

```

using System;

namespace ConsoleAppValorVenda
{
    // Desenvolva um algoritmo para calcular o valor de venda de cada item e o valor total da venda.

    0 referências | miguel.galego, Há 20 minutos | 1 autor, 2 alterações
    class Program
    {
        0 referências | miguel.galego, Há 20 minutos | 1 autor, 2 alterações
        static void Main(string[] args)
        {
            var venda = VendaRepositorio.GerarVendaCapa();

            var vendaServico = new VendaServico(venda);

            foreach (var produto in ProdutoRepositorio.Produtos)
            {
                vendaServico.AdicionarItem(produto);
            }

            vendaServico.Totalizar();

            Console.WriteLine("Pressione uma tecla para continuar...");
            Console.ReadKey();
        }
    }
}

```

```

1 referência | miguel.galego, Há 21 minutos | 1 autor, 2 alterações
public void AdicionarItem(Produto produto, double qtde = 1)
{
    var vendaItem = new VendaItem
    (
        venda: Venda,
        produto: produto,
        qtde: qtde,
        valorUnitario: produto.ValorVenda
    );

    var regraFiscal = vendaItem.BuscarRegraFiscal();
    if (regraFiscal is null)
        throw new Exception("Regra fiscal não encontrada");

    var conceito = vendaItem.BuscarConceito();
    if (conceito is null)
        throw new Exception("Conceito não encontrado");

    Venda.Itens.Add(vendaItem);

    vendaItem.CalcularValorDesconto(conceito);

    vendaItem.CalcularValorImposto(regraFiscal);
}

```

1 referência | miguel.galego, Há 21 minutos | 1 autor, 2 alterações

```
public void Totalizar()
{
    Venda.CalcularValorAcrescimo();

    Venda.TotalizarVenda();

    Venda.GerarPagtos();
}
```

0 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração

```
public static class ImpostoServico
```

```
{
    1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração
    private static Func<RegraFiscal, bool> FiltrarRegraFiscal(VendaItem vendaItem)
    {
        return (regra) =>
            regra.Finalidade.Equals(vendaItem.Venda.Finalidade) &&
            regra.Grupo.Equals(vendaItem.Produto.Grupo) &&
            regra.UfOrigem.Equals(vendaItem.Venda.Cooperativa.Municipio.Uf) &&
            regra.UfConsumo.Equals(vendaItem.Venda.Cooperado.Municipio.Uf) &&
            regra.TipoPessoa.Equals(vendaItem.Venda.Cooperado.TipoPessoa);
    }
}
```

1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração

```
public static RegraFiscal BuscarRegraFiscal(this VendaItem vendaItem)
```

```
{
    return RegraFiscalRepositorio.RegraFiscals.FirstOrDefault(FiltrarRegraFiscal(vendaItem));
}
```

1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração

```
public static void CalcularValorImposto(this VendaItem vendaItem, RegraFiscal regraFiscal)
```

```
{
    if (regraFiscal.IndIsento)
        return;

    vendaItem.ValorImposto =
        vendaItem.ValorLiquido * regraFiscal.PercImposto / 100;
}
```

```

private const int _qtdeMaximaDescontoGrupo = 5;

1 referência | miguel.galego, Há 23 minutos | 1 autor, 4 alterações
public static void CalcularValorDesconto(this VendaItem vendaItem, Conceito conceito)
{
    var percDesconto = conceito.PercDesconto;

    var percDescontoPorQtdeGrupos = vendaItem.Venda.Itens
        .Select(item => item.Produto.Grupo)
        .GroupBy(grup => grup.Codigo)
        .Select(s => s.Key)
        .Count();

    if (percDescontoPorQtdeGrupos > _qtdeMaximaDescontoGrupo)
    {
        percDescontoPorQtdeGrupos = _qtdeMaximaDescontoGrupo;
    }

    percDesconto += percDescontoPorQtdeGrupos;

    vendaItem.ValorDesconto =
        percDesconto * vendaItem.ValorBruto / 100;

    vendaItem.ValorLiquido =
        vendaItem.ValorBruto - vendaItem.ValorDesconto;
}
}

```

```

using System;
using System.Linq;

namespace ConsoleAppValorVenda
{
    0 referências | miguel.galego, Há 24 minutos | 1 autor, 2 alterações
    public static class acrescimoServico
    {
        1 referência | miguel.galego, Há 24 minutos | 1 autor, 2 alterações
        public static void CalcularValorAcrescimo(this Venda venda)
        {
            //Nas vendas a prazo é acrescentado 0,05 % de juros ao dia conforme a seguinte fórmula:
            //Valor de Juros = Valor principal * (( 1 + (taxa de juros/ 100)) ^nro de dias) - 1), onde nro
            //a quantidade de dias úteis entre a data de compra e data de vencimento;

            var taxaJuros = 0.05;
            var numeroDiasUteis = venda.FormaPagto.CalcularNumeroDiasUteis(venda.Data);
            var prazoMedioPagto = numeroDiasUteis / venda.FormaPagto.Parcelas.Count();
            var valorTotalLiquido = venda.Itens.Sum(x => x.ValorLiquido);
            var valorTotalAcrescimo =
                valorTotalLiquido * (Math.Pow(1 + (taxaJuros / 100), prazoMedioPagto) - 1);
            venda.ValorAcrescimo = Math.Round(valorTotalAcrescimo, 2);
        }
    }
}

```

```

using System.Linq;

namespace ConsoleAppValorVenda
{
    0 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração
    public static class TotalServico
    {
        1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração
        public static void TotalizarVenda(this Venda venda)
        {
            venda.ValorProduto = venda.Itens.Sum(x => x.ValorLiquido);
            venda.ValorDesconto = venda.Itens.Sum(x => x.ValorDesconto);
            venda.ValorImposto = venda.Itens.Sum(x => x.ValorImposto);
            venda.ValorTotal = venda.Itens.Sum(x => x.ValorLiquido + x.ValorImposto) + venda.ValorAcrescimo;
        }
    }
}

```

```

namespace ConsoleAppValorVenda
{
    0 referências | miguel.galego, Há 26 minutos | 1 autor, 3 alterações
    public static class PagtoServico
    {
        1 referência | miguel.galego, Há 26 minutos | 1 autor, 3 alterações
        public static void GerarPagtos(this Venda venda)
        {
            venda.Pagtos.Clear();

            var qtdeParcela = venda.FormaPagto.Parcelas.Count;
            var valorParcela = Math.Round(venda.ValorTotal / qtdeParcela, 2);
            var valorResto = Math.Round(venda.ValorTotal - (valorParcela * qtdeParcela), 2);

            foreach (var parcela in venda.FormaPagto.Parcelas)
            {
                var dataVencito = venda.Data.AddDays(parcela.NumeroDias);
                var vendaPagto = new VendaPagto(parcela.SeqParcela, dataVencito, Math.Round(valorParcela, 2));
                venda.Pagtos.Add(vendaPagto);

                if (parcela.SeqParcela == 1)
                {
                    vendaPagto.ValorParcela = Math.Round(vendaPagto.ValorParcela + valorResto, 2);
                    valorResto = 0;
                }
            }
        }
    }
}

```

```

namespace ConsoleAppValorVenda
{
    3 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração
    public static class ConceitoRepositorio
    {
        public static Conceito[] Conceitos =
        {
            new Conceito(ConceitoTipo.A, 5.0),
            new Conceito(ConceitoTipo.B, 3.0),
            new Conceito(ConceitoTipo.C, 0.0),
        };
    }
}

```

```

using System;

namespace ConsoleAppValorVenda
{
    1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração
    public static class CooperadoRepositorio
    {
        public static Cooperado[] Cooperados =
        {
            new CooperadoFisica(1, "Jose", "Rua", MunicipioRepositorio.Municipios[0],
                ConceitoRepositorio.Conceitos[0],
                "11122233344", "67778889", EstCivilTipo.Casado, new DateTime(1980, 1, 1)),
            new CooperadoJuridica(1, "Jose", "Rua", MunicipioRepositorio.Municipios[0],
                ConceitoRepositorio.Conceitos[0],
                new DateTime(1980, 1, 1), null),
        };
    }
}

```

namespace ConsoleAppValorVenda

```
{  
    1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração  
    public static class CooperativaRepositorio  
    {  
        public static Cooperativa[] Cooperativas =  
        {  
            new Cooperativa(1, "Campo Mourao", MunicipioRepositorio.Municipios[0]),  
        };  
    }  
}
```

namespace ConsoleAppValorVenda

```
{  
    2 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração  
    public static class FinalidadeRepositorio  
    {  
        public static Finalidade[] Finalidades =  
        {  
            new Finalidade(1, "Venda")  
        };  
    }  
}
```

```

namespace ConsoleAppValorVenda
{
    1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração
    public class FormaPagtoRepositorio
    {
        public static FormaPagtoParcela[] FormaPagtoParcelasAVista =
        {
            new FormaPagtoParcela(1, 0),
        };

        public static FormaPagto FormaPagtoAVista =
            new FormaPagto(1, "A Vista", FormaPagtoParcelasAVista);

        public static FormaPagtoParcela[] FormaPagtoParcelasAPrazo =
        {
            new FormaPagtoParcela(1, 30),
            new FormaPagtoParcela(2, 60),
            new FormaPagtoParcela(3, 90),
        };

        public static FormaPagto FormaPagtoAPrazo =
            new FormaPagto(1, "A Prazo", FormaPagtoParcelasAPrazo);
    }
}

```

```

namespace ConsoleAppValorVenda
{
    6 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração
    public static class GrupoRepositorio
    {
        public static Grupo[] Grupos =
        {
            new Grupo(1, "Fertilizantes"),
            new Grupo(2, "Corretivos"),
            new Grupo(3, "Herbicidas"),
            new Grupo(4, "Fungicidas"),
            new Grupo(5, "Inseticidas")
        };
    }
}

```

```
namespace ConsoleAppValorVenda
```

```
{  
    3 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração  
    public static class MunicipioRepositorio  
    {  
        public static Municipio[] Municipios =  
        {  
            new Municipio(1, "Campo Mourao", "PR"),  
            new Municipio(2, "Florianopolis", "SC"),  
            new Municipio(3, "Campo Grande", "MS"),  
        };  
    }  
}
```

```
namespace ConsoleAppValorVenda
```

```
{  
    1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração  
    public static class ProdutoRepositorio  
    {  
        public static Produto[] Produtos =  
        {  
            new Produto(1, GrupoRepositorio.Grupos[0], "Fertilizante - A", null, null, "KG", 150),  
            new Produto(1, GrupoRepositorio.Grupos[1], "Corretivo - A", null, null, "KG", 150),  
            new Produto(1, GrupoRepositorio.Grupos[2], "Herbicida - A", null, null, "KG", 150),  
            new Produto(1, GrupoRepositorio.Grupos[3], "Fungicida - A", null, null, "KG", 150),  
            new Produto(1, GrupoRepositorio.Grupos[4], "Inseticida - A", null, null, "KG", 150),  
        };  
    }  
}
```



```
public static RegraFiscal[] RegraFiscals;
```

```
0 referências | miguel.galego, há 15 horas | 1 autor, 1 alteração  
static RegraFiscalRepositorio()
```

```
{  
    RegraFiscals = GetRegraFiscals();  
}
```

```
1 referência | miguel.galego, há 15 horas | 1 autor, 1 alteração
```

```
private static RegraFiscal[] GetRegraFiscals()
```

```
{  
    var regraFiscals = new List<RegraFiscal>();  
  
    foreach (var finalidade in FinalidadeRespositorio.Finalidades)  
    {  
        foreach (var grupos in GrupoRepositorio.Grupos)  
        {  
            foreach (var uf in _ufs)  
            {  
                var regraFiscal = new RegraFiscal(finalidade, grupos, uf.Key, uf.Key, PessoaTipo.Fisi  
                regraFiscals.Add(regraFiscal);  
            }  
        }  
    }  
  
    return regraFiscals.ToArray();  
}
```

```
}
```

```
using System;
```

```
namespace ConsoleAppValorVenda
```

```
{
```

```
1 referência | miguel.galego, há 1 hora | 1 autor, 2 alterações
```

```
public static class VendaRepositorio
```

```
{  
    1 referência | miguel.galego, há 1 hora | 1 autor, 2 alterações  
    public static Venda GerarVendaCapa()  
    {  
        return new Venda(  
            codigo: 1,  
            cooperativa: CooperativaRepositorio.Cooperativas[0],  
            cooperado: CooperadoRepositorio.Cooperados[0],  
            finalidade: FinalidadeRespositorio.Finalidades[0],  
            formaPagto: FormaPagtoRepositorio.FormaPagtoAPrazo,  
            data: DateTime.Today  
        );  
    }  
}
```

```
}
```

