

Upload e Armazenamento de Imagens

Desenvolvimento de Sistemas I
Prof. Jezer Machado de Oliveira



Necessidade

Criar estrutura para upload e visualização das capas dos livros:

- Adicionar upload de arquivos no cadastro de livros
- Salvar imagem
- Recuperar e exibir imagem no Primefaces



Necessidade

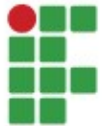
Duas possibilidades de armazenamento:

- Arquivo gravado no banco
- Arquivo gravado em diretório



Métodos de Armazenamento

Qual o mais vantajoso !?



INSTITUTO FEDERAL
Rio Grande do Sul



Armazenamento no Banco

```
@Entity
@Table(name = "livros")
public class Livro implements Serializable {
    @Id
    @GeneratedValue
    private int id;
    private String titulo;
    private String sinopse;
    private int anoPublicacao;
    private String editora;
    private boolean doacao;

    @Lob
    @Basic(fetch = FetchType.EAGER)
    private byte[] capa;
```



Livros.xhtml

```
<p:dialog header="Incluir/Editar Livro" widgetVar="dlgEdicaoLivro">
<h:form id="imagem" >
    <p:graphicImage width="100px" cache="false"
        rendered="#{livrosBean.capa!=null}"
        value="#{livrosBean.capa}"></p:graphicImage>
    <br/>
<p:outputLabel value="Capa do Livro:" for="upload"/>
<p:fileUpload id="upload" fileUploadListener="#{livrosBean.upload}"
    auto="true"
    fileLimit="1"
    label="Escolher"
    update="@form"
    uploadLabel="Enviar"
    cancelLabel="Cancelar"
/>
</h:form>
```

CadastroLivrosBean








```
@ManagedBean(name = "livrosBean")
@SessionScoped
public class CadastroLivrosBean {
    public void upload(FileUploadEvent event) throws IOException {
        byte[] imgCapa = new byte[(int) event.getFile().getSize()];
        event.getFile().getInputStream().read(imgCapa);
        livro.setCapa(imgCapa);
    }

    public StreamedContent getCapa() throws IOException, SQLException {
        if (livro.getCapa() == null) {
            return null;
        }
        return new DefaultStreamedContent(
            new ByteArrayInputStream(livro.getCapa()));
    }
}
```



Banco

+ Opções

					id	capa		
<input type="checkbox"/>		Edita		Copiar		Apagar	4	[BLOB - 43.4 KB]
<input type="checkbox"/>		Edita		Copiar		Apagar	5	NULL

Armazenamento em Diretório

```
@Entity
@Table(name="livros")
public class Livro implements Serializable {
    @Id
    @GeneratedValue
    private int id;
    private String titulo;
    private String sinopse;
    private int anoPublicacao;
    private String editora;
    private boolean doacao;

    private String urlCapa;
```



CadastroLivrosBean

```
@ManagedBean(name = "livrosBean")
@SessionScoped
public class CadastroLivrosBean {

    private Livro livro;
    private List<Livro> listaLivros;
    private LivroDAO dao = new LivroDAO();
    private AutorDAO autorDAO = new AutorDAO();

    byte[] imgCapa;
    String nomeArquivoCapa;
```



CadastroLivrosBean

```
public void upload(FileUploadEvent event)
    throws IOException {
    imgCapa = new byte[(int)
        event.getFile().getSize()];
    event.getFile().getInputStream().read(imgCapa);
    nomeArquivoCapa = event.getFile().getFileName();
}
```



CadastroLivrosBean.salvar()

```
public void salvar() throws IOException {
    boolean novo = livro.isNovo();
    Autor autor = autorDAO.carregar(idAutor);
    livro.setAutor(autor);
    if (imgCapa != null) {
        int index = nomeArquivoCapa.lastIndexOf('.');
        String extension = "";
        if (index >= 0)
            extension = nomeArquivoCapa.substring(index);
        String nomeArq = (Math.random() * 1000)
            + System.currentTimeMillis() + extension;
        File dir = new File( PATH_IMG + File.separator + "capas");
        if(!dir.exists())
            dir.mkdirs();
        FileOutputStream out = new FileOutputStream(new File(dir, nomeArq));
        out.write(imgCapa);
        out.close();
        livro.setUrlCapa(nomeArq);
    }
}
```



CadastroLivrosBean

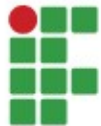
```
public void carregar(Livro livro)
    throws SQLException {
    this.livro = livro;
    imgCapa = null;
    nomeArquivoCapa = null;
}
```

```
public void novoLivro() {
    livro = new Livro();
    imgCapa=null;
    nomeArquivoCapa=null;
}
```


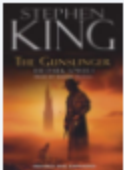


CadastroLivrosBean

```
public StreamedContent getCapa() throws IOException, SQLException {  
    if (imgCapa != null) {  
        return new DefaultStreamedContent(  
            new ByteArrayInputStream(imgCapa));  
    } else if (livro.getUrlCapa() != null) {  
        return new DefaultStreamedContent(  
            new FileInputStream(  
                PATH_IMG + File.separator + "capas"  
                + File.separator + livro.getUrlCapa()  
            ));  
    }  
    return null;  
}
```



Exibir Imagem no dataTable

ID ↕	Título ↕ <input type="text"/>	Capa
4	Cinquenta tons de cinza	
5	A Torre Negra	

⏪ ⏴ 1 ⏵ ⏩



Servidor de Imagens

```
@WebServlet("/images/*")  
public class ImageServlet extends HttpServlet {  
  
    public final static String PATH_IMG = "C:\\img";  
}
```



Servidor de Imagens

```
@Override
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    String filename = request.getPathInfo().substring(1);
    File file = new File(PATH_IMG, filename);
    if(file.exists())
    {
        response.setHeader("Content-Type",
            getServletContext().getMimeType(filename));
        response.setHeader("Content-Length", String.valueOf(file.length()));
        response.setHeader("Content-Disposition",
            "inline; filename=\"" + filename + "\"");
        Files.copy(file.toPath(), response.getOutputStream());
    }
    else
        response.setStatus(HttpServletResponse.SC_NOT_FOUND);
}
```



DataTable

```
<p:dataTable value="#{livrosBean.listaLivros}" var="livro" pagina=1>
  <p:column headerText="ID" width="70" sortBy="#{livro.id}">
    #{livro.id}
  </p:column>
  <p:column headerText="Título" sortBy="#{livro.titulo}"
    filterBy="#{livro.titulo}">
    #{livro.titulo}
  </p:column>
  <p:column headerText="Capa" style="text-align: center" >
    <p:graphicImage width="60px" cache="false"
      rendered="#{livro.urlCapa!=null}"
      value="/images/capas/#{livro.urlCapa}">
    </p:graphicImage>
  </p:column>
</p:dataTable>
```

