



## DESAFIO PARA O CARGO DE PROGRAMADOR

### Informações sobre o Desafio

- As questões devem ser respondidas em word porém o teste deve ser entregue em versão .pdf constando nome completo e data de realização
- Em caso de dúvidas encaminhar e-mail para [renato.alves@be3.co](mailto:renato.alves@be3.co)

**Nome Completo: Henrique Hernandes Da Silva Rodrigues Data: 19/07/2021**

- 1) No código abaixo há 3 erros de lógica (não de sintaxe). Identifique-os, transcrevendo as respectivas linhas defeituosas, descrevendo por que estão erradas, e reescrevendo-as corretamente. (2.5)

```
public bool verificadisponibilidade(datetime datahoracandidata, int códmedico, int tproc)

{

    datetime[,] hi = new datetime[8, 4];

    datetime[,] hf = new datetime[8, 4];

    db = new database();

    system.data.sqlclient.sqldatareader rstabdispstd;

    system.data.sqlclient.sqlparameter[] pramsdisppadrão = new system.data.sqlclient.sqlparameter[1];
    pramsdisppadrão[0] = db.makeinparam("@codmed", system.data.sqlDbType.int, 12, códmedico);

    db.runproc("disppadrãoopormédico", pramsdisppadrão, out rstabdispstd);

    db.close();

    int i = 0;

    if (rstabdispstd.hasrows)

    {
```

```

while (rstabdispstd.read())

{

    hi[i, 1] = rstabdispstd.getdatetime(rstabdispstd.getordinal("hi1"));

    hf[i, 1] = rstabdispstd.getdatetime(rstabdispstd.getordinal("hf1"));

    hi[i, 2] = rstabdispstd.getdatetime(rstabdispstd.getordinal("hi2"));

    hf[i, 2] = rstabdispstd.getdatetime(rstabdispstd.getordinal("hf2"));

    hi[i, 3] = rstabdispstd.getdatetime(rstabdispstd.getordinal("hi3"));

    hf[i, 3] = rstabdispstd.getdatetime(rstabdispstd.getordinal("hf3"));

    // aqui tem que adicionar para rodar o array, se diminuir ele vai dar exception no array por ser um valor
    negativo

    i++;

}

rstabdispstd.close();

db.close();

}

datetime hl1 = datahoracandidata;

bool found = false;

for (int y = 0; y < 7; y++)

{

    for (int z = 1; z <= 3; z++)

    {

```

```

if (hl1.timeofday >= hi[y, z].timeofday & hl1.timeofday <= hf[y, z].addminutes(-

tproc).addseconds(1).timeofday)

{

    found = true;

    // pode um exception no for pois está colocando 1 + 3 = 4

    // se o array não tiver essa posição vai gerar uma exception no segundo array

    // e o primeiro array so vai rodar mais 4 vezes se cair nesse if

    //z = 3;

    //y = 3;

}

}

}

//não tem como retornar i pois não está dentro das chaves, não é um boolean

return found;

}

```



- 2) Crie uma função chamada ValidaCPF, booleana, que retorne o status da validação de um suposto CPF sem formatação. Regra de validação (fictícia): os dois últimos dígitos à direita devem corresponder ao resto R da divisão de um somatório S por 11, onde S é o somatório do valor da multiplicação de cada dígito pela sua posição (da primeira à nona, da esquerda para a direita). Caso R tenha apenas um dígito, preencher o dígito à esquerda com "0".

Utilize a seguinte linguagem para a codificação: C#. (2.5)

```

public static bool validarCPF(string cpf)
{
    int[] digitos = new int[11];
    int somatorio = 0;
    for (int i = 0; i < 11; i++)
        digitos[i] = Convert.ToInt32(cpf.Substring(i, 1));

    for (int i = 0; i < 8; i++)
    {
        int let = i * digitos[i];
        somatorio = somatorio + let;
    }
    int soma = Convert.ToInt32($"{digitos[9]}" + $"{digitos[10]}");
    int R = somatorio / 11;
    R = (R < 10) ? Convert.ToInt32("0" + R) : R;
    if (R == soma)
        return true;
    else
        return false;
}

```

- 3) Considere que uma agenda de consultas de uma clínica médica é uma lista de horários de início e fim dos compromissos dessa agenda, e que os horários dessa agenda não podem se sobrepor.  
 Faça uma função que receba dois parâmetros: lista de DateTime com horários de início e fim (que pode estar desordenada), e um par de horários de início e fim. A função deve retornar se o par de horários candidato pode ser inserido na lista de horários da agenda.  
 Crie uma segunda função que recebe a lista de horários e calcule a taxa de ocupação dessa agenda considerando que o atendimento ocorre das 08h às 18h.  
 Utilize a seguinte linguagem para a codificação: C#. (2.5)
- 4) Considere uma lista de 100 elementos contendo o código do procedimento, descrição, tipo, valor e forma de pagamento, onde tipo pode ser consulta, exame ou cirurgia e forma de pagamento cartão de crédito, cartão de débito, cheque ou dinheiro. Faça um algoritmo que calcule o total, por tipo e forma de pagamento. Utilize a seguinte linguagem para a codificação: C#. (2.5)

Exemplos de registros da lista:

Código do	Descrição Tipo Valor Forme de pagamentoprocedimento			
10101012	Consulta em consultório	Consulta	50,00	Cartão de débito
10101012	Consulta em consultório	Consulta	100,00	Dinheiro
10101012	Consulta em consultório	Consulta	100,00	Cartão de débito
32102544	Campimetria	Exame	150,00	Cartão de débito
55654525	Retinografia	Exame	200,00	Cartão de débito
33656545	Facectomia	Cirurgia	1000,00	Cartão de crédito
44545856	Lasik	Cirurgia	2000,00	Cartão de débito

Exemplos de saída esperada do algoritmo:

Tipo Forma de pagamento Valor total		
Consulta	Cartão de débito	150,00
Consulta	Dinheiro	100,00

Exame	Cartão de débito	350,00
Cirurgia	Cartão de crédito	1000,00
Cirurgia	Cartão de débito	2000,00

```

public class registro{
    public int codigoDoProcedimento { get; set; }
    public string descricao { get; set; }
    public string tipo { get; set; }
    public double valor { get; set; }
    public string formaDePagamento { get; set; }
}

public static void Faturamento(List<registro> registros)
{
    string[] formasDePagamento = new string[] { "Dinheiro", "Cartão de crédito", "Cartão de débito" };

    string[] Procedimentos = new string[] {"Cirurgia", "Exame", "Consulta"};

    string resultado = "";
    for (int i = 0; i < Procedimentos.Length; i++)
    {
        for (int j = 0; j < formasDePagamento.Length; j++)
        {
            double valor = 0;
            foreach (var registro in registros)
            {
                if (Procedimentos[i] == registro.tipo)
                {
                    if (formasDePagamento[j] == registro.formaDePagamento)
                    {
                        valor += registro.valor;
                    }
                }
            }
            if (valor != 0)
                resultado += $"Tipo:{Procedimentos[i]} forma de pagamento:{formasDePagamento[j]} Valor:
{valor}\n";
        }
    }
    Console.WriteLine(resultado);
}

```