

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)
- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Administrador

[java.lang.Object](#)

[com.example.pedidoentrega.Usuario](#)

[com.example.pedidoentrega.Administrador](#)

public class Administrador extends [Usuario](#)

Classe Administrador herda a classe abstrata Usuario, possui uma lista de carrinhos que representa os pedidos dos clientes e guarda os dados do administrador.

• Field Summary

Fields

Modifier and Type

Field

Description

private ArrayList<Carrinho>
carrinhos

private String
cnpj

• Constructor Summary

Constructors

Constructor

Description

Administrador(String email, String cnpj, String senha)
Construtor do Administrador.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

adicionarCarrinho(Carrinho carrinho)

MÃ©todo para adicionar carrinho na lista.

ArrayList<Carrinho>

getCarrinhos()

MÃ©todo para retornar a lista de carrinhos.

Methods inherited from class com.example.pedidoentrega.Usuario

getEmail, getSenha

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ cnpj

private String cnpj

♦ **carrinhos**

private ArrayList<Carrinho> carrinhos

• **Constructor Details**

♦ **Administrador**

public Administrador(String email, String cnpj, String senha)
Construtor do Administrador.

Parameters:

email - String- email do administrador

cnpj - String - cnpj do estabelecimento

senha - String - senha do administrador

• **Method Details**

♦ **adicionarCarrinho**

public void adicionarCarrinho(Carrinho carrinho)
MÃ©todo para adicionar carrinho na lista.

Parameters:

carrinho - Carrinho - novo carrinho do cliente

♦ **getCarrinhos**

public ArrayList<Carrinho> getCarrinhos()
MÃ©todo get para retornar a lista de carrinhos.

Returns:

Carrinho - lista de carrinhos

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class **AlterarEnderecoController**

[java.lang.Object](#)

`com.example.pedidoentrega.AlterarEnderecoController`

public class `AlterarEnderecoController` extends [Object](#)

Classe Controller da interface `alterarEndereco` para alterar o endereço de entrega.

• Field Summary

Fields

Modifier and Type

Field

Description

`private javafx.scene.control.Label`
[erro](#)

`private javafx.scene.control.TextField`
[tf_bairro](#)

`private javafx.scene.control.TextField`
[tf_numero](#)

`private javafx.scene.control.TextField`
[tf_rua](#)

• Constructor Summary

Constructors

Constructor

Description

[AlterarEnderecoController\(\)](#)

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

`void`

[lidarBotaoCardapio\(javafx.event.ActionEvent e\)](#)
MÃ©todo para mudar para interface cardapio.

`void`

[lidarBotaoCarrinho\(javafx.event.ActionEvent e\)](#)
MÃ©todo para mudar para interface carrinho.

`void`

[lidarBotaoConfirmar\(javafx.event.ActionEvent e\)](#)
MÃ©todo para verificar se todos os campos do endereÃ§o foram preenchidos e, entÃ£o, chamar a interface de pagamento.

`void`

[lidarBotaoMeuPedido\(javafx.event.ActionEvent e\)](#)
MÃ©todo para mudar para interface meu-pedido.

`void`

[lidarBotaoSair\(javafx.event.ActionEvent e\)](#)
MÃ©todo para sair da conta e mudar para tela inicial.

```
void  
lidarBotaoVoltar(javaafx.event.ActionEvent e)  
MÃ©todo para voltar para interface pagamento.
```

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

◆ tf_rua

```
private javafx.scene.control.TextField tf_rua
```

◆ tf_bairro

```
private javafx.scene.control.TextField tf_bairro
```

◆ tf_numero

```
private javafx.scene.control.TextField tf_numero
```

◆ erro

```
private javafx.scene.control.Label erro
```

• Constructor Details

◆ AlterarEnderecoController

```
public AlterarEnderecoController()
```

• Method Details

◆ lidarBotaoMeuPedido

```
public void lidarBotaoMeuPedido(javaafx.event.ActionEvent e)  
MÃ©todo para mudar para interface meu-pedido.  
Parameters:  
    e - ActionEvent - botÃ£o Meu Pedido
```

◆ lidarBotaoCarrinho

```
public void lidarBotaoCarrinho(javaafx.event.ActionEvent e)  
MÃ©todo para mudar para interface carrinho.  
Parameters:  
    e - ActionEvent - botÃ£o Carrinho
```

◆ lidarBotaoCardapio

```
public void lidarBotaoCardapio(javaafx.event.ActionEvent e)  
MÃ©todo para mudar para interface cardapio.  
Parameters:
```

e - ActionEvent - botão Cardapio

♦ **lidarBotaoSair**

```
public void lidarBotaoSair(javafx.event.ActionEvent e)
```

MÃ©todo para sair da conta e mudar para tela inicial.

Parameters:

e - ActionEvent - botão Sair

♦ **lidarBotaoVoltar**

```
public void lidarBotaoVoltar(javafx.event.ActionEvent e)
```

MÃ©todo para voltar para interface pagamento.

Parameters:

e - ActionEvent - botão Voltar

♦ **lidarBotaoConfirmar**

```
public void lidarBotaoConfirmar(javafx.event.ActionEvent e)
```

MÃ©todo para verificar se todos os campos do endereço foram preenchidos e, então, chamar a interface de pagamento.

Parameters:

e - ActionEvent - botão Confirmar

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class CardapioController

[java.lang.Object](#)

com.example.pedidoentrega.CardapioController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class CardapioController extends [Object](#) implements javafx.fxml.Initializable
 Classe Controller da interface cardapio.

Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao_addMilkshake

private javafx.scene.control.Button
botao_addPicole

private javafx.scene.control.Button
botao_addSorvete

private javafx.scene.control.RadioButton
botao_agua

private javafx.scene.control.Button
botao_carrinho

private javafx.scene.control.RadioButton
botao_G

private javafx.scene.control.RadioButton
botao_leite

private javafx.scene.control.RadioButton
botao_M

private javafx.scene.control.RadioButton
botao_P

private javafx.scene.control.Button
botao_voltar

private javafx.scene.control.Label
erroMilkshake

private javafx.scene.control.Label
erroPicole

private javafx.scene.control.Label
erroSorvete

private Estoque
estoque

private javafx.scene.control.Label
label_precoMilkshake

•

```
private javafx.scene.control.Label  
label_precoPicole  
  
private javafx.scene.control.Label  
label_precoSorvete  
  
private javafx.scene.control.ChoiceBox<String>  
milkshakeChoiceBox  
  
private javafx.scene.control.Spinner<Integer>  
milkshakeSpinner  
  
private javafx.scene.layout.Pane  
notificacaoPane  
  
private javafx.scene.control.ChoiceBox<String>  
picoleChoiceBox  
  
private javafx.scene.control.Spinner<Integer>  
picoleSpinner  
  
private javafx.scene.control.ChoiceBox<String>  
saboresChoiceBox  
  
private javafx.scene.control.Spinner<Integer>  
sorveteSpinner  
  
private javafx.scene.control.ComboBox<String>  
sorveteTipo
```

• Constructor Summary

Constructors

Constructor

Description

CardapioController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

String

getBasePicole()

MÃ©todo para pegar a base do picolÃ©.

String

getTamanhoMilkshake()

MÃ©todo para pegar o tamanho do milkshake.

void

initialize(URL url, ResourceBundle resourceBundle)

MÃ©todo de inicializaÃ§Ã£o, pega todas as informaÃ§Ãµes dos produtos: sabores, tamanhos, tipos, etc.

void

lidarBotaoAddMilkshake(javafx.event.ActionEvent e)

MÃ©todo para adicionar um milkshake no carrinho, caso todos os campos necessÃ¡rios estejam selecionados.

void

lidarBotaoAddPicole(javafx.event.ActionEvent e)

MÃ©todo para adicionar um picolÃ© no carrinho, caso todos os campos necessÃ¡rios estejam selecionados.

void

lidarBotaoAddSorvete(javafx.event.ActionEvent e)

MÃ©todo para adicionar um sorvete no carrinho, caso todos os campos necessÃ¡rios estejam selecionados.

void

lidarBotaoCarrinho(javafx.event.ActionEvent e)

MÃ©todo para entrar na interface do carrinho, caso o usuÃ¡rio esteja logado, ou pede login caso contrÃ¡rio.

void

lidarBotaoMilkshake()

MÃ©todo para mostrar o preÃ§o do milkshake.

void

lidarBotaoPicole()

MÃ©todo para mostrar o preÃ§o do picolÃ©.

void

lidarBotaoSorvete()

MÃ©todo para mostrar o preÃ§o do sorvete.

void

lidarBotaoVoltar(javafx.event.ActionEvent e)

MÃ©todo para sair do carrinho, volta a tela do usuÃ¡rio caso esteja logado, ou para tela de login caso contrÃ¡rio.

void

setVisible()

MÃ©todo para aparecer mensagem avisando que foi adicionado ao carrinho.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

• Field Details

♦ saboresChoiceBox

private javafx.scene.control.ChoiceBox<String> saboresChoiceBox

♦ **milkshakeChoiceBox**

```
private javafx.scene.control.ChoiceBox<String> milkshakeChoiceBox
```

♦ **picoleChoiceBox**

```
private javafx.scene.control.ChoiceBox<String> picoleChoiceBox
```

♦ **sorveteTipo**

```
private javafx.scene.control.ComboBox<String> sorveteTipo
```

♦ **sorveteSpinner**

```
private javafx.scene.control.Spinner<Integer> sorveteSpinner
```

♦ **milkshakeSpinner**

```
private javafx.scene.control.Spinner<Integer> milkshakeSpinner
```

♦ **picoleSpinner**

```
private javafx.scene.control.Spinner<Integer> picoleSpinner
```

♦ **botao_addSorvete**

```
private javafx.scene.control.Button botao_addSorvete
```

♦ **botao_addMilkshake**

```
private javafx.scene.control.Button botao_addMilkshake
```

♦ **botao_addPicole**

```
private javafx.scene.control.Button botao_addPicole
```

♦ **botao_P**

```
private javafx.scene.control.RadioButton botao_P
```

♦ **botao_M**

```
private javafx.scene.control.RadioButton botao_M
```

♦ **botao_G**

```
private javafx.scene.control.RadioButton botao_G
```

♦ **botao_agua**

```
private javafx.scene.control.RadioButton botao_agua
```

♦ **botao_leite**

```
private javafx.scene.control.RadioButton botao_leite
```

♦ **botao_voltar**

```
private javafx.scene.control.Button botao_voltar
```

♦ **botao_carrinho**

```
private javafx.scene.control.Button botao_carrinho
```

♦ **label_precoPicole**

```
private javafx.scene.control.Label label_precoPicole
```

♦ **label_precoMilkshake**

```
private javafx.scene.control.Label label_precoMilkshake
```

♦ **label_precoSorvete**

```
private javafx.scene.control.Label label_precoSorvete
```

♦ **notificacaoPane**

```
private javafx.scene.layout.Pane notificacaoPane
```

♦ **erroSorvete**

```
private javafx.scene.control.Label erroSorvete
```

♦ **erroMilkshake**

```
private javafx.scene.control.Label erroMilkshake
```

♦ **erroPicole**

```
private javafx.scene.control.Label erroPicole
```

♦ **estoque**

```
private Estoque estoque
```

• **Constructor Details**

♦ **CardapioController**

```
public CardapioController()
```

• **Method Details**

♦ **initialize**

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

Método de inicialização, pega todas as informações dos produtos: sabores, tamanhos, tipos, etc.

Specified by:

```
initialize in interface javafx.fxml.Initializable
```

Parameters:

url -

resourceBundle -

♦ **lidarBotaoAddSorvete**

public void lidarBotaoAddSorvete(javafx.event.ActionEvent e)

MÃ©todo para adicionar um sorvete no carrinho, caso todos os campos necessÃ¡rios estejam selecionados.

Parameters:

e - ActionEvent - botÃ£o Adicionar ao Carrinho

♦ **lidarBotaoAddMilkshake**

public void lidarBotaoAddMilkshake(javafx.event.ActionEvent e)

MÃ©todo para adicionar um milkshake no carrinho, caso todos os campos necessÃ¡rios estejam selecionados.

Parameters:

e - ActionEvent - botÃ£o Adicionar ao Carrinho

♦ **lidarBotaoAddPicole**

public void lidarBotaoAddPicole(javafx.event.ActionEvent e)

MÃ©todo para adicionar um picolÃ© no carrinho, caso todos os campos necessÃ¡rios estejam selecionados.

Parameters:

e - ActionEvent - botÃ£o Adicionar ao Carrinho

♦ **lidarBotaoCarrinho**

public void lidarBotaoCarrinho(javafx.event.ActionEvent e)

MÃ©todo para entrar na interface do carrinho, caso o usuÃ¡rio esteja logado, ou pede login caso contrÃ¡rio.

Parameters:

e - ActionEvent - botÃ£o Carrinho

♦ **lidarBotaoVoltar**

public void lidarBotaoVoltar(javafx.event.ActionEvent e) throws IOException

MÃ©todo para sair do carrinho, volta a tela do usuÃ¡rio caso esteja logado, ou para tela de login caso contrÃ¡rio.

Parameters:

e - ActionEvent - botÃ£o Voltar

Throws:

IOException

♦ **lidarBotaoMilkshake**

public void lidarBotaoMilkshake()

MÃ©todo para mostrar o preÃ§o do milkshake.

♦ **getTamanhoMilkshake**

public String getTamanhoMilkshake()

MÃ©todo para pegar o tamanho do milkshake.

Returns:

retorna otamanho do milkshake (P,M,G)

♦ **lidarBotaoPicole**

```
public void lidarBotaoPicole()  
MÃ©todo para mostrar o preÃ§o do picolÃ©.
```

♦ **getBasePicole**

```
public String getBasePicole()  
MÃ©todo para pegar a base do picolÃ©.  
Returns:  
    retorna a base do picolÃ©
```

♦ **lidarBotaoSorvete**

```
public void lidarBotaoSorvete()  
MÃ©todo para mostrar o preÃ§o do sorvete.
```

♦ **setVisible**

```
public void setVisible()  
MÃ©todo para aparecer mensagem avisando que foi adicionado ao carrinho.
```

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Carrinho

[java.lang.Object](#)

[com.example.pedidoentrega.Carrinho](#)

public class Carrinho extends [Object](#)
Classe Carrinho define o status do pedido.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
codigo

private Endereco
endereco

private float
precoTotal

protected int
status

private static final int
statusCancelado

private static final int
statusConcluido

private static int
statusConfirmado

private static final int
statusEntregando

private static int
statusPendente

private static float
taxaEntrega

• Constructor Summary

Constructors

Constructor

Description

Carrinho(ArrayList<Pedido<?>> listaPedidos, Endereco endereco)

Construtor do carrinho, recebe uma lista de pedidos e endereço

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

calcularPrecoTotal(ArrayList<Pedido<?>> listaPedidos)

MÃ©todo para calcular o preÃ§o total do pedido.

void

cancelar()

MÃ©todo para mudar o status para Cancelado, caso seja possÃ-vel.

void

concluir()

MÃ©todo para mudar o status para ConcluÃ-do, caso seja possÃ-vel.

void

confirmar()

MÃ©todo para mudar o status para Confirmado.

void

entregar()

MÃ©todo para entregar, caso o pedido nÃo tenha sido cancelado.

String

getCodigo()

MÃ©todo para acessar o cÃdigo.

Endereco

getEndereco()

MÃ©todo para acessar o endereÃo.

float

getPrecoTotal()

MÃ©todo para acessar o preÃ§o total.

int

getStatus()

MÃ©todo para acessar o status.

void

setEndereco(Endereco endereco)

MÃ©todo para modificar o endereÃo.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ precoTotal

private float precoTotal

♦ taxaEntrega

private static float taxaEntrega

♦ statusPendente

private static int statusPendente

♦ **statusConfirmado**

private static int statusConfirmado

♦ **statusEntregando**

private static final int statusEntregando

See Also:

- [Constant Field Values](#)

♦ **statusConcluido**

private static final int statusConcluido

See Also:

- [Constant Field Values](#)

♦ **statusCancelado**

private static final int statusCancelado

See Also:

- [Constant Field Values](#)

♦ **status**

protected int status

♦ **codigo**

private [String](#) codigo

♦ **endereco**

private [Endereco](#) endereco

• **Constructor Details**

♦ **Carrinho**

public Carrinho([ArrayList](#)<[Pedido](#)<?>> listaPedidos, [Endereco](#) endereco)

Construtor do carrinho, recebe uma lista de pedidos e endereço

Parameters:

listaPedidos - [ArrayList](#) - lista de pedidos

endereco - [Endereco](#) - endereço

• **Method Details**

♦ **confirmar**

public void confirmar()

Método para mudar o status para Confirmado.

♦ **entregar**

```
public void entregar()
```

Método para entregar, caso o pedido não tenha sido cancelado.

♦ **concluir**

```
public void concluir()
```

Método para mudar o status para Concluído, caso seja possível.

♦ **cancelar**

```
public void cancelar()
```

Método para mudar o status para Cancelado, caso seja possível.

♦ **calcularPrecoTotal**

```
public void calcularPrecoTotal(ArrayList<Pedido>> listaPedidos)
```

Método para calcular o preço total do pedido.

Parameters:

listaPedidos - ArrayList - lista de pedidos

♦ **getStatus**

```
public int getStatus()
```

Método para acessar o status.

Returns:

retorna o status

♦ **getPrecoTotal**

```
public float getPrecoTotal()
```

Método para acessar o preço total.

Returns:

retorna o preço total

♦ **getCodigo**

```
public String getCodigo()
```

Método para acessar o código.

Returns:

retorna o código

♦ **getEndereco**

```
public Endereco getEndereco()
```

Método para acessar o endereço.

Returns:

retorna o endereço

♦ **setEndereco**

```
public void setEndereco(Endereco endereco)
```

Método para modificar o endereço.

Parameters:

endereco - Endereco - endereço

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class CarrinhoController

[java.lang.Object](#)

com.example.pedidoentrega.CarrinhoController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class CarrinhoController extends [Object](#) implements `javafx.fxml.Initializable`
 Classe controller da interface carrinho.

• Field Summary

Fields

Modifier and Type

Field

Description

```
private javafx.scene.control.Button  
botao_cardapio
```

```
private javafx.scene.control.Button  
botao_finalizar
```

```
private javafx.scene.control.Button  
botao_meuPedido
```

```
private javafx.scene.control.Button  
botao_remove
```

```
private javafx.scene.control.Button  
botao_sair
```

```
private Cliente  
cliente
```

```
private ArrayList<Pedido>  
listaPedidos
```

```
private javafx.scene.control.TableColumn<Pedido, String>  
produto
```

```
private javafx.scene.control.TableColumn<Pedido, Integer>  
quantidade
```

```
private javafx.scene.control.TableView<Pedido>  
tabela_pedidos
```

```
private javafx.scene.control.TableColumn<Pedido, Float>  
total
```

```
private javafx.scene.control.TableColumn<Pedido, Float>  
unidade
```

• Constructor Summary

Constructors

Constructor

Description

```
CarrinhoController()
```

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[initialize\(URL url, ResourceBundle resourceBundle\)](#)

MÃ©todo para inicializar a tabela do carrinho.

void

[lidarBotaoCardapio\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface cardapio.

void

[lidarBotaoFinalizar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface pagamento.

void

[lidarBotaoMeuPedido\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface meu-pedido.

void

[lidarBotaoRemover\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para remover pedido do carrinho.

void

[lidarBotaoSair\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface inicial.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

• Field Details

♦ **botao_meuPedido**

private javafx.scene.control.Button botao_meuPedido

♦ **botao_cardapio**

private javafx.scene.control.Button botao_cardapio

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_finalizar**

private javafx.scene.control.Button botao_finalizar

♦ **botao_remover**

private javafx.scene.control.Button botao_remover

♦ **tabela_pedidos**

```
private javafx.scene.control.TableView<Pedido> tabela_pedidos
```

♦ **produto**

```
private javafx.scene.control.TableColumn<Pedido,String> produto
```

♦ **quantidade**

```
private javafx.scene.control.TableColumn<Pedido,Integer> quantidade
```

♦ **unidade**

```
private javafx.scene.control.TableColumn<Pedido,Float> unidade
```

♦ **total**

```
private javafx.scene.control.TableColumn<Pedido,Float> total
```

♦ **cliente**

```
private Cliente cliente
```

♦ **listaPedidos**

```
private ArrayList<Pedido<?>> listaPedidos
```

• **Constructor Details**

♦ **CarrinhoController**

```
public CarrinhoController()
```

• **Method Details**

♦ **lidarBotaoMeuPedido**

```
public void lidarBotaoMeuPedido(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para interface meu-pedido.

Parameters:

e - ActionEvent - botÃ£o Meu Pedido

♦ **lidarBotaoCardapio**

```
public void lidarBotaoCardapio(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para interface cardapio.

Parameters:

e - ActionEvent - botÃ£o Cardapio

♦ **lidarBotaoSair**

```
public void lidarBotaoSair(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para interface inicial.

Parameters:

e - ActionEvent - botÃ£o Sair

♦ **lidarBotaoRemover**

```
public void lidarBotaoRemover(javafx.event.ActionEvent e)
```

MÃ©todo para remover pedido do carrinho.

Parameters:

e - ActionEvent - botÃ£o Remover

♦ **lidarBotaoFinalizar**

```
public void lidarBotaoFinalizar(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para interface pagamento.

Parameters:

e - ActionEvent - botÃ£o Finalizar

♦ **initialize**

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

MÃ©todo para inicializar a tabela do carrinho.

Specified by:

```
initialize in interface javafx.fxml.Initializable
```

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Casquinha

[java.lang.Object](#)

[com.example.pedidoentrega.Sorvete](#)

[com.example.pedidoentrega.Casquinha](#)

public class Casquinha extends [Sorvete](#)

Classe Casquinha herda a classe abstrata Sorvete, guarda os dados do sorvete do tipo casquinha.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
tamanho

• Constructor Summary

Constructors

Constructor

Description

Casquinha(String sabor, Estoque estoque, String tamanho)
Construtor da Casquinha.

• Method Summary

Methods inherited from class com.example.pedidoentrega.Sorvete

getPreco, getSabor, setPreco, setPreco

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ tamanho

private String tamanho

• Constructor Details

♦ Casquinha

public Casquinha(String sabor, Estoque estoque, String tamanho)
Construtor da Casquinha.

Parameters:

sabor - String - sabor

estoque - Estoque - quantidade no estoque

tamanho - String - tamanho (casquinha, cascÃ£o)

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Cliente

[java.lang.Object](#)

[com.example.pedidoentrega.Usuario](#)

[com.example.pedidoentrega.Cliente](#)

public class Cliente extends [Usuario](#)

Classe Cliente herda a classe abstrata Usuario, possui métodos para cancelar/pagar/fazer pedidos e alterar endereço.

• Field Summary

Fields

Modifier and Type

Field

Description

private Carrinho
carrinho

private Endereco
endereco

private ArrayList<Carrinho>
meusPedidos

private String
nome

private ArrayList<Pedido<?>>
pedidos

private String
telefone

• Constructor Summary

Constructors

Constructor

Description

Cliente(String nome, String email, String telefone, String senha,
Endereco endereco)

Construtor do Cliente.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

adicionarPedido(Sorvete sorvete, int quantidade)

MÃ©todo para adicionar pedido a lista de pedidos do cliente.

void

alterarEndereco(Endereco novoEndereco)

MÃ©todo para alterar o endereÃ§o do cliente.

void

cancelar(Carrinho carrinho)

MÃ©todo para cancelar o pedido.

```

void
finalizarPedido()
MÃ©todo para finalizar o pedido.
Carrinho
getCarrinho()
MÃ©todo para acessar o carrinho do cliente.
Endereco
getEndereco()
MÃ©todo para acessar o endereÃ§o do cliente.
ArrayList<Carrinho>
getMeusPedidos()
MÃ©todo para acessar o carrinho com a lista de pedidos do cliente.
String
getNome()
MÃ©todo para acessar o nome do cliente.
ArrayList<Pedido<?>>
getPedidos()
MÃ©todo para acessar a lista de pedidos.
String
getTelefone()
MÃ©todo para acessar o telefone do cliente.
void
pagarEFinalizar()
MÃ©todo para confirmar o pedido e fazer o pagamento.
void
removerPedido(Pedido pedido)
MÃ©todo para remover pedido da lista de pedidos do cliente.
Methods inherited from class com.example.pedidoentrega.Usuario

```

getEmail, getSenha

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

◆ nome

private String nome

◆ telefone

private String telefone

◆ endereco

private Endereco endereco

♦ **carrinho**

private Carrinho carrinho

♦ **meusPedidos**

private ArrayList<Carrinho> meusPedidos

♦ **pedidos**

private ArrayList<Pedido<?>> pedidos

• **Constructor Details**

♦ **Cliente**

public Cliente(String nome, String email, String telefone, String senha, Endereco endereco)

Construtor do Cliente.

Parameters:

nome - String - nome do cliente

email - String - email do cliente

telefone - String - telefone do cliente

senha - String - senha do cliente

endereco - Endereco - endereço do cliente

• **Method Details**

♦ **alterarEndereco**

public void alterarEndereco(Endereco novoEndereco)

Método para alterar o endereço do cliente.

Parameters:

novoEndereco - Endereco - novo endereço

♦ **getTelefone**

public String getTelefone()

Método para acessar o telefone do cliente.

Returns:

retorna o telefone do cliente

♦ **getNome**

public String getNome()

Método para acessar o nome do cliente.

Returns:

retorna o nome do cliente

♦ **getEndereco**

public Endereco getEndereco()

Método para acessar o endereço do cliente.

Returns:

retorna o endereço do cliente.

♦ **getCarrinho**

public Carrinho getCarrinho()

MÃ©todo para acessar o carrinho do cliente.

Returns:

retorna o carrinho do cliente

♦ **getPedidos**

public ArrayList<Pedido<?>> getPedidos()

MÃ©todo para acessar a lista de pedidos.

Returns:

retorna a lista de pedidos

♦ **getMeusPedidos**

public ArrayList<Carrinho> getMeusPedidos()

MÃ©todo para acessar o carrinho com a lista de pedidos do cliente.

Returns:

retorna o carrinho com a lista de pedidos do cliente

♦ **pagarEFinalizar**

public void pagarEFinalizar()

MÃ©todo para confirmar o pedido e fazer o pagamento.

♦ **cancelar**

public void cancelar(Carrinho carrinho)

MÃ©todo para cancelar o pedido.

Parameters:

carrinho - Carrinho - carrinho com os pedidos do cliente

♦ **adicionarPedido**

public void adicionarPedido(Sorvete sorvete, int quantidade)

MÃ©todo para adicionar pedido a lista de pedidos do cliente.

Parameters:

sorvete - Sorvete - sorvete

quantidade - int - quantidade de itens

♦ **finalizarPedido**

public void finalizarPedido()

MÃ©todo para finalizar o pedido.

♦ **removerPedido**

public void removerPedido(Pedido pedido)

MÃ©todo para remover pedido da lista de pedidos do cliente.

Parameters:

pedido - Pedido - pedido a ser removido

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- Detail:
- [Field |](#)
- [Constr |](#)
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Copinho

[java.lang.Object](#)

[com.example.pedidoentrega.Sorvete](#)

[com.example.pedidoentrega.Copinho](#)

public class Copinho extends [Sorvete](#)

Classe Copinho herda a classe abstrata Sorvete, guarda os dados do sorvete de copinho.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
tamanho

• Constructor Summary

Constructors

Constructor

Description

Copinho(String sabor, Estoque estoque, String tamanho)
Contrutor do Copinho.

• Method Summary

Methods inherited from class com.example.pedidoentrega.Sorvete

getPreco, getSabor, setPreco, setPreco

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ tamanho

private String tamanho

• Constructor Details

♦ Copinho

public Copinho(String sabor, Estoque estoque, String tamanho)

Contrutor do Copinho.

Parameters:

sabor - String - sabor

estoque - Estoque - quantidade no estoque

tamanho - String - tamanho (P,M,G)

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class DisplaySorveteria

[java.lang.Object](#)

com.example.pedidoentrega.DisplaySorveteria

public class DisplaySorveteria extends [Object](#)

Classe DisplaySorveteria para fazer login, ler e atualizar arquivos.

• Field Summary

Fields

Modifier and Type

Field

Description

private final SorveteriaSistema
sorveteria

• Constructor Summary

Constructors

Constructor

Description

DisplaySorveteria()
Construtor para ler os arquivos.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

atualizarArquivoEntregadores(javafx.collections.ObservableList<Entregador>
MÃ©todo para atualizar o arquivo de entregadores.

void

atualizarArquivoEstoque(javafx.scene.control.TableView<ItemEstoque> tabela
MÃ©todo para atualizar o arquivo de estoque.

void

cadastroGui(javafx.event.ActionEvent e, String nome, String email,
String telefone, String senha)

MÃ©todo para cadastrar novo entregador.

void

cadastroGui(javafx.event.ActionEvent e, String nome, String email,
String telefone, String senha, String rua, String bairro,
String num)

MÃ©todo para cadastrar novo cliente.

ClienteclienteLogado()

MÃ©todo para acessar o cliente logado.

void

desloggedGui()

MÃ©todo para deslogar da conta.

EntregadorentregadorLogado()

MÃ©todo para acessar o entregador logado.

Administrador

Administrador

[getAdministradorSorveteria\(\)](#)

MÃ©todo para acessar o administrador da sorveteria.

[SorveteriaSistema](#)

[getSorveteria\(\)](#)

MÃ©todo para acessar o sistema da sorveteria.

void

[lerClientes\(File arquivoClientes\)](#)

MÃ©todo para ler arquivo de clientes.

void

[lerEntregadores\(File arquivoEntregadores\)](#)

MÃ©todo para ler arquivo de entregadores.

void

[lerEstoque\(File arquivoEstoque\)](#)

MÃ©todo para ler arquivo do estoque.

void

[lerSabores\(File arquivoSabores\)](#)

MÃ©todo para ler arquivo de sabores.

void

[loggedGui\(javafx.event.ActionEvent e\)](#)

MÃ©todo para entrar na interface certa (cliente, entregador, administrador) apÃ³s o login.

void

[loginGui\(javafx.event.ActionEvent e, String email, String senha\)](#)

MÃ©todo para fazer o login do usuÃ¡rio.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

• Field Details

◆ **sorveteria**

private final [SorveteriaSistema](#) sorveteria

• Constructor Details

◆ **DisplaySorveteria**

public DisplaySorveteria()
Construtor para ler os arquivos.

• Method Details

◆ **lerSabores**

public void lerSabores([File](#) arquivoSabores) throws [Exception](#)
MÃ©todo para ler arquivo de sabores.
Parameters:
arquivoSabores - File - arquivo de sabores

Throws:

Exception

♦ lerEstoque

public void lerEstoque(File arquivoEstoque) throws Exception

MÃ©todo para ler arquivo do estoque.

Parameters:

arquivoEstoque - File - arquivo do estoque

Throws:

Exception

♦ lerClientes

public void lerClientes(File arquivoClientes) throws Exception

MÃ©todo para ler arquivo de clientes.

Parameters:

arquivoClientes - File - arquivo de clientes

Throws:

Exception

♦ lerEntregadores

public void lerEntregadores(File arquivoEntregadores) throws Exception

MÃ©todo para ler arquivo de entregadores.

Parameters:

arquivoEntregadores - File - arquivo de entregadores

Throws:

Exception

♦ getSoreteria

public SoreteriaSistema getSoreteria()

MÃ©todo para acessar o sistema da soreteria.

Returns:

retorna o sistema da soreteria

♦ loginGui

public void loginGui(javaafx.event.ActionEvent e, String email, String senha) throws

UsuarioInexistenteException

MÃ©todo para fazer o login do usuÃ¡rio.

Parameters:

e - ActionEvent - logar

email - String - email do usuÃ¡rio

senha - String - senha do usuÃ¡rio

Throws:

UsuarioInexistenteException

♦ cadastroGui

public void cadastroGui(javaafx.event.ActionEvent e, String nome, String email, String telefone, String senha, String rua, String bairro, String num) throws Exception

MÃ©todo para cadastrar novo cliente.

Parameters:

e - ActionEvent - cadastrar

Administrador

nome - String - nome do usuário
email - String - email do usuário
telefone - String - telefone do usuário
senha - String - senha do usuário
rua - String - rua do usuário
bairro - String - bairro do usuário
num - String - número da residência do usuário

Throws:

Exception

♦ cadastroGui

public void cadastroGui(javafx.event.ActionEvent e, String nome, String email, String telefone, String senha) throws Exception

Método para cadastrar novo entregador.

Parameters:

e - e ActionEvent - cadastrar
nome - String - nome do entregador
email - String - email do entregador
telefone - String - telefone do entregador
senha - String - senha do entregador

Throws:

Exception

♦ loggedGui

public void loggedGui(javafx.event.ActionEvent e)

Método para entrar na interface certa (cliente, entregador, administrador) após o login.

Parameters:

e - ActionEvent login

♦ desloggedGui

public void desloggedGui()

Método para deslogar da conta.

♦ clienteLogado

public Cliente clienteLogado()

Método para acessar o cliente logado.

Returns:

retorna a conta do cliente logado

♦ entregadorLogado

public Entregador entregadorLogado()

Método para acessar o entregador logado.

Returns:

retorna a conta do entregador logado

♦ getAdministradorSoreteria

public Administrador getAdministradorSoreteria()

Método para acessar o administrador da soreteria.

Returns:

retorna a conta do administrador

♦ **atualizarArquivoEntregadores**

```
public void atualizarArquivoEntregadores(javafx.collections.ObservableList<Entregador> items)
```

MÃ©todo para atualizar o arquivo de entregadores.

Parameters:

`items` - Lista de entregadores

♦ **atualizarArquivoEstoque**

```
public void atualizarArquivoEstoque(javafx.scene.control.TableView<ItemEstoque> tabela_estoque)
```

MÃ©todo para atualizar o arquivo de estoque.

Parameters:

`tabela_estoque` - Tabela de estoque

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Endereco

[java.lang.Object](#)

com.example.pedidoentrega.Endereco

public class Endereco extends [Object](#)

Classe para guardar os dados do endereço dos clientes.

• Field Summary

Fields

Modifier and Type

Field

Description

`private String
bairro`

`private String
num`

`private String
rua`

• Constructor Summary

Constructors

Constructor

Description

`Endereco(String rua, String bairro, String num)`

Construtor do Endereco.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

`String`

`getBairro()`

MÃ©todo para acessar o bairro.

`String`

`getNum()`

MÃ©todo para acessar o nÃºmero.

`String`

`getRua()`

MÃ©todo para acessar a rua.

`String`

`toString()`

MÃ©todo para escrever o endereÃ§o na interface.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait`

• Field Details

♦ rua

private String rua

♦ bairro

private String bairro

♦ num

private String num

• Constructor Details

♦ Endereco

public Endereco(String rua, String bairro, String num)

Construtor do Endereco.

Parameters:

rua - String - rua

bairro - String - bairro

num - String - número

• Method Details

♦ getRua

public String getRua()

Método para acessar a rua.

Returns:

retorna a rua

♦ getBairro

public String getBairro()

Método para acessar o bairro.

Returns:

retorna o bairro

♦ getNum

public String getNum()

Método para acessar o número.

Returns:

retorna o número

♦ toString

public String toString()

Método para escrever o endereço na interface.

Overrides:

Administrador

toString in class Object

Returns:

retorna rua, bairro, numero

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Entregador

[java.lang.Object](#)

[com.example.pedidoentrega.Usuario](#)

[com.example.pedidoentrega.Entregador](#)

public class Entregador extends [Usuario](#)

Classe Entregador herda a classe abstrata Usuario, guarda os dados dos entregadores.

• Field Summary

Fields

Modifier and Type

Field

Description

private ArrayList<Carrinho>
carrinhos

private String
nome

private String
telefone

• Constructor Summary

Constructors

Constructor

Description

Entregador(String email, String senha, String nome, String telefone)

Construtor do Entregador.

• Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

void

adicionarCarrinho(Carrinho carrinho)

MÃ©todo para adicionar carrinho de pedidos na lista.

ArrayList<Carrinho>

getCarrinhos()

MÃ©todo para acessar a lista de carrinhos com os pedidos.

String

getNome()

MÃ©todo para acessar o nome de entregador.

String

getTelefone()

MÃ©todo para acessar o telefone de entregador.

void

removerCarrinho(Carrinho carrinho)

MÃ©todo para remover carrinho de pedidos da lista.

String

toString()

MÃ©todo para escrever os dados dos entregadores na interface do administrador com a lista de entregadores

Methods inherited from class `com.example.pedidoentrega.Usuario`

`getEmail`, `getSenha`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
`wait`, `wait`, `wait`

• Field Details

◆ `nome`

private `String` `nome`

◆ `telefone`

private `String` `telefone`

◆ `carrinhos`

private `ArrayList<Carrinho>` `carrinhos`

• Constructor Details

◆ `Entregador`

public `Entregador(String email, String senha, String nome, String telefone)`

Construtor do `Entregador`.

Parameters:

`email` - `String` - email do entregador

`senha` - `String` - senha do entregador

`nome` - `String` - nome do entregador

`telefone` - `String` - telefone do entregador

• Method Details

◆ `adicionarCarrinho`

public void `adicionarCarrinho(Carrinho carrinho)`

MÃ©todo para adicionar carrinho de pedidos na lista.

Parameters:

`carrinho` - `Carrinho` - carrinho de pedidos

◆ `removerCarrinho`

public void `removerCarrinho(Carrinho carrinho)`

MÃ©todo para remover carrinho de pedidos da lista.

Parameters:

`carrinho` - `Carrinho` - carrinho de pedidos

♦ **getNome**

```
public String getNome()  
MÃ©todo para acessar o nome de entregador.  
Returns:  
    retorna o nome do entregador
```

♦ **getTelefone**

```
public String getTelefone()  
MÃ©todo para acessar o telefone de entregador.  
Returns:  
    retorna o telefone do entregador
```

♦ **getCarrinhos**

```
public ArrayList<Carrinho> getCarrinhos()  
MÃ©todo para acessar a lista de carrinhos com os pedidos.  
Returns:  
    retorna a lista de carrinhos
```

♦ **toString**

```
public String toString()  
MÃ©todo para escrever os dados dos entregadores na interface do administrador com a lista de entregadores  
Overrides:  
    toString in class Object  
Returns:  
    retorna nome / email / telefone do entregador
```


JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class EntregadoresController

[java.lang.Object](#)

com.example.pedidoentrega.EntregadoresController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class EntregadoresController extends [Object](#) implements `javafx.fxml.Initializable`
 Classe controller da interface com a lista de entregadores do administrador.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao_cadastrar

private javafx.scene.control.Button
botao_demitir

private javafx.scene.control.Button
botao_estoque

private javafx.scene.control.Button
botao_listaPedidos

private javafx.scene.control.Button
botao_mudarCardapio

private javafx.scene.control.Button
botao_sair

private javafx.scene.control.ListView<Entregador>
lista_entregadores

private SorveteriaSistema
sorveteria

• Constructor Summary

Constructors

Constructor

Description

EntregadoresController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

initialize(URL url, ResourceBundle resourceBundle)

MÃ©todo para inicializar a lista de entregadores.

void

Administrador

```
lidarBotaoCadastrar(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a interface sign-up-entregador.  
void  
lidarBotaoDemitir(javafx.event.ActionEvent e)  
MÃ©todo para demitir um entregador.  
void  
lidarBotaoEstoque(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a interface estoque.  
void  
lidarBotaoListaPedidos(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a interface logged-administrador, com a lista de pedidos da sorveteria.  
void  
lidarBotaoMudarCardapio(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a interface mudar-cardapio.  
void  
lidarBotaoSair(javafx.event.ActionEvent e)  
MÃ©todo para sair da conta do administrador e voltar para a tela inicial.
```

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_cadastrar**

private javafx.scene.control.Button botao_cadastrar

♦ **botao_demitir**

private javafx.scene.control.Button botao_demitir

♦ **botao_listaPedidos**

private javafx.scene.control.Button botao_listaPedidos

♦ **botao_mudarCardapio**

private javafx.scene.control.Button botao_mudarCardapio

♦ **botao_estoque**

private javafx.scene.control.Button botao_estoque

♦ **lista_entregadores**

private javafx.scene.control.ListView<Entregador> lista_entregadores

♦ **sorveteria**

private SorveteriaSistema sorveteria

• **Constructor Details**

♦ **EntregadoresController**

public EntregadoresController()

• **Method Details**

♦ **lidarBotaoSair**

public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException
MÃ©todo para sair da conta do administrador e voltar para a tela inicial.

Parameters:

e - ActionEvent - sair

Throws:

IOException

♦ **lidarBotaoCadastrar**

public void lidarBotaoCadastrar(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface sign-up-entregador.

Parameters:

e - ActionEvent - cadastrar entregador

♦ **lidarBotaoListaPedidos**

public void lidarBotaoListaPedidos(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface logged-administrador, com a lista de pedidos da sorveteria.

Parameters:

e -

♦ **lidarBotaoMudarCardapio**

public void lidarBotaoMudarCardapio(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface mudar-cardapio.

Parameters:

e - ActionEvent mudar cardÃ¡pio

♦ **lidarBotaoEstoque**

public void lidarBotaoEstoque(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface estoque.

Parameters:

e - ActionEvent botÃ£o estoque

♦ **lidarBotaoDemitir**

```
public void lidarBotaoDemitir(javafx.event.ActionEvent e)
```

MÃ©todo para demitir um entregador.

Parameters:

e -

♦ **initialize**

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

MÃ©todo para inicializar a lista de entregadores.

Specified by:

```
initialize in interface javafx.fxml.Initializable
```

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Estoque

[java.lang.Object](#)

com.example.pedidoentrega.Estoque

public class Estoque extends [Object](#)

Classe Estoque com métodos para seu gerenciamento.

• Field Summary

Fields

Modifier and Type

Field

Description

private int

quantCascao

private int

quantCasquinha

private int

quantCopinhoG

private int

quantCopinhoM

private int

quantCopinhoP

private int

quantCopoG

private int

quantCopoM

private int

quantCopoP

private int

quantPicoleAgua

private int

quantPicoleLeite

(package private) Set<String>

saboresSorvete

• Constructor Summary

Constructors

Constructor

Description

Estoque()

Construtor do Estoque.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[adicionarSaborSorvete](#) ([String](#) sabor)

MÃ©todo para adicionar sabor de sorvete na lista (Set) de sabores.

int

[getQuantCascao](#) ()

MÃ©todo para acessar a quantidade de cascÃ£o.

int

[getQuantCasquinha](#) ()

MÃ©todo para acessar a quantidade de casquinha.

int

[getQuantCopinhoG](#) ()

MÃ©todo para acessar a quantidade de copinho G.

int

[getQuantCopinhoM](#) ()

MÃ©todo para acessar a quantidade de copinho M.

int

[getQuantCopinhoP](#) ()

MÃ©todo para acessar a quantidade de copinho P.

int

[getQuantCopoG](#) ()

MÃ©todo para acessar a quantidade de copo G.

int

[getQuantCopoM](#) ()

MÃ©todo para acessar a quantidade de copo M.

int

[getQuantCopoP](#) ()

MÃ©todo para acessar a quantidade de copo P.

int

[getQuantPicoleAgua](#) ()

MÃ©todo para acessar a quantidade de picolÃ© a base de Ãgua.

int

[getQuantPicoleLeite](#) ()

MÃ©todo para acessar a quantidade de picolÃ© a base de leite.

[Set](#)<[String](#)>

[getSabores](#) ()

MÃ©todo para acessar a lista (Set) de sabores.

void

[setQuantCascao](#) (int quantidade)

MÃ©todo para alterar a quantidade de cascÃ£o do estoque.

void

[setQuantCasquinha](#) (int quantidade)

MÃ©todo para alterar a quantidade de casquinha do estoque.

void

[setQuantCopinhoG](#) (int quantidade)

Administrador

- **MÃ©todo** para alterar a quantidade de copinho G do estoque.

void

setQuantCopinhoM(int quantidade)

- **MÃ©todo** para alterar a quantidade de copinho M do estoque.

void

setQuantCopinhoP(int quantidade)

- **MÃ©todo** para alterar a quantidade de copinho P do estoque.

void

setQuantCopoG(int quantidade)

- **MÃ©todo** para alterar a quantidade de copo G do estoque.

void

setQuantCopoM(int quantidade)

- **MÃ©todo** para alterar a quantidade de copo M do estoque.

void

setQuantCopoP(int quantidade)

- **MÃ©todo** para alterar a quantidade de copo P do estoque.

void

setQuantidade(int quantCascao, int quantCasquinha, int quantCopoP, int quantCopoM, int quantCopoG, int quantCopinhoP, int quantCopinhoM, int quantCopinhoG, int quantPicoleAgua, int quantPicoleLeite)

- **MÃ©todo** para receber a quantidade do estoque.

void

setQuantPicoleAgua(int quantidade)

- **MÃ©todo** para alterar a quantidade de picolÃ© a base de Ã¡gua do estoque.

void

setQuantPicoleLeite(int quantidade)

- **MÃ©todo** para alterar a quantidade de picolÃ© a base de leite do estoque.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

• Field Details

♦ **saboresSorvete**

Set<String> saboresSorvete

♦ **quantCascao**

private int quantCascao

♦ **quantCasquinha**

private int quantCasquinha

♦ **quantCopoP**

private int quantCopoP

♦ **quantCopoM**

```
private int quantCopoM
```

♦ **quantCopoG**

```
private int quantCopoG
```

♦ **quantCopinhoP**

```
private int quantCopinhoP
```

♦ **quantCopinhoM**

```
private int quantCopinhoM
```

♦ **quantCopinhoG**

```
private int quantCopinhoG
```

♦ **quantPicoleAgua**

```
private int quantPicoleAgua
```

♦ **quantPicoleLeite**

```
private int quantPicoleLeite
```

• **Constructor Details**♦ **Estoque**

```
public Estoque()  
Construtor do Estoque.
```

• **Method Details**♦ **setQuantidade**

```
public void setQuantidade(int quantCascao, int quantCasquinha, int quantCopoP,  
int quantCopoM, int quantCopoG, int quantCopinhoP, int quantCopinhoM,  
int quantCopinhoG, int quantPicoleAgua, int quantPicoleLeite)  
Método para receber a quantidade do estoque.
```

Parameters:

```
quantCascao - int - quantidade de cascaõ  
quantCasquinha - int - quantidade de casquinha  
quantCopoP - int - quantidade de copo P  
quantCopoM - int - quantidade de copo M  
quantCopoG - int - quantidade de copo G  
quantCopinhoP - int - quantidade de copinho P  
quantCopinhoM - int - quantidade de copinho M  
quantCopinhoG - int - quantidade de copinho G  
quantPicoleAgua - int - quantidade de picolÃ© a base de Ãgua  
quantPicoleLeite - int - quantidade de picolÃ© a base de leite
```

♦ **getQuantCascao**

public int getQuantCascao()

MÃ©todo para acessar a quantidade de cascÃ£o.

Returns:

retorna a quantidade de cascÃ£o do estoque

♦ **getQuantCasquinha**

public int getQuantCasquinha()

MÃ©todo para acessar a quantidade de casquinha.

Returns:

retorna a quantidade de casquinha do estoque

♦ **getQuantCopoP**

public int getQuantCopoP()

MÃ©todo para acessar a quantidade de copo P.

Returns:

retorna a quantidade de copo P do estoque

♦ **getQuantCopoM**

public int getQuantCopoM()

MÃ©todo para acessar a quantidade de copo M.

Returns:

retorna a quantidade de copo M do estoque

♦ **getQuantCopoG**

public int getQuantCopoG()

MÃ©todo para acessar a quantidade de copo G.

Returns:

retorna a quantidade de copo G do estoque

♦ **getQuantCopinhoP**

public int getQuantCopinhoP()

MÃ©todo para acessar a quantidade de copinho P.

Returns:

retorna a quantidade de copinho P do estoque

♦ **getQuantCopinhoM**

public int getQuantCopinhoM()

MÃ©todo para acessar a quantidade de copinho M.

Returns:

retorna a quantidade de copinho M do estoque

♦ **getQuantCopinhoG**

public int getQuantCopinhoG()

MÃ©todo para acessar a quantidade de copinho G.

Returns:

retorna a quantidade de copinho G do estoque

♦ **getQuantPicoleAgua**

```
public int getQuantPicoleAgua()
```

Método para acessar a quantidade de picolé a base de água.

Returns:

retorna a quantidade de picolé a base de água do estoque

♦ **getQuantPicoleLeite**

```
public int getQuantPicoleLeite()
```

Método para acessar a quantidade de picolé a base de leite.

Returns:

retorna a quantidade de picolé a base de leite do estoque

♦ **setQuantCascao**

```
public void setQuantCascao(int quantidade)
```

Método para alterar a quantidade de cascaõ do estoque.

Parameters:

quantidade - int - quantidade de cascaõ

♦ **setQuantCasquinha**

```
public void setQuantCasquinha(int quantidade)
```

Método para alterar a quantidade de casquinha do estoque.

Parameters:

quantidade - int - quantidade de casquinha

♦ **setQuantCopoP**

```
public void setQuantCopoP(int quantidade)
```

Método para alterar a quantidade de copo P do estoque.

Parameters:

quantidade - int - quantidade de copo P

♦ **setQuantCopoM**

```
public void setQuantCopoM(int quantidade)
```

Método para alterar a quantidade de copo M do estoque.

Parameters:

quantidade - int - quantidade de copo M

♦ **setQuantCopoG**

```
public void setQuantCopoG(int quantidade)
```

Método para alterar a quantidade de copo G do estoque.

Parameters:

quantidade - int - quantidade de copo G

♦ **setQuantCopinhoP**

```
public void setQuantCopinhoP(int quantidade)
```

Método para alterar a quantidade de copinho P do estoque.

Parameters:

quantidade - int - quantidade de copinho P

♦ **setQuantCopinhoM**

public void setQuantCopinhoM(int quantidade)
MÃ©todo para alterar a quantidade de copinho M do estoque.
Parameters:
quantidade - int - quantidade de copinho M

♦ **setQuantCopinhoG**

public void setQuantCopinhoG(int quantidade)
MÃ©todo para alterar a quantidade de copinho G do estoque.
Parameters:
quantidade - int - quantidade de copinho G

♦ **setQuantPicoleAgua**

public void setQuantPicoleAgua(int quantidade)
MÃ©todo para alterar a quantidade de picolÃ© a base de Ãgua do estoque.
Parameters:
quantidade - int - quantidade de picolÃ© a base de Ãgua

♦ **setQuantPicoleLeite**

public void setQuantPicoleLeite(int quantidade)
MÃ©todo para alterar a quantidade de picolÃ© a base de leite do estoque.
Parameters:
quantidade - int - quantidade de picolÃ© a base de leite

♦ **adicionarSaborSorvete**

public void adicionarSaborSorvete(String sabor)
MÃ©todo para adicionar sabor de sorvete na lista (Set) de sabores.
Parameters:
sabor - String - novo sabor

♦ **getSabores**

public Set<String> getSabores()
MÃ©todo para acessar a lista (Set) de sabores.
Returns:
retorna a lista de sabores

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class EstoqueController

[java.lang.Object](#)

com.example.pedidoentrega.EstoqueController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class EstoqueController extends [Object](#) implements `javafx.fxml.Initializable`
 Classe controller da interface estoque do administrador.

• Field Summary

Fields

Modifier and Type

Field

Description

```
private javafx.scene.control.Button  
botao_add
```

```
private javafx.scene.control.Button  
botao_entregadores
```

```
private javafx.scene.control.Button  
botao_mudarCardapio
```

```
private javafx.scene.control.Button  
botao_sair
```

```
private javafx.scene.control.TableColumn<ItemEstoque, Integer>  
coluna_disponivel
```

```
private javafx.scene.control.TableColumn<ItemEstoque, String>  
coluna_item
```

```
private Estoque  
estoque
```

```
private javafx.scene.control.Spinner  
quantidadeSpinner
```

```
private javafx.scene.control.TableView<ItemEstoque>  
tabela_estoque
```

• Constructor Summary

Constructors

Constructor

Description

```
EstoqueController()
```

• Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

void

Administrador

```
initialize(URL url, ResourceBundle resourceBundle)
MÃ©todo para inicializar a tabela de estoque.
void
lidarBotaoAdicionar(javafx.event.ActionEvent e)
MÃ©todo para adicionar quantidade de itens no estoque.
void
lidarBotaoEntregador(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface com a lista de entregadores.
void
lidarBotaoListaPedidos(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface de lista de pedidos da sorveteria.
void
lidarBotaoMudarCardapio(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface de mudar cardÃ¡pio.
void
lidarBotaoSair(javafx.event.ActionEvent e)
MÃ©todo para sair da conta do administrador e mudar para a tela inicial.
```

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_add**

private javafx.scene.control.Button botao_add

♦ **botao_entregadores**

private javafx.scene.control.Button botao_entregadores

♦ **botao_mudarCardapio**

private javafx.scene.control.Button botao_mudarCardapio

♦ **tabela_estoque**

private javafx.scene.control.TableView<ItemEstoque> tabela_estoque

♦ **coluna_item**

private javafx.scene.control.TableColumn<ItemEstoque,String> coluna_item

♦ **coluna_disponivel**

private javafx.scene.control.TableColumn<ItemEstoque,Integer> coluna_disponivel

♦ **quantidadeSpinner**

private javafx.scene.control.Spinner quantidadeSpinner

♦ **estoque**

private Estoque estoque

• **Constructor Details**

♦ **EstoqueController**

public EstoqueController()

• **Method Details**

♦ **lidarBotaoSair**

public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException
MÃ©todo para sair da conta do administrador e mudar para a tela inicial.

Parameters:

e - ActionEvent - botÃ£o sair

Throws:

IOException

♦ **lidarBotaoEntregador**

public void lidarBotaoEntregador(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface com a lista de entregadores.

Parameters:

e - ActionEvent botÃ£o entregadores

♦ **lidarBotaoMudarCardapio**

public void lidarBotaoMudarCardapio(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface de mudar cardÃ¡pio.

Parameters:

e - ActionEvent botÃ£o mudar cardÃ¡pio

♦ **lidarBotaoListaPedidos**

public void lidarBotaoListaPedidos(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface de lista de pedidos da sorveteria.

Parameters:

e - ActionEvent botÃ£o lista de pedidos

♦ **lidarBotaoAdicionar**

public void lidarBotaoAdicionar(javafx.event.ActionEvent e)
MÃ©todo para adicionar quantidade de itens no estoque.

Parameters:

e - ActionEvent - botÃ£o adicionar

♦ **initialize**

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

MÃ©todo para inicializar a tabela de estoque.

Specified by:

```
initialize in interface javafx.fxml.Initializable
```

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class HelloApplication

[java.lang.Object](#)

[javafx.application.Application](#)

[com.example.pedidoentrega.HelloApplication](#)

```
public class HelloApplication extends javafx.application.Application
```

• Nested Class Summary

Nested classes/interfaces inherited from class `javafx.application.Application`

`javafx.application.Application.Parameters`

• Field Summary

Fields

Modifier and Type

Field

Description

protected static `DisplaySorveteria`
`display`

Fields inherited from class `javafx.application.Application`

`STYLE SHEET_CASPIAN`, `STYLE SHEET_MODENA`

• Constructor Summary

Constructors

Constructor

Description

`HelloApplication()`

• Method Summary

All MethodsStatic MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

static void
`main(String[] args)`

void

`start(javafx.stage.Stage stage)`

Methods inherited from class `javafx.application.Application`

`getHostServices`, `getParameters`, `getUserAgentStylesheet`, `init`,
`launch`, `launch`, `notifyPreloader`, `setUserAgentStylesheet`, `stop`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
`toString`, `wait`, `wait`, `wait`

• Field Details

◆ `display`

protected static `DisplaySorveteria` `display`

• Constructor Details

◆ `HelloApplication`

public `HelloApplication()`

• Method Details

◆ `start`

public void `start(javafx.stage.Stage stage)` throws `IOException`

Specified by:

`start` in class `javafx.application.Application`

Throws:

`IOException`

◆ `main`

public static void `main(String[] args)`

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class HelloController

[java.lang.Object](#)

com.example.pedidoentrega>HelloController

public class HelloController extends [Object](#)
Classe controller da tela inicial.

• Field Summary

Fields

Modifier and Type

Field

Description

`private javafx.scene.control.Button`
`botao_cadastrar`

`private javafx.scene.control.Button`
`botao_cardapio`

`private javafx.scene.control.Button`
`botao_login`

`private javafx.scene.Parent`
`root`

`private javafx.scene.Scene`
`scene`

`private javafx.stage.Stage`
`stage`

• Constructor Summary

Constructors

Constructor

Description

`HelloController()`

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

`void`

`lidarCadastroButton(javafx.event.ActionEvent e)`
 MÃ©todo para mudar para a tela de cadastro.

`void`

`lidarCardapioButton(javafx.event.ActionEvent e)`
 MÃ©todo para mudar para a tela do cardÃ¡pio.

`void`

`lidarLoginButton(javafx.event.ActionEvent e)`
 MÃ©todo para mudar para a tela de login.

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
`toString`, `wait`, `wait`, `wait`

• Field Details

◆ `botao_login`

`private javafx.scene.control.Button botao_login`

◆ `botao_cadastrar`

`private javafx.scene.control.Button botao_cadastrar`

◆ `botao_cardapio`

`private javafx.scene.control.Button botao_cardapio`

◆ `stage`

`private javafx.stage.Stage stage`

◆ `scene`

`private javafx.scene.Scene scene`

◆ `root`

`private javafx.scene.Parent root`

• Constructor Details

◆ `HelloController`

`public HelloController()`

• Method Details

◆ `lidarLoginButton`

`public void lidarLoginButton(javafx.event.ActionEvent e) throws IOException`

MÃ©todo para mudar para a tela de login.

Parameters:

`e` - `ActionEvent` login

Throws:

`IOException`

◆ `lidarCadastroButton`

`public void lidarCadastroButton(javafx.event.ActionEvent e) throws IOException`

MÃ©todo para mudar para a tela de cadastro.

Parameters:

Administrador

e - `ActionEvent` cadastrar

Throws:

`IOException`

◆ **lidarCardapioButton**

`public void lidarCardapioButton(javafx.event.ActionEvent e) throws IOException`
MÃ©todo para mudar para a tela do cardÃ¡pio.

Parameters:

e - `ActionEvent` cardapio

Throws:

`IOException`

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)
- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class InterfaceUtil

[java.lang.Object](#)

com.example.pedidoentrega.InterfaceUtil

public class InterfaceUtil extends [Object](#)

• Constructor Summary

Constructors
 Constructor

Description

InterfaceUtil()

• Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type

Method

Description

static void

chanceScenes(javafx.event.ActionEvent e, String fxmlFile, String titulo)

MÃ©todo para trocar de tela.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

• Constructor Details

♦ InterfaceUtil

public InterfaceUtil()

• Method Details

♦ chanceScenes

public static void chanceScenes(javafx.event.ActionEvent e, String fxmlFile, String titulo)

MÃ©todo para trocar de tela.

Parameters:

e -

fxmlFile -

titulo -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class ItemEstoque

[java.lang.Object](#)

com.example.pedidoentrega.ItemEstoque

public class ItemEstoque extends [Object](#)

Classe ItemEstoque personalizada para colocar na tabela do estoque.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
item

private int
quantidade

• Constructor Summary

Constructors

Constructor

Description

ItemEstoque(String item, int quantidade)
Construtor do ItemEstoque.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

String

getItem()

MÃ©todo para acessar o item.

int

getQuantidade()

MÃ©todo para acessar a quantidade do item.

void

setQuantidade(int quantidade)

MÃ©todo para alterar a quantidade de itens do estoque.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ item

private String item

♦ **quantidade**

private int quantidade

• **Constructor Details**

♦ **ItemEstoque**

public ItemEstoque(String item, int quantidade)
Construtor do ItemEstoque.

Parameters:

item - String - item

quantidade - int - quantidade do item

• **Method Details**

♦ **getItem**

public String getItem()
MÃ©todo para acessar o item.
Returns:

retorna o nome do item

♦ **getQuantidade**

public int getQuantidade()
MÃ©todo para acessar a quantidade do item.
Returns:

retorna a quantidade do item

♦ **setQuantidade**

public void setQuantidade(int quantidade)
MÃ©todo para alterar a quantidade de itens do estoque.
Parameters:

quantidade - int - quantidade de itens

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class LogInController

[java.lang.Object](#)

com.example.pedidoentrega.LogInController

public class LogInController extends [Object](#)
 Classe controller da interface log-in.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao entrar

private javafx.scene.control.Button
botao voltar

private javafx.scene.control.Label
erro

private javafx.scene.control.TextField
tf_email

private javafx.scene.control.PasswordField
tf_senha

• Constructor Summary

Constructors

Constructor

Description

LogInController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

lidarBotaoEntrar(javafx.event.ActionEvent e)
MÃ©todo para entrar na conta do usuÃ¡rio.

void

lidarBotaoVoltar(javafx.event.ActionEvent e)
MÃ©todo para voltar para a tela inicial.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ **tf_email**

```
private javafx.scene.control.TextField tf_email
```

♦ **tf_senha**

```
private javafx.scene.control.PasswordField tf_senha
```

♦ **botao_entrar**

```
private javafx.scene.control.Button botao_entrar
```

♦ **botao_voltar**

```
private javafx.scene.control.Button botao_voltar
```

♦ **erro**

```
private javafx.scene.control.Label erro
```

• Constructor Details

♦ **LogInController**

```
public LogInController()
```

• Method Details

♦ **lidarBotaoEntrar**

```
public void lidarBotaoEntrar(javafx.event.ActionEvent e)
```

MÃ©todo para entrar na conta do usuÃ¡rio.

Parameters:

e - ActionEvent - botÃ£o entrar

♦ **lidarBotaoVoltar**

```
public void lidarBotaoVoltar(javafx.event.ActionEvent e) throws IOException
```

MÃ©todo para voltar para a tela inicial.

Parameters:

e - ActionEvent - botÃ£o voltar

Throws:

IOException

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class LoggeInController

[java.lang.Object](#)

com.example.pedidoentrega.LoggeInController

public class LoggeInController extends [Object](#)
 Classe controller da interface logged-in do cliente.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao_cardapio

private javafx.scene.control.Button
botao_carrinho

private javafx.scene.control.Button
botao_meuPedido

private javafx.scene.control.Button
botao_sair

private Cliente
cliente

private javafx.scene.control.Label
label_email_user

private javafx.scene.control.Label
label_nome_user

private javafx.scene.control.Label
label_telefone

• Constructor Summary

Constructors

Constructor

Description

LoggeInController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

displayUsuario(String nome, String email, String telefone)

MÃ©todo para mostrar na tela os dados da conta do usuÃ¡rio logado.

void

Administrador

lidarBotaoCardapio(javafx.event.ActionEvent e)

MÃ©todo para mudar para a interface do cardapio.

void

lidarBotaoCarrinho(javafx.event.ActionEvent e)

MÃ©todo para mudar para a interface carrinho ou pagamento, caso pedido tenha sido feito e estÃ¡ aguardando pagamento.

void

lidarBotaoMeuPedido(javafx.event.ActionEvent e)

MÃ©todo para mudar para a interface meu-pedido.

void

lidarBotaoSair(javafx.event.ActionEvent e)

MÃ©todo para sair da conta do cliente e voltar para a tela inicial.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

• Field Details

♦ **label_nome_user**

private javafx.scene.control.Label label_nome_user

♦ **label_telefone**

private javafx.scene.control.Label label_telefone

♦ **label_email_user**

private javafx.scene.control.Label label_email_user

♦ **botao_meuPedido**

private javafx.scene.control.Button botao_meuPedido

♦ **botao_cardapio**

private javafx.scene.control.Button botao_cardapio

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_carrinho**

private javafx.scene.control.Button botao_carrinho

♦ **cliente**

private Cliente cliente

• Constructor Details

◆ LoggeInController

```
public LoggeInController()
```

• Method Details

◆ displayUsuario

```
public void displayUsuario(String nome, String email, String telefone)
```

MÃ©todo para mostrar na tela os dados da conta do usuÃ¡rio logado.

Parameters:

nome - String - nome do cliente

email - String - email do cliente

telefone - String - telefone do cliente

◆ lidarBotaoMeuPedido

```
public void lidarBotaoMeuPedido(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface meu-pedido.

Parameters:

e - ActionEvent - botÃ£o meu pedido

◆ lidarBotaoCarrinho

```
public void lidarBotaoCarrinho(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface carrinho ou pagamento, caso pedido tenha sido feito e estÃ¡ aguardando pagamento.

Parameters:

e - ActionEvent - botÃ£o carrinho

◆ lidarBotaoCardapio

```
public void lidarBotaoCardapio(javafx.event.ActionEvent e) throws IOException
```

MÃ©todo para mudar para a interface do cardapio.

Parameters:

e - ActionEvent - botÃ£o cardapio

Throws:

IOException

◆ lidarBotaoSair

```
public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException
```

MÃ©todo para sair da conta do cliente e voltar para a tela inicial.

Parameters:

e - ActionEvent - botÃ£o sair

Throws:

IOException

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class LoggedAdministrador

[java.lang.Object](#)

com.example.pedidoentrega.LoggedAdministrador

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class LoggedAdministrador extends [Object](#) implements javafx.fxml.Initializable
 Classe controller da interface logged-administrador, com a lista de pedidos da sorveteria.

• Field Summary

Fields

Modifier and Type

Field

Description

private Administrador
administrador

private javafx.scene.control.Button
botao entregadores

private javafx.scene.control.Button
botao estoque

private javafx.scene.control.Button
botao mudarCardapio

private javafx.scene.control.Button
botao sair

private ArrayList<Carrinho>
listaCarrinho

private javafx.scene.control.TableColumn<Carrinho, String>
pedido

private javafx.scene.control.TableColumn<Carrinho, String>
status

private javafx.scene.control.TableView<Carrinho>
tabela pedidos

private javafx.scene.control.TableColumn<Carrinho, Float>
total

• Constructor Summary

Constructors

Constructor

Description

LoggedAdministrador()

• Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

void

initialize(URL url, ResourceBundle resourceBundle)

MÃ©todo para inicializar a tabela de pedidos da sorveteria.

void

lidarBotaoEntregador(javafx.event.ActionEvent e)

MÃ©todo para mudar para a tela da lista de entregadores.

void

lidarBotaoEstoque(javafx.event.ActionEvent e)

MÃ©todo para mudar para a tela da tabela de estoque.

void

lidarBotaoMudarCardapio(javafx.event.ActionEvent e)

MÃ©todo para mudar para a tela de adicionar novo sabor.

void

lidarBotaoSair(javafx.event.ActionEvent e)

MÃ©todo para sair da conta do administrador.

Methods inherited from class java.lang.Objectclone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait• **Field Details**♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_entregadores**

private javafx.scene.control.Button botao_entregadores

♦ **botao_mudarCardapio**

private javafx.scene.control.Button botao_mudarCardapio

♦ **botao_estoque**

private javafx.scene.control.Button botao_estoque

♦ **tabela_pedidos**private javafx.scene.control.TableView<Carrinho> tabela_pedidos♦ **pedido**private javafx.scene.control.TableColumn<Carrinho,String> pedido♦ **total**private javafx.scene.control.TableColumn<Carrinho,Float> total

♦ **status**

private javafx.scene.control.TableColumn<Carrinho,String> status

♦ **administrador**

private Administrador administrador

♦ **listaCarrinho**

private ArrayList<Carrinho> listaCarrinho

• **Constructor Details**

♦ **LoggedAdministrador**

public LoggedAdministrador()

• **Method Details**

♦ **lidarBotaoSair**

public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException
MÃ©todo para sair da conta do administrador.

Parameters:

e - ActionEvent - botÃ£o sair

Throws:

IOException

♦ **lidarBotaoEntregador**

public void lidarBotaoEntregador(javafx.event.ActionEvent e)
MÃ©todo para mudar para a tela de entregadores.

Parameters:

e - ActionEvent - botÃ£o entregadores

♦ **lidarBotaoMudarCardapio**

public void lidarBotaoMudarCardapio(javafx.event.ActionEvent e)
MÃ©todo para mudar para a tela de adicionar novo sabor.

Parameters:

e - ActionEvent - botÃ£o mudar cardÃ¡pio

♦ **lidarBotaoEstoque**

public void lidarBotaoEstoque(javafx.event.ActionEvent e)
MÃ©todo para mudar para a tela da tabela de estoque.

Parameters:

e - ActionEvent - botÃ£o estoque

♦ **initialize**

public void initialize(URL url, ResourceBundle resourceBundle)

MÃ©todo para inicializar a tabela de pedidos da sorveteria.

Specified by:

initialize in interface `javafx.fxml.Initializable`

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class MenuEntregadorController

[java.lang.Object](#)

com.example.pedidoentrega.MenuEntregadorController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class MenuEntregadorController extends [Object](#) implements `javafx.fxml.Initializable`
 Classe controller da interface menu-entregador, com a lista de pedidos para entregar.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao concluir

private javafx.scene.control.Button
botao entregar

private javafx.scene.control.Button
botao sair

private javafx.scene.control.TableColumn<Carrinho, String>
endereco

private Entregador
entregador

private ArrayList<Carrinho>
listaCarrinho

private javafx.scene.control.TableColumn<Carrinho, String>
pedido

private javafx.scene.control.TableColumn<Carrinho, String>
status

private javafx.scene.control.TableView<Carrinho>
tabela entregador

• Constructor Summary

Constructors

Constructor

Description

MenuEntregadorController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

Administrador

initialize(URL url, ResourceBundle resourceBundle)

MÃ©todo para inicializar a tabela de pedidos para entregar.

void

lidarBotaoConcluir(javafx.event.ActionEvent e)

MÃ©todo para concluir uma entrega, mudando o status para concluÃ­do.

void

lidarBotaoEntregar(javafx.event.ActionEvent e)

MÃ©todo para entregar o pedido selecionado na tabela.

void

lidarBotaoSair(javafx.event.ActionEvent e)

MÃ©todo para sair da conta do entregador e voltar para a tela inicial.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ **tabela_entregador**

private javafx.scene.control.TableView<Carrinho> tabela_entregador

♦ **pedido**

private javafx.scene.control.TableColumn<Carrinho,String> pedido

♦ **endereco**

private javafx.scene.control.TableColumn<Carrinho,String> endereco

♦ **status**

private javafx.scene.control.TableColumn<Carrinho,String> status

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_entregar**

private javafx.scene.control.Button botao_entregar

♦ **botao_concluir**

private javafx.scene.control.Button botao_concluir

♦ **entregador**

private Entregador entregador

♦ **listaCarrinho**

private ArrayList<Carrinho> listaCarrinho

• Constructor Details

◆ MenuEntregadorController

```
public MenuEntregadorController()
```

• Method Details

◆ lidarBotaoSair

```
public void lidarBotaoSair(javafx.event.ActionEvent e)
MÃ©todo para sair da conta do entregador e voltar para a tela inicial.
Parameters:
    e - ActionEvent - botÃ£o sair
```

◆ lidarBotaoEntregar

```
public void lidarBotaoEntregar(javafx.event.ActionEvent e)
MÃ©todo para entregar o pedido selecionado na tabela.
Parameters:
    e - ActionEvent - botÃ£o entregar
```

◆ lidarBotaoConcluir

```
public void lidarBotaoConcluir(javafx.event.ActionEvent e)
MÃ©todo para concluir uma entrega, mudando o status para concluÃ-do.
Parameters:
    e - ActionEvent - botÃ£o concluir
```

◆ initialize

```
public void initialize(URL url, ResourceBundle resourceBundle)
MÃ©todo para inicializar a tabela de pedidos para entregar.
Specified by:
    initialize in interface javafx.fxml.Initializable
Parameters:
    url -
    resourceBundle -
```

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class MeuPedidoController

[java.lang.Object](#)

com.example.pedidoentrega.MeuPedidoController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class MeuPedidoController extends [Object](#) implements `javafx.fxml.Initializable`

Classe controller da interface meu-pedido, que mostra o status dos pedidos, o histórico dos pedidos e possui a função de cancelar pedidos.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao_cancelar

private javafx.scene.control.Button
botao_cardapio

private javafx.scene.control.Button
botao_carrinho

private javafx.scene.control.Button
botao_outrosPedidos

private javafx.scene.control.Button
botao_sair

private Cliente
cliente

private javafx.scene.control.Label
codigoLabel

private javafx.scene.control.Label
erro

private javafx.scene.layout.Pane
pedidoPane

private javafx.scene.control.Label
statusLabel

private javafx.scene.control.Label
totalLabel

• Constructor Summary

Constructors

Constructor

Description

MeuPedidoController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[initialize\(URL url, ResourceBundle resourceBundle\)](#)

MÃ©todo para inicializar a tela com os dados do pedidos efetuado.

void

[lidarBotaoCancelar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para cancelar pedidos, caso seja possÃ­vel.

void

[lidarBotaoCardapio\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para a interface cardÃ¡pio.

void

[lidarBotaoCarrinho\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para a interface carrinho.

void

[lidarBotaoOutrosPedidos\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para acessar o histÃ³rico de pedidos do cliente, muda para a interface meuPedido-cliente.

void

[lidarBotaoSair\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para sair da conta do cliente e voltar para a tela inicial.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#),
[toString](#), [wait](#), [wait](#), [wait](#)

• Field Details

♦ **botao_carrinho**

private javafx.scene.control.Button botao_carrinho

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_cardapio**

private javafx.scene.control.Button botao_cardapio

♦ **botao_cancelar**

private javafx.scene.control.Button botao_cancelar

♦ **botao_outrosPedidos**

private javafx.scene.control.Button botao_outrosPedidos

♦ **pedidoPane**

```
private javafx.scene.layout.Pane pedidoPane
```

♦ **statusLabel**

```
private javafx.scene.control.Label statusLabel
```

♦ **codigoLabel**

```
private javafx.scene.control.Label codigoLabel
```

♦ **totalLabel**

```
private javafx.scene.control.Label totalLabel
```

♦ **erro**

```
private javafx.scene.control.Label erro
```

♦ **cliente**

```
private Cliente cliente
```

• **Constructor Details**♦ **MeuPedidoController**

```
public MeuPedidoController()
```

• **Method Details**♦ **lidarBotaoCardapio**

```
public void lidarBotaoCardapio(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface cardÃ¡pio.

Parameters:

e - ActionEvent - botÃ£o cardÃ¡pio

♦ **lidarBotaoCarrinho**

```
public void lidarBotaoCarrinho(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface carrinho.

Parameters:

e - ActionEvent - botÃ£o carrinho

♦ **lidarBotaoSair**

```
public void lidarBotaoSair(javafx.event.ActionEvent e)
```

MÃ©todo para sair da conta do cliente e voltar para a tela inicial.

Parameters:

e - ActionEvent - botÃ£o sair

♦ **lidarBotaoCancelar**

```
public void lidarBotaoCancelar(javafx.event.ActionEvent e)
```

MÃ©todo para cancelar pedidos, caso seja possÃ­vel.

Parameters:

e - ActionEvent - botÃ£o cancelar

♦ **lidarBotaoOutrosPedidos**

```
public void lidarBotaoOutrosPedidos(javafx.event.ActionEvent e)
```

MÃ©todo para acessar o histÃ³rico de pedidos do cliente, muda para a interface meuPedido-cliente.

Parameters:

e - ActionEvent - botÃ£o histÃ³rico

♦ **initialize**

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

MÃ©todo para inicializar a tela com os dados do pedidos efetuado.

Specified by:

initialize in interface `javafx.fxml.Initializable`

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Milkshake

[java.lang.Object](#)

[com.example.pedidoentrega.Sorvete](#)

[com.example.pedidoentrega.Milkshake](#)

public class Milkshake extends [Sorvete](#)

Classe Milkshake herda a classe abstrata Sorvete, guarda os dados do milkshake.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
tamanho

• Constructor Summary

Constructors

Constructor

Description

Milkshake(String sabor, Estoque estoque, String tamanho)
Construtor do Milkshake.

• Method Summary

Methods inherited from class com.example.pedidoentrega.Sorvete

getPreco, getSabor, setPreco, setPreco

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ tamanho

private String tamanho

• Constructor Details

♦ Milkshake

public Milkshake(String sabor, Estoque estoque, String tamanho)
Construtor do Milkshake.

Parameters:

sabor - String - sabor do milkshake

estoque - Estoque - quantidade do estoque

tamanho - String - tamanho do milkshake (P, M, G)

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class MudarCardapioController

[java.lang.Object](#)

com.example.pedidoentrega.MudarCardapioController

public class MudarCardapioController extends [Object](#)

Classe controller da interface mudar-cardapio, permite adicionar novos sabores de sorvete.

• Field Summary

Fields

Modifier and Type

Field

Description

```
private javafx.scene.control.Button
botao confirmar
```

```
private javafx.scene.control.Button
botao entregadores
```

```
private javafx.scene.control.Button
botao estoque
```

```
private javafx.scene.control.Button
botao listaPedidos
```

```
private javafx.scene.control.Button
botao sair
```

```
private javafx.scene.control.Label
erro
```

```
private javafx.scene.control.TextField
tf novoSabor
```

• Constructor Summary

Constructors

Constructor

Description

```
MudarCardapioController()
```

• Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

void

```
lidarBotaoConfirmar(javafx.event.ActionEvent e)
```

```
MÃ©todo para confirmar a aÃ§Ã£o de adicionar sabor de sorvete.
```

void

```
lidarBotaoEntregador(javafx.event.ActionEvent e)
```

```
MÃ©todo para mudara para a interface com a lista de entregadores.
```

void

Administrador

```
lidarBotaoEstoque(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a interface estoque, com a tabela do estoque.  
void  
lidarBotaoListaPedidos(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a tela com a lista de pedidos da sorveteria.  
void  
lidarBotaoSair(javafx.event.ActionEvent e)  
MÃ©todo para sair da conta do administrador e voltar para a tela inicial.
```

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ **botao_sair**

```
private javafx.scene.control.Button botao_sair
```

♦ **botao_entregadores**

```
private javafx.scene.control.Button botao_entregadores
```

♦ **botao_listaPedidos**

```
private javafx.scene.control.Button botao_listaPedidos
```

♦ **botao_confirmar**

```
private javafx.scene.control.Button botao_confirmar
```

♦ **botao_estoque**

```
private javafx.scene.control.Button botao_estoque
```

♦ **tf_novoSabor**

```
private javafx.scene.control.TextField tf_novoSabor
```

♦ **erro**

```
private javafx.scene.control.Label erro
```

• Constructor Details

♦ **MudarCardapioController**

```
public MudarCardapioController()
```

• Method Details

♦ **lidarBotaoSair**

public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException
MÃ©todo para sair da conta do administrador e voltar para a tela inicial.

Parameters:

e - ActionEvent - botÃ£o sair

Throws:

IOException

♦ **lidarBotaoEntregador**

public void lidarBotaoEntregador(javafx.event.ActionEvent e)
MÃ©todo para mudara para a interface com a lista de entregadores.

Parameters:

e - ActionEvent - botÃ£o entregadores

♦ **lidarBotaoListaPedidos**

public void lidarBotaoListaPedidos(javafx.event.ActionEvent e)
MÃ©todo para mudar para a tela com a lista de pedidos da sorveteria.

Parameters:

e - ActionEvent - botÃ£o lista de pedidos

♦ **lidarBotaoConfirmar**

public void lidarBotaoConfirmar(javafx.event.ActionEvent e) throws IOException
MÃ©todo para confirmar a aÃ§Ã£o de adicionar sabor de sorvete.

Parameters:

e - ActionEvent - botÃ£o confirmar

Throws:

IOException

♦ **lidarBotaoEstoque**

public void lidarBotaoEstoque(javafx.event.ActionEvent e)
MÃ©todo para mudar para a interface estoque, com a tabela do estoque.

Parameters:

e - ActionEvent - botÃ£o estoque

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class PagamentoController

[java.lang.Object](#)

com.example.pedidoentrega.PagamentoController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class PagamentoController extends [Object](#) implements `javafx.fxml.Initializable`

Classe controller da interface pagamento, para preencher os dados do cartão para pagamento, com opção de trocar o endereço de entrega.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao_cardapio

private javafx.scene.control.Button
botao_meuPedido

private javafx.scene.control.Button
botao_sair

private Cliente
cliente

private javafx.scene.control.Label
enderecoLabel

private javafx.scene.control.Label
erro

private javafx.scene.control.TextField
tf_cpfTitular

private javafx.scene.control.TextField
tf_cvv

private javafx.scene.control.TextField
tf_nomeTitular

private javafx.scene.control.TextField
tf_numeroCartao

private javafx.scene.control.TextField
tf_validade

private javafx.scene.control.Label
totalLabel

• Constructor Summary

Constructors

Constructor

Description

PagamentoController()

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[initialize\(URL url, ResourceBundle resourceBundle\)](#)

MÃ©todo para inicializar o label com o endereÃ§o de entrega salvo no cadastro do cliente, e mostra ele atualizado caso haja alteraÃ§Ã£o.

void

[lidarBotaoAlterarEndereco\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para a interface alterarEndereco.

void

[lidarBotaoCardapio\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface cardapio.

void

[lidarBotaoCarrinho\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface carrinho.

void

[lidarBotaoFinalizar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para verificar se todos os campos do pagamento foram preenchidos e, entÃ£o, chamar a interface para mostrar o pedido efetuado e seu status.

void

[lidarBotaoMeuPedido\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para interface meu-pedido.

void

[lidarBotaoSair\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para sair da conta e mudar para tela inicial.

void

[lidarBotaoVoltar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para voltar para interface pagamento.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

• Field Details

♦ **tf_numeroCartao**

private javafx.scene.control.TextField tf_numeroCartao

♦ **tf_validade**

private javafx.scene.control.TextField tf_validade

♦ **tf_cvv**

```
private javafx.scene.control.TextField tf_cvv
```

♦ **tf_cpfTitular**

```
private javafx.scene.control.TextField tf_cpfTitular
```

♦ **tf_nomeTitular**

```
private javafx.scene.control.TextField tf_nomeTitular
```

♦ **botao_meuPedido**

```
private javafx.scene.control.Button botao_meuPedido
```

♦ **botao_cardapio**

```
private javafx.scene.control.Button botao_cardapio
```

♦ **botao_sair**

```
private javafx.scene.control.Button botao_sair
```

♦ **erro**

```
private javafx.scene.control.Label erro
```

♦ **enderecoLabel**

```
private javafx.scene.control.Label enderecoLabel
```

♦ **totalLabel**

```
private javafx.scene.control.Label totalLabel
```

♦ **cliente**

```
private Cliente cliente
```

• Constructor Details

♦ **PagamentoController**

```
public PagamentoController()
```

• Method Details

♦ **lidarBotaoMeuPedido**

```
public void lidarBotaoMeuPedido(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para interface meu-pedido.

Parameters:

e - ActionEvent - botÃ£o Meu Pedido

◆ lidarBotaoCarrinho

public void lidarBotaoCarrinho(javafx.event.ActionEvent e)

MÃ©todo para mudar para interface carrinho.

Parameters:

e - ActionEvent - botÃ£o Carrinho

◆ lidarBotaoCardapio

public void lidarBotaoCardapio(javafx.event.ActionEvent e) throws IOException

MÃ©todo para mudar para interface cardapio.

Parameters:

e - ActionEvent - botÃ£o Cardapio

Throws:

IOException

◆ lidarBotaoSair

public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException

MÃ©todo para sair da conta e mudar para tela inicial.

Parameters:

e - ActionEvent - botÃ£o Sair

Throws:

IOException

◆ lidarBotaoAlterarEndereco

public void lidarBotaoAlterarEndereco(javafx.event.ActionEvent e)

MÃ©todo para mudar para a interface alterarEndereco.

Parameters:

e - ActionEvent - botÃ£o Alterar endereco

◆ lidarBotaoFinalizar

public void lidarBotaoFinalizar(javafx.event.ActionEvent e)

MÃ©todo para verificar se todos os campos do pagamento foram preenchidos e, entÃ£o, chamar a interface para mostrar o pedido efetuado e seu status.

Parameters:

e - ActionEvent - botÃ£o finalizar

◆ lidarBotaoVoltar

public void lidarBotaoVoltar(javafx.event.ActionEvent e) throws IOException

MÃ©todo para voltar para interface pagamento.

Parameters:

e - ActionEvent - botÃ£o Voltar

Throws:

IOException

◆ initialize

public void initialize(URL url, ResourceBundle resourceBundle)

MÃ©todo para inicializar o label com o endereco de entrega salvo no cadastro do cliente, e mostra ele atualizado caso haja alteraÃ§Ã£o.

Specified by:

Administrador

initialize in interface javafx.fxml.Initializable

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Pedido<T extends [Sorvete](#)>

[java.lang.Object](#)

com.example.pedidoentrega.Pedido<T>

Type Parameters:

T -

public class Pedido<T extends [Sorvete](#)> extends [Object](#)

Classe Pedido usa Generics para poder receber qualquer tipo de Sorvete (Casquinha, Milkshake, PicolÃ©).

• Field Summary

Fields

Modifier and Type

Field

Description

private String
descricao

private T
item

private int
quantidade

private Float
unidade

private float
valor

• Constructor Summary

Constructors

Constructor

Description

Pedido(T item, int quantidade, String descricao)

Construtor do Pedido.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

String

getDescricao()

MÃ©todo para acessar a descriÃ§Ã£o do pedido.

T

getItem()

MÃ©todo para acessar o item pedido.

int

getQuantidade()

MÃ©todo para acessar a quantidade de itens pedidos.

float

getUnidade()

MÃ©todo para acessar o valor por unidade do pedido.

float

getValor()

MÃ©todo para acessar o valor do pedido.

void

setDescricao(String descricao)

MÃ©todo para acessar a descriÃ§Ã£o do pedido.

void

setValor()

MÃ©todo para alterar o valor do item, conforme a quantidade pedida.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ item

private T extends Sorvete item

♦ quantidade

private int quantidade

♦ valor

private float valor

♦ descricao

private String descricao

♦ unidade

private Float unidade

• Constructor Details

♦ Pedido

public Pedido(T item, int quantidade, String descricao)

Construtor do Pedido.

Parameters:

item - T - Casquiha/Milkshake/PicolÃ©

quantidade - int - quantidade de itens

descricao - String - descriÃ§Ã£o do pedido

• Method Details

◆ **setValor**

```
public void setValor()
```

Método para alterar o valor do item, conforme a quantidade pedida.

◆ **getValor**

```
public float getValor()
```

Método para acessar o valor do pedido.

Returns:

retorna o valor do pedido - float

◆ **getQuantidade**

```
public int getQuantidade()
```

Método para acessar a quantidade de itens pedidos.

Returns:

retorna a quantidade de itens pedidos - int

◆ **getDescricao**

```
public String getDescricao()
```

Método para acessar a descrição do pedido.

Returns:

retorna a descrição do pedido

◆ **getUnidade**

```
public float getUnidade()
```

Método para acessar o valor por unidade do pedido.

Returns:

retorna o valor por unidade do pedido

◆ **getItem**

```
public T getItem()
```

Método para acessar o item pedido.

Returns:

retorna o item pedido

◆ **setDescricao**

```
public void setDescricao(String descricao)
```

Método para acessar a descrição do pedido.

Parameters:

descricao - String - descrição do pedido

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Picole

[java.lang.Object](#)

[com.example.pedidoentrega.Sorvete](#)

[com.example.pedidoentrega.Picole](#)

public class Picole extends [Sorvete](#)

Classe Picole herda a classe abstrata Sorvete, guarda os dados do picolÃ©.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
base

• Constructor Summary

Constructors

Constructor

Description

Picole(String sabor, Estoque estoque, String base)
Construtor do PicolÃ©.

• Method Summary

Methods inherited from class com.example.pedidoentrega.Sorvete

getPreco, getSabor, setPreco, setPreco

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ base

private String base

• Constructor Details

♦ Picole

public Picole(String sabor, Estoque estoque, String base)
Construtor do PicolÃ©.

Parameters:

sabor - String - sabor do picolÃ©

estoque - Estoque - quantidade do estoque

base - String - base do picolÃ© (Ã¡gua ou leite)

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class SignUpController

[java.lang.Object](#)

`com.example.pedidoentrega.SignUpController`

public class SignUpController extends [Object](#)

Classe controller da interface sign-up, que faz o cadastro de novos clientes.

• Field Summary

Fields

Modifier and Type

Field

Description

private javafx.scene.control.Button
botao confirmar

private javafx.scene.control.Button
botao voltar

private javafx.scene.control.Label
erro

private javafx.scene.Parent
root

private javafx.scene.Scene
scene

private javafx.stage.Stage
stage

private javafx.scene.control.TextField
tf bairro

private javafx.scene.control.TextField
tf email

private javafx.scene.control.TextField
tf nome

private javafx.scene.control.TextField
tf numero

private javafx.scene.control.TextField
tf rua

private javafx.scene.control.PasswordField
tf senha

private javafx.scene.control.TextField
tf telefone

• Constructor Summary

Constructors

Constructor

Description

[SignUpController\(\)](#)

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[lidarBotaoConfirmar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para confirmar o cadastro apÃ³s preencher todos os campos necessÃ¡rios.

void

[lidarBotaoVoltar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para sair do cadastro e voltar para a tela inicial.

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

• Field Details

♦ **tf_nome**

private javafx.scene.control.TextField tf_nome

♦ **tf_email**

private javafx.scene.control.TextField tf_email

♦ **tf_senha**

private javafx.scene.control.PasswordField tf_senha

♦ **tf_telefone**

private javafx.scene.control.TextField tf_telefone

♦ **tf_rua**

private javafx.scene.control.TextField tf_rua

♦ **tf_numero**

private javafx.scene.control.TextField tf_numero

♦ **tf_bairro**

```
private javafx.scene.control.TextField tf_bairro
```

♦ **botao_confirmar**

```
private javafx.scene.control.Button botao_confirmar
```

♦ **botao_voltar**

```
private javafx.scene.control.Button botao_voltar
```

♦ **erro**

```
private javafx.scene.control.Label erro
```

♦ **stage**

```
private javafx.stage.Stage stage
```

♦ **scene**

```
private javafx.scene.Scene scene
```

♦ **root**

```
private javafx.scene.Parent root
```

• **Constructor Details**

♦ **SignUpController**

```
public SignUpController()
```

• **Method Details**

♦ **lidarBotaoConfirmar**

```
public void lidarBotaoConfirmar(javafx.event.ActionEvent e)
```

Método para confirmar o cadastro após preencher todos os campos necessários.

Parameters:

e - ActionEvent - botão confirmar

♦ **lidarBotaoVoltar**

```
public void lidarBotaoVoltar(javafx.event.ActionEvent e) throws IOException
```

Método para sair do cadastro e voltar para a tela inicial.

Parameters:

e - ActionEvent - botão voltar

Throws:

IOException

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class SignUpEntregadorController

[java.lang.Object](#)

com.example.pedidoentrega.SignUpEntregadorController

public class SignUpEntregadorController extends [Object](#)

Classe controller da interface sign-up-entregador, para fazer o cadastro de novos entregadores, aÃ§Ã£o possÃ-vel apenas para o administrador.

• Field Summary

Fields

Modifier and Type

Field

Description

```
private javafx.scene.control.Button
botao confirmar
```

```
private javafx.scene.control.Button
botao estoque
```

```
private javafx.scene.control.Button
botao listaPedidos
```

```
private javafx.scene.control.Button
botao mudarCardapio
```

```
private javafx.scene.control.Button
botao sair
```

```
private javafx.scene.control.Label
erro
```

```
private javafx.scene.control.TextField
tf email
```

```
private javafx.scene.control.TextField
tf nome
```

```
private javafx.scene.control.PasswordField
tf senha
```

```
private javafx.scene.control.TextField
tf telefone
```

• Constructor Summary

Constructors

Constructor

Description

```
SignUpEntregadorController()
```

• Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type

Administrador

Method

Description

void

lidarBotaoConfirmar(javafx.event.ActionEvent e)

MÃ©todo para confirmar os dados do novo entregador e cadastrÃ¡-lo.

void

lidarBotaoEntregador(javafx.event.ActionEvent e)

MÃ©todo para mudar para a tela da lista de entregadores.

void

lidarBotaoEstoque(javafx.event.ActionEvent e)

MÃ©todo para mudar para a tela da tabela de estoque.

void

lidarBotaoListaPedidos(javafx.event.ActionEvent e)

MÃ©todo para mudar para a interface logged-administrador, com a lista de pedidos da sorveteria.

void

lidarBotaoMudarCardapio(javafx.event.ActionEvent e)

MÃ©todo para mudar para a tela de adicionar novo sabor.

void

lidarBotaoSair(javafx.event.ActionEvent e)

MÃ©todo para sair da conta do administrador.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ **botao_sair**

private javafx.scene.control.Button botao_sair

♦ **botao_confirmar**

private javafx.scene.control.Button botao_confirmar

♦ **botao_listaPedidos**

private javafx.scene.control.Button botao_listaPedidos

♦ **botao_mudarCardapio**

private javafx.scene.control.Button botao_mudarCardapio

♦ **botao_estoque**

private javafx.scene.control.Button botao_estoque

♦ **tf_nome**

private javafx.scene.control.TextField tf_nome

♦ **tf_email**

```
private javafx.scene.control.TextField tf_email
```

♦ **tf_telefone**

```
private javafx.scene.control.TextField tf_telefone
```

♦ **tf_senha**

```
private javafx.scene.control.PasswordField tf_senha
```

♦ **erro**

```
private javafx.scene.control.Label erro
```

• **Constructor Details**

♦ **SignUpEntregadorController**

```
public SignUpEntregadorController()
```

• **Method Details**

♦ **lidarBotaoConfirmar**

```
public void lidarBotaoConfirmar(javafx.event.ActionEvent e)  
MÃ©todo para confirmar os dados do novo entregador e cadastrÃ-lo.  
Parameters:  
e - ActionEvent - botÃ£o confirmar
```

♦ **lidarBotaoSair**

```
public void lidarBotaoSair(javafx.event.ActionEvent e) throws IOException  
MÃ©todo para sair da conta do administrador.  
Parameters:  
e - ActionEvent - botÃ£o sair  
Throws:  
IOException
```

♦ **lidarBotaoMudarCardapio**

```
public void lidarBotaoMudarCardapio(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a tela de adicionar novo sabor.  
Parameters:  
e - ActionEvent - botÃ£o mudar cardÃpio
```

♦ **lidarBotaoEstoque**

```
public void lidarBotaoEstoque(javafx.event.ActionEvent e)  
MÃ©todo para mudar para a tela da tabela de estoque.  
Parameters:  
e - ActionEvent - botÃ£o estoque
```

♦ **lidarBotaoListaPedidos**

```
public void lidarBotaoListaPedidos(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface logged-administrador, com a lista de pedidos da sorveteria.

Parameters:

e -

♦ **lidarBotaoEntregador**

```
public void lidarBotaoEntregador(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a tela da lista de entregadores.

Parameters:

e - ActionEvent - botÃ£o entregadores

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Sorvete

[java.lang.Object](#)

com.example.pedidoentrega.Sorvete

Direct Known Subclasses:

[Casquinha](#), [Copinho](#), [Milkshake](#), [Picole](#)

public abstract class Sorvete extends [Object](#)

Classe abstrata Sorvete, guarda atributos em comum dos tipos de sorvetes que herdam essa classe, como o sabor, preço e quantidade de estoque.

• Field Summary

Fields

Modifier and Type

Field

Description

private Estoque
estoque

private static float
preco

private String
sabor

• Constructor Summary

Constructors

Constructor

Description

Sorvete(String sabor, Estoque estoque)
Construtor de Sorvete.

• Method Summary

All MethodsStatic MethodsInstance MethodsConcrete Methods

Modifier and Type

Method

Description

static float

getPreco()

MÃ©todo para acessar o preÃ§o do sorvete.

String

getSabor()

MÃ©todo para acessar o sabor do sorvete.

void

setPreco(float preco)

MÃ©todo para alterar o preÃ§o do sorvete.

static float

setPreco(String tamanho)

MÃ©todo para alterar o preÃ§o dos itens de acordo com o tipo/tamanho.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ sabor

private String sabor

♦ preco

private static float preco

♦ estoque

private Estoque estoque

• Constructor Details

♦ Sorvete

public Sorvete(String sabor, Estoque estoque)

Construtor de Sorvete.

Parameters:

sabor - String - sabor do sorvete

estoque - Estoque - quantidade no estoque

• Method Details

♦ getPreco

public static float getPreco()

MÃ©todo para acessar o preÃ§o do sorvete.

Returns:

retorna o preÃ§o do sorvete

♦ setPreco

public static float setPreco(String tamanho)

MÃ©todo para alterar o preÃ§o dos itens de acordo com o tipo/tamanho.

Parameters:

tamanho - String - tamanho/tipo do sorvete

Returns:

retorna o preÃ§o do sorvete

♦ setPreco

public void setPreco(float preco)

MÃ©todo para alterar o preÃ§o do sorvete.

Parameters:

preco - float - preÃ§o

♦ **getSabor**

```
public String getSabor()
```

Método para acessar o sabor do sorvete.

Returns:

retorna o sabor do sorvete

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
 - [Nested](#) |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class SorveteriaSistema

[java.lang.Object](#)

com.example.pedidoentrega.SorveteriaSistema

public class SorveteriaSistema extends [Object](#)

Classe para cadastrar usu rios, fazer login e deslogar.

• Field Summary

Fields

Modifier and Type

Field

Description

private Administrador
administradorLogado

private Administrador
administradorSorveteria

private Cliente
clienteLogado

private ArrayList<Cliente>
clientes

private ArrayList<Entregador>
entregadores

private Entregador
entregadorLogado

private Estoque
estoque

• Constructor Summary

Constructors

Constructor

Description

SorveteriaSistema(String email, String cnpj, String senha)
Construtor da SorveteriaSistema que cadastra o administrador.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void
adicionarNovoCliente(Cliente novo)

Método para cadastrar novo cliente, verificando se a conta já não foi cadastrada, e adicionando ele no arquivo.

void
adicionarNovoEntregador(Entregador novo)

Método para cadastrar novo entregador, verificando se já não foi cadastrado, e adicionando ele

Administrador

no arquivo.

void

adicionarSaborSorvete(String sabor)

MÃ©todo para adicionar os sabores de sorvete.

Usuario

buscarUsuarios(String email, String senha)

MÃ©todo para buscar usuÃ¡rios na lista.

void

cadastrarCliente(String nome, String email, String telefone, String senha, String rua, String bairro, String num)

MÃ©todo para cadastrar clientes lidos no arquivo.

void

cadastrarEntregador(String nome, String email, String senha, String telefone)

MÃ©todo para cadastrar entregadores lidos no arquivo.

void

deslogar()

MÃ©todo para deslogar da conta.

Administrador

getAdministradorLogado()

MÃ©todo para acessar o administrador logado.

Administrador

getAdministradorSorveteria()

MÃ©todo para acessar o administrador logado.

Cliente

getClienteLogado()

MÃ©todo para acessar o cliente logado.

ArrayList<Entregador>

getEntregadores()

MÃ©todo para acessar a lista de entregadores.

Entregador

getEntregadorLogado()

MÃ©todo para acessar o entregador logado.

Estoque

getEstoque()

MÃ©todo para acessar o estoque.

void

iniciarEstoque(int cascao, int casquinha, int copoP, int copoM, int copoG, int copinhoP, int copinhoM, int copinhoG, int picoleAgua, int picoleLeite)

MÃ©todo para inicializar o estoque, atribuindo as quantidades.

void

logar(String email, String senha)

MÃ©todo para fazer login do usuÃ¡rio.

void

removerEntregador(Entregador entregador)

MÃ©todo para remover da lista de entregadores.

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

• Field Details

◆ `clientes`

private `ArrayList<Cliente>` clientes

◆ `entregadores`

private `ArrayList<Entregador>` entregadores

◆ `estoque`

private `Estoque` estoque

◆ `administradorSorveteria`

private `Administrador` administradorSorveteria

◆ `clienteLogado`

private `Cliente` clienteLogado

◆ `entregadorLogado`

private `Entregador` entregadorLogado

◆ `administradorLogado`

private `Administrador` administradorLogado

• Constructor Details

◆ `SorveteriaSistema`

public `SorveteriaSistema(String email, String cnpj, String senha)`

Construtor da `SorveteriaSistema` que cadastra o administrador.

Parameters:

`email` - `String` - email do administrador da sorveteria

`cnpj` - `String` - cnpj da sorveteria

`senha` - `String` - senha do administrador da sorveteria

• Method Details

◆ `cadastrarCliente`

public void `cadastrarCliente(String nome, String email, String telefone, String senha, String rua, String bairro, String num)`

MÃ©todo para cadastrar clientes lidos no arquivo.

Administrador

Parameters:

nome - String - nome do cliente
email - String - email do cliente
telefone - String - telefone do cliente
senha - String - senha do cliente
rua - String - rua do cliente
bairro - String - bairro do cliente
num - String - número do cliente

♦ adicionarNovoCliente

public void adicionarNovoCliente(Cliente novo) throws Exception

Método para cadastrar novo cliente, verificando se a conta já não foi cadastrada, e adicionando ele no arquivo.

Parameters:

novo - Cliente - novo cliente

Throws:

Exception

♦ cadastrarEntregador

public void cadastrarEntregador(String nome, String email, String senha, String telefone)

Método para cadastrar entregadores lidos no arquivo.

Parameters:

nome - String - nome do entregador
email - String - email do entregador
senha - String - senha do entregador
telefone - String - telefone do entregador

♦ adicionarNovoEntregador

public void adicionarNovoEntregador(Entregador novo) throws Exception

Método para cadastrar novo entregador, verificando se já não foi cadastrado, e adicionando ele no arquivo.

Parameters:

novo - Entregador - novo entregador

Throws:

Exception

♦ buscarUsuarios

public Usuario buscarUsuarios(String email, String senha)

Método para buscar usuários na lista.

Parameters:

email - String - email do usuário
senha - String - senha do usuário

Returns:

retorna a conta do usuário caso a encontre

♦ logar

public void logar(String email, String senha) throws UsuarioInexistenteException

Método para fazer login do usuário.

Parameters:

email - String - email do usuário

Administrador

senha - String - senha do usuário

Throws:

UsuarioInexistenteException

♦ iniciarEstoque

public void iniciarEstoque(int cascao, int casquinha, int copoP, int copoM, int copoG, int copinhoP, int copinhoM, int copinhoG, int picoleAgua, int picoleLeite)

Método para inicializar o estoque, atribuindo as quantidades.

Parameters:

cascao - int - quantidade cascao

casquinha - int - quantidade casquinha

copoP - int - quantidade copo P

copoM - int - quantidade copo M

copoG - int - quantidade copo G

copinhoP - int - quantidade copinho P

copinhoM - int - quantidade copinho M

copinhoG - int - quantidade copinho G

picoleAgua - int - quantidade picolé a base de água

picoleLeite - int - quantidade picolé a base de leite

♦ adicionarSaborSorvete

public void adicionarSaborSorvete(String sabor)

Método para adicionar os sabores de sorvete.

Parameters:

sabor - String - sabor

♦ deslogar

public void deslogar()

Método para deslogar da conta.

♦ getClienteLogado

public Cliente getClienteLogado()

Método para acessar o cliente logado.

Returns:

retorna o cliente logado

♦ getEntregadorLogado

public Entregador getEntregadorLogado()

Método para acessar o entregador logado.

Returns:

retorna o entregador logado

♦ getAdministradorLogado

public Administrador getAdministradorLogado()

Método para acessar o administrador logado.

Returns:

retorna o administrador logado

♦ **getEstoque**

```
public Estoque getEstoque()  
MÃ©todo para acessar o estoque.  
Returns:  
    retorna o estoque
```

♦ **getEntregadores**

```
public ArrayList<Entregador> getEntregadores()  
MÃ©todo para acessar a lista de entregadores.  
Returns:  
    retorna a lista de entregadores
```

♦ **removerEntregador**

```
public void removerEntregador(Entregador entregador)  
MÃ©todo para remover da lista de entregadores.  
Parameters:  
    entregador - Entregador - entregador a ser removido
```

♦ **getAdministradorSorveteria**

```
public Administrador getAdministradorSorveteria()  
MÃ©todo para acessar o administrador logado.  
Returns:  
    retorna o administrador logado
```

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class Usuario

[java.lang.Object](#)

com.example.pedidoentrega.Usuario

Direct Known Subclasses:

[Administrador](#), [Cliente](#), [Entregador](#)

public abstract class Usuario extends [Object](#)

Classe abstrata usu rio com atributos em comum entre todos os usu rios que herdam a classe.

• Field Summary

Fields

Modifier and Type

Field

Description

private String
email

private String
senha

• Constructor Summary

Constructors

Constructor

Description

Usuario(String email, String senha)
Construtor do uSUARIO.

• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

String

getEmail()

MÃ©todo para acessar o email do usuÃ¡rio.

String

getSenha()

MÃ©todo para acessar a senha do usuÃ¡rio.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

• Field Details

♦ email

private String email

♦ senha

private String senha

• Constructor Details

♦ Usuario

```
public Usuario(String email, String senha)
```

Construtor do uSUARIO.

Parameters:

email - String - email

senha - String - senha

• Method Details

♦ getEmail

```
public String getEmail()
```

MÃ©todo para acessar o email do usuÃ¡rio.

Returns:

retorna o email do usuÃ¡rio

♦ getSenha

```
public String getSenha()
```

MÃ©todo para acessar a senha do usuÃ¡rio.

Returns:

retorna a senha do usuÃ¡rio

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)

- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)
- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class UsuarioInexistenteException

[java.lang.Object](#)

[java.lang.Throwable](#)

[java.lang.Exception](#)

com.example.pedidoentrega.UsuarioInexistenteException

All Implemented Interfaces:

[Serializable](#)

public class UsuarioInexistenteException extends Exception
 Classe para lançar exceção caso o usuário não exista.

See Also:

◊ Serialized Form

• Constructor Summary

Constructors

Constructor

Description

UsuarioInexistenteException(String mensagem)

• Method Summary

Methods inherited from class java.lang.Throwable

addSuppressed, fillInStackTrace, getCause, getLocalizedMessage,
getMessage, getStackTrace, getSuppressed, initCause,
printStackTrace, printStackTrace, printStackTrace, setStackTrace,
toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
wait, wait, wait

• Constructor Details

◆ UsuarioInexistenteException

public UsuarioInexistenteException(String mensagem)

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
 - ◆ [Nested](#)
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Detail:
 - ◆ [Field](#)
 - ◆ [Constr](#)
 - ◆ [Method](#)
- Summary:
- Nested |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package [com.example.pedidoentrega](#)

Class historicoController

[java.lang.Object](#)

com.example.pedidoentrega.historicoController

All Implemented Interfaces:

`javafx.fxml.Initializable`

public class historicoController extends [Object](#) implements `javafx.fxml.Initializable`
 Classe controller da interface de histÃ³rico de pedidos do cliente.

• Field Summary

Fields

Modifier and Type

Field

Description

```
private javafx.scene.control.Button  
botao cancelar
```

```
private javafx.scene.control.Button  
botao cardapio
```

```
private javafx.scene.control.Button  
botao carrinho
```

```
private javafx.scene.control.Button  
botao sair
```

```
private javafx.scene.control.Button  
botao voltar
```

```
private Cliente  
cliente
```

```
private javafx.scene.control.TableColumn<Carrinho, String>  
coluna pedido
```

```
private javafx.scene.control.TableColumn<Carrinho, String>  
coluna status
```

```
private javafx.scene.control.TableColumn<Carrinho, Float>  
coluna total
```

```
private javafx.scene.control.Label  
erroLabel
```

```
private ArrayList<Carrinho>  
listaCarrinho
```

```
private javafx.scene.control.TableView<Carrinho>  
tabela pedidos
```

• Constructor Summary

Constructors

Constructor

Description

```
historicoController()
```


• Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type

Method

Description

void

[initialize\(URL url, ResourceBundle resourceBundle\)](#)

MÃ©todo para inicializar a tabela do histÃ³rico de pedidos.

void

[lidarBotaoCancelar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para cancelar um pedido do histÃ³rico de pedidos.

void

[lidarBotaoCardapio\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para a interface do cardÃ¡pio.

void

[lidarBotaoCarrinho\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para mudar para a interface do carrinho.

void

[lidarBotaoSair\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para sair da conta do cliente.

void

[lidarBotaoVoltar\(javaafx.event.ActionEvent e\)](#)

MÃ©todo para voltar para a interface do meu-pedido.

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

• Field Details

♦ erroLabel

private javafx.scene.control.Label erroLabel

♦ botao_carrinho

private javafx.scene.control.Button botao_carrinho

♦ botao_voltar

private javafx.scene.control.Button botao_voltar

♦ botao_sair

private javafx.scene.control.Button botao_sair

♦ botao_cardapio

private javafx.scene.control.Button botao_cardapio

♦ **botao_cancelar**

```
private javafx.scene.control.Button botao_cancelar
```

♦ **tabela_pedidos**

```
private javafx.scene.control.TableView<Carrinho> tabela_pedidos
```

♦ **coluna_pedido**

```
private javafx.scene.control.TableColumn<Carrinho,String> coluna_pedido
```

♦ **coluna_total**

```
private javafx.scene.control.TableColumn<Carrinho,Float> coluna_total
```

♦ **coluna_status**

```
private javafx.scene.control.TableColumn<Carrinho,String> coluna_status
```

♦ **cliente**

```
private Cliente cliente
```

♦ **listaCarrinho**

```
private ArrayList<Carrinho> listaCarrinho
```

• Constructor Details

♦ **historicoController**

```
public historicoController()
```

• Method Details

♦ **lidarBotaoCardapio**

```
public void lidarBotaoCardapio(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface do cardÃ¡pio.

Parameters:

e - ActionEvent botÃ£o cardapio

♦ **lidarBotaoCarrinho**

```
public void lidarBotaoCarrinho(javafx.event.ActionEvent e)
```

MÃ©todo para mudar para a interface do carrinho.

Parameters:

e - ActionEvent botÃ£o carrinho

♦ **lidarBotaoSair**

```
public void lidarBotaoSair(javafx.event.ActionEvent e)
```

MÃ©todo para sair da conta do cliente.

Parameters:

e - ActionEvent botÃ£o sair

♦ **lidarBotaoVoltar**

```
public void lidarBotaoVoltar(javafx.event.ActionEvent e)
```

MÃ©todo para voltar para a interface do meu-pedido.

Parameters:

e - ActionEvent botÃ£o voltar

♦ **lidarBotaoCancelar**

```
public void lidarBotaoCancelar(javafx.event.ActionEvent e)
```

MÃ©todo para cancelar um pedido do histÃ³rico de pedidos.

Parameters:

e - ActionEvent botÃ£o cancelar

♦ **initialize**

```
public void initialize(URL url, ResourceBundle resourceBundle)
```

MÃ©todo para inicializar a tabela do histÃ³rico de pedidos.

Specified by:

```
initialize in interface javafx.fxml.Initializable
```

Parameters:

url -

resourceBundle -

JavaScript is disabled on your browser.

Skip navigation links

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Package:
 - ◆ [Description](#)
 - ◆ [Related Packages](#)
 - ◆ [Classes and Interfaces](#)
- Package:
- [Description |](#)
- [Related Packages |](#)
- [Classes and Interfaces](#)

SEARCH

Module [com.example.pedidoentrega](#)

Package com.example.pedidoentrega

package com.example.pedidoentrega

- All Classes and InterfacesClassesException Classes
 - [Class](#)
 - [Description](#)
 - [Administrador](#)
Classe Administrador herda a classe abstrata Usuario, possui uma lista de carrinhos que representa os pedidos dos clientes e guarda os dados do administrador.
 - [AlterarEnderecoController](#)
Classe Controller da interface alterarEndereco para alterar o endereço de entrega.
 - [CardapioController](#)
Classe Controller da interface cardapio.
 - [Carrinho](#)
Classe Carrinho define o status do pedido.
 - [CarrinhoController](#)
Classe controller da interface carrinho.
 - [Casquinha](#)
Classe Casquinha herda a classe abstrata Sorvete, guarda os dados do sorvete do tipo casquinha.
 - [Cliente](#)
Classe Cliente herda a classe abstrata Usuario, possui métodos para cancelar/pagar/fazer pedidos e alterar endereço.
 - [Copinho](#)
Classe Copinho herda a classe abstrata Sorvete, guarda os dados do sorvete de copinho.
 - [DisplaySorveteria](#)

Administrador

Classe DisplaySorveteria para fazer login, ler e atualizar arquivos.

Endereco

Classe para guardar os dados do endereço dos clientes.

Entregador

Classe Entregador herda a classe abstrata Usuario, guarda os dados dos entregadores.

EntregadoresController

Classe controller da interface com a lista de entregadores do administrador.

Estoque

Classe Estoque com métodos para seu gerenciamento.

EstoqueController

Classe controller da interface estoque do administrador.

HelloApplication

HelloController

Classe controller da tela inicial.

historicoController

Classe controller da interface de histórico de pedidos do cliente.

InterfaceUtil

ItemEstoque

Classe ItemEstoque personalizada para colocar na tabela do estoque.

LoggedAdministrador

Classe controller da interface logged-administrador, com a lista de pedidos da sorveteria.

LoggeInController

Classe controller da interface logged-in do cliente.

LogInController

Classe controller da interface log-in.

MenuEntregadorController

Classe controller da interface menu-entregador, com a lista de pedidos para entregar.

MeuPedidoController

Classe controller da interface meu-pedido, que mostra o status dos pedidos, o histórico dos pedidos e possui a função de cancelar pedidos.

Milkshake

Classe Milkshake herda a classe abstrata Sorvete, guarda os dados do milkshake.

MudarCardapioController

Classe controller da interface mudar-cardapio, permite adicionar novos sabores de sorvete.

PagamentoController

Classe controller da interface pagamento, para preencher os dados do cartão para pagamento, com opção de trocar o endereço de entrega.

Pedido<T extends Sorvete>

Classe Pedido usa Generics para poder receber qualquer tipo de Sorvete (Casquinha, Milkshake, Picolé).

Picole

Classe Picole herda a classe abstrata Sorvete, guarda os dados do picolé.

SignUpController

Classe controller da interface sign-up, que faz o cadastro de novos clientes.

SignUpEntregadorController

Classe controller da interface sign-up-entregador, para fazer o cadastro de novos entregadores, a função possui apenas para o administrador.

Sorvete

Administrador

Classe abstrata Sorvete, guarda atributos em comum dos tipos de sorvetes que herdam essa classe, como o sabor, preço e quantidade de estoque.

SorveteriaSistema

Classe para cadastrar usuários, fazer login e deslogar.

Usuario

Classe abstrata usuário com atributos em comum entre todos os usuários que herdam a classe.

UsuarioInexistenteException

Classe para lançar exceção caso o usuário não exista.

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Module](#)
- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

[SEARCH](#)

Hierarchy For Package com.example.pedidoentrega

Class Hierarchy

- [java.lang.Object](#)
 - ◆ [com.example.pedidoentrega.AlterarEnderecoController](#)
 - ◆ [javafx.application.Application](#)
 - ◇ [com.example.pedidoentrega.HelloApplication](#)
 - ◆ [com.example.pedidoentrega.CardapioController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.Carrinho](#)
 - ◆ [com.example.pedidoentrega.CarrinhoController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.DisplaySorveteria](#)
 - ◆ [com.example.pedidoentrega.Endereco](#)
 - ◆ [com.example.pedidoentrega.EntregadoresController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.Estoque](#)
 - ◆ [com.example.pedidoentrega.EstoqueController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.HelloController](#)
 - ◆ [com.example.pedidoentrega.historicoController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.InterfaceUtil](#)
 - ◆ [com.example.pedidoentrega.ItemEstoque](#)
 - ◆ [com.example.pedidoentrega.LoggedAdministrador](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.LoggeInController](#)
 - ◆ [com.example.pedidoentrega.LogInController](#)
 - ◆ [com.example.pedidoentrega.MenuEntregadorController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.MeuPedidoController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.MudarCardapioController](#)
 - ◆ [com.example.pedidoentrega.PagamentoController](#) (implements [javafx.fxml.Initializable](#))
 - ◆ [com.example.pedidoentrega.Pedido<T>](#)
 - ◆ [com.example.pedidoentrega.SignUpController](#)
 - ◆ [com.example.pedidoentrega.SignUpEntregadorController](#)
 - ◆ [com.example.pedidoentrega.Sorvete](#)
 - ◇ [com.example.pedidoentrega.Casquinha](#)
 - ◇ [com.example.pedidoentrega.Copinho](#)
 - ◇ [com.example.pedidoentrega.Milkshake](#)
 - ◇ [com.example.pedidoentrega.Picole](#)
 - ◆ [com.example.pedidoentrega.SorveteriaSistema](#)
 - ◆ [java.lang.Throwable](#) (implements [java.io.Serializable](#))

Administrador

- ◇ java.lang.Exception
 - com.example.pedidoentrega.UsuarioInexistenteException
- ◆ com.example.pedidoentrega.Usuario
 - ◇ com.example.pedidoentrega.Administrador
 - ◇ com.example.pedidoentrega.Cliente
 - ◇ com.example.pedidoentrega.Entregador