

Aula 10 – Upload

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: allbert.almeida@fatec.sp.gov.br

Site: <http://www.allbert.com.br>



/allbert.almeida

Objetivo

- O objetivo da aula é realizar upload de arquivos com aplicação web e gravar no banco de dados. Será realizado o upload de duas formas:
 - Um único arquivo (foto perfil);
 - Vários arquivos (álbum);

Modelo



Criar uma classe Pessoa

```
public class Pessoa
{
    [Key]
    public int Id { get; set; }
    [MaxLength(255)]
    public string Nome { get; set; }
    [MaxLength(100)]
    public string Foto { get; set; }
    public ICollection<Album> Albums { get; set; }
}
```


Criar uma classe Album

```
public class Album
{
    [Key]
    public int Id { get; set; }
    public int PessoaId { get; set; }
    public virtual Pessoa Pessoa { get; set; }
    [MaxLength(100)]
    public string Foto { get; set; }
}
```

Adicionar 2 pacotes para solução

- EntityFramework
- MySql Data Entity 6.9.11

EntityFramework por Microsoft

v6.2.0



Entity Framework is Microsoft's recommended data access technology for new applications.



MySql.Data.Entity por Oracle

v6.10.6



Entity Framework 6.0 supported

Criar uma classe Contexto

```
public class Contexto:DbContext
{
    public Contexto() : base(nameOrConnectionString:
"StringConexao") { }
    public DbSet<Pessoa> Pessoa { get; set; }
    public DbSet<Album> Album { get; set; }
}
```

Adicionar a String de Conexão no WebConfig

```
<connectionStrings>
```

```
  <add name="StringConexao"  
  providerName="MySql.Data.MySqlClient"  
  connectionString="server=localhost;dat  
  abase=teste;uid=root;pwd='root'" />
```

```
</connectionStrings>
```


Criar Classe Funcoes

```
public static bool CriarDiretorio()
{
    string dir = HttpContext.Current.Request.PhysicalApplicationPath + "Uploads\\";
    if (!Directory.Exists(dir))
    {
        //Caso não exista devemos criar
        Directory.CreateDirectory(dir);
        return true;
    }
    else
        return false;
}
public static bool ExcluirArquivo(string arq)
{
    if (File.Exists(arq))
    {
        File.Delete(arq);
        return true;
    }
    else
        return false;
}
public static string UploadArquivo(HttpPostedFileBase flpUpload, string nome)
{
    try
    {
        double permitido = 999;
        if (flpUpload != null)
        {
            string arq = Path.GetFileName(flpUpload.FileName);
            double tamanho = Convert.ToDouble(flpUpload.ContentLength) / 1024;
            string extensao = Path.GetExtension(flpUpload.FileName).ToLower();
            string diretorio = HttpContext.Current.Request.PhysicalApplicationPath + "Uploads\\" + nome;
            if (tamanho > permitido)
                return "Tamanho Máximo permitido é de " + permitido + " kb!";
            else if ((extensao != ".png" && extensao != ".jpg"))
                return "Extensão inválida, só são permitidas .png e .jpg!";
            else
            {
                flpUpload.SaveAs(diretorio);
                return "sucesso";
            }
        }
        else
            return "Erro no Upload!";
    }
    catch { return "Erro no Upload"; }
}
```

Compilar a solução

- Compilar a solução para que os modelos sejam validados e seja possível criar o “CRUD” da classe Pessoa e Album

Gerar os controllers

- Gerar o controller das classes Pessoa e Album;

View Create - Pessoa

- No BeginForm:

```
Html.BeginForm("Create", "Pessoas", FormMethod.Post,  
new { enctype = "multipart/form-data" })
```

- No lugar do editor de Foto:

```
@*@Html.EditorFor(model => model.Foto, new {  
htmlAttributes = new { @class = "form-control" } })  
    @Html.ValidationMessageFor(model =>  
model.Foto, "", new { @class = "text-danger" })*@  
<input name="arquivo" class="form-control"  
type="file" />
```


Controller Create - Pessoa

```
public ActionResult Create([Bind(Include = "Id,Nome")] Pessoa pessoa, HttpPostedFileBase arquivo)
{
    string valor = "";
    if (ModelState.IsValid)
    {
        db.Pessoa.Add(pessoa);
        db.SaveChanges();
        if (arquivo != null)
        {
            Funcoes.CriarDiretorio();
            string nomearg = "FotoPessoa" + pessoa.Id + ".png";
            valor = Funcoes.UploadArquivo(arquivo, nomearg);
            if (valor == "sucesso")
            {
                pessoa.Foto = nomearg;
                db.Entry(pessoa).State = EntityState.Modified;
                db.SaveChanges();
                TempData["MSG"] = "success|Usuário cadastrado com sucesso";
                return RedirectToAction("Index");
            }
            else
            {
                System.IO.File.Copy(Request.PhysicalApplicationPath + "Uploads\\user.png", Request.PhysicalApplicationPath +
                "Uploads\\FotoPessoa" + pessoa.Id + ".png");
                pessoa.Foto = "FotoPessoa" + pessoa.Id + ".png";
                db.Entry(pessoa).State = EntityState.Modified;
                db.SaveChanges();
                TempData["MSG"] = "warning|Problema no Upload da foto: " + valor;
                return RedirectToAction("Index");
            }
        }
        else
        {
            System.IO.File.Copy(Request.PhysicalApplicationPath + "Uploads\\user.png", Request.PhysicalApplicationPath +
            "Uploads\\FotoPessoa"+pessoa.Id+".png");
            pessoa.Foto = "FotoPessoa" + pessoa.Id + ".png";
            db.Entry(pessoa).State = EntityState.Modified;
            db.SaveChanges();
            TempData["MSG"] = "success|Usuário cadastrado com sucesso";
            return RedirectToAction("Index");
        }
    }
    return View(pessoa);
}
```

View Index - Pessoa

```
@*@Html.DisplayFor(modelItem =>
item.Foto)*@

```


Testar Solução

- Executar a aplicação e cadastrar uma pessoa sem foto e outra com foto;

View Delete - Pessoa

- Comentar DisplayFor e inserir tag da imagem:

```
@*@Html.DisplayFor(model => model.Foto)*@  

```


Controller DeleteConfirmed - Pessoa

```
public ActionResult DeleteConfirmed(int id) {
    try {
        Pessoa pessoa = db.Pessoa.Find(id);
        Funcoes.ExcluirArquivo(Request.PhysicalApplicationPath
+ "Uploads\\" + pessoa.Foto);
        db.Pessoa.Remove(pessoa);
        db.SaveChanges();
        TempData["MSG"] = "success|Usuário removido com
sucesso";
    } catch {
        TempData["MSG"] = "error|Erro ao remover usuário";
    }
    return RedirectToAction("Index");
}
```

View Edit - Pessoa

- No BeginForm:

```
Html.BeginForm("Edit", "Pessoas", FormMethod.Post, new {  
  enctype = "multipart/form-data" })
```

- Adicionar Input e Exibição de Foto:

```
<div class="form-group">  
  @Html.LabelFor(model => model.Foto, htmlAttributes: new { @class =  
  "control-label col-md-2" })  
  <div class="col-md-10">  
    @Html.EditorFor(model => model.Foto, new { htmlAttributes = new {  
    @class = "form-control", style="display:none;" } })  
    @*@Html.ValidationMessageFor(model => model.Foto, "", new {  
    @class = "text-danger" })*@  
    <input name="arquivo" class="form-control" type="file" />  
  </div>  
</div>  
<div class="form-group">  
  @Html.LabelFor(model => model.Foto, htmlAttributes: new { @class =  
  "control-label col-md-2" })  
  <div class="col-md-10">  
      
  </div>  
</div>
```


Controller Edit [HttpPost] - Pessoa

```
public ActionResult Edit([Bind(Include = "Id, Nome, Foto")] Pessoa pessoa, HttpPostedFileBase arquivo)
{
    string valor = "";
    if (ModelState.IsValid)
    {
        db.Entry(pessoa).State = EntityState.Modified;
        db.SaveChanges();
        if (arquivo != null)
        {
            Funcoes.CriarDiretorio();
            string nomearq = "FotoPessoa" + pessoa.Id + ".png";
            valor = Funcoes.UploadArquivo(arquivo, nomearq);
            if (valor == "sucesso")
            {
                TempData["MSG"] = "success|Usuário editado com sucesso";
                return RedirectToAction("Index");
            }
            else
            {
                Funcoes.ExcluirArquivo(Request.PhysicalApplicationPath + "Uploads\\FotoPessoa" + pessoa.Id + ".png");
                System.IO.File.Copy(Request.PhysicalApplicationPath + "Uploads\\user.png",
                Request.PhysicalApplicationPath + "Uploads\\FotoPessoa" + pessoa.Id + ".png");
                TempData["MSG"] = "success|Usuário editado com sucesso";
                return RedirectToAction("Index");
            }
        }
    }
    else
    {
        Funcoes.ExcluirArquivo(Request.PhysicalApplicationPath + "Uploads\\FotoPessoa" + pessoa.Id + ".png");
        System.IO.File.Copy(Request.PhysicalApplicationPath + "Uploads\\user.png",
        Request.PhysicalApplicationPath + "Uploads\\FotoPessoa" + pessoa.Id + ".png");
        TempData["MSG"] = "success|Usuário editado com sucesso";
        return RedirectToAction("Index");
    }
}
return View(pessoa);
}
```

Testar Solução

- Executar a aplicação, testar:
 - Editar foto;
 - Excluir;

Editar Foto com plugin

- Utilizando o script Croppie para redimensionar e cortar a foto do sistema;

View Index - Pessoa

- Adicionar o link para edição:

```
| @Html.ActionLink("Editar Foto",  
"EditarFoto", new { id = item.Id })
```


Adicionar no Controller Pessoa

```
public ActionResult EditarFoto(int? id) {
    if (id == null) {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest)
        ;
    }
    Pessoa pessoa = db.Pessoa.Find(id);
    if (pessoa == null) {
        return HttpNotFound();
    }
    return View(pessoa);
}
```

Add View EditarFoto - Pessoa

```
@model WebApplication1.Models.Pessoa
@{
    ViewBag.Title = "EditarFoto";
}

<h2>EditarFoto</h2>
@Html.HiddenFor(model => model.Id)
@Html.HiddenFor(model => model.Foto)
<div>
    <div id="main-cropper"></div>
    <a class="btn btn-primary" action="select">
        <input type="file" id="select" value="Escolha Imagem" accept="image/*">
    </a>
    <a href="#" class="btn btn-primary" action="upload">Upload</a>
</div>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/croppie/2.6.5/croppie.min.css" />
<style>
    .croppie-container {
        width: 100%;
        height: 80%;
    }
</style>
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
    <script src="https://cdnjs.cloudflare.com/ajax/libs/croppie/2.6.5/croppie.min.js"></script>
    <script>
        var basic = $('#main-cropper').croppie({
            viewport: { width: 250, height: 320 },
            boundary: { width: 300, height: 370 },
            showZoomer: true,
            url: '/Uploads/@Model.Foto?@DateTime.Now.ToString("yyyyMMddHHmmssfff")',
            format: 'jpeg|png|webp'
        });

        function readFile(input) {
            if (input.files && input.files[0]) {
                var reader = new FileReader();
                reader.onload = function (e) {
                    $('#main-cropper').croppie('bind', {
                        url: e.target.result
                    });
                }
                reader.readAsDataURL(input.files[0]);
            }
        }

        $('actionSelect input').on('change', function () { readFile(this); });
        $('actionUpload').on('click', function () {
            basic.croppie('result', 'blob').then(function(blob) {
                var formData = new FormData();
                formData.append('id', @Model.Id);
                formData.append('imagem', blob);
                var MyAppUrlSettings = {
                    MyUsefulUrl: '@Url.Action("SalvarFoto", "Pessoas")'
                }

                var request = new XMLHttpRequest();
                request.open('POST', MyAppUrlSettings.MyUsefulUrl);
                request.send(formData);

                request.onreadystatechange = function () {
                    if (request.readyState < 4) // está à espera de resposta
                        console.log("A carregar...");
                    else if (request.readyState === 4) { // 4 = A resposta do servidor está carregada
                        if (request.status === 200 && request.status < 300) // http status entre 200 e 299 quer dizer sucesso
                            window.location.href = '/Pessoas/Index/'; //maneira
                    }
                }
            });
        });
    </script>
}
```


Adicionar no Controller Pessoa

```
[HttpPost]
public ActionResult SalvarFoto(int id,
    HttpPostedFileBase imagem) {
    try {
        string nomearquivo = "FotoPessoa" + id + ".png";
        string diretorio =
            HttpContext.Request.PhysicalApplicationPath +
            "Uploads\\";
        imagem.SaveAs(diretorio + nomearquivo);
        return Json("ok");
    } catch {
        return Json("erro");
    }
}
```

Testar Solução

- Executar a aplicação, testar:

View Create - Album

- No BeginForm:

```
Html.BeginForm("Create", "Albums", FormMethod.Post,  
new { enctype = "multipart/form-data" })
```

- No lugar do editor de Foto:

```
@*@Html.EditorFor(model => model.Foto, new {  
htmlAttributes = new { @class = "form-control" } })  
@Html.ValidationMessageFor(model => model.Foto, "",  
new { @class = "text-danger" })*@  
<input name="arquivos" class="form-control"  
type="file" multiple="multiple" />
```

Controller Create - Album

```
public ActionResult Create([Bind(Include = "Id,PessoaId")] Album album, HttpPostedFileBase[] arquivos)
{
    string nomearq, valor;
    if (ModelState.IsValid)
    {
        if (arquivos != null)
        {
            Funcoes.CriarDiretorio();
            foreach (HttpPostedFileBase flb in arquivos)
            {
                nomearq = DateTime.Now.ToString("yyyyMMddHHmmssfff") +
                Path.GetExtension(flb.FileName);
                valor = Funcoes.UploadArquivo(flb, nomearq);
                if (valor == "sucesso")
                {
                    album.Foto = nomearq;
                    db.Album.Add(album);
                    db.SaveChanges();
                }
            }
            return RedirectToAction("Index");
        }
    }
    ViewBag.PessoaId = new SelectList(db.Pessoa, "Id", "Nome", album.PessoaId);
    return View(album);
}
```


View Index - Album

```
<h2>Index</h2>

<p>
  @Html.ActionLink("Create New", "Create")
</p>
@{string nome = ""; int cont = 0; }
@foreach (var item in Model)
{
  if (nome != item.Pessoa.Nome)
  {
    if (nome != "")
    {
      @Html.Raw("</div></div>");
      cont = 0;
    }
    @Html.Raw("<div class='row'>");
    <h3>@item.Pessoa.Nome</h3>
  }
  if (cont % 4 == 0)
  {
    if (cont != 0)
    {
      @Html.Raw("</div>");
    }
    @Html.Raw("<div class='row'>");
  }
  <div class="col-xs-6 col-md-3">
    <a href="#" class="thumbnail">
      
    </a>
    <a href="@Url.Action("Delete", "Albums", new { id = item.Id })">Excluir</a>
  </div>
  nome = item.Pessoa.Nome;
  cont++;
}
```

View Delete - Album

- Comentar DisplayFor e inserir tag da imagem:

```
@*@Html.DisplayFor(model => model.Foto)*@  

```


Controller DeleteConfirmed - Album

```
public ActionResult DeleteConfirmed(int id)
{
    Album album = db.Album.Find(id);
    Funcoes.ExcluirArquivo(Request.PhysicalApplicationPath
+ "Uploads\\" + album.Foto);
    db.Album.Remove(album);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```