

Condicionais, expressões

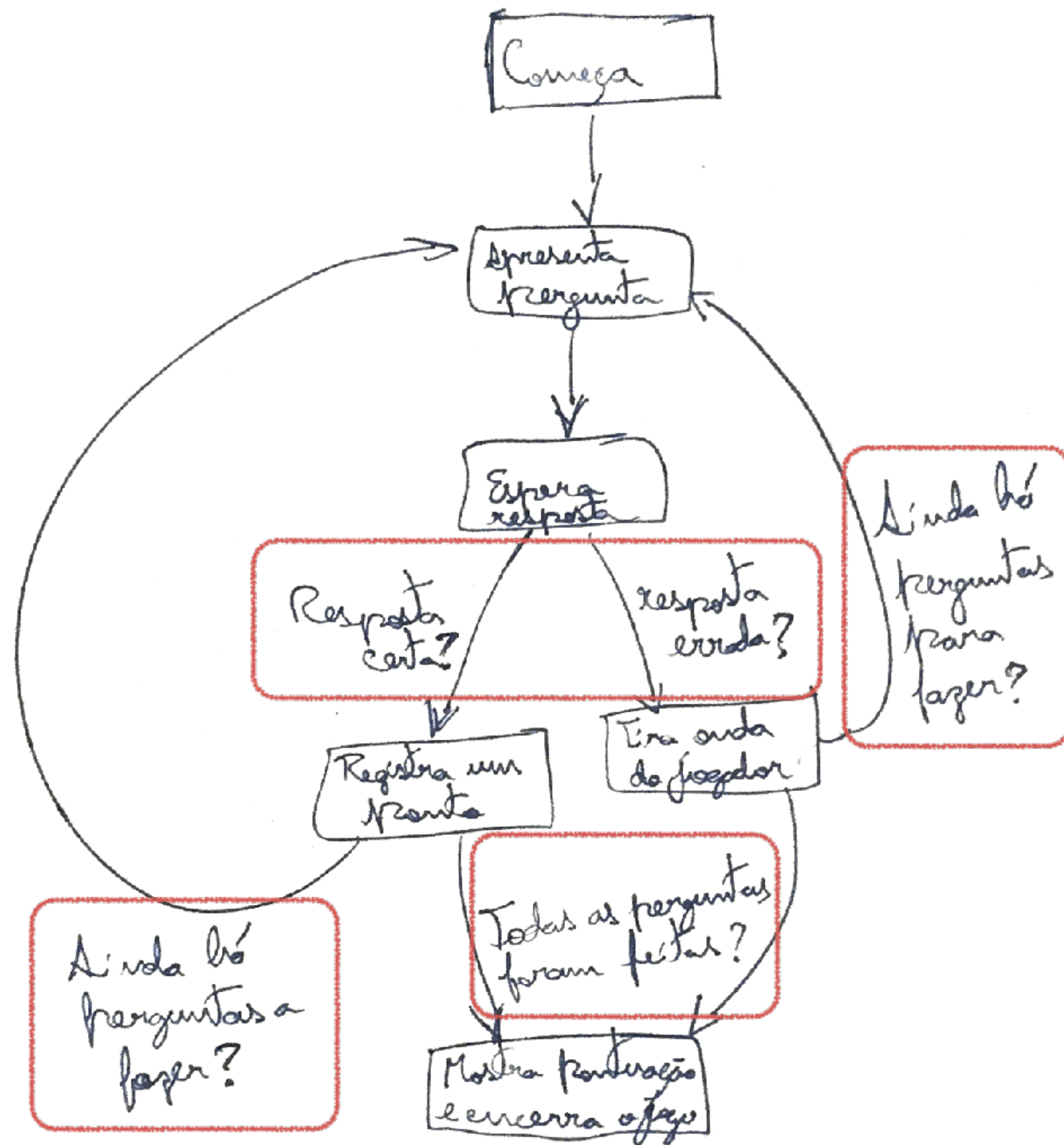
```
// Várias perguntas
let p1 = "Qual a capital da Croácia?"
let respostaP1 = "Zagreb"

let p2 = "Em que ano Swift 1.0 tornou-se disponível?"
let respostaP2 = "2014"

let p3 = "Qual o nome do segundo álbum do Vampire
Weekend?"
let respostaP3 = "Contra"

// Mostrando as perguntas e esperando as respostas:
print(p1)
var palpite = readLine()
print(p2)
palpite = readLine()
print(p3)
palpite = readLine()
```

O que falta ainda?



O comando `if`

Verifica se uma condição é verdadeira

Próximo passo depende dessa verificação

```
if respostaP1 == palpite {  
    ... // o que fazer quando o jogador acerta?  
}
```

Forma geral do comando if

```
if condição {  
    comandos1  
} else {  
    comandos2  
}  
comandos3
```

Forma geral do comando if

```
if condição {  
  comandos1  
} else {  
  comandos2  
}  
comandos3
```


Antes de continuar, defina uma variável para registrar os pontos do jogador.

Em que ela difere das variáveis que criamos antes?

```
// variável que guarda um número inteiro  
var pontos = 0
```

```
// variável que guarda um número de ponto flutuante  
// números de ponto flutuante têm uma parte decimal  
var pontos = 0.0
```

Swift é estaticamente tipificada

```
var pontos = 0.0
```

```
// Erro de compilação!!!!
```

```
pontos = "Isso não funciona!"
```

```
var pontos:Float = 0.0
```

```
// Erro de compilação!!!!
```

```
pontos = "Isso não funciona!"
```

O que fazer quando um jogador acerta?

```
if respostaP1 == palpite {  
    ... // o que fazer quando está certo?  
}
```

O que fazer quando um jogador acerta?

```
if respostaP1 == palpite {  
    pontos = ...  
}
```


O que fazer quando um jogador acerta?

```
if respostaP1 == palpite {  
    pontos = pontos + 1  
}
```

E quando o jogador erra?

```
if respostaP1 == palpite {  
    pontos = pontos + 1  
} else {  
    ... // xingue o jogador!!!  
}
```

Vários tipos de expressões e operadores

Aritméticos: produzem números (inteiros, de ponto flutuante, etc.) como resultado:

+ - * / % pow

De **comparação e lógicos**: produzem valores do tipo verdadeiro ou falso como resultado:

== != < <= > >=

&& || !

Adicione duas questões ao jogo.

Faça seu programa checar as respostas para as perguntas e guardar um ponto para cada resposta certa.

Ao final, mostre o número total de pontos.

Modifique seu programa para que ele dê parabéns ao jogador caso sua pontuação seja maior ou igual a 5

Agora faça seu programa **subtrair**
um ponto para cada resposta
errada.

Apresente ao jogador uma pergunta de múltipla escolha (com quatro opções). O jogo deve receber apenas a letra correspondente à opção correta (sem diferenciar maiúsculas e minúsculas). Caso o jogador acerte, deve registrar o ponto normalmente.

Construa um programa que lê um string contendo apenas um caractere e informa se ele é alfabético, numérico ou nenhum dos dois. Se o string tiver mais que um caractere, seu programa deve imprimir uma mensagem de erro. O tamanho de um String pode ser obtido assim:

```
let a = "abc"  
a.count // obtem o tamanho: 3
```


Construa um programa que lê três números a partir do teclado e ao final diz quantos desses são pares.

Exercício 4.10 – Escreva um programa que calcule o preço a pagar pelo fornecimento de energia elétrica. Pergunte a quantidade de kWh consumidas e o tipo de instalação: R para residências, I para indústrias e C para comércios. Calcule o preço a pagar de acordo com a tabela a seguir.

Preço por tipo e faixa de consumo		
Tipo	Faixa (kWh)	Preço
Residencial	Até 500	R\$ 0,40
	Acima de 500	R\$ 0,65
Comercial	Até 1000	R\$ 0,55
	Acima de 1000	R\$ 0,60
Industrial	Até 5000	R\$ 0,55
	Acima de 5000	R\$ 0,60

Problema extraído do Capítulo 4 de:

Nilo Ney Menezes. Introdução à Programação com Python. Editora NOVATEC, 2010.

Construa um programa que lê os nomes de três pokémon e suas respectivas forças de ataque (um número entre 10 e 200) e tipos (terra, fogo, air, água). O programa deve imprimir o nome, o tipo e a força de ataque do mais forte dos três. Se o tipo do pokémon for inválido ou a sua força estiver fora da faixa esperada, seu programa deve informar o usuário que tal pokémon não é válido.



Construa um programa que lê quatro números entre 100 e 999 a partir do teclado e imprime o maior número que seja ímpar, divisível por três e que não tenha raiz quadrada inteira. Se nenhum número com essas características foi fornecido, seu programa deve imprimir uma mensagem informando o usuário sobre isso. Além disso, se algum dos números tem mais que três dígitos, seu programa deve **parar imediatamente**, imprimindo uma mensagem de erro. Para saber se um número tem raiz quadrada inteira, compare sua raiz quadrada com o valor da própria arredondada:

```
round(sqrt(100)) == sqrt(100) // true
```