



Português estruturado

"Português estruturado" ou "Portugol" é algo que chamamos de pseudocódigo, uma linguagem voltada a didática de algoritmos, ou seja, uma excelente opção para iniciar os estudos da base da programação.

Usar essa linguagem faz com que você entenda a ideia por trás de programar e poder usar essa ideia em qualquer outra linguagem, você vai aos poucos entendendo o que é de fato programar: realizar tarefas e resolver problemas não importando qual a linguagem, mas sim a forma que você pensa. Portanto foque em como as coisas funcionam pois assim você irá conseguir reproduzir o que foi aprendido em outras linguagens que for estudar.

Português estruturado tem uma fácil compreensão para nós justamente por ser em nosso idioma nativo, todos os comandos são em português, portanto você só foca em entender como tudo funciona, mas antes de começar a usar o português estruturado precisamos instalar um editor que o compreenda, o vscode vamos usar em breve para HTML, CSS e JavaScript, nesse momento para o Português Estruturado vamos utilizar o Portugol Studio.

Portugol Studio

O Portugol Studio é um ambiente para aprender a programar, voltado para os iniciantes em programação que falam o idioma português. Possui uma sintaxe fácil, diversos exemplos e materiais de apoio à aprendizagem. Também possibilita a criação de jogos e outras aplicações.

<http://ite.acad.univali.br/portugol/>



A instalação é bastante simples, após a instalação você pode abrir o Portugol Studio e com isso será exibida a tela de início onde você pode criar um novo arquivo, abrir um arquivo, pedir ajuda ou instalar plug-ins e algumas outras coisas, o que nos interessa agora é criar um novo arquivo.



E logo ao criar um novo arquivo será gerado um pequeno trecho de código que vamos entender

```
programa
{
```

```
funcao inicio()
{
    escreva("Olá Mundo")
}
}
```

Muito interessante, é bom entender já como nós fazemos para dar o "start" no que foi escrito, para isso é bem simples, basta clicar no sinal de **"play" verde** no topo do lado esquerdo da tela ou usar o atalho **Shift + F6**:



Ao clicar nesse botão o programinha será executado na parte de baixo da tela, onde são exibidas as saídas do nosso programa, onde ele é executado, chamamos essa parte de "console"



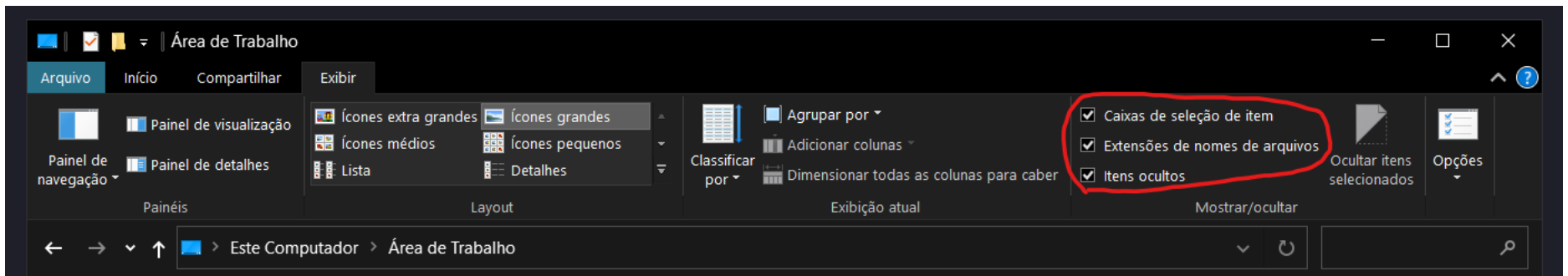
Experimente alterar o texto e dar o "start" novamente, a alteração será refletida no console, tente algo como abaixo:

```
programa
{

    funcao inicio()
```

```
{
  escreva("Olá Beginners PRO")
}
```

Muito interessante, porém e se fecharmos o Portugol Studio? Será exibida uma mensagem como **"O documento Sem título1" possui modificações, deseja salvá-las?"**, clique em "Sim" e prossiga em salvar o arquivo em um local do qual você se lembre, seria bom criar uma pasta só para os arquivos que você irá criar no Portugol Studio, dê nomes legais aos seus arquivos, e note que eles são salvos com a extensão .por, assim como uma imagem é salva com .png ou .jpg. Para ver as extensões dos arquivos abra o explorador de arquivos e na aba exibir selecione "Extensões de nomes de arquivos"



Agora vamos aprender a sintaxe do Português Estruturado para podermos praticar alguns exercícios afim de que você compreenda cada vez mais como as coisas funcionam na programação.

Estrutura básica

Observe a estrutura básica de um programa criado no Portugol Studio

```
programa
{
    funcao inicio()
    {
        escreva("Olá Mundo")
    }
}
```

Por volta do código todo existe o seguinte

```
programa
{
}
```

Ou seja, o seu programa em si, isso faz parte da sintaxe do Portugol, e logo temos uma função chamada inicio, essa função sempre é executada quando iniciamos o programinha, ou seja, quando clicamos naquele ícone de "start" verde.

```
programa
{
    funcao inicio()
    {
        // tudo que estiver aqui é executado no inicio da execução
    }
}
```

Caso queira fazer comentários em seu código basta usar **//** para isso, os comentários não são executados, são apenas comentários mesmo, inclusive faça comentários em seu código para lembrar o que você está estudando, como anotações.

Criação de variáveis

Espera, você sabe bem o que são variáveis?

Variáveis são umas das coisas mais importantes em um código, seja de qual linguagem for, inclusive aqui com Portugol, variáveis são espaços na memória do computador que recebem um nome → identificador e um determinado valor, elas existem em tempo de execução, ou seja, enquanto o programa está "rodando", ficará mais claro com exemplos reais, para criar variáveis no Portugol

precisamos também informar qual o tipo dessa variável, se é um número, uma palavra...vamos ver 3 principais tipos que temos disponíveis no Portugol:

cadeia → cadeia de caracteres, ou seja, palavras em geral

real → números que podem ter casas decimais positivos ou negativos

inteiro → números inteiros positivos ou negativos

Com os 3 tipos acima podemos fazer muita coisa interessante, então vamos logo iniciar com alguns exemplos e ir aprendendo com o passar do conteúdo.

A criação de variáveis ocorre logo no início do código, antes da função início, observe um exemplo

```
programa
{
    cadeia nomeAluno
    real nota1, nota2, nota3, media

    funcao inicio()
    {
    }
}
```

Durante a escrita do código você irá atribuir valores correspondentes aos tipos dessas variáveis, por exemplo

```
programa
{
    cadeia nomeAluno
    real nota1, nota2, nota3, media

    funcao inicio()
    {
        nomeAluno = "Lucas"
    }
}
```

Comandos do Portugol para leitura e escrita de dados

Em um programa muitas (na maioria das) vezes queremos receber dados vindos do usuário, ou seja, dados que ele digitar, informar ao programar para esses dados serem processados, por exemplo um programinha que recebe notas de um determinado aluno e devolve para o usuário a média, temos 2 momentos nesse programinha que são importantes notarmos: entrada e saída de dados, no caso a entrada é o usuário informar as notas e a saída é o programa após o processamento mostrar em tela a média dessas notas.

```
programa
{
    cadeia nomeAluno
    real nota1, nota2, media

    funcao inicio()
    {
        escreva("Qual o nome do aluno? ")
        leia(nomeAluno)

        escreva("Insira a primeira nota: ")
        leia(nota1)

        escreva("Insira a segunda nota: ")
        leia(nota2)

        media = (nota1 + nota2) / 2

        escreva("A média de " + nomeAluno + " é " + media)
    }
}
```

O código acima faz exatamente o que foi descrito, que é um programa que lê 3 dados: nome do aluno e duas notas, esses valores são atribuídos nas variáveis criadas: nomeAluno, nota1 e nota2 respectivamente, após a leitura dos dados informados pelo usuário a variável média recebe o cálculo da média $(\text{nota1} + \text{nota2}) / 2$, e em seguida é mostrado ao usuário um texto formatado com o nome e a média do aluno. Observe que para escrever coisas para o usuário usamos o comando **escreva**, e para captar dados usamos o **leia**.

Outra coisa bacana que fizemos acima foi a formatação de textos junto de variáveis, isso é muito comum e é bem fácil de ser feito, isso se chama concatenação, unimos textos e variáveis afim de que ambos se completem e façam sentido.

Note também que na declaração de variáveis colocamos todas elas em uma linha após informar o tipo delas, isso é possível se todas as variáveis forem daquele tipo.

Vamos fazer um loop?

Os loops são usados em várias linguagens de programação, como o JavaScript que você irá aprender em breve, eles são usados para executar alguma tarefa por quantas vezes você definir, para que você entenda melhor vamos criar um programinha que se beneficiará do loop, no caso um sistema que também calcula a média das notas de um aluno, porém com o diferencial que o programa pergunta quantas notas serão calculadas:

```
programa
{
    cadeia nomeAluno
    inteiro qtdNotas, contador
    real somaNotas = 0.0, nota, media

    funcao inicio()
    {
        escreva("Qual o nome do aluno? ")
        leia(nomeAluno)

        escreva("Quantas notas serão calculadas? ")
        leia(qtdNotas)

        para (contador = 0; contador < qtdNotas; contador += 1) {
            escreva("Insira uma nota calculada: ")
            leia(nota)
            somaNotas = somaNotas + nota
        }

        media = somaNotas / qtdNotas
        escreva("A média de " + nomeAluno + " é: " + media)
    }
}
```

No Portugol existe o loop chamado "para", em linguagens de verdade esse loop se chama "for", como no JavaScript, ele funciona com base em uma condição, observe:

```
para (contador = 0; contador < qtdNotas; contador += 1) {
    // faça algo
}
```

- Temos a variável **contador** que inicia com o **valor 0**;
- Enquanto **contador** for menor que **qtdNotas** o loop irá fazer algo;
- A cada loop **contador** ganha **+1** em seu valor até que seu valor seja = **qtdNotas** e o loop termina;
- A cada loop o que está entre as **{ }** é executado;

Outra coisa legal de aprender é sobre o comprimentos de estruturas de dados, ou então de textos, isso é algo bastante visto no dia a dia de um programador, abaixo temos um programinha que recebe um texto e devolve o número de caracteres digitados

```
programa
{
    inclua biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto
        inteiro comprimento

        escreva("Digite um texto Qualquer: ")
    }
}
```

```

leia(texto)

comprimento = txt.numero_caracteres(texto)
escreva("Comprimento da entrada de dados: ", comprimento)
}
}

```

Além dos caracteres digitados que podem ser vistos os espaços em branco também são contabilizados, pois de qualquer forma também são caracteres, para esse cálculo do comprimento de um texto o Portugol conta com o **numero_caracteres**, em linguagens como o JavaScript temos o **length**, outra hora falamos sobre ele, porém a ideia é exatamente a mesma.

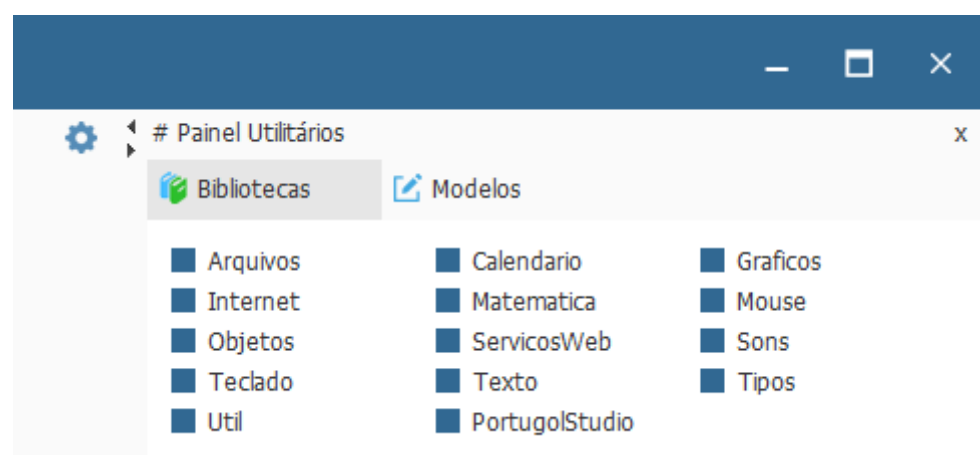
Observe que no topo do nosso código há algo diferente:

```

inclua biblioteca Texto --> txt

```

Esse tipo de coisa é o que chamamos de inclusão de bibliotecas, para o caso de fazer algumas funções especiais, nesse caso um biblioteca para trabalhar com textos, você pode encontrar e adicionar bibliotecas no canto superior direito da tela do Portugol Studio:



Não vamos focar muito em bibliotecas do Portugol, apenas se formos usar alguma, na vida real você irá se deparar com esse tipo de situação, onde você irá "instalar" algo em seu projeto e usá-lo em algum arquivo específico, mas basicamente o que fizemos foi incluir a biblioteca "Texto" e em seguida demos um nome a ela, no caso "txt", poderia ser qualquer outro nome que quiséssemos.

Vamos aprender um pouco mais com outro programinha, observe:

```

programa
{
    inclua biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto
        caracter letra
        inteiro tamanhoTexto

        escreva("Digite uma frase: ")
        leia(texto)

        tamanhoTexto = txt.numero_caracteres(texto)

        para(inteiro i=0; i < tamanhoTexto; i++) {

            letra = txt.obter_caracter(texto, i)
            escreva(letra, "\n")

        }
    }
}

```

Esse programinha recebe um texto e o imprime uma letra por vez, para que isso seja possível utilizamos o **obter_caracter** da biblioteca "Texto", essa pequena função espera 2 argumentos: um texto e a posição do caracter, como isso está em um loop temos a variável **i** que foi criada dentro do para, essa variável representa a posição de cada caracter dentro da palavra, e para quebrar a linha usamos **"/n"**, é como se fosse a tecla ENTER.

Condições

Vamos aprender agora sobre condicionais, por exemplo no calculo de média que fizemos, poderíamos direcionar o código para 2 lados: aprovado ou reprovado, para isso precisamos de uma condição, por exemplo: se a media for maior ou igual a 6 o aluno é aprovado, senão ele é reprovado, existe uma estrutura específica para isso na programação, no Portugol se chama **se**, no mundo real se chama **if**, observe

```
programa
{
    inteiro numero1, numero2

    funcao inicio()
    {
        escreva("Insira um numero inteiro: ")
        leia(numero1)

        escreva("Insira um numero inteiro: ")
        leia(numero2)

        se (numero1 > numero2) {
            escreva(numero1 + " é maior que " + numero2)
        } senao {
            escreva(numero2 + " é maior que " + numero1)
        }
    }
}
```

Claro que é um exemplo bem bobinho e simples, mas dá para perceber que dependendo de uma condição algo é executado, senão, outro algo será executado, essa é a ideia.

Observe bem a sintaxe do **se**

```
se (essa condição for verdadeira) {
    acontece isso
} senao {
    acontece isso outro
}
```

Essa ideia é comum em todas as linguagens, o que muda apenas é a sintaxe, ou seja, como é escrito, basta entender que se a primeira condição for verdadeira acontece o que está entre **{ }** do primeiro bloco, senão acontece o que está no outro bloco.

Vamos fazer o programinha da média das notas agora um pouquinho mais elaborado

```
programa
{
    cadeia nomeAluno
    inteiro qtdNotas, contador
    real somaNotas = 0.0, nota, media

    funcao inicio()
    {
        escreva("Qual o nome do aluno? ")
        leia(nomeAluno)

        escreva("Quantas notas serão calculadas? ")
        leia(qtdNotas)

        para (contador = 0; contador < qtdNotas; contador += 1) {
            escreva("Insira uma nota calculada: ")
            leia(nota)
            somaNotas = somaNotas + nota
        }

        media = somaNotas / qtdNotas

        se (media >= 6) {
            escreva("A média de " + nomeAluno + " é: " + media, " aprovado")
        } senao {
            escreva("A média de " + nomeAluno + " é: " + media, " reprovado")
        }
    }
}
```

Agora dependendo do valor da média uma mensagem diferente é mostrada, no caso se a média for maior ou igual a 6 aparece algo como *"A média de Lucas é: 7, aprovado"*, se for menor que 6 aparece algo como: *"A média de Lucas é: 4, reprovado"*.

Funções

Uma coisa EXTREMAMENTE importante na programação são as funções, seja no JavaScript que você irá aprender ou em outras linguagens, você pode ouvir também alguém as chamando de "método", mas pode ser método ou função, a ideia é a mesma.

Para que serve a função? Como eu crio uma? Como eu uso uma função?

Com as funções você cria um trecho de código que é executado sempre que você precisar sem precisar reescrever aquele trecho, apenas "chamando" a função que você criou para essa tarefa, observe:

```
programa
{

    funcao inicio()
    {
        minhaFuncao()
    }

    funcao minhaFuncao() {
        escreva("Essa é minha função")
    }
}
```

Assim como em algumas linguagens de programação, no Portugol você cria as funções/ métodos fora do main (que também é uma função, no caso a função principal, que é iniciada assim que você dá o "start" no seu programinha), e para executá-las você as "chama!" dentro da main.

As funções podem ficar mais elaboradas ao usarmos parâmetros, uma função com parâmetro funciona como um liquidificador, o liquidificador é a função, os parâmetros são água, açúcar e alguma fruta, e no fim você tem o suco, que é o que a função te devolve/ retorna.

Posso dar o nome que eu quiser para um parâmetro, ele é como uma variável criada apenas para funcionar dentro da função, e quando eu chamar essa função devemos passar entre os parênteses valores que façam sentido para aquela função.

```
programa
{
    inteiro resultado
    funcao inicio()
    {
        somaNumeros(3, 5)
    }

    funcao somaNumeros(inteiro n1, inteiro n2) {
        resultado = n1 + n2
        escreva("A soma de ", n1, " + ", n2, " é: ", resultado)
    }
}
```

Observe outro exemplo bacana do uso de funções, nesse caso nós colocamos toda a lógica dentro da função e na main apenas chamamos a função que faz o trabalho pesado

```
programa
{
    inteiro resultado, n1, n2

    funcao inicio()
    {
        somaNumeros()
    }

    funcao somaNumeros() {
        escreva("Digite um número inteiro: ")
        leia(n1)

        escreva("Digite outro número inteiro: ")
        leia(n2)

        resultado = n1 + n2
        escreva("A soma de ", n1, " + ", n2, " é ", resultado)
    }
}
```


Observe os símbolos

Na programação iremos nos deparar com alguns operadores que podem nos confundir no começo dos estudos, portanto vamos desde já entender como alguns dos mais importantes funcionam.

Imagine que você quer fazer uma comparação, verificar se um item é igual ao outro, do mesmo tipo ou se tem o mesmo valor, ou então você quer atribuir um valor a algo, ou ainda como faço para saber se um valor é maior ou igual a outro? Menor ou igual? Tem como fazer isso na programação? A resposta é sim, e é bastante simples

```
programa
{
    inteiro numero1, numero2

    funcao inicio()
    {
        verificaIgualdade()
    }

    funcao verificaIgualdade() {

        escreva("Informe um número: ")
        leia(numero1)

        escreva("Informe outro número: ")
        leia(numero2)

        se (numero1 == numero2) {
            escreva("Os números digitados são iguais")
        } senao {
            escreva("Os números digitados são diferentes")
        }
    }
}
```

Veja que interessante, usamos == para fazer uma comparação, na programação é assim, se usarmos apenas = estaremos atribuindo um valor, e não comparando.

Veja abaixo um exemplo de menor ou igual

```
programa
{
    inteiro idade

    funcao inicio()
    {
        verificaIdade()
    }

    funcao verificaIdade() {

        escreva("Informe a idade: ")
        leia(idade)

        se (idade >= 18) {
            escreva("Entrada permitida")
        } senao {
            escreva("Entrada proibida")
        }
    }
}
```

Podemos ainda incrementar esse programinha acima utilizando conhecimentos que vimos anteriormente

```
programa
{
    inteiro idade
    cadeia resposta

    funcao inicio()
    {
        entradaPermitida()
    }

    funcao entradaPermitida() {

        escreva("Informe a idade: ")
```

```

leia(idade)

se (idade < 18) {
    escreva("Está acompanhado dos pais? (Responda com SIM ou NÃO): ")
    leia(resposta)

    se (resposta == "SIM") {
        escreva("Entrada permitida")
    } senao {
        escreva("Entrada proibida")
    }
} senao {
    escreva("Entrada permitida")
}
}
}

```

Observe como inserimos um **se** dentro do outro.

Outra coisa bacana que podemos fazer é utilizar o enquanto, ele funciona como uma condicional também, algo como "enquanto isso for verdade, faça tal coisa", veja um exemplo com o portugol

```

programa
{

    cadeia resposta

    funcao inicio()
    {
        escreva("Escolha uma opção: 1 - iniciar: ")
        leia(resposta)

        enquanto (resposta != "0") {
            escreva("Escolha uma opção: 1 - continuar | 0 - sair: ")
            leia(resposta)
        }

        escreva("Fim")
    }

}

```

E você ainda pode notar o **!=** que significa "diferente", se a resposta for diferente de 0, continua.

Há muito mais do que podemos aprender sobre portugol, assim como há muito mais programinhas que você pode desenvolver com o que aprendeu, ensinamos um pouco do portugol não para que você o decore, não que você vá usar portugol no mercado de trabalho, mas sim para que você entenda que programar é unir seu pensamento com uma linguagem de programação e resolver problemas.