

# Componente Cartao

---

O componente Cartao oferece uma interface de verificação de dados da bandeira de um determinado cartão. Ele faz parte do pagamento e pode ou não ser utilizado durante o processo.

## Interfaces

- ICartao

## Detalhamento das Interfaces

---

### Interface ICartao

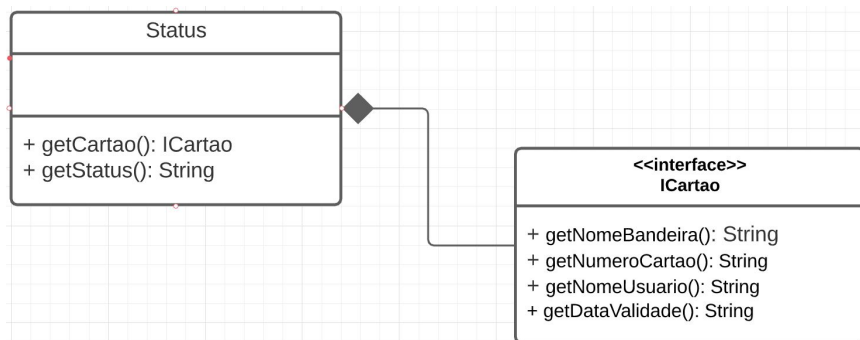
<<interface>> ICartao
+ getNomeBandeira(): String + getNumeroCartao(): String + getNomeUsuario(): String + getDataValidade(): String

Utilizado para agrupar os dados fornecidos pelo usuário.

Método	Objetivo
getNomeBandeira()	Recupera o nome da bandeira do cartão utilizado
getNumeroCartao()	Recupera o número do cartão inserido pelo usuário
getNomeUsuario()	Recupera o nome preenchido na interface gráfica
getDataValidade()	Recupera a data de validade do cartão utilizado

### Assina: *resposta/dados/{cartao}/{status}*

- Este tópico faz parte de um barramento que entrega mensagens de status referentes ao cartão utilizado. Os dados transitados neste tópico são:



```
{
  "Cartao": {
    "Bandeira": "Teste",
    "Numero": "1234 1234 1234 1234",
    "Nome": "Joao da Silva",
    "Validade": date
  },
  "Status": "OK"
}
```

Atributo	Descrição
Cartao	Dados referentes ao cartão do usuário
Status	Utilizado para verificar se o cartão possui algum problema

### Posta: *consulta/dados/{cartao}*

- Este tópico é utilizado para que seja possível pedir ao controller que valide os dados do cartão. Os dados transitados neste tópico são exatamente os dados de cartão já descritos acima.

```
{
  "Bandeira": "Teste",
  "Numero": "1234 1234 1234 1234",
  "Nome": "Joao da Silva",
  "Validade": date
}
```

# Componente Pagamento

---

O componente Pagamento oferece serviços voltados ao uso de boletos ou cartões dentro do sistema. Internamente possui o papel de transitar os dados de usuário com o objetivo de validar os mesmos. Por default, a compra pode ser feita por boleto (lógica interna) entretanto existe uma comunicação com o componente de **Cartao** no caso do usuário escolher a opção de pagamento por cartão. Para computar o valor total do serviço, o componente requer os dados de Frete, que contém o valor do mesmo.

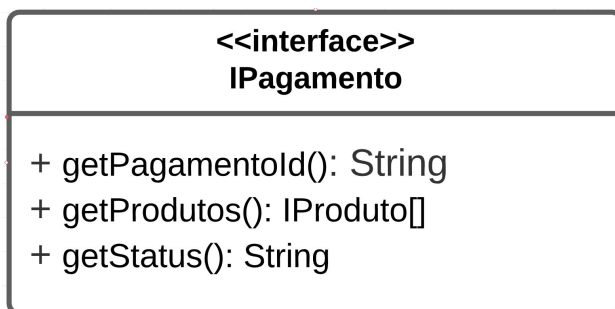
## Interfaces

- IPagamento
- IFrete

## Detalhamento das Interfaces

---

### Interface IPagamento

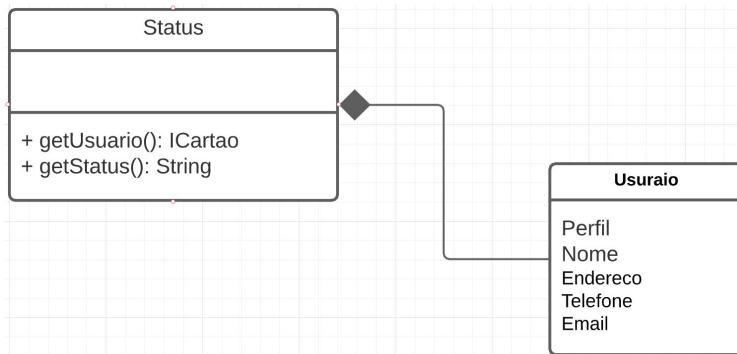


Utilizado para encapsular os dados referentes ao pagamento que o usuário deve realizar para prosseguir com o pedido.

Método	Objetivo
getPagamentId()	Utilizado para vincular o ID do pagamento ao banco
getProdutos()	Utilizado para verificar a lista de produtos daquele pagamento
getStatus()	Verifica o status atual de pagamento

### Assina: *resposta/dados/{usuario}/{status}*

- Este tópico tem o objetivo de transitar informações sobre a integridade do usuário que está utilizando o sistema.



```

{
  "Usuario":
  {
    "perfil": "cliente",
    "nome": "",
    "endereco": "",
    "telefone": "",
    "email": ""
  },
  "Status": "OK"
}

```

Atributo	Descrição
Usuario	Dados referentes diretamente ao usuário
Status	Utilizado para verificar se o usuário possui algum problema

### **Posta: *consulta/dados/{usuario}***

- Tópico utilizado para garantir uma validação assíncrona do usuário utilizando o controller. Os dados que transitam neste tópico tem o formato Usuario:

```

{
  "perfil": "cliente",
  "nome": "",
  "endereco": "",
  "telefone": "",
  "email": ""
}

```

# Componente Frete

---

O componente de frete tem o objetivo de fornecer ao usuário a transparência de se obter um cálculo do valor de transporte para seus produtos. Para seu funcionamento é necessário apenas o CEP do endereço.

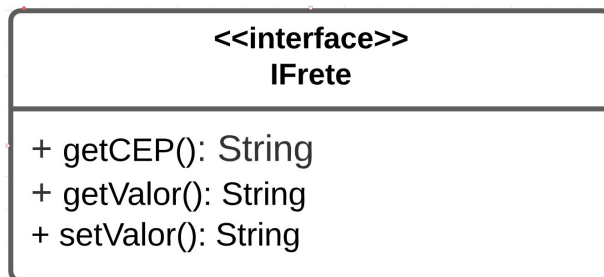
## Interfaces

- IFrete

## Detalhamento das Interfaces

---

### Interface IFrete



Esta interface é utilizada para transitar os dados do preço sugerido para transportar o produto até o seu destino.

Método	Objetivo
getCEP()	Recupera o CEP da região escolhida
getValor()	Recupera o valor total do transporte
setValor()	Insere o valor do frete

### Assina: *resposta/dados/{cep}/{valor}*

- Este tópico foi criado com objetivo de transitar os dados de valor do frete para um determinado CEP.

Frete
+ CEP: String + Valor: String

```
{
  "CEP": "08071-190",
  "Valor": "R$ 00,00"
}
```

Atributo	Descrição
CEP	Campo contendo o CEP do destinatário
Valor	Campo onde será atribuído o valor do frete

### **Posta: *consulta/dados/{CEP}***

- Este tópico é utilizado para disparar a consulta de preço do frete para um determinado CEP. O único dado necessário para este fluxo é o CEP do destinatário.

```
{
  "CEP": "08071-190"
}
```

# Componente Gerencia Preenchimento

Este componente é responsável por iniciar o fluxo de pagamento através da interface gráfica disponibilizada para o usuário. Aqui o usuário poderá preencher os formulários referentes ao seu requerimento de criação de um novo pedido.

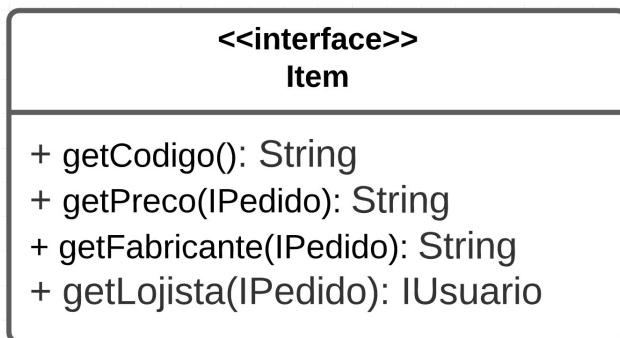
## Interfaces

- IPagamento
- IPedido
- Item[]

## Detalhamento das Interfaces

---

### Interface Item[]



Esta interface é utilizada para armazenar os itens que o usuário pretende adicionar em seu pedido.

Método	Objetivo
getCodigo()	Recupera o código do produto selecionado
getPreco()	Recupera o valor do produto
getFabricante()	Recupera o fabricante original do produto
getLojista()	Campo para informar o fornecedor do produto

### Interface IPagamento

A interface de pagamento é requerida neste componente durante o processo de criação do formulário que o usuário deve preencher para realizar seu pedido.

**Interface IPedido**

Esta interface (Já descrita) simboliza o pedido completo para que seja possível mostrar ao usuário.



# Componente Armazena Pedido

---

O componente Armazena Pedido oferece uma interface de pedido para persistência em banco de dados e posterior consulta do histórico de vendas. Ele se comunica com o componente PedidoDAO representado na camada Model do modelo MVC.

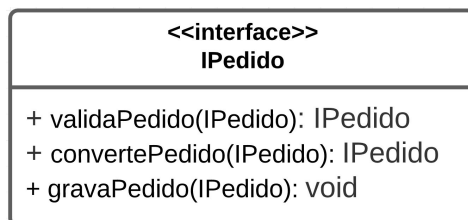
## Interfaces

- IPedido

## Detalhamento das Interfaces

---

### Interface IPedido



Utilizado para agrupar os dados relacionados ao pedido fornecidos pelo usuário.

Método	Objetivo
validaPedido(IPedido)	Valida todos os atributos de IPedido seguindo as regras definidas pela área de negócio
convertePedido(IPedido)	Aplica o padrão Objeto de Transferência de Dados (DTO) para converter o IPedido da camada de Controller para a camada Model
gravaPedido(IPedido)	Envia o IPedido para a camada Model persistir no banco de dados

# Componente PedidoDAO

---

O componente de PedidoDAO tem o objetivo de receber o pedido da camada Controller para persistência no banco de dados. Para seu funcionamento é necessário que o objeto pedido seja fornecido com as devidas validações dos atributos e conversão para o modelo de dados da camada Model.

## Interfaces

- IPedido

## Detalhamento da Interfaces

---

### Interface IPedido

Na camada Model essa interface é utilizada para transitar os dados de pedido entre a camada Controller e a camada Model, nesta etapa os dados do objeto foram transformados pelos métodos validaPedido e convertePedido, responsáveis por garantir que a informação enviada do Controller esteja fiel ao modelo esperado na camada Model.

# Componente Valida Cartao

---

O componente Valida Cartao oferece uma interface de cartão com as devidas validações aplicadas com o objetivo de assegurar vendas fidedignas ao lojista, minimizando assim possíveis fraudes. Para seu funcionamento é necessário o objeto cartão fornecido pelo usuário.

## Interfaces

- ICartao

## Detalhamento das Interfaces

---

### Interface ICartao

No componente Valida Cartao a interface cartão é utilizada para transitar os dados fornecidos pelo usuário para que os mesmos sejam validados por uma Autoridade Certificadora e publica no barramento os dados do cartão validados.

#### **Assina: *resposta/dados/{cartao}/{status}***

- Este tópico foi criado com objetivo de transitar os dados do cartão para para uma Autoridade Certificadora.

```
{
  "Cartao": {
    "Bandeira": "Teste",
    "Numero": "1234 1234 1234 1234",
    "Nome": "Joao da Silva",
    "Validade": date
  }, "Status": "Pendente"
}
```

#### **Posta: *consulta/dados/{cartao}***

- Este tópico é utilizado para disparar os dados validados do cartão do usuário. As informações necessárias para este fluxo são os dados homologados.

```
{
  "Cartao": {
    "Bandeira": "Teste",
    "Numero": "1234 1234 1234 1234",
    "Nome": "Joao da Silva",
    "Validade": date
  }, "Status": "Ok"
}
```

# Componente Valida Usuario

---

O componente Valida Usuario oferece uma interface de usuário para validação dos dados com o objetivo de assegurar vendas fidedignas ao lojista, minimizando assim possíveis fraudes. Para seu funcionamento é necessário o objeto usuário fornecido pelo cliente.

## Interfaces

- IUsuario

## Detalhamento das Interfaces

---

### Interface IUsuario

No componente Valida Usuario a interface usuário é utilizada para transitar os dados fornecidos pelo usuário para que os mesmos sejam validados com base nas regras fornecidos pela área de negócio e publica no barramento os dados do usuário validados.

#### **Assina: *resposta/dados/{usuario}/{status}***

- Este tópico foi criado com objetivo de transitar os dados do usuário.

```
{
  "Usuario": {
    "perfil": "cliente",
    "nome": "",
    "endereco": "",
    "telefone": "",
    "email": ""
  }, "Status": "Pendente"
}
```

#### **Posta: *consulta/dados/{usuario}***

- Este tópico é utilizado para disparar os dados do usuário. As informações necessárias para este fluxo são os dados do usuário homologados.

```
{
  "Usuario": {
    "perfil": "cliente",
    "nome": "",
    "endereco": "",
    "telefone": "",
    "email": ""
  }, "Status": "Ok"
}
```

# Componente Valida CEP

---

O componente Valida CEP oferece uma interface de frete para validação dos dados relacionados ao local de entrega com o objetivo de assegurar vendas fidedignas ao lojista, minimizando assim possíveis fraudes. Para seu funcionamento é necessário o CEP.

## Interfaces

- IFrete

## Detalhamento das Interfaces

---

### Interface IFrete

No componente Valida CEP a interface frete é utilizada para transitar o CEP fornecido pelo usuário para que o mesmo seja validado e publica no barramento os endereço validado.

#### **Assina:** *resposta/dados/{cep}/{valor}*

- Este tópico foi criado com objetivo de transitar os dados do frete.

```
{
  "CEP": "08071-190",
  "Valor": "R$ 00,00"
}
```

#### **Posta:** *consulta/dados/{cep}*

- Este tópico é utilizado para disparar os dados do usuário. A informação necessária para este fluxo é o endereço homologado.

```
{
  "CEP": "08071-190"
}
```

# Componente Gerar Pedido

Este componente é responsável por iniciar o fluxo de geração do pedido através da lista de produtos disponibilizada pela camada View. Com base na lista de produtos o componente inicia execuções assíncronas para os componentes: Valida Usuario, Valida Cartao, Valida CEP e Armazena Pedido culminando na criação do pedido com todas as validações necessárias.

## Interfaces

- IPedido
- Item[]
- ICartao
- IUsuario
- IFrete

## Detalhamento das Interfaces

---

### Interface IPedido

Esta interface (Já descrita) simboliza o pedido completo para que seja possível mostrar ao usuário.

### Interface Item[]

Esta interface é utilizada para armazenar os itens que o usuário pretende adicionar em seu pedido.

### Interface ICartao

Esta interface (Já descrita) simboliza os dados do cartão validados por uma Autoridade Certificadora.

### Interface IUsuario

Esta interface (Já descrita) simboliza os dados do usuário validados com base nas regras definidas pela área de negócio.

### Interface IFrete

Esta interface (Já descrita) simboliza o endereço validado através do CEP.