

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO RIO GRANDE DO NORTE  
CAMPUS NATAL CENTRAL  
DIRETORIA ACADÊMICA DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO

## **SISTEMA PARA LANCHONETE**

NEEMIAS RENAN SANTOS DE OLIVEIRA - 20211014040062

HILQUIAS ABIAS FIGUEIREDO SILVA – 20201014040081

Natal-RN, Agosto de 2022

## Projeto POO – FINAL

O nosso projeto tem como intuito desenvolver um software voltado para uma lanchonete que terá como usuário final o vendedor/garçom do estabelecimento.

Área de gerenciamento - nessa área, o usuário é responsável pelas funções abaixo:

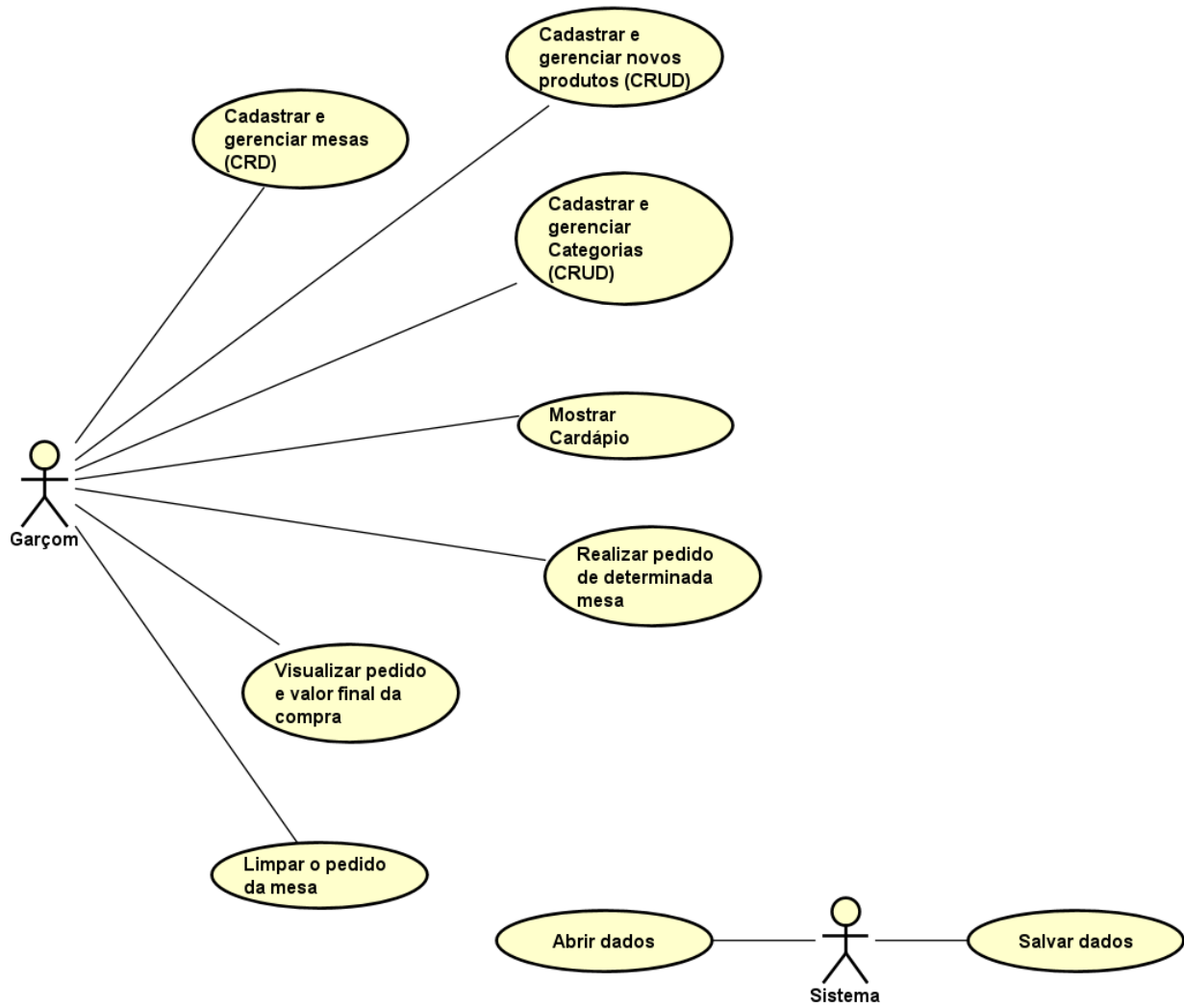
- Cadastrar, listar e remover mesas (abrir e fechar);
- Cadastrar, atualizar, listar e remover uma categoria de produtos;
- Cadastrar, atualizar, listar e remover um determinado produto;

Área de Realização de pedidos - nessa área, o usuário definirá qual mesa deseja operar e em seguida será possível realizar as seguintes ações:

- Visualizar o cardápio;
- Realizar o pedido;
- Visualizar os pedidos realizados pela mesa, calculando a soma total dos valores dos produtos;
- Limpar os pedidos realizados pela mesa;

O código do projeto, junto com toda documentação estará hospedado no Github e será possível acessá-lo a partir do seguinte endereço;

<https://github.com/neemias-renan/poo-projetofinal>



**MainClass**

- ncategoria : NCategoria
- nproduto : NProduto
- nmesa : NMesa
- nvenda : NVenda
- mesaLogin : Mesa
- MesaVenda : Venda

- + Main() : void
- + MenuUsuario() : void
- + MenuVendedorCategoria() : void
- + MenuVendedorProduto() : void
- + MenuVendedorMesa() : void
- + MenuVendedorOpcao() : void
- + MenuMesaLogout() : void
- + MenuMesaLogin() : void
- + CategoriaListar() : void
- + CategoriaInserir() : void
- + CategoriaAtualizar() : void
- + CategoriaExcluir() : void
- + ProdutoListar() : void
- + ProdutoInserir() : void
- + ProdutoAtualizar() : void
- + ProdutoExcluir() : void
- + MesaListar() : void
- + MesaInserir() : void
- + MesaExcluir() : void
- + VendaListar() : void
- + DefinirMesa() : void
- + MesaLogout() : void
- + MesaProdutoListar() : void
- + MesaProdutoInserir() : void
- + MesaPedidoVisualizar() : void
- + MesaPedidoLimpar() : void

**Arquivo**

+ Abrir(Arquivo : string) : T  
 + Salvar(Arquivo : List<Venda>, obj : int) : void

**NCategoria**

- Categorias : Categoria[]  
 - NCategoria : int  
 - obj : NCategoria  
 + <<get>> Singleton : NCategoria

- NCategoria() : void  
 + Listar() : Categoria  
 + Listar(id : int) : Categoria  
 + Inserir(c : Categoria) : void  
 + Atualizar(c : Categoria) : void  
 - Indice(c : Categoria) : int  
 + Excluir(c : Categoria) : void  
 + Abrir() : void  
 + Salvar() : void

**NMesa**

- Mesas : List<Mesa>  
 - obj : NMesa  
 + <<get>> Singleton : NMesa

- NMesa() : void  
 + Listar() : List<Mesa>  
 + Listar(id : int) : Mesa  
 + Inserir(c : Mesa) : void  
 + Excluir(c : Mesa) : void  
 + Abrir() : void  
 + Salvar() : void

**NVendas**

- Vendas : List<Venda>  
 - obj : NVendas  
 + <<get>> Singleton : NVendas

- NVenda()  
 + Listar() : List<Venda>  
 + Listar(c : Mesa) : List<Venda>  
 + ListarPedido(p : pedido) : Venda  
 + Inserir(v : Venda, carrinho : boolean) : void  
 + ItemListar(v : Venda) : List<VendaItem>  
 + ItemInserir(v : Venda, Qtd : int, p : Produto) : void  
 + ItemExcluir(v : Venda) : void  
 + Abrir() : void  
 + Salvar() : void

**NProdutos**

- Produtos : Produto[]  
 - contadordeprodutos : int  
 - obj : NProdutos  
 + <<get>> Singleton : NProdutos

- NProduto()  
 + Listar() : Produto[]  
 + Listar(id : int) : Produto  
 + Inserir(p : Produto) : void  
 + Atualizar(p : Produto) : void  
 + Indice(p : Produto) : void  
 + Excluir(p : Produto) : void  
 + Abrir() : void  
 + AtualizarCategoria() : void  
 + Salvar() : void

**Categoria**

- id : int  
 - descricao : string  
 - produtos : produtos[]  
 - contadorproduto : int  
 - <<get,set>> Id : int  
 - <<get,set>> Descricao : int

+ Categoria()  
 + Categoria(Id : int, Descricao : string) : void  
 + SetId(id : int) : void  
 + SetDescricao(descricao : string) : void  
 + GetId() : int  
 + GetDescricao() : string  
 + ProdutoListar() : produto[]  
 + ProdutoInserir(p : produto) : void  
 + ProdutoIndice(p : produto) : int  
 + ProdutoExcluir(p : produto) : void  
 + ToString() : string

**Mesa**

- id : int  
 - <<get,set>> Id : int

+ CompareTo(Mesa : obj) : int  
 + ToString() : string

**Produto**

- id : int  
 - descricao : string  
 - qtd : int  
 - preco : double  
 - categoriald : int  
 - <<get,set>> Id : int  
 - <<get,set>> Descricao : string  
 - <<get,set>> Qtd : int  
 - <<get,set>> Preco : double  
 - <<get,set>> Categoricald : int

+ Produto()  
 + SetId(id : int) : void  
 + SetDescricao(descricao : string) : void  
 + SetQtd(qtd : int) : void  
 + SetPreco(preco : double) : void  
 + SetCategoria(categoria : Categoria) : void  
 + GetId() : int  
 + GetDescricao() : string  
 + GetQtd() : int  
 + GetPreco() : double  
 + GetCategoria() : Categoria  
 + ToString() : string

**Venda**

- id : int  
 - data : DateTime  
 - pedido : boolean  
 - mesa : Mesa  
 - mesald : int  
 - itens : List<Vendaltem>  
 - <<get,set>> Id : int  
 - <<get,set>> Data : DateTime  
 - <<get,set>> Pedido : boolean  
 - <<get,set>> Mesald : int  
 - <<get,set>> Itens : List<Vendaltem>

+ Venda()  
 + Venda(DateTime : int, mesa : Mesa)  
 + SetId(id : int) : void  
 + SetData(data : DateTime) : void  
 + SetPedido(pedido : boolean) : void  
 + SetMesa(mesa : Mesa) : void  
 + GetId() : int  
 + GetData() : DateTime  
 + GetMesa() : Mesa  
 + ItemListar(p : Produto) : Vendaltem  
 + ItemListar() : List<Vendaltem>  
 + ItemInserir(qtd : int, p : Produto) : void  
 + ItemExcluir() : void  
 + ToString() : string

**Vendaltem.**

- qtd : int  
 - preco : double  
 - produto : Produto  
 - produtold : int  
 - <<get,set>> Qtd : int  
 - <<get,set>> Preco : double  
 - <<get,set>> Produtold : int

+ Vendaltem()  
 + Vendaltem(qtd : int, produto : Produto)  
 + SetQtd(qtd : int) : void  
 + SetPreco(preco : double) : void  
 + SetProduto(produto Produto : int) : void  
 + GetQtd() : int  
 + GetPreco() : double  
 + GetProduto() : Produto