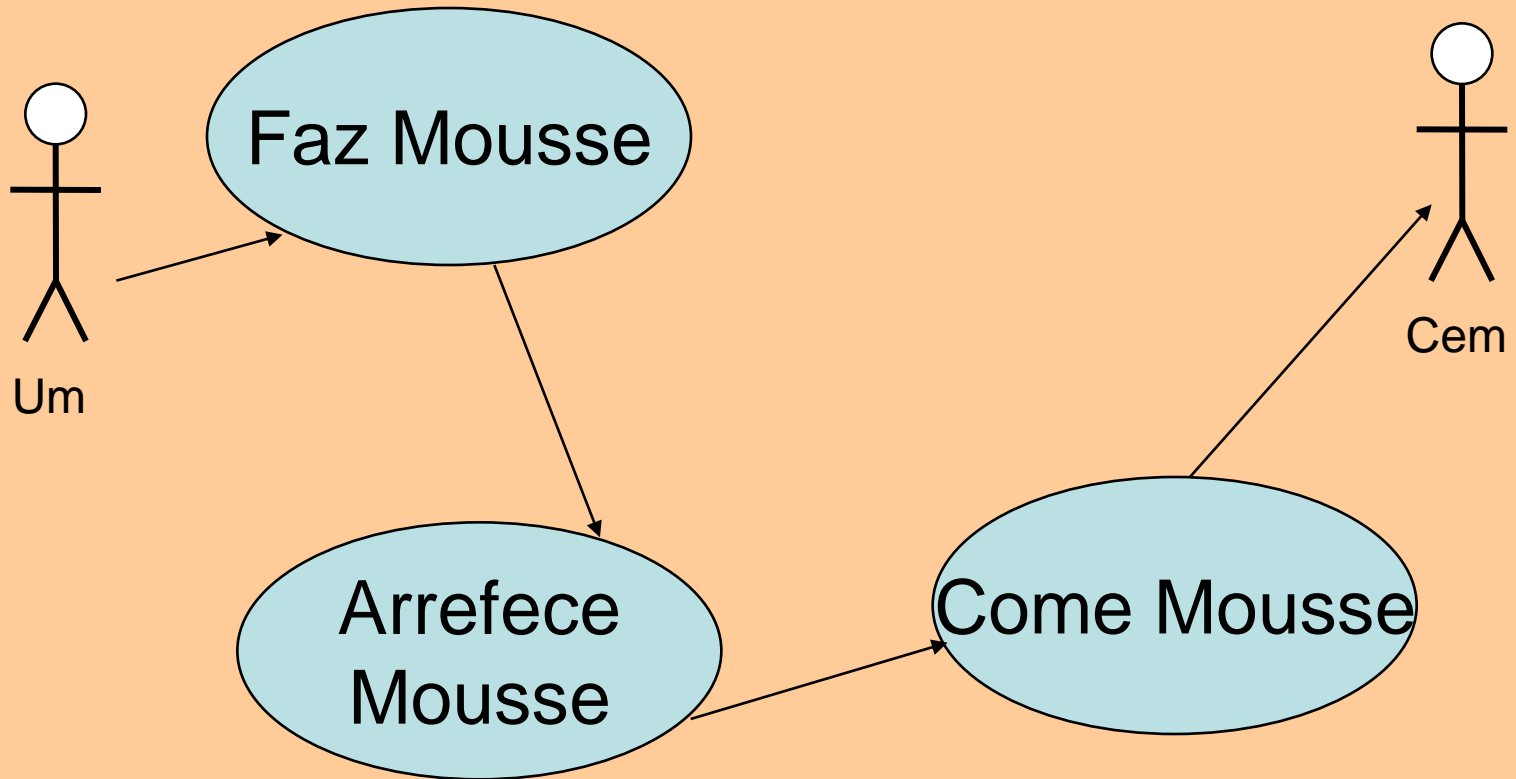


POO

```
bacia.conteudo=ovos.gemas+acucar.quantidade;  
batedeira.taca.conteudo=bacia.conteudo;  
batedeira.velocidade=2;  
batedeira.ligar();  
tacho.conteudo=manteiga.quantidade +  
                chocolate.quantidade;  
fogao.boca.pequena.recipiente=tacho;  
fogao.boca.pequena.ligar();  
fogao.boca.pequena.temperatura=3;
```



Diagrama



Linguagem

- Vocabulário
- Gramática
- A gramática da língua portuguesa está escrita em

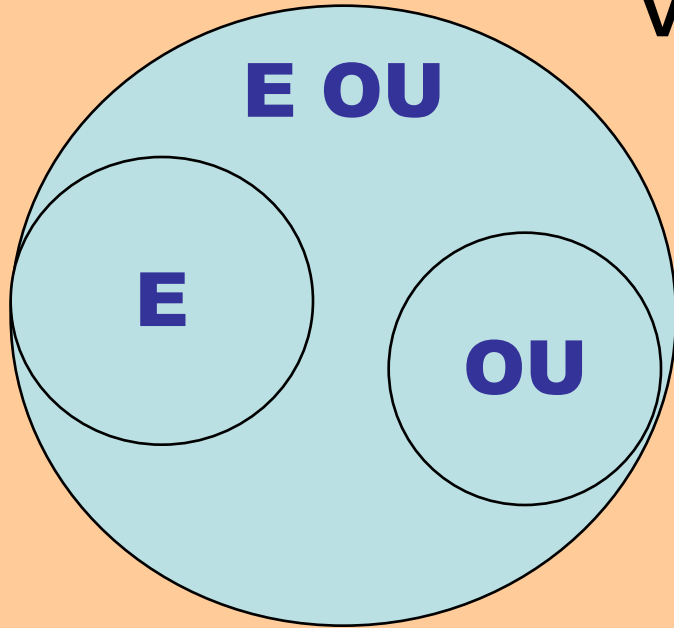
Linguagem

A gramática da língua portuguesa está escrita em português.

O compilador da linguagem C está escrito em C.

O que apareceu primeiro,
a língua ou a gramática?

Vocabulário



NÃO

SIM

SE

SENÃO

Maior

Menor

Igual

ENQUANTO

Início

Fim

Diferente

Linguagens

Lingua- gem	SE	SENÃO	ENQUANTO	INÍCIO	FIM	SIM	NÃO
PASCAL	If	else	while	begin	End	TRUE	FALSE
C	if	else	while	{	}	Qualquer coisa excepto 0	0

Linguagens

Lingua- gem	MENOR	IGUAL	MAIOR	DIFERENTE	E	OU	E OU	NEGAÇÃO
PASCAL	<	=	>	<>	and	xor	or	not
C	<	==	>	!=	&& &	^	 	~

IF

Se Então Senão

if (.....) else

Se o dia estiver quente Então vou à praia Senão vou ao cinema

IF

Se Então
if (.....)

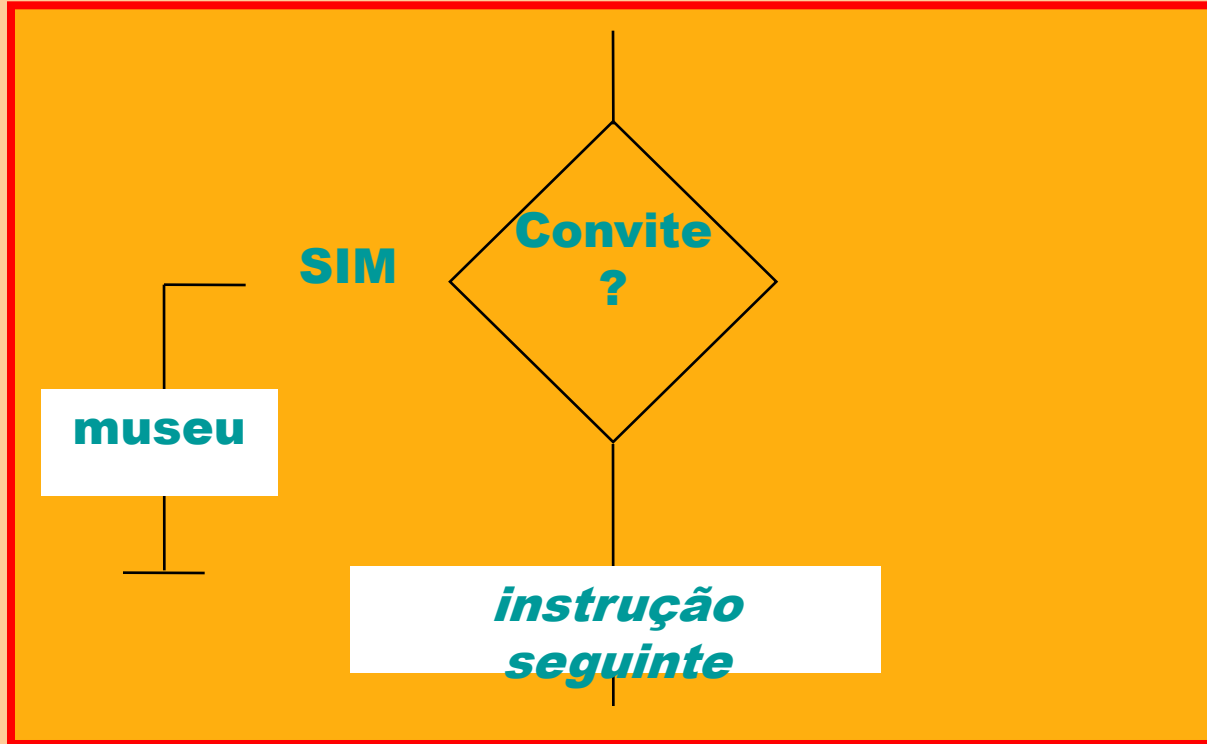
Se tiver convite Então vou ao museu

IF



O IF serve para conduzir o fluxo do programa. Escolhe o caminho do sim ou do não, e depois continua para executar a *instrução seguinte*.

IF



O IF serve para conduzir o fluxo do programa. Quer escolha ou não o caminho do sim depois continua para executar a *instrução seguinte*.

IF

Estrutura do IF:

```
if ( <condição> )  
    <instruções devido à condição ser verdadeira>  
else  
    <instruções devido à condição ser falsa>
```

- O if só aceita uma instrução verdadeira e uma falsa (a falsa é opcional)

```
if (autocarro==140) printf("Apanha o autocarro");  
else printf("Espera!");
```

IF

Para colocar mais do que uma instrução é preciso abrir um bloco { }

```
{
```

```
    <aqui podemos colocar 1 ou mais instruções>
```

```
}
```

```
if (autocarro==140) {  
    printf("Apanha o autocarro");  
    printf ("Hehehe, desta vez vou mais cedo 😊");  
} else {  
    printf ("Má onda!");  
    printf ("Espera!");  
}
```

While

Enquanto

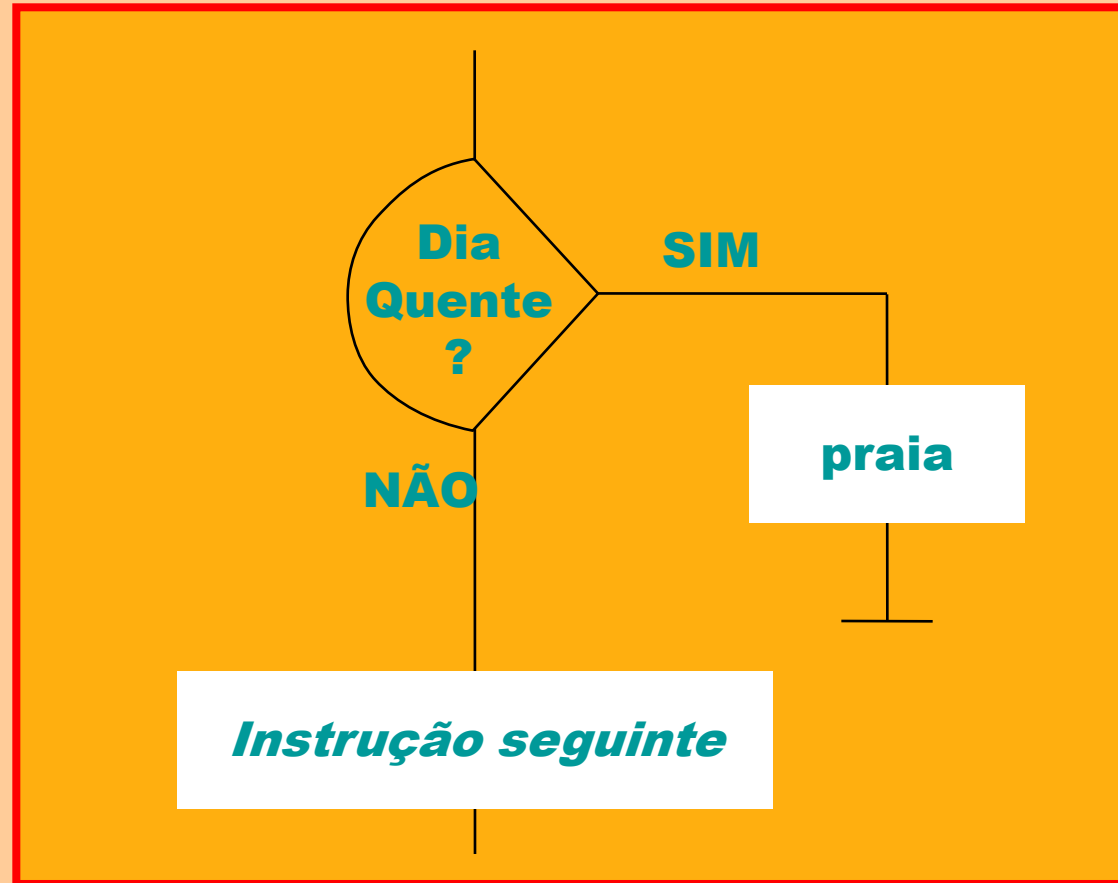
Faz ...

while (.....)

**Enquanto o dia estiver quente fico
na praia**

While

O WHILE serve para repetir várias vezes as mesmas acções até se verificar a condição de fim. Só depois é que continua para executar a *instrução seguinte*.



WHILE

Estrutura do WHILE

```
while ( <condição> )
```

```
<instruções devido à condição ser verdadeira>
```

O while só aceita a repetição de uma instrução.

```
while ( balde < cheio )
```

```
    printf("Continua a encher o balde");
```

WHILE

Para colocar mais do que uma instrução é preciso abrir um bloco { }

```
{  
    <aqui podemos colocar 1 ou mais  
    instruções>  
}
```

```
while ( balde < cheio ) {  
    printf("Continua a encher o balde");  
    balde = balde+1;  
}
```

EBNF

Extended Backus-Naur Form

Linguagem para descrever linguagens

Uma frase é um conjunto de
palavras separadas por sinais de pontuação,
uma palavra um conjunto de caracteres,
os caracteres são símbolos
os sinais de pontuação são símbolos

<frase> ::= {<palavra> <pontuação>}⁺

<palavra> ::= {<caracter>}⁺

<caracter> ::= {a b c d e f ... z A B C ... Z 0 1 2 ... 7 8 9}

<pontuação> ::= { . , ; + - ' ? ! " ... }

EBNF

::=	definido como
<a>	declaração de a (símbolo não terminal)
{a}	declaração de a (símbolo terminal)
{ }	usado uma vez
{ }⁺	usado uma ou mais vezes
[]	opcional
{ }[*]	usado zero ou mais vezes
a b	ou a ou b (exclusive)
a b	a e b