



**TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA**



Curso Técnico em Informática

Desenvolvimento de Sistemas II – 180h

Profª: Francisleide Almeida

Relatórios usando asp

- Consultar e analisar dados gerados por sistemas de gestão;
- Transformar HTML em PDF
- Informações compiladas de dados do sistema.

Para começar...

- Utilizaremos os plug-ins:
 - PagedList - Renderiza o relatório na página (paginado);
 - Rotativa- Gera PDF
- Nuget:
 - Install-Package PagedList.Mvc
 - Install-Package Rotativa

- Clique com botão direito na pasta Controller>Add>Controller e selecione a opção “MVC 5 Controller – Empty”. Clique em Add e nomeie o controller como Relatorios;
- Esse controller será criado apenas com uma action Index, porém iremos realizar algumas modificações e implementar a action ListagemClientes, que tratará de gerar nosso primeiro relatório.


```

05 using System.Web;
06 using System.Web.Mvc;
07 using PagedList;
08 using Rotativa;
09 using Rotativa.Options;
10
11 namespace Relatorios_ASPNET_MVC.Controllers
12 {
13     public class RelatoriosController : Controller
14     {
15         private VendaRelatorioDbContext db = new VendaRelatorioDbContext();
16         public ActionResult ListagemClientes(int? pagina, Boolean? gerarPDF)
17         {
18             var listagemClientes = db.clientes.OrderBy(n => n.ClienteId).ToList<Cliente>();
19
20             if (gerarPDF != true)
21             {
22                 //Definindo a paginação
23                 int paginaQdteRegistros = 10;
24                 int paginaNumeroNavegacao = (pagina ?? 1);
25
26                 return View(listagemClientes.TopagedList(paginaNumeroNavegacao, paginaQdteRegistros));
27             }
28             else
29             {
30                 int paginaNumero = 1;
31
32                 var pdf = new ViewsPdf
33                 {
34                     ViewName = "ListagemClientes",
35                     PageSize = Size.A4,
36                     IsGrayscale = true,
37                     Model = listagemClientes.TopagedList(paginaNumero, listagemClientes.Count)
38                 };
39                 return pdf;
40             }
41         }
42     }
43 }

```

Com essa action pronta para retornar a listagem de clientes, agora precisamos criar a view que de fato irá renderizar a página HTML com alguns recursos do Bootstrap.

The screenshot shows the 'Add View' dialog box in Visual Studio. The fields are as follows:

- View name: ListagemClientes
- Template: List
- Model class: Cliente (Relatorios_ASPNET_MVC.Models)
- Data context class: VendaRelatorioDbContext (Relatorios_ASPNET_MVC.Models)

Under 'View options':

- ☐ Create as a partial view
- ☒ Reference script libraries
- ☒ Use a layout page

The file path is set to ~/Views/Shared/_Layout.cshtml. Below the path, it says '(Leave empty if it is set in a Razor _viewstart file)'. The dialog has 'Add' and 'Cancel' buttons.


```

01 @model PagedList<Relatorios_ASPNET_MVC.Models.Cliente>
02
03 @{ ViewBag.Title = "ListagemClientes";
04 Layout = "~/Views/Shared/_Layout.cshtml"; }
05
06 <div class="panel panel-default">
07   <div class="panel-heading"><h5>Relatório Listagem de Clientes</h5></div>
08   <div class="panel-body">
09     <div class="row">
10       <div class="col-md-12">
11         <table class="table">
12           <tr>
13             <th>ID </th><th>Nome </th><th>CPF/CNPJ </th><th>Endereço </th><th>Bairro </th><
14             <th>CEP </th><th>Email </th><th>Telefone </th><th>Status </th><th>Data cadastro
15             </tr>
16             @foreach (var item in Model)
17             {
18               <tr>
19                 <td>@Html.DisplayFor(modelItem => item.ClienteId)</td>
20                 <td>@Html.DisplayFor(modelItem => item.Nome)</td>
21                 <td>@Html.DisplayFor(modelItem => item.CpfCnpj)</td>
22                 <td>@Html.DisplayFor(modelItem => item.Endereco)</td>
23                 <td>@Html.DisplayFor(modelItem => item.Bairro)</td>
24                 <td>@Html.DisplayFor(modelItem => item.Cidade)</td>
25                 <td>@Html.DisplayFor(modelItem => item.Cep)</td>
26                 <td>@Html.DisplayFor(modelItem => item.Email)</td>
27                 <td>@Html.DisplayFor(modelItem => item.Telefone)</td>
28                 <td>@Html.DisplayFor(modelItem => item.StatusCliente)</td>
29                 <td>@Html.DisplayFor(modelItem => item.DataCadastro)</td>
30               </tr> }
31             </tr>
32             <td><b>@Model.Count registros de @Model.TotalItemCount</b></td>
33             <td><a href="/Relatorios/ListagemClientes?generarPDF=true"><b>GERAR PDF</b></a></
34             </tr>

```

```

35 </table>
36 </div>
37 </div>
38 @{
39     if (Model.TotalItemCount != Model.Count)
40     {
41         <div class="row">
42             <div class="col-md-12">
43                 Página @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) de @(Model.P
44
45                 @if (Model.HasPreviousPage)
46                 {
47                     @Html.ActionLink("<<", "ListagemClientes", new { pagina = 1, sortOrder = Vi
48                     @Html.Raw(" ");
49                     @Html.ActionLink("< Anterior", "ListagemClientes", new { pagina = Model.Pag
50                 }
51                 else
52                 {
53                     @:<<
54                     @Html.Raw(" ");
55                     @:< Anterior
56                 }
57
58                 @if (Model.HasNextPage)
59                 {
60                     @Html.ActionLink("Próxima >", "ListagemClientes", new { pagina = Model.page
61                     @Html.Raw(" ");
62                     @Html.ActionLink(">", "ListagemClientes", new { pagina = Model.PageCount,
63                 }
64                 else
65                 {
66                     @:Próxima >
67                     @Html.Raw(" ")
68                     @:>>
69                 }
70             </div>
71         </div>
72     }
73 }
74 </div>

```

Ainda podemos personalizar melhor nossos
relatórios

CONTINUA...