

Template para Elaboração de Material Didático Multimídia com AsciiDoc, git e GitHub

Sumário

1. Instalação das Ferramentas no seu Ambiente de Desenvolvimento	2
2. Resumo de Comandos AsciiDoc	2
2.1. Seções/Títulos, Numeração	2
2.2. Marcadores	3
2.2.1. Para criar uma lista de marcadores, basta usar um hífen em cada linha:	3
2.2.2. Para numerar os itens, basta usar números sequenciais ou apenas um 1 na frente de cada item:	3
2.3. <i>Itálico</i> e Negrito	3
2.4. Links	3
2.4.1. Link para documento local	3
2.4.2. Link para página externa	3
2.4.3. Link para outros arquivos asciidoc	4
2.5. Tabelas	4
2.6. Imagens	4
2.6.1. Imagem em bloco (em um linha separada)	5
2.6.2. Imagem com Título	5
2.6.3. Gifs animados	7
2.7. Vídeos	7
2.8. Notas	7
2.8.1. Notas em Blocos (com quebra de linha)	8
2.9. Exibindo código	8
2.9.1. Exibindo blocos de código	8
2.10. Incluindo arquivos externos no documento adoc	8
2.10.1. Incluindo apenas trechos de um arquivo	9
2.10.2. Destacando código fonte incluído a partir de um arquivo	10
3. GitHub Pages	11
4. Licença	11
5. Referências	11



Você pode baixar uma versão sempre atualizada em PDF deste material [aqui](#).

Este repositório serve como um template para a criação de material didático multimídia utilizando

AsciiDoc e hospedando no GitHub Pages. Ele apresenta instruções de como configurar seu ambiente de desenvolvimento para poder escrever conteúdo em AsciiDoc e publicar no GitHub Pages. Adicionalmente, há um resumo dos comandos AsciiDoc para quem estiver iniciando na produção de conteúdo neste formato.

Assim, utilize este arquivo README.adoc como exemplo pois ele já está configurado. Para criar seu próprio conteúdo, basta apagar o texto neste arquivo e incluir o seu. Quando for criar novos arquivos, basta copiar o README.adoc anterior e iniciar a partir daí.

1. Instalação das Ferramentas no seu Ambiente de Desenvolvimento

Execute o script [ferramentas.sh](#) em um terminal com sistema Ubuntu ou qualquer derivado de Debian:

```
sh ferramentas.sh
```

Se sua distribuição for antiga ou não for 64 bits, pode ser que a instalação do Visual Studio code não funcione. Se ocorrer erro, baixe o Visual Studio Code manualmente em <http://code.visualstudio.com>

Crie uma conta no GitHub em <https://github.com/signup>. O nome de usuário se torna o link para sua conta pública, como <http://github.com/manoelcampos>. Por isso, escolha o nome apropriadamente.

Para qualquer alteração em um projeto que você desejar enviar para o GitHub a partir da sua máquina, você precisará informar nome de usuário e senha. Isso se tornará um processo bastante chato e toma tempo. Para evitar isso, na nossa máquina pessoal, podemos configurar a autenticação por meio de um par de chaves pública/privada.

Para isto, siga o tutorial do vídeo abaixo.

► <https://www.youtube.com/watch?v=j7lX9ff9rC8> (YouTube video)

Autenticação baseada em chave no GitHub a partir do Linux

2. Resumo de Comandos AsciiDoc

2.1. Seções/Títulos, Numeração

```
= Título
```

```
== Título 1
```

```
=== Título 2
```

```
==== Título 3
```

Para numerar os títulos, basta adicionar a linha abaixo no topo do documento:

```
:numbered:
```

2.2. Marcadores

2.2.1. Para criar uma lista de marcadores, basta usar um hífen em cada linha:

```
- Item A  
- Item B  
- Item C
```

2.2.2. Para numerar os itens, basta usar números sequenciais ou apenas um 1 na frente de cada item:

```
1. Item 1  
1. Item 2  
1. Item 3
```

2.3. *Itálico* e **Negrito**

```
_Itálico_
```

```
*Negrito*
```

2.4. Links

2.4.1. Link para documento local

```
link:documento.ext[Documento]
```

[Apresentação sobre WebSockets](#)

2.4.2. Link para página externa

```
http://palmas.ifto.edu.br  
http://palmas.ifto.edu.br[IFTO Palmas]
```

<http://ifto.edu.br>

2.4.3. Link para outros arquivos asciidoc

Se você usar a tag link para outro arquivo `.adoc`, tal link só funcionará na interface do GitHub. Por outro lado, se você está compilando os AsciiDocs para HTML e hospedando como um site convencional em qualquer lugar como o GitHub Pages (que não exibe a interface do GitHub), tais links não funcionarão, pois o navegador não interpreta arquivos `.adoc`, somente sites como o GitHub tem essa capacidade.

Por outro lado, se você colocar o link com extensão `.html`, vai funcionar na sua hospedagem mas não vai funcionar na interface padrão do GitHub. Para resolver esse dilema, neste caso, use o comando `xref` no lugar de `link`. A estrutura é exatamente a mesma:

```
xref:documento.adoc[Documento AsciiDoc]
```

Tal comando vai gerar um link, incluindo a extensão apropriada, dependendo de onde e como o arquivo AsciiDoc está sendo aberto. Mas para isso funcionar, é preciso incluir o comando `if` abaixo no topo de cada documento:

```
ifdef::env-github,env-browser[]
:outfilesuffix: .adoc
endif::[]
```

2.5. Tabelas

```
|===
| Coluna A | Coluna B | Coluna C
| Célula A1 | Célula B1 | Célula C1
| Célula A2 | Célula B2 | Célula C2
|===
```

Coluna A	Coluna B	Coluna C
Célula A1	Célula B1	Célula C1
Célula A2	Célula B2	Célula C2

Veja que basta separar as colunas por `|` e iniciar e terminar a tabela com `|===`. Você pode destacar os títulos usando negrito, por exemplo, como **Coluna A**.

2.6. Imagens

Imagem no meio do texto.

Um texto image:url-ou-endereço-local[] qualquer aqui.

Esta imagem [evolution of man and computer] mostra a evolução do homem e dos computadores.

2.6.1. Imagem em bloco (em um linha separada)

Um texto aqui.

.Título Opcional
image::url-ou-endereço-local[]

Outro texto aqui. Observe as linhas em branco antes e depois.

A imagem não pode ficar junto com o texto.

[thread selectors] | *thread-selectors.png*

Figura 1. Seletores para execução de tarefas concorrentes em uma Thread

Ela deve ficar em uma linha separada.

2.6.2. Imagem com Título

.Título da Imagem aqui
image::url-ou-endereço-local[]

[314px Laptop hard drive exposed] |

<https://upload.wikimedia.org/wikipedia/commons/thumb/f/f8/Laptop-hard-drive-exposed.jpg/314px->

Laptop-hard-drive-exposed.jpg

Figura 2. Hard Disk Drive - HDD (Fonte: Wikipedia)

2.6.3. Gifs animados

Pode-se usar qualquer uma das formas acima, por exemplo:

```
.Titulo  
image:url-ou-endereço-local[]
```

Escalonamento horizontal de serviços

[horizontal scaling small]

2.7. Vídeos

```
.Título Opcional  
video::SALUHrZz2OQ[youtube]  
  
video::SALUHrZz2OQ[youtube]  
  
video::SALUHrZz2OQ[youtube, width=640, height=480]
```

▶ <https://www.youtube.com/watch?v=SALUHrZz2OQ> (YouTube video)

Locks otimistas em aplicações de bancos de dados

2.8. Notas

Existem 5 tipos de notas, que são blocos no meio do texto para chamar a atenção para algo. A forma simplificada não permite quebra de linha no texto.

NOTE: Texto da nota aqui.

TIP: Texto da dica aqui.

WARNING: Texto do aviso aqui.

CAUTION: Texto de alerta aqui.

IMPORTANT: Texto importante aqui.



Esta é uma nota qualquer.



Esta é uma dica qualquer.



Este é um aviso qualquer.



Este é um alerta qualquer.



Esta é uma informação importante.

2.8.1. Notas em Blocos (com quebra de linha)

[NOTE]

====

Sua nota aqui.

Ela pode contar quantas linhas desejar.

Você pode até ter linhas em branco.

====



Sua nota aqui. Ela pode contar quantas linhas desejar.

Você pode até ter linhas em branco.

2.9. Exibindo código

Podemos exibir código em qualquer linguagem que desejarmos dentro de um arquivo adoc. Se precisamos exibir apenas uma única linha de código, basta envolver tal linha entre apóstrofes, como ``linha de código aqui``. O exemplo abaixo mostra um texto com um comando Linux no meio, tal comando envolvido em apóstrofes. Assim, o comando fica destacado.

O comando `sudo apt-get install asciidoctor` instala o compilador de AsciiDoc em sistemas Ubuntu e derivados de debian.

2.9.1. Exibindo blocos de código

[source, nome-linguagem-programacao-aqui]

seu
código
fonte
aqui

2.10. Incluindo arquivos externos no documento adoc

Podemos incluir conteúdo de qualquer outro arquivo dentro de um arquivo adoc. Tais arquivos

devem estar em formato de texto (como arquivos txt, arquivos de configuração, de código fonte, xml, json, sh, etc).

```
.Titulo opcional  
include::caminho-ou-url-do-arquivo[]
```

Se usarmos o comando acima para incluirmos um documento adoc dentro de outro, não será exibido o código fonte adoc, mas sim o conteúdo final após o adoc ser compilado.



O comando **include** ainda não funciona na interface do GitHub, apenas no GitHub Pages. No GitHub, ele vai apenas criar um link para o arquivo, no lugar de exibir seu conteúdo.

Exemplo de cálculo do quadrado de um número em Java

```
public class Quadrado{ /** * Calcula o quadrado de um número. */ public static double  
quadrado(double valor){ return valor * valor; }
```

```
    public static void main(String[] args) {  
        double valor = 4;  
        System.out.printf("O quadrado de %.2f é %.2f\n", valor, quadrado(valor));  
    }  
}
```

2.10.1. Incluindo apenas trechos de um arquivo

Se desejarmos incluir apenas um determinado trecho de um arquivo no nosso documento adoc, podemos delimitar tal trecho dentro do arquivo por meio de uma tag. A tag deve ter um nome, como preferirmos. Se desejarmos incluir um trecho de um arquivo de código fonte como Java, a abertura e fechamento da tag deve estar comentada. Como a tag representa código AsciiDoc e não código Java, se não comentarmos a tag, o código Java não irá compilar.

Depois de escolher o nome para sua tag, para delimitar o bloco de código que podemos selecionar para exibir no documento adoc, basta delimitar tal bloco com as linhas **// tag::nome_da_tag[]** e **// end::nome_da_tag[]**, como no exemplo a seguir.

```
// tag::nome_da_tag[]  
Linhas de código  
dentro de arquivo de código fonte,  
que desejamos exibir apenas o trecho envolvido na tag  
// end::nome_da_tag[]
```

Veja que foi usado **//** para comentar a tag pois nosso exemplo é com Java. Assim, o arquivo Java fica como abaixo:

```
import java.util.Scanner;

public class Teclado{

    // tag::main[]
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite seu nome: ");
        String nome = scanner.nextLine();
        System.out.println("Olá " + nome);
    }
    // end::main[]
}
```

Veja que o nome da tag foi chamada de main, justamente por envolver apenas o método `main()` da classe Java de exemplo. Para então exibir apenas o trecho delimitado pela tag no arquivo de código fonte, podemos fazer:

```
.Título opcional
include::caminho-ou-url-do-arquivo[tag=nome_da_tag]
```

O resultado é então como abaixo:

Capturando dados do teclado em Java

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Digite seu nome: ");
    String nome = scanner.nextLine();
    System.out.println("Olá " + nome);
}
```

2.10.2. Destacando código fonte incluído a partir de um arquivo

Em todos os exemplos de inclusão de arquivos de código fonte externo, você pode notar que o código não foi destacado (colorido). Para fazer isto, basta colocar o `include` dentro de um bloco `[source]`, indicando o tipo de conteúdo do arquivo (no exemplo, java) como abaixo:

```
.Título opcional
[source, java]
----
include::caminho-ou-url-do-arquivo[tag=nome_da_tag]
----
```

Observe que o título (se tiver), agora deve ficar no bloco `source` e não no comando `include`. O

resultado fica então:

Listagem 1. Capturando dados do teclado em Java (veja que o código agora está colorido)

```
import java.util.Scanner;

public class Teclado{

    // tag::main[]
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite seu nome: ");
        String nome = scanner.nextLine();
        System.out.println("Olá " + nome);
    }
    // end::main[]
}
```

3. GitHub Pages

O repositório usa a [AsciiDoctor GitHub Pages Action](#) para converter cada arquivo `adoc` para `html`, enviando tais arquivos para o branch `gh-pages`. Este processo ocorre automaticamente sempre que um push é executado ou pull request enviada. Com isto, o material estará disponível em um site no GitHub Pages em <http://seu-nome-de-usuario.github.io/nome-do-seu-repositorio>.



É necessário incluir o branch `gh-pages` como fonte de publicação para seu site GitHub Pages nas configurações do seu repositório. [Instruções aqui](#).

4. Licença

Este repositório não possui uma licença definida e você pode utilizar o conteúdo dele livremente para criar seu próprio repositório. Assim, não é preciso dar qualquer tipo de crédito por ter utilizado este projeto (a menos que realmente queira).

Você pode utilizar tal repositório como bem desejar, sem qualquer tipo de limitação. No entanto, o repositório é fornecido como está e nenhuma responsabilidade será atribuída ao autor do mesmo.

Ao produzir seu conteúdo multimídia a partir deste repositório, é importante adicionar um arquivo LICENSE e assim definir explicitamente uma licença para o conteúdo que você produzir.

5. Referências

- [AsciiDoc Official Writer's Guide](#)
- [Toda a Documentação oficial do AsciiDoc](#)
- [Temas em CSS](#)