02

Cadastro de movimentações

Cadastro de movimentações

Agora que já temos o cadastro de usuários da aplicação, vamos começar a trabalhar no cadastro de movimentações, para isso criaremos um novo DAO chamado `MovimentacaoDAO` :

```
public class MovimentacaoDAO
{
    private FinancasContext context;
    public MovimentacaoDAO(FinancasContext context)
    {
        this.context = context;
    }
    public void Adiciona(Movimentacao movimentacao)
    {
        context.Movimentacoes.Add(movimentacao);
        context.SaveChanges();
    }
    public IList<Movimentacao> Lista()
    {
        return context.Movimentacoes.ToList();
    }
}
```

E agora precisamos implementar o controller que cuidará do cadastro e lista de movimentações:

```
public class MovimentacaoController : Controller
{
    private MovimentacaoDAO movimentacaoDAO;
    public MovimentacaoController(MovimentacaoDAO movimentacaoDAO)
    {
        this.movimentacaoDAO = movimentacaoDAO;
    }
}
```

Dentro desse controller colocaremos o método `Form` que novamente mostrará o formulário de cadastro para o usuário:

```
public ActionResult Form()
{
    return View();
}
```

Mas no formulário de cadastro de movimentações, queremos escolher qual é o usuário dessa movimentação utilizando um combo box, mas para isso, precisaremos da lista de usuários na camada de visualização, portanto o construtor do `MovimentacaoController` precisa receber também um objeto do tipo `UsuarioDAO` além do `MovimentacaoDAO` :

```
public class MovimentacaoController : Controller
{
    private MovimentacaoDAO movimentacaoDAO;
    private UsuarioDAO usuarioDAO;
    public MovimentacaoController(MovimentacaoDAO movimentacaoDAO, UsuarioDAO usuarioDAO)
    {
        this.movimentacaoDAO = movimentacaoDAO;
        this.usuarioDAO = usuarioDAO;
    }
}
```

E no método `Form` enviaremos a lista de usuários para a view através da `ViewBag` :

```
public ActionResult Form()
{
    ViewBag.Usuarios = usuarioDAO.Lista();
    return View();
}
```

E agora na view do método `Form` , utilizaremos novamente o `HtmlHelper` para montar o formulário:

```
@model Financas.Entidades.Movimentacao

@using (Html.BeginForm("Adiciona", "Movimentacao", FormMethod.Post))
{
    @Html.LabelFor(m => m.Valor, "Valor")
    @Html.TextBoxFor(m => m.Valor, new { @class = "form-control" })
    @Html.ValidationMessageFor(m => m.Valor)

    @Html.LabelFor(m => m.UsuarioId, "Usuario")
    @Html.DropDownListFor(m => m.UsuarioId,
        new SelectList(ViewBag.Usuarios, "Id", "Nome"),
        new { @class = "form-control" })
    @Html.ValidationMessageFor(m => m.UsuarioId)

    @Html.LabelFor(m => m.Tipo, "Tipo")
    @Html.EnumDropDownListFor(m => m.Tipo, new { @class="form-control"})
    @Html.ValidationMessageFor(m => m.Tipo)

    @Html.LabelFor(m => m.Data, "Data")
    @Html.TextBoxFor(m => m.Data, new { @class = "form-control" })
    @Html.ValidationMessageFor(m => m.Data)

    <input type="submit" value="Cadastrar"/>
}
```

Para terminarmos o cadastro, precisamos apenas criar o método `Adiciona` no `MovimentacaoController` :

```
public ActionResult Adiciona(Movimentacao movimentacao)
{
    if(ModelState.IsValid)
    {
```

```
            movimentacaoDAO.Adiciona(movimentacao);
            return RedirectToAction("Index");
        }
        else
        {
            ViewBag.Usuarios = usuarioDAO.Lista();
            return View("Form");
        }
    }
```

Agora resta apenas implementar a lógica para listar as informações no método `Index` desse controller:

```
public ActionResult Index()
{
    return View(movimentacaoDAO.Lista());
}
```

Na view para essa action utilizaremos novamente uma tabela para mostrar as informações:

```
@model IList<Financas.Entidades.Movimentacao>

@Html.ActionLink("Nova movimentação", "Form")

<table class="table table-hover">
    <thead>
        <tr>
            <th>Id</th>
            <th>Usuario</th>
            <th>Valor</th>
            <th>Tipo</th>
        </tr>
    </thead>
    <tbody>
        @foreach (var m in Model)
        {
            <tr>
                <td>@m.Id</td>
                <td>@m.Usuario.Nome</td>
                <td>@m.Valor</td>
                <td>@m.Tipo</td>
            </tr>
        }
    </tbody>
</table>
```