

**Documentação Técnica do Projeto - JCOFFE**

**Ensino Médio Integrado à Informática**

**Prática de Laboratório - PL2**

Alicia da S. Rodrigues

Cibelly Angel da Silva

Erick Amando C.

Melissa Alves dos Santos

Rebeca Virtuosa dos S.

Renata Gardenia da S. C.

**2022**

**SUMÁRIO**

[**1. SOBRE O SISTEMA**](#_kcdulfegr21e) **3**

[1.1. IDE](#_blmkx4jxifu) 3

[1. 2. Linguagem](#_cox39hyv8o2v) 3

[**3. DIAGRAMA DE CLASSES**](#_igx4yclymdnp) **5**

[**4. CLASSES E MÉTODOS**](#_m6abaqtpk1o4) **5**

[4.1. Classe Pedido](#_qbpn1ngz31k0) 5

[Atributos](#_kuxid3iv25yz) 6

[Métodos](#_w91lrn659r6m) 6

[4.2. Classe Cafe](#_69jz4pxpdx3k) 7

[Atributos](#_u4yca6tale04) 7

[Métodos](#_adag447flk5i) 7

[**4.3. Classe Item**](#_hzmloxs4wsa5) **7**

[Atributos](#_fj0phr5heljj) 7

[Métodos](#_lcosj3sacnw) 7

[**5. FORMULÁRIOS**](#_r87i8ug16z3z) **8**

[5.1.Form Menu](#_qhcnv6f37h7c) 8

[Actionperformed](#_3g78d0rj2a7a) 8

[Métodos](#_ufkjsaeprl5) 8

[MouseClicked](#_5d58mnxqn68n) 9

[5.2.Form Caixa](#_r3oiuj5u2i2a) 9

[Actionperformed](#_aaidl9v5uzva) 9

[Método](#_grdb3o7v1551) 10

# SOBRE O SISTEMA

Este software foi desenvolvido com o objetivo de simular uma máquina de cafeteria, onde é possível fazer um pedido de uma forma fácil e rápida. Denominamos o sistema de JCOFFE, a combinação de Java e Café em inglês.

## **1.1. IDE**

Utilizamos inicialmente a IDE Netbeans 8.0, no ambiente escolar e no particular com o jdk 8.

Abaixo, são apresentados os sites para instalação da IDE e do JDK:

**- netbeans** : https://netbeans.apache.org

**- jdk:** https://www.oracle.com/br/java/technologies/downloads/

Nos sites tem versões mais atualizadas do que utilizamos no nosso sistema pois infelizmente a Oracle não fornece mais o netbeans 8.2, portanto dependendo do uso, recomendamos uma IDE mais atualizada.

## 

## 1**. 2. Linguagem**

A linguagem de programação utilizada no sistema foi Java. Uma linguagem orientada a objetos de alto nível, que aprendemos durante o ano.

# **3. DIAGRAMA DE CLASSES**

# 

# **4. CLASSES E MÉTODOS**

## **4.1. Classe Pedido**

Dentro dessa classe é armazenado o pedido realizado pelo usuário, pelos atributos privados, getters e setters e métodos:

### Atributos

* **Cliente:** tipo string , que armazena o nome do responsável pelo pedido, que será digitado pelo usuário;
* **Itens:** ArrayList de objetos da classe Café , que armazena todos cafés selecionados pelo usuário;
* **Total:** tipo double, que armazena o total do pedido, após chamada do método ***CalcularTotal();***
* **Troco:** tipo double, que armazena o troco do pedido, após a chamada do método***Pagamento();***
* **Comanda:** tipo inteiro e estático, que armazena a quantidade de pedidos feito pelos usuários;
* **FormaPag**: tipo caracter, que armazena a forma de pagamento selecionada pelo usuário.

### Métodos

* **Construtor:** construtor padrão sem argumentos, que instancia o arraylist e acrescenta +1 ao atributo estático comanda.
* **CalcularTotal():** método que faz a soma de preços do pedido, percorrendo pelo arraylist, multiplicando pela quantidade e armazenado no atributo total.
* **Pagamento()**: método que tem como um argumento um valor double, no qual verifica se este valor é o suficiente para realizar o pagamento, retornando um valor booleano.

## 

## 

## 

## **4.2. Classe Cafe**

### Atributos

* **Nome** : armazena o nome do café.
* **Preço** : armazena o preço do café .

### **Métodos**

* **Construtor**: construtor sobrecarregado com argumentos que atribui valores para os atributos nome e preço.

# 

# 4.3. Classe Item

### **Atributos**

* **Cafe** : armazena o objeto do tipo Cafe(classe);
* **Qtd** : armazena a quantidade de cafés desejados pelo usuário .

### **Métodos**

* **Construtor:** construtor sobrecarregado com argumentos que atribui valores para os atributos café e qtd.

# 

# 

# 

# 

# **5. FORMULÁRIOS**

## **5.1.Form Menu**

### Actionperformed

* **Botão “ +” - BtnMais** : Verifica se a quantidade do item é maior que um e aumenta +1.
* **Botão “-” - BtnMenos**: Verifica se a quantidade do item é maior que um e se for diminui um, senão o valor da quantidade é um.
* **Botão Adicionar - BtnAdd:**  Verifica se os campos(nome e item) estão limpos, senão instância os objetos cafe e item chamando o método *adicionar(item)* :
* **Botão Confirmar - BtnConfirmar :** verifica se o pedido foi feito, caso sim, instancia um objeto da classe Pedido e atribui nele o arrayList pedido, abrindo um novo form(Form Caixa) com o objeto como argumento.
* **Botão Cancelar - BtnCancelar :** Limpa todos campos (labels, campo de texto, arraylist, lista), cancelando o pedido que foi feito.

### Métodos

* **adicionar() :** Verifica se o objeto a ser adicionado já não está na lista, caso não estiver ele adiciona no arrayList e no *listPedido*(JList).

### MouseClicked

Painéis com cada tipo de Café que ao ser clicado, direciona o nome, preço do café selecionado para os campos de texto *TxtItem* e *TxtPreco*.

* Cafe Tradicional - R$ 8.50
* Cafe com Leite - R$ 9.50
* Expresso - R$ 10.50
* Achocolatado - R$ 12.00
* Latte - R$ 14.00
* Mocha - R$ 14.00
* Cappuccino - R$ 15.00
* Macchiato - R$ 15.00
* Descafeinado - R$ 18.00

## **5.2.Form Caixa**

### Actionperformed

* **Botão Pagar - BtnPagar :**
  + Se o botão rádio cartão for selecionado, no objeto pedido irá ser setado a forma de pagamento “Cartão” e chamará o método do ***pagamento()*** da classe Pedido com o valor do campo de texto valor (*TxtValor)*, que se retornar o valor booleano *true* irá chamar o método ***mostrarComprovante()*.**
  + Se o botão radio dinheiro for selecionado, no objeto pedido irá ser setado a forma de pagamento “Dinheiro” e se o campo txtvalor não estiver limpo ocorrerá um tratamento de erro para verificar se o valor dele está incorreto(Exception NumberFormat), caso não esteja chamará o método da classe Pedido  ***pagamento()*** que retornando verdadeiro irá exibir o comprovante e o troco(*TxtTroco*).
  + Caso nenhum dos botões rádios forem selecionados, será apresentado um ``JOptionPane'', com a mensagem de erro.

* **Botão Voltar - BtnVoltar :** Volta para o form anterior (*FormMenu*)
* **Botão Radio Cartão - RbtnCartao :** atribui no campo de texto Valor(*Txtvalor*), o valor do total do pedido.

### Método

* **mostrarComprovante() :** quando esse método for chamado, o painel comprovante irá mostrar as informações principais do pedido, como o nome do responsável pela compra, o número do pedido, a lista de itens e o total da compra.

# 

### 