

OBS.: Não esquecer de executar no terminal os comandos:

Ao criar o modelo MVC :

```
dotnet new mvc --no-https
```

Para poder conectar ao Banco de dados:

```
dotnet add package MySQLConnector
```

Não esquecer de adicionar o pacote Newtonsoft.Json na video aula 3:

```
dotnet add package Newtonsoft.Json
```

```
*****  
*****FARMACIA*****  
*****  
*****Startup.cs*****  
*****
```

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using Microsoft.AspNetCore.Builder;  
using Microsoft.AspNetCore.Hosting;  
using Microsoft.AspNetCore.Http;  
using Microsoft.AspNetCore.Mvc;  
using Microsoft.Extensions.Configuration;  
using Microsoft.Extensions.DependencyInjection;  
using Microsoft.Extensions.Hosting;
```

```
namespace Farmacia
```

```
{  
    public class Startup  
    {  
        public Startup(IConfiguration configuration)  
        {  
            Configuration = configuration;  
        }  
  
        public IConfiguration Configuration { get; }  
  
        // This method gets called by the runtime. Use this method to add services to the container.  
        [Obsolete]  
        public void ConfigureServices(IServiceCollection services)  
        {  
            services.Configure<CookiePolicyOptions>(options =>  
            {  
                options.CheckConsentNeeded = context => false;  
                options.MinimumSameSitePolicy = SameSiteMode.None;  
            });  
  
            services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);  
            services.AddMemoryCache();  
            services.AddSession();  
        }  
    }  
}
```

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
```

```

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    else
    {
        app.UseExceptionHandler("/Home/Error");
        //Https
        app.UseHsts();
    }
    //Https
    app.UseHttpsRedirection();
    app.UseCookiePolicy();
    app.UseSession();
    //dotnet dev-certs https --trust

    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
}

*****
*****CONTROLLERS*****
*****
*****HomeController.cs*****
*****

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using Farmacia.Models;

namespace Farmacia.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {

```

```

        _logger = logger;
    }

    public IActionResult Index()
    {
        return View();
    }

    public IActionResult Privacy()
    {
        return View();
    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier
});
    }
}
}
}
}
*****
*****MedicamentoController.cs*****
*****
using System;
using System.Collections.Generic;
using Farmacia.Models;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Newtonsoft.Json;

namespace Farmacia.Controllers
{
    public class MedicamentoController : Controller
    {
        public IActionResult Lista()
        {
            MedicamentoRepository medicamentoRepo = new MedicamentoRepository();
            List<Medicamento> medicamentos = medicamentoRepo.Lista();
            return View(medicamentos);
        }
        [HttpPost]
        public IActionResult AdicionaASacola(ItemPedido novoItem)
        {
            List<ItemPedido> itensSacola = new List<ItemPedido>();
            string listaEmTexto = HttpContext.Session.GetString("sacola");
            if (!String.IsNullOrEmpty(listaEmTexto))
            {
                itensSacola = JsonConvert.DeserializeObject<List<ItemPedido>>(listaEmTexto);
            }
            itensSacola.Add(novoItem);
            listaEmTexto = JsonConvert.SerializeObject(itensSacola);
            HttpContext.Session.SetString("sacola", listaEmTexto);
            return RedirectToAction("Lista");
        }
    }
}

```

```

}
*****
*****PedidoController.cs*****
*****
using System;
using System.Collections.Generic;
using Farmacia.Models;
using Farmacia.Models.ViewModels;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Newtonsoft.Json;

namespace Farmacia.Controllers
{
    public class PedidoController : Controller
    {
        public IActionResult Sacola()
        {
            List<ItemPedido> itensSacola = new List<ItemPedido>();
            string listaEmTexto = HttpContext.Session.GetString("sacola");
            if(!String.IsNullOrEmpty(listaEmTexto))
            {
                itensSacola = JsonConvert.DeserializeObject<List<ItemPedido>>(listaEmTexto);
            }

            MedicamentoRepository medicamentoRepo = new MedicamentoRepository();
            List<Medicamento> resultadoBusca;
            foreach(ItemPedido item in itensSacola)
            {
                resultadoBusca = medicamentoRepo.Lista(item.medicamento.Id);
                if(resultadoBusca != null && resultadoBusca.Count > 0)
                {
                    item.medicamento = resultadoBusca[0];
                }
            }
            ClinicaRepository clinicaRepo = new ClinicaRepository();
            List<Clinica> listaClinicas = clinicaRepo.Lista(0);

            SacolaViewModel sacola = new SacolaViewModel();
            sacola.Clinicas = listaClinicas;
            sacola.Itens = itensSacola;

            return View(sacola);
        }
        [HttpPost]
        public IActionResult FecharPedido(SacolaViewModel sacola)
        {
            PedidoRepository pedidoRepo = new PedidoRepository();
            Pedido novoPedido = new Pedido();
            novoPedido.DataPedido = DateTime.Now;
            novoPedido.clinica = new Clinica();
            novoPedido.clinica.Id = sacola.IdClinica;
            novoPedido.Entregue = false;
            novoPedido.Id = pedidoRepo.Cadastra(novoPedido);

            ItemPedidoRepository itemPedidoRepo = new ItemPedidoRepository();

```

```

        foreach(ItemPedido item in sacola.Itens)
        {
            item.pedido = novoPedido;
            itemPedidoRepo.Cadastra(item);
        }
        HttpContext.Session.Remove("sacola");
        return View();
    }

    public IActionResult Lista()
    {
        PedidoRepository pedidoRepo = new PedidoRepository();
        List<Pedido> pedidos = pedidoRepo.Lista();

        return View(pedidos);
    }
    public IActionResult Detalhe(int id)
    {
        Pedido pedido = null;
        PedidoRepository pedidoRepo = new PedidoRepository();
        List<Pedido> pedidos = pedidoRepo.Lista(id);
        if(pedidos != null && pedidos.Count > 0)
        {
            pedido = pedidos[0];

            ClinicaRepository clinicaRepo = new ClinicaRepository();
            List<Clinica> clinicas = clinicaRepo.Lista(pedido.clinica.Id);
            pedido.clinica = clinicas[0];

            ItemPedidoRepository itemPedidoRepo = new ItemPedidoRepository();
            List<ItemPedido> itens = itemPedidoRepo.Lista(pedido.Id);
            pedido.Itens = itens;
        }
        return View(pedido);
    }
}
}

*****
*****MODELS*****
*****
*****Clinica.cs*****
*****

namespace Farmacia.Models
{
    public class Clinica
    {
        public int Id {get;set;}
        public string Nome {get;set;}
        public string Detalhes {get;set;}
    }
}

*****
*****ClinicaRepository.cs*****
*****

using System.Collections.Generic;
using MySqlConnection;

```

```

namespace Farmacia.Models
{
    public class ClinicaRepository : Repository
    {
        public void Cadastra (Clinica c)
        {
            conexao.Open();
            string sql = "INSERT INTO Clinica (nome, detalhes) VALUES (@nome, @detalhes)";

            MySqlCommand comando = new MySqlCommand(sql, conexao);
            comando.Parameters.AddWithValue("@nome", c.Nome);
            comando.Parameters.AddWithValue("@detalhes", c.Detalhes);
            comando.ExecuteNonQuery();
            conexao.Close();
        }
        public List<Clinica> Lista(int id)
        {
            conexao.Open();
            string sql = "SELECT * FROM Clinica " + (id > 0 ? "WHERE id = @id : """) + " ORDER BY
nome";

            MySqlCommand comando = new MySqlCommand(sql, conexao);
            if(id > 0)
            {
                comando.Parameters.AddWithValue("@id", id);
            }
            MySqlDataReader reader = comando.ExecuteReader();
            List<Clinica> lista = new List<Clinica>();

            while(reader.Read())
            {
                Clinica clinica = new Clinica();
                clinica.Id = reader.GetInt32("id");
                if (!reader.IsDBNull(reader.GetOrdinal("nome")))
                {
                    clinica.Nome = reader.GetString("nome");
                }
                if (!reader.IsDBNull(reader.GetOrdinal("detalhes")))
                {
                    clinica.Detalhes = reader.GetString("detalhes");
                }
                lista.Add(clinica);
            }
            conexao.Close();
            return lista;
        }
        public List<Clinica> Lista()
        {
            return Lista(0);
        }
    }
}

```

```

*****
*****ItemPedidoRepository.cs*****
*****

```

```

using System.Collections.Generic;
using MySqlConnectionor;

namespace Farmacia.Models
{
    public class ItemPedidoRepository : Repository
    {
        public void Cadastra(ItemPedido ip)
        {
            conexao.Open();
            string sql = "INSERT INTO ItemPedido (id_pedido, id_medicamento, quantidade) VALUES
(@id_pedido, @id_medicamento, @quantidade);";
            MySqlCommand comando = new MySqlCommand(sql, conexao);
            comando.Parameters.AddWithValue("@id_pedido", ip.pedido.Id);
            comando.Parameters.AddWithValue("@id_medicamento", ip.medicamento.Id);
            comando.Parameters.AddWithValue("@quantidade", ip.Quantidade);
            comando.ExecuteNonQuery();
            conexao.Close();
        }
        public List<ItemPedido> Lista(int idPedido)
        {
            conexao.Open();
            string sql = "SELECT ip.id as IdItemPedido, ip.id_pedido, ip.quantidade, m.id as IdMedicamento,
m.nome, m.valor, FROM itempedido ip JOIN medicamento m ON ip.id_medicamento = m.id WHERE
id_pedido = @idPedido;";

            MySqlCommand comando = new MySqlCommand(sql, conexao);
            comando.Parameters.AddWithValue("@idPedido", idPedido);
            MySqlDataReader reader = comando.ExecuteReader();
            List<ItemPedido> lista = new List<ItemPedido>();

            while(reader.Read())
            {
                ItemPedido item = new ItemPedido();
                item.Id = reader.GetInt32("IdItemPedido");
                item.Quantidade = reader.GetInt32("quantidade");

                item.medicamento = new Medicamento();
                item.medicamento.Id = reader.GetInt32("IdMedicamento");

                if (!reader.IsDBNull(reader.GetOrdinal("nome")))
                {
                    item.medicamento.Nome = reader.GetString("nome");
                }
                if (!reader.IsDBNull(reader.GetOrdinal("valor")))
                {
                    item.medicamento.Valor = reader.GetDecimal("valor");
                }
                lista.Add(item);
            }
            conexao.Close();
            return lista;
        }
    }
}

```

\*\*\*\*\*

\*\*\*\*\*Medicamento.cs\*\*\*\*\*

\*\*\*\*\*

namespace Farmacia.Models

```
{
    public class Medicamento
    {
        public int Id {get;set;}
        public string Nome {get;set;}
        public string Descricao {get;set;}
        public decimal Valor {get;set;}
        public string Dosagem {get;set;}
        public int QtdEstoque {get;set;}
    }
}
```

\*\*\*\*\*

\*\*\*\*\*MedicamentoRepository.cs\*\*\*\*\*

\*\*\*\*\*

using System.Collections.Generic;

using MySqlConnection;

namespace Farmacia.Models

```
{
    public class MedicamentoRepository : Repository
    {
        public void Cadastra(Medicamento m)
        {
            conexao.Open();
            string sql = "INSERT INTO Medicamento (nome, descricao, valor, dosagem, qtdEstoque) VALUES (@nome, @descricao, @valor, @dosagem, @qtdEstoque)";
            MySqlCommand comando = new MySqlCommand(sql, conexao);
            comando.Parameters.AddWithValue("@nome", m.Nome);
            comando.Parameters.AddWithValue("@descricao", m.Descricao);
            comando.Parameters.AddWithValue("@valor", m.Valor);
            comando.Parameters.AddWithValue("@dosagem", m.Dosagem);
            comando.Parameters.AddWithValue("@qtdEstoque", m.QtdEstoque);
            comando.ExecuteNonQuery();
            conexao.Close();
        }
        public List<Medicamento> Lista(int id)
        {
            conexao.Open();
            string sql = "SELECT * FROM Medicamento " + (id > 0 ? "WHERE id = @id : " + " ORDER BY nome";

            MySqlCommand comando = new MySqlCommand(sql, conexao);
            if(id > 0)
            {
                comando.Parameters.AddWithValue("@id", id);
            }
            MySqlDataReader reader = comando.ExecuteReader();
            List<Medicamento> lista = new List<Medicamento>();

            while(reader.Read())
            {
                Medicamento medicamento = new Medicamento();
                medicamento.Id = reader.GetInt32("id");
```



```

        if (!reader.IsDBNull(reader.GetOrdinal("nome")))
        {
            medicamento.Nome = reader.GetString("nome");
        }
        if (!reader.IsDBNull(reader.GetOrdinal("descricao")))
        {
            medicamento.Descricao = reader.GetString("descricao");
        }
        if (!reader.IsDBNull(reader.GetOrdinal("valor")))
        {
            medicamento.Valor = reader.GetDecimal("valor");
        }
        if (!reader.IsDBNull(reader.GetOrdinal("dosagem")))
        {
            medicamento.Dosagem = reader.GetString("dosagem");
        }
        if (!reader.IsDBNull(reader.GetOrdinal("qtdEstoque")))
        {
            medicamento.QtdEstoque = reader.GetInt32("qtdEstoque");
        }
        lista.Add(medicamento);
    }
    conexao.Close();
    return lista;
}
public List<Medicamento> Lista()
{
    return Lista(0);
}
}
}

*****
*****Pedido.cs*****
*****

using System;
using System.Collections.Generic;

namespace Farmacia.Models
{
    public class Pedido
    {
        public int Id {get;set;}
        public DateTime DataPedido {get;set;}
        public bool Entregue {get;set;}
        public Clinica clinica {get;set;}
        public List<ItemPedido> Itens {get;set;}
    }
}

*****
*****PedidoRepository.cs*****
*****

using System.Collections.Generic;
using MySqlConnection;

namespace Farmacia.Models
{

```

```

public class PedidoRepository : Repository
{
    public int Cadastra(Pedido p)
    {
        conexao.Open();
        string sql = "INSERT INTO Pedido (dataPedido, entregue, id_clinica) VALUES (@dataPedido, 0,
@id_clinica)";
        MySqlCommand comando = new MySqlCommand(sql, conexao);
        comando.Parameters.AddWithValue("@dataPedido", p.DataPedido);
        comando.Parameters.AddWithValue("@descricao", p.clinica.Id);
        comando.ExecuteNonQuery();
        int idNovoPedido = (int)comando.LastInsertedId;
        conexao.Close();
        return idNovoPedido;
    }
    public List<Pedido> Lista(int id)
    {
        conexao.Open();
        string sql = "SELECT * FROM Pedido p " + (id > 0 ? "WHERE id = @id : """) + " ORDER BY
dataPedido DESC";

        MySqlCommand comando = new MySqlCommand(sql, conexao);
        if(id > 0)
        {
            comando.Parameters.AddWithValue("@id", id);
        }
        MySqlDataReader reader = comando.ExecuteReader();
        List<Pedido> lista = new List<Pedido>();

        while(reader.Read())
        {
            Pedido pedido = new Pedido();
            pedido.Id = reader.GetInt32("id");
            if (!reader.IsDBNull(reader.GetOrdinal("dataPedido")))
            {
                pedido.DataPedido = reader.GetDateTime("dataPedido");
            }
            if (!reader.IsDBNull(reader.GetOrdinal("entregue")))
            {
                pedido.Entregue = reader.GetBoolean("entregue");
            }
            if (!reader.IsDBNull(reader.GetOrdinal("id_clinica")))
            {
                pedido.clinica = new Clinica();
                pedido.clinica.Id = reader.GetInt32("id_clinica");
            }
            lista.Add(pedido);
        }
        conexao.Close();
        return lista;
    }
    public List<Pedido> Lista()
    {
        return Lista(0);
    }
}

```

```

}
*****
*****Repository.cs*****
*****
using MySqlConnection;

namespace Farmacia.Models
{
    public class Repository
    {
        protected const string _strConexao = "DATABASE=farmacia;Data Source=localhost;User Id=root;";
        protected MySqlConnection conexao = new MySqlConnection(_strConexao);
    }
}
*****
*****ItemPedido.cs*****
*****
namespace Farmacia.Models
{
    public class ItemPedido
    {
        public int Id {get;set;}
        public int Quantidade {get;set;}
        public Pedido pedido {get;set;}
        public Medicamento medicamento {get;set;}
    }
}
*****
*****ViewModels***(Pasta dentro de Models)*****
*****
*****SacolaViewModel.cs*****
*****
using System.Collections.Generic;

namespace Farmacia.Models.ViewModels
{
    public class SacolaViewModel
    {
        public List<Clinica> Clinicas {get;set;}
        public List<ItemPedido> Itens {get;set;}
        public int IdClinica {get;set;}
    }
}
*****
*****VIEWS*****
*****
*****Medicamento*****
*****
*****Lista.cshtml*****
*****
@model List<Medicamento>

```

<h2>Adicione Medicamentos ao Pedido:</h2>

```

<table>
  <thead>

```

```

<tr>
  <th>Nome</th>
  <th>Valor</th>
  <th>Quantidade</th>
  <th>Ação</th>
</tr>
</thead>
<tbody>
  @foreach (Medicamento medicamento in Model)
  {
    <tr>
      <td>@medicamento.Nome</td>
      <td>@medicamento.Valor</td>
      <form asp-action="AdicionaASacola" method="post">
        <input type="hidden" name="Medicamento.Id" value="@medicamento.Id">
        <td><input type="number" name="Quantidade"></td>
        <td><input type="submit" value="Adicionar"></td>
      </form>
    </tr>
  }
</tbody>
</table>
*****
*****Pedido*****
*****
*****Detalhe.cshtml*****
*****
@model Pedido

<h2>Pedido #@Model.Id</h2>

<ul>
  <li>Clínica: @Model.clinica.Nome</li>
  <li>Data: @Model.DataPedido</li>
  <li>Entregue: @Model.Entregue</li>
  <li>Itens: </li>
  <table>
    <tr>
      <th>Medicamento</th>
      <th>Valor</th>
      <th>Quantidade</th>
    </tr>
    @foreach (ItemPedido item in Model.Itens)
    {
      <tr>
        <td>@item.medicamento.Nome</td>
        <td>@item.medicamento.Valor</td>
        <td>@item.Quantidade</td>
      </tr>
    }
  </table>
</ul>
*****
*****FecharPedido.cshtml*****
*****
<h1>Pedido registrado com sucesso!</h1>

```

```
*****
*****Lista.cshtml*****
*****
```

@model List<Pedido>

<h2>Lista de Pedidos</h2>

```
<table>
  <thead>
    <th>Id</th>
    <th>Data</th>
    <th>Clinica (id)</th>
    <th></th>
  </thead>
  <tbody>
    @foreach (Pedido item in Model)
    {
      <tr>
        <td>@item.Id</td>
        <td>@item.DataPedido</td>
        <td>@item.clinica.Id</td>
        <td><a href="/Pedido/Detalhe?id=@item.Id">Detalhes</a></td>
      </tr>
    }
  </tbody>
</table>
```

```
*****
*****Sacola.cshtml*****
*****
```

@model Farmacia.Models.ViewModels.SacolaViewModel

<h2>Medicamentos escolhidos até agora para o pedido</h2>

```
@if (Model != null && Model.Itens.Count > 0)
{
  <form action="/Pedido/FecharPedido" method="POST">
    <label asp-for="IdClinica">Selecione a Clínica</label>
    <select asp-for="IdClinica">
      @foreach (Clinica cli in Model.Clinicas)
      {
        <option value="@cli.Id">@cli.Nome</option>
      }
    </select>
    <br>
    <table>
      <thead>
        <tr>
          <th>Nome</th>
          <th>Valor</th>
          <th>Quantidade</th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        @for (int i = 0; i < Model.Itens.Count; i++)
        {
          <tr>
```

```

        <td>@Model.Itens[i].medicamento.Nome</td>
        <td>@Model.Itens[i].medicamento.Valor</td>
        <td>@Model.Itens[i].Quantidade</td>
        <input type="hidden" asp-for="@Model.Itens[i].medicamento.Id">
        <td><input type="number" asp-for="@Model.Itens[i].Quantidade"></td>
    </tr>
}
</tbody>
</table>
<input type="submit" value="Fechar Pedido">
</form>
}
else
{
    <span>Nenhum item selecionado</span>
}
*****
*****Shared*****
*****
*****_Layout.cshtml*****
*****
@using Microsoft.AspNetCore.Http;
@using Newtonsoft.Json;
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - Farmacia</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom
box-shadow mb-3">
            <div class="container">
                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Farmacia</a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-
collapse" aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-
action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-
action="Privacy">Privacy</a>
                        </li>
                    </ul>
                </div>
                @if (Context.Session.GetString("sacola") != null)

```

```

        {
            List<ItemPedido> itensSacola =
JsonConvert.DeserializeObject<List<ItemPedido>>(Context.Session.GetString("sacola"));
            <a href="/Pedido/Sacola" >@itensSacola.Count Itens na Sacola</a>
        }
        else
        {
            <span>Sacola Vazia</span>
        }
    </div>
</div>
</div>
</nav>
</header>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2021 - Farmacia - <a asp-area="" asp-controller="Home"
asp-action="Privacy">Privacy</a>
    </div>
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
    @await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```