

Análise do programa e do manual

Antônio Bernábio Pereira Dos Santos Júnior, Sidney Gadelha de Oliveira.

Instituto Federal de Educação, Ciência e Tecnologia do Ceará | Campus Maracanaú
Av. Parque Central, 1315 - Distrito Industrial I, Maracanaú - CE,
61939-140

sidney.gadelha.oliveira08@aluno.ifce.edu.br;

antonio.bernabio.pereira08@aluno.ifce.edu.br

Resumo: *uma breve análise do desenvolvimento de um programa funcional, um quiz, e informações de seu funcionamento como um todo*

Abstract. *a brief analysis of the development of a functional program, a quiz, and information on its functioning as a whole*

Análise da funcionalidades em geral

A linguagem utilizada para produção do programa foi Portugol, que se trata de uma pseudolinguagem onde se pode desenvolver algoritmos em português estruturado, que foi aplicado em um programa para computador, um interpretador chamado “portugol studio” que permite interpretar linguagens próximas ao português.

O programa se trata de um quiz, onde o participante/jogador escolherá um tema e responderá perguntas de acordo com a opção escolhida.

Estrutura

A princípio, temos no topo do programa uma função, onde a mesma abre uma sessão de opções. Os comandos utilizadas no início do programa foram “escreva” se trata de um comando de saída, onde vai apresentar na tela do usuário as informações colocadas entre aspas orientando a escolher uma opção e logo após, a indicação de duas possibilidades onde a primeira oferece a escolha de iniciar o quiz e a segunda possibilita ver a quantidade máxima de pontos obtidos durante uma outra partida. A seguir temos o comando “Leia”, um comando de entrada que recebe a informação, no caso uma variável de um número inteiro que será digitada pelo jogador/participante onde ele irá informar ao script os dados necessários para entrar em uma das opções supracitadas.

```

“ funcao
opcao()
{ inteiro
numero
escreva("Esc
    olha uma
    opção\n")

```

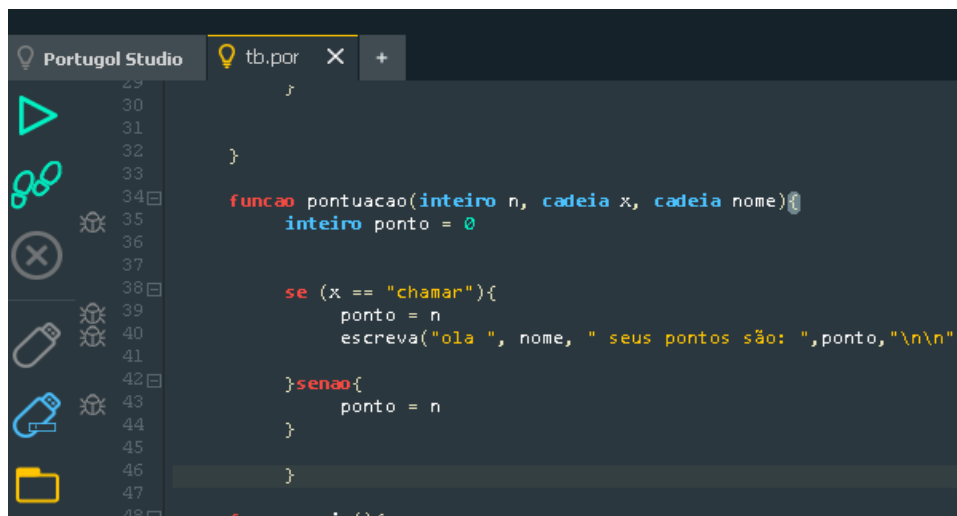
```

escreva("1-
    Quiz \n")
escreva("2
- Pontos \
n")
leia(ume
ro) se
(numero
== 1)
{ quiz() }
se(numero
== 2)
{ pontuac
ao(0,"x","
") } senao
{
escreva("") }
}”

```

O comando “Se” utilizada, refere-se a abertura de uma estrutura de decisão onde o participante vai ser condicionado a uma escolha, no mesmo caso o comando senão, porém, oferecendo uma contraproposta. Se o numero digitado for igual a 1 ele adentrara em outra estrutura para dar inicio ao quis propriamente dito, caso escolhido o numero 2, irá ver o percentual de acertos e quantidade máxima de pontos, e caso não for escolhida nenhuma das opções, se não for escolhido nem o numero 1 ou 2 o resultado que será apresentado na tela do jogador será de “opção inválida”.

Função na programação remete a um subprograma que retorna um determinado resultado em casos específicos.



```
}  
  
funcao pontuacao(inteiro n, cadeia x, cadeia nome){  
    inteiro ponto = 0  
  
    se (x == "chamar"){  
        ponto = n  
        escreva("ola ", nome, " seus pontos são: ",ponto,"\n\n")  
    }senao{  
        ponto = n  
    }  
  
}
```

A “função de pontuação”, aquela que apresenta a quantidade de pontos adquiridos ao decorrer das questões. Detalhamento da função propriamente dita: Declaração das variáveis como “inteiro n” , “cadeia x” , “cadeia nome”, uma estrutura “se” e “senão” para condicionar os comandos. A utilização do “Se” determina a possibilidade da variável X, na qual foi declarada conjunto de caracteres, ser igual ao nome CHAMAR, se sim, mostra a informação com o nome do participante e a quantidade de pontos conseguidos até então, senão apenas determina os pontos.

Esta função nomeada como “função Quiz”, resulta na maior parte do código onde é introduzido variáveis de cadeia: nome, alternativa. Variável de números inteiros: p = 0 , contador. Vetor de 12 secções chamado “cadeia questões” que representa a lista de questões. Declaração de dois comandos de saídas que fornecem na tela informações que pedem dados à ser reconhecidos e interpretados.

```

funcao quiz(){

    cadeia nome

    cadeia alternativa
    inteiro p = 0
    inteiro contador
    escreva("quiz selecionado \n\n\n")
    escreva("Digite seu nome: ")
    leia(nome)
    cadeia questoes[12]
    cadeia
alternativaA[12],alternativaB[12],alternativaC[12],alternativaD[12],alternativaE[12]
    cadeia respostas[12]
    respostas[0] = "D"
    respostas[1] = "A"
    respostas[2] = "B"
    respostas[3] = "E"
    respostas[4] = "B"
    respostas[5] = "D"
    respostas[6] = "A"
    respostas[7] = "E"
    respostas[8] = "C"
    respostas[9] = "E"
    respostas[10] = "E"
    respostas[11] = "A"

```

Incrementação de cadeias de vetores correlacionados aos dados de respostas das perguntas em geral do quiz com um total de 12 secções e 5 vetores: alternativaA, alternativaB, alternativaC, alternativaD, alternativaE, com 12 secções, para ser os identificadores do itens, armazenando texto e/ou numero que represente as respostas das perguntas fornecidas.

“

```

alternativaA[12],alternativaB[12],alternativaC[12],alternativaD[12],alternativaE[12
] cadeia respostas[12] ”

```

```
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72

funcao quiz(){
    cadeia nome

    cadeia alternativa
    inteiro p = 0
    inteiro contador
    escreva("quiz selecionado \n\n\n")
    escreva("Digite seu nome: ")
    leia(nome)
    cadeia questoes[12]
    cadeia alternativaA[12], alternativaB[12], alternativaC[12], alternativaD[12], alternativaE[12]
    cadeia respostas[12]
    respostas[0] = "D"
    respostas[1] = "A"
    respostas[2] = "B"
    respostas[3] = "E"
    respostas[4] = "B"
    respostas[5] = "D"
    respostas[6] = "A"
    respostas[7] = "E"
    respostas[8] = "C"
    respostas[9] = "E"
    respostas[10] = "E"
```

A finalização do programa a criação de um vetor “para (contador = 0; contador <= 11; contador++)” onde vai analisar todas os itens escolhidos durante o trajeto da primeira e ultima questão

```
para (contador = 0; contador <= 11; contador++)
{
    escreva(questoes[contador])
    escreva("A) ", alternativaA[contador], "\n")
    escreva("B) ", alternativaB[contador], "\n")
    escreva("C) ", alternativaC[contador], "\n")
    escreva("D) ", alternativaD[contador], "\n")
    escreva("E) ", alternativaE[contador], "\n\n")
    escreva("selecione uma alternativa\n")
    leia(alternativa)
```

```
por
{
    para (contador = 0; contador <= 11; contador++)
    {
        escreva(questoes[contador])
        escreva("A) ", alternativaA[contador], "\n")
        escreva("B) ", alternativaB[contador], "\n")
        escreva("C) ", alternativaC[contador], "\n")
        escreva("D) ", alternativaD[contador], "\n")
        escreva("E) ", alternativaE[contador], "\n\n")
        escreva("selecione uma alternativa\n")
        leia(alternativa)
```

para cada questão, há itens com vetores de 0 a 11, e das 12 questões os comandos de saída citados acima vão ser realizados em cada uma delas e também um dado de entrada para indicar a questão escolhida com comando “leia(alternativa)”.

Uma estrutura condicional

```
se(alternativa == respostas[contador]){  
    escreva("acertou a questão \n\n\n\n\n")  
    p = p + 1  
    pontuacao(p,"x","")
```

```
se(alternativa == respostas[contador]){  
    escreva("acertou a questão \n\n\n\n\n")  
    p = p + 1  
    pontuacao(p,"x","")
```

que impõe um condição de que se a alternativa for igual a respostas do contador ou seja
, se estiver com a resposta correta, o comando “escreva” de saída mandará uma mensagem dizendo que acertou a questão e a variável “P” irá somar 1 ponto na quantidade total da pontuação que no caso é cumulativo .

```
se(alternativa != respostas[contador]){  
    escreva("errou \n\n\n\n\n")  
    p = p + 0  
    pontuacao(p,"x","")
```

```
}se(alternativa != respostas[contador]){  
    escreva("errou \n\n\n\n\n")  
    p = p + 0  
    pontuacao(p,"x","")
```

onde apenas entra para executar os comandos se , somente se a alternativa for diferente da resposta correta do vetor “contador” e irá ser somado 0 pontos o que indica que naquela questão especificamente o jogador não irá ganhar nenhum ponto por errar a questão mantendo assim os pontos da questão anterior e irá resultar em um comando de saída representando que o dado inserido pelo participante não está certo.

MANUAL DE INSTALAÇÃO

Manual de Instalação

*Para fins de execução do programa
faz-se necessário a instalação de um
programa cujo nome é portugol
studio.
O download pode ser feito pelo link
abaixo que vai redirecionar para o
download*



LINK

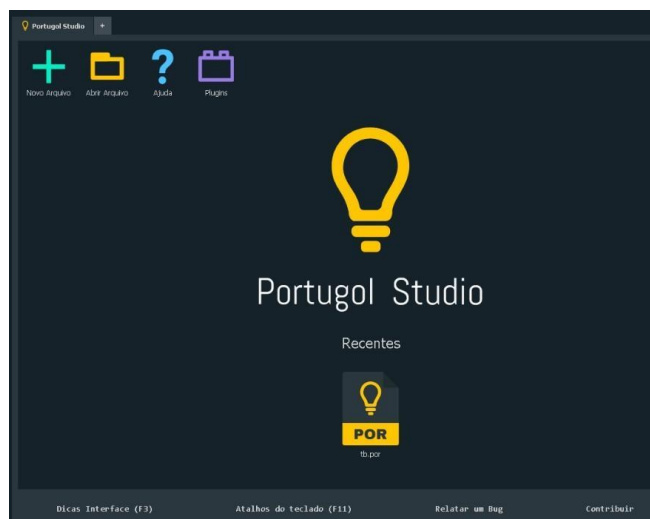
*ou entrando pelo seu
navegador e pesquisar por
portugol studio download*

A TELA APRESENTADA PARA
AS DUAS OCASIÕES É ESTA



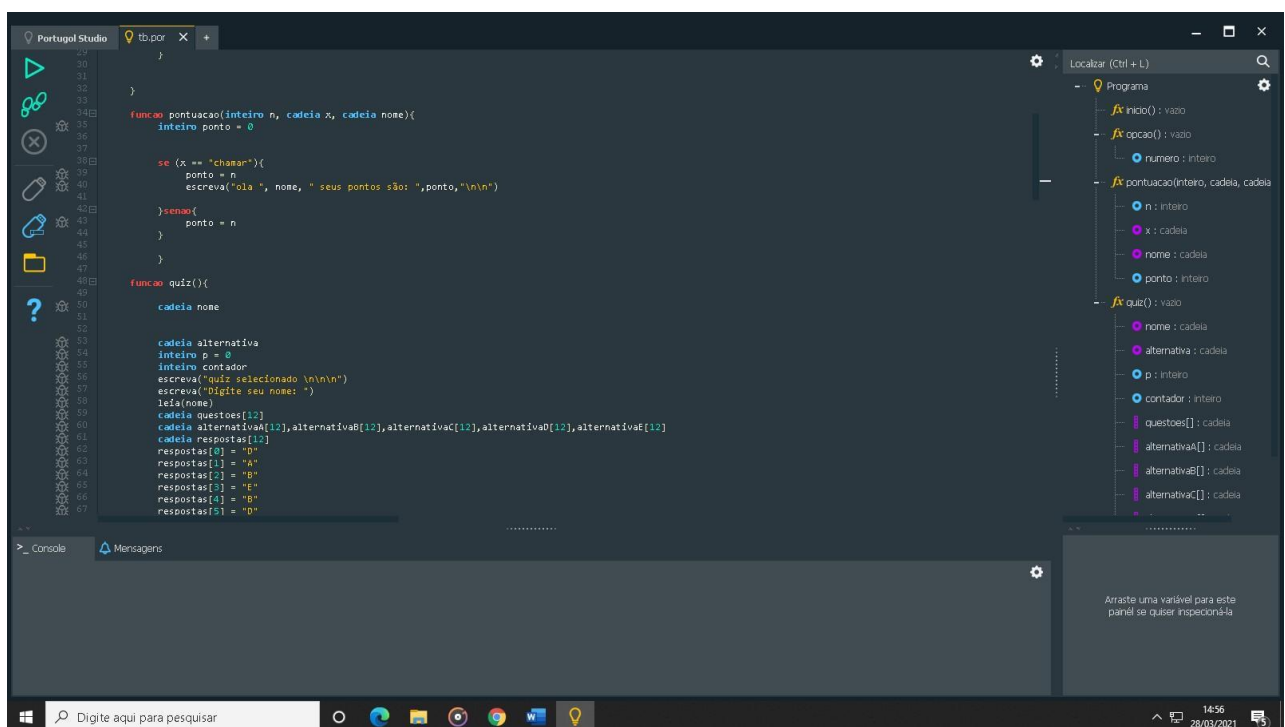
basta apenas
clicar sobre o
botão download
que será baixado
após o clique

INICIO DO SOFTWARE

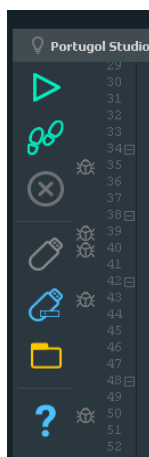


1. Para iniciar o quiz, deve-se primeiro iniciar o software Portugol no PC.
2. Identificar o programa que tem os algoritmos do Quiz.
3. Abrir o quiz.exe que aparecerá a tela seguinte.

TELA INICIAL APRESENTADA



COMANDOS DO PROGRAMA PARA A EXECUÇÃO



A cada ícone de cima para baixo, logo abaixo estará a função de cada um.

1. Executar o programa
2. Executar o programa passo a passo
3. Interromper a execução
4. Salvar o programa atual
5. Salvar uma nova cópia do programa
6. Abrir um arquivo .por
7. Ajuda.

Clique na função 1, executar.

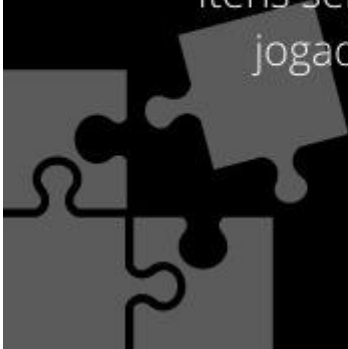
MENU PRINCIPAL



MENU

- 1 o jogador poderá selecionar duas opções onde a primeira dará início ao jogo e a segunda opção verá a pontuação

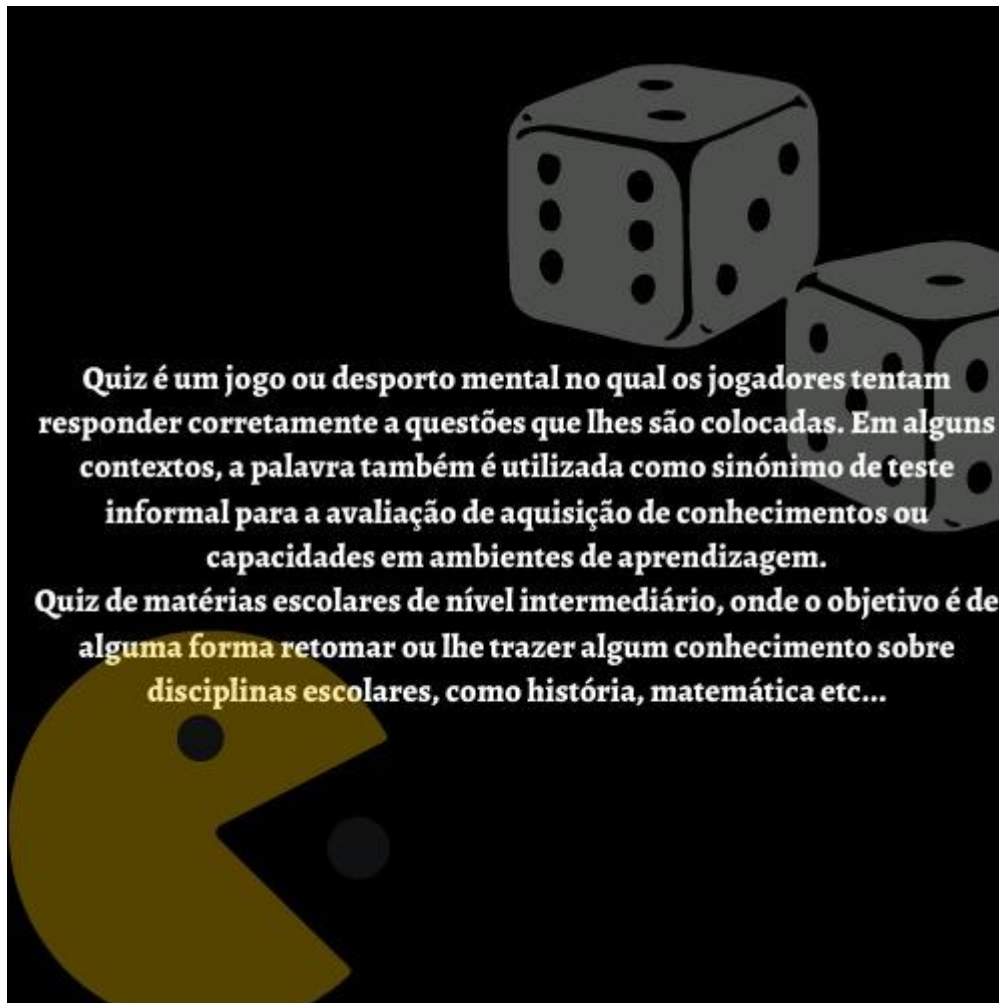
- 1.1 inicia a jogo de quiz. após a seleção terá algumas perguntas e dependendo dos itens selecionados como respostas o jogador irá ou não ganhar uma pontuação.



2 - PONTUAÇÃO

A pontuação valerá de acordo com o andamento do participante no jogo onde a cada questão correta irá ganhar 1 ponto que se acumulará até a última questão sem ocorrer perdas, ou seja, se o jogador errar eles receberá um total de 0 ponto válidos para aquela questão errada.





Link Github:

<https://github.com/Sidney-Aluno-IFCE/trabalho>