

<<Interface>> ICalcular	<<Interface>> IProduto	<<Interface>> IVenda	<<Interface>> IRelatorio
+ calcularTotal(): double + aplicarDesconto(double): void	+ adicionarEstoque(int): void + removerEstoque(int): void	+ receberPagamento(double): void + emitirNotaFiscal(): void	+ imprimirRelatorio(): void

Carrinho	Cliente	Funcionario	PagamentoImpl
- produtos: list<Produto> - total: double - quantidadeProdutos: int - fechado: boolean - desconto: double - cliente: string	- endereco: string - telefone: string - email: string - historicoCompras: List<String> - saldo: double - statusAtivo: boolean	- nivelFuncionario: NivelFuncionario - cargo: string - salario: double - horasTrabalhadas: int - vendasRealizadas: int - ativo: boolean - turno: string	- idTransacao: string - dataPagamento: LocalDateTime - nomeCliente: string - confirmado: boolean - troco: double
+ calcularTotal(type): double + adicionarProduto(Produto): void + removerProduto(Produto): void + aplicarDesconto(double): void + finalizarCarrinho(): void + imprimirRelatorio(type): void	+ atualizarEndereco(String): void + adicionarSaldo(double): void + registrarCompra(String): void + realizarCompra(double): void + alterarStatus(boolean): void	+ registrarHoras(int): void + registrarVenda(): void + calcularSalarioFinal(): double + alterarStatus(boolean): void + imprimirRelatorio(): void	+ gerarIdTransacao(): string + processarPagamento(): void + cancelarPagamento(): void + calcularTroco(double): void + imprimirRelatorio(): void + verificarStatusPagamento(): boolean

ProdutoImpl
- quantidadeEstoque: int - desconto: double - fornecedor: String - dataValidade: String - disponivel: boolean - produtos: Map<Integer, Produto>
+ calcularPrecoFinal(): double + adicionarEstoque(int): void + removerEstoque(int): void + verificarDisponibilidade(): boolean + imprimirRelatorio(): void

<<enum>> NivelFuncionario
+ CAIXA + GERENTE + DIRETOR

<<enum>> TipoPagamento
+ DINHEIRO + CARTAO + PIX

VendaImpl
- dataVenda: LocalDateTime - metodoPagamento: TipoPagamento - notaFiscalEmitida: boolean - desconto: double - valorPago: double - erros: List<String>
+ receberPagamento(double): void + realizarVenda(): void + aplicarDesconto(double): void + emitirNotaFiscal(): void + imprimirRelatorio(): void + getValorPago(): double

<<enum>> StatusVenda
+ PENDENTE + REALIZADA + PAGA

<<enum>> TipoProduto
+ ALIMENTO + BEBIDA + ROUPA + ELETRONICO

<<abstract>> Pagamento
valor: double # tipoPagamento: TipoPagamento
+ processarPagamento(): void

<<abstract>> Produto
nome: string # preco: double # tipoProduto: TipoProduto # codigo: int # descricao: string
+ calcularPrecoFinal(): double + getNome(): string + getCodigo(): int + getPreco(): double + getDescricao(): string + getTipoProduto(): TipoProduto

<<abstract>> Classname
nome: string # cpf: string
+ getNome(): string + getCpf(): string

<<abstract>> Venda
cliente: Cliente # funcionario: Funcionario # statusVenda: StatusVenda # carrinho: Carrinho # total: double
+ method(type): type