

# TEMA PROJETO: Cadeia de Restaurantes(Bob's ou Subway)

Discente: João Batista Araújo de Lima Filho.

## Questão 2

2. Descrição textual das funcionalidades do seu projeto:

a) (10 pontos) Quais objetos o usuário vai poder cadastrar, editar ou excluir?

O usuário vai poder manipular os objetos Sanduiche.

b) (10 pontos) Que operações cada objeto pode realizar para o usuário?

Cadastrar nome, preço, tamanho, quantidade de carne em cada sanduiche e o ponto dela. Podendo editar e excluir os mesmos.

## Questões 3, 4, 5 e 6

### Classe Sanduiche

```
package Atividade11;

import java.io.Serializable;

// 3- Crie uma classe básica do projeto.

    "Essa é a classe básica do projeto"

public class Sanduiche implements Item, Serializable {

    private String nome;
    private String tamanho;
    private double preco;
    int numeroDeCarnes;
    String pontoDaCarne;

    int id;

    // 4- Crie o construtor da classe básica.

    "Esse e o construtor da classe Sanduiche"

    public Sanduiche(String nome, String tamanho, double preco) {
        setNome(nome);
        setTamanho(tamanho);
        setPreco(preco);
    }

    // 5- Crie um método que realiza operações úteis para o usuário (não pode ser
    //métodos do tipo get(), set() ou toString()).

    "Esse é um método da classe Sanduiche, que adiciona o numero de carnes do
    sanduiche"

    void numCarnes(int numeroDeCarnes) {
        this.numeroDeCarnes = numeroDeCarnes;
        if (numeroDeCarnes > 1) {
            this.preco += (numeroDeCarnes - 1) * 3.99;
        }
    }
}
```

```

}

// 6- Crie um segundo método que realiza operações úteis para o usuário (não pode
//ser métodos do tipo get(), set() ou toString()).

    “Esse é outro metodo da classe Sanduiche que recebe o ponto da carne do sanduiche”

void pontoCarne(String ponto) {
    this.pontoDaCarne = ponto;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getTamanho() {
    return tamanho;
}

public void setTamanho(String tamanho) {
    this.tamanho = tamanho;
}

public double getPreco() {
    return preco;
}

public void setPreco(double preco) {
    this.preco = preco;
}

@Override
public String toString() {
    return "Sanduiche: "+nome +", Tamanho: "+ tamanho+", Preco: "+preco;
}

@Override
public int getID() {
    return id;
}

@Override
public void setID(int id) {
    this.id = id;
}
}
}

```

Questões 7, 8 e 9

Classe RepositorioSanduiche

```
package Atividade11;
```

```
import java.io.*;
```

```
import java.util.ArrayList;
```

```
// 7- Um repositório da sua classe básica.
```

“Aqui temos o repositório da classe básica Sanduiche”

```
public class RepositorioSanduiche implements Serializable {
```

```
    ArrayList<Item> sanduiches;
```

```
    int cont;
```

```
    public RepositorioSanduiche() {  
        this.sanduiches = new ArrayList<Item>();  
        cont = 0;  
    }
```

```
    public void cadastraSanduiche(Item sanduiche){  
        sanduiche.setID(cont);  
        this.sanduiches.add(sanduiche);  
        cont++;  
    }
```

```
    public int ultimoID(){  
        return cont;  
    }
```

```
    public Item recuperar(int id) {  
        for (Item inter : sanduiches) {  
            if (inter.getID() == id)  
                return inter;  
        }  
        return null;  
    }
```

```
    public static void mostraSanduiche(ArrayList<Sanduiche> sanduiches){  
        for (Sanduiche i: sanduiches){  
            System.out.println(i.getNome());  
        }  
    }
```

```
// 8- O repositório é salvo em arquivo, utilizando a interface Serializable.
```

“Aqui temos o método que salva o repositório de sanduiches num arquivo utilizando a interface Serializable”

```
    public void salvarRepositorio(String caminhoArquivo) throws IOException {
```

```
        FileOutputStream fout = new FileOutputStream(caminhoArquivo);
```

```
        ObjectOutputStream oos = new ObjectOutputStream(fout);
```

```
        oos.writeObject(this);
```

```
        oos.close();
```

```
    }
```

```
// 9- O repositório é carregado do arquivo.
```

“Aqui temos o metodo que carrega os arquivos que foram salvos no arquivo utilizando a interface Serializable”

```
public static RepositorioSanduiche carregarRepositorio(String caminhoArquivo) throws
IOException, ClassNotFoundException {

    FileInputStream fin = new FileInputStream(caminhoArquivo);
    ObjectInputStream ois = new ObjectInputStream(fin);
    RepositorioSanduiche resp = (RepositorioSanduiche) ois.readObject();
    ois.close();
    return resp;
}

}
```

### Questão 10

#### Classe Main com Interface em linha de comando

```
package Atividade11;

import java.io.IOException;
import java.util.Scanner;

public class Main extends RepositorioSanduiche{

    // 10- GUI (interface com usuário), pode ser uma interface em linha de comando.

    “Aqui temos a GUI em linha de comando que recebe as entradas do usuario e faz as
    operações”

    static final String opcoes = "Digite qual operacao voce deseja fazer: \n1 - Cadastrar
Sanduiche. \n2 - Mostrar Sanduiches. " +
        "\n3 - Sair.\n";

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        boolean continuar = true;

        RepositorioSanduiche repositorioSanduiche = null;
        String arquivoRepositorio = "sandwiches.txt";
        try {
            repositorioSanduiche =
RepositorioSanduiche.carregarRepositorio(arquivoRepositorio);
        } catch (IOException e) {
            System.out.println("Nao foi possivel carregar o repositorio.");
            repositorioSanduiche = new RepositorioSanduiche();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }

        while (continuar) {
```

```

System.out.println(opcoes);
int leitura = sc.nextInt();

if (leitura == 1) {
    System.out.println("Digite o nome do Sanduiche: ");
    String a = sc.nextLine();
    String nome = sc.nextLine();

    System.out.println("Digite o tamanho do Sanduiche: ");
    String tamanho = sc.next();

    System.out.println("Digite o preco do Sanduiche: ");
    double preco = sc.nextDouble();

    Item sanduiche = new Sanduiche2(nome, tamanho, preco);
    repositorioSanduiche.cadastraSanduiche(sanduiche);
    // salvando repositorio no arquivo
    try {
        repositorioSanduiche.salvarRepositorio(arquivoRepositorio);
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("Nao foi possível salvar o arquivo.");
    }

} else if (leitura == 2) {
    for (int i = 0; i < repositorioSanduiche.ultimoID(); i++) {
        Item a = repositorioSanduiche.recuperar(i);
        System.out.println(a);
    }
} else if (leitura == 3) {
    continuar = false;
}

}

}

}

```

**Imagem das Entradas e Saídas na Interface com Usuário**

```
/usr/lib/jvm/java-11-openjdk-amd64/bin/java -javaagent:/snap/int
Digite qual operacao voce deseja fazer:
1 - Cadastrar Sanduiche.
2 - Mostrar Sanduiches.
3 - Sair.

2
Sanduiche: Bobs Bacon, Tamanho: Medio, Preco: 12.0
Digite qual operacao voce deseja fazer:
1 - Cadastrar Sanduiche.
2 - Mostrar Sanduiches.
3 - Sair.

1
Digite o nome do Sanduiche:
Bobs Cheddar
Digite o tamanho do Sanduiche:
Grande
Digite o preco do Sanduiche:
11,99
Digite qual operacao voce deseja fazer:
1 - Cadastrar Sanduiche.
2 - Mostrar Sanduiches.
3 - Sair.

2
Sanduiche: Bobs Bacon, Tamanho: Medio, Preco: 12.0
Sanduiche: Bobs Cheddar, Tamanho: Grande, Preco: 11.99
Digite qual operacao voce deseja fazer:
1 - Cadastrar Sanduiche.
2 - Mostrar Sanduiches.
3 - Sair.

3

Process finished with exit code 0
```

Classe Sanduiche2

```

package Atividade11;

import java.io.Serializable;

public class Sanduiche2 implements Item, Serializable {

    int id;
    Sanduiche sanduiche;

    public Sanduiche2(String nome, String tamanho, double preco){
        this.sanduiche = new Sanduiche(nome,tamanho,preco);
    }

    @Override
    public String toString() {
        return sanduiche.toString();
    }

    @Override
    public int getID() {
        return id;
    }

    @Override
    public void setID(int id) {
        this.id = id;
    }
}

```

### Interface Item

```

package Atividade11;

public interface Item {

    int getID();

    void setID(int id);
}

```