

Desenvolvimento Aberto



Fluxo de trabalho distribuído e Comunidades de Software

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)

Ordem das apresentações

1. Docker
2. GIMP
3. Kubernetes
4. flatpak
5. Godot engine
6. GNOME
7. TensorFlow
8. digiKam
9. Shotcut

10 minutos cada uma

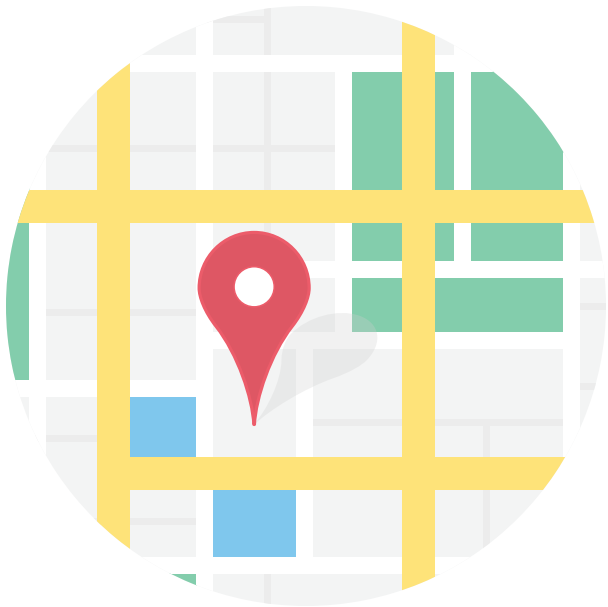
Atividade: Explorando o entorno de um projeto



```
"metadata": {  
  "filename": "nome do arquivo na pasta da aula passada",  
  "group": ["login1", "login2"]  
}
```

- No `group` incluir os outros membros do grupo somente
- Um membro da equipe deve fazer um PR para o projeto incluindo o arquivo da apresentação. O mesmo deve ser colocado na pasta `apresentacoes/2022`

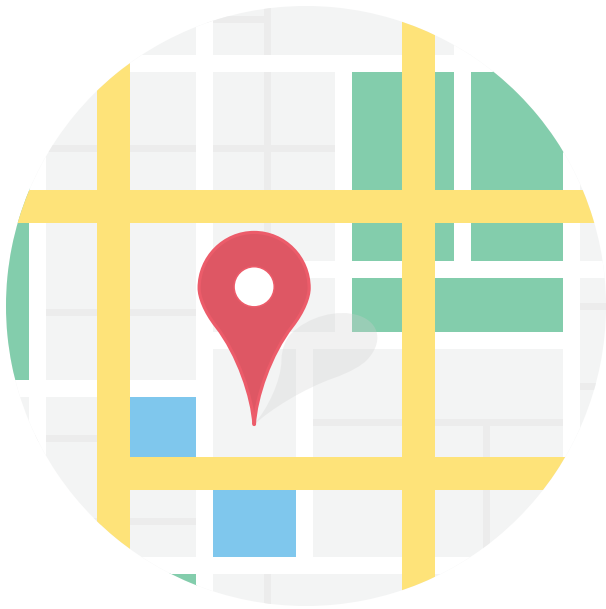
Desenvolvimento Aberto



Documentação de software

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)

Desenvolvimento Aberto



(A ausência de d)ocumentação de software

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)

Documentação de software

Documentação de usuário:

Documentação de desenvolvimento:

Documentação de software

Documentação de usuário:

- Instalação
- Funcionalidades
- Onde obter ajuda

Documentação de software

Documentação de desenvolvimento:

- Como compilar (dependências, ferramentas usadas, etc)
- Como testar (dependências, ferramentas usadas, etc)
- Estilo de código e outras orientações relacionadas
- Organização do código e arquitetura da aplicação

Documentação de software

Às vezes as coisas se confundem! O que vocês colocariam na documentação de usuário do *Python*?

E na de desenvolvedor?

Exemplo: Spyder



HOME BLOG

Navigation

[Overview](#)

[Installation](#)

[Command Line Options](#)

[Editor](#)

[IPython Console](#)

[Variable Explorer](#)

[Help](#)

[Debugging](#)

[Static Code Analysis](#)

[Profiler](#)

[Projects](#)

[File Explorer](#)

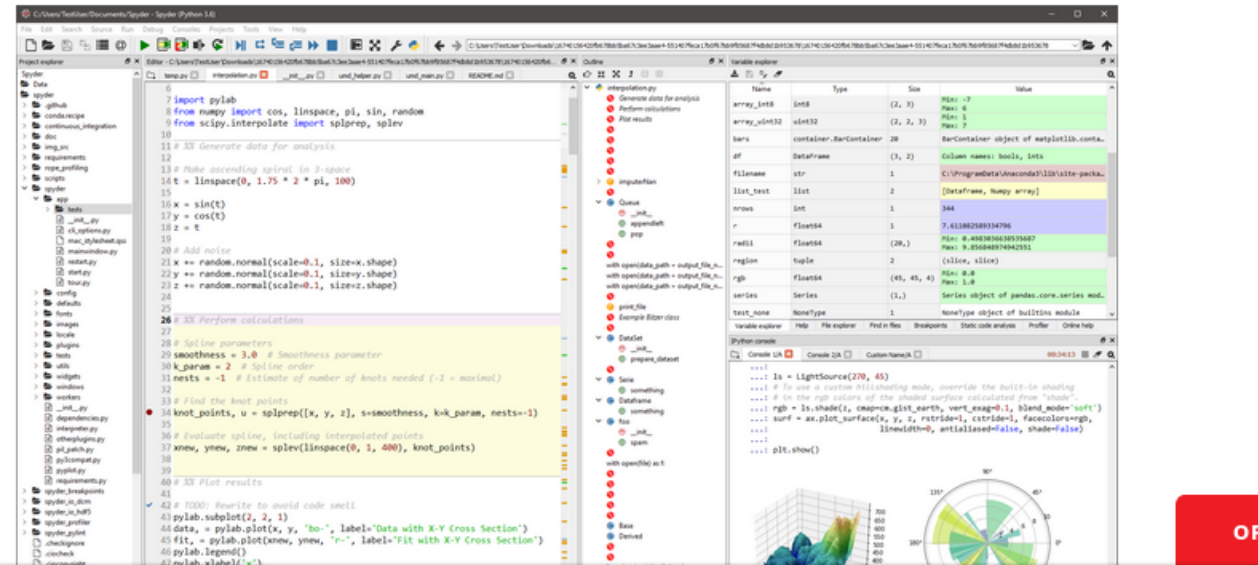
[Find in Files](#)

[Online Help](#)

[History Log](#)

[Internal Console](#)

Spyder: The Scientific Python Development Environment — Documentation



<https://www.spyder-ide.org/>

Igor Montagner













Exemplo: Spyder

Welcome to the Spyder IDE Wiki!

Spyder is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features.

This wiki contains

-  [Contributing to spyder](#)
-  [Current Funding and Development Status](#)
-  [Development information](#)
-  [Troubleshooting Guide and FAQ](#)
-  [SEPs: Spyder Enhancement Proposals](#)
-  [Roadmap](#)
-  [Projects using Spyder](#)
-  [Frequently asked questions](#)
-  [Plugin ideas](#)
-  [User Plugins](#)

▼ Pages 42

[Home](#)

[Anaconda stopped funding
Spyder](#)

[API decoupling](#)

[Beta version changelog](#)

[Changelog](#)

[Contributing to Spyder](#)

[Current Funding and
Development Status](#)

[Dev: Coding Style](#)

[Dev: Cookbook](#)

[Dev: Debugging Spyder](#)

[Dev: Documentation](#)


[Dev: Github Workflow](#)

[Dev: Index](#)

[Dev: Spyder Internals](#)

<https://github.com/spyder-ide/spyder/wiki>

Sistemas de documentação



SPHINX

Python Documentation Generator

HomeGet itDocsExtend/Develop

Welcome

Sphinx is a tool that makes it easy to create intelligent and beautiful documentation, written by Georg Brandl and licensed under the BSD license.


It was originally created for [the Python documentation](#), and it has excellent facilities for the documentation of software projects in a range of languages. Of course, this site is also created from reStructuredText sources using Sphinx! The following features should be highlighted:

- **Output formats:** HTML (including Windows HTML Help), LaTeX (for printable PDF versions), ePub, Texinfo, manual pages, plain text
- **Extensive cross-references:** semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information
- **Hierarchical structure:** easy definition of a document tree, with automatic links to siblings, parents and children
- **Automatic indices:** general index as well as a language-specific module indices
- **Code handling:** automatic highlighting using the [Pygments](#) highlighter
- **Extensions:** automatic testing of code snippets, inclusion of docstrings from Python modules (API docs), and [more](#)
- **Contributed extensions:** more than 50 extensions [contributed by users](#) in a second repository; most of them installable from PyPI

Sphinx uses [reStructuredText](#) as its markup language, and many of its strengths come from the power and straightforwardness of reStructuredText and its parsing and translating suite, the [Docutils](#).

What users say:

"Cheers for a great tool that actually makes programmers **want** to write documentation!"

A  project

Download

Current version: **pypi v1.7.7**

Install Sphinx with:

```
pip install -U Sphinx
```

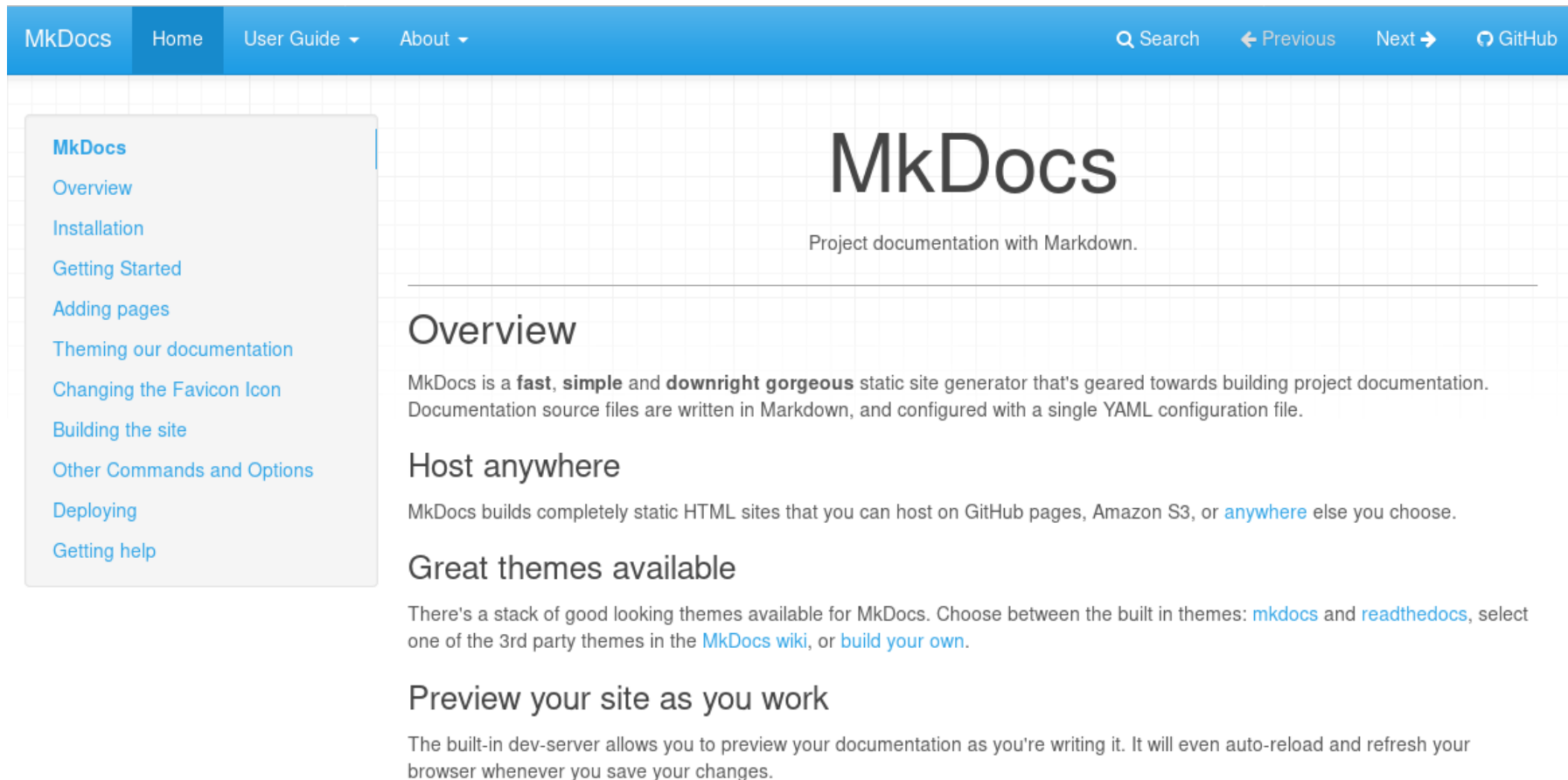
Questions? Suggestions?

Join the [sphinx-users](#) mailing list on Google Groups:

or come to the [#sphinx-doc](#) channel on FreeNode.

Sphinx

Sistemas de documentação



The screenshot shows the MkDocs website interface. At the top is a blue navigation bar with links for 'MkDocs', 'Home', 'User Guide' (with a dropdown arrow), 'About' (with a dropdown arrow), a search icon and 'Search', 'Previous' (with a left arrow), 'Next' (with a right arrow), and a 'GitHub' link with an icon. On the left side, there is a grey sidebar menu with a vertical line on its right edge. The menu items are: 'MkDocs' (highlighted in blue), 'Overview', 'Installation', 'Getting Started', 'Adding pages', 'Theming our documentation', 'Changing the Favicon Icon', 'Building the site', 'Other Commands and Options', 'Deploying', and 'Getting help'. The main content area has a light grey grid background. It features the 'MkDocs' logo in a large, dark font, followed by the tagline 'Project documentation with Markdown.' Below this is a horizontal line. The 'Overview' section follows, with a sub-header 'Overview' and a paragraph describing MkDocs as a 'fast, simple and downright gorgeous' static site generator. The 'Host anywhere' section has a sub-header and a paragraph stating that MkDocs builds static HTML sites hostable on GitHub pages, Amazon S3, or 'anywhere'. The 'Great themes available' section has a sub-header and a paragraph about available themes, including 'mkdocs' and 'readthedocs'. The 'Preview your site as you work' section has a sub-header and a paragraph about the built-in dev-server.

MkDocs

Project documentation with Markdown.

Overview

MkDocs is a **fast, simple** and **downright gorgeous** static site generator that's geared towards building project documentation. Documentation source files are written in Markdown, and configured with a single YAML configuration file.

Host anywhere

MkDocs builds completely static HTML sites that you can host on GitHub pages, Amazon S3, or [anywhere](#) else you choose.

Great themes available

There's a stack of good looking themes available for MkDocs. Choose between the built in themes: [mkdocs](#) and [readthedocs](#), select one of the 3rd party themes in the [MkDocs wiki](#), or [build your own](#).

Preview your site as you work

The built-in dev-server allows you to preview your documentation as you're writing it. It will even auto-reload and refresh your browser whenever you save your changes.

Mkdocs

Sistemas de documentação (hospedagem)



Read the Docs

Create, host, and browse documentation.

Sign up

or [Log in](#)

Technical documentation lives here

Read the Docs simplifies software documentation by automating building, versioning, and hosting of your docs for you.

Free docs hosting

We will host your documentation for free forever. There are no tricks. We help 89,019 open source projects share their docs.

Webhooks

Whenever you push code to your favorite version control system, whether that is Git, Mercurial, Bazaar, or Subversion, we will automatically build your docs so your code and documentation are never out of sync.

<https://readthedocs.org/>

Atividade

Vocês receberão um *zip* com o código de um software e zero instruções. Vocês deverão:

1. Aprender como rodar o software
2. Documentar os passos que vocês seguiram para fazê-lo
3. Fazer uma análise crítica do projeto com relação a
 - Arquitetura
 - Qualidade do código
 - Segurança da solução

Objetivos desta atividade

1. Entender na prática a necessidade de documentação;
2. Identificar quais são as dependências de um software. Isto inclui bibliotecas/frameworks usados e serviços aos quais esse software se conecta.
3. Rodar sua própria versão de um software de terceiros

Não são objetivos desta atividade

1. Mostrar nenhum tipo de boa prática de programação
2. Modificar o software analisado
3. Criticar o desenvolvedor que trabalhou antes no projeto

Atividade prática: Projeto não documentado



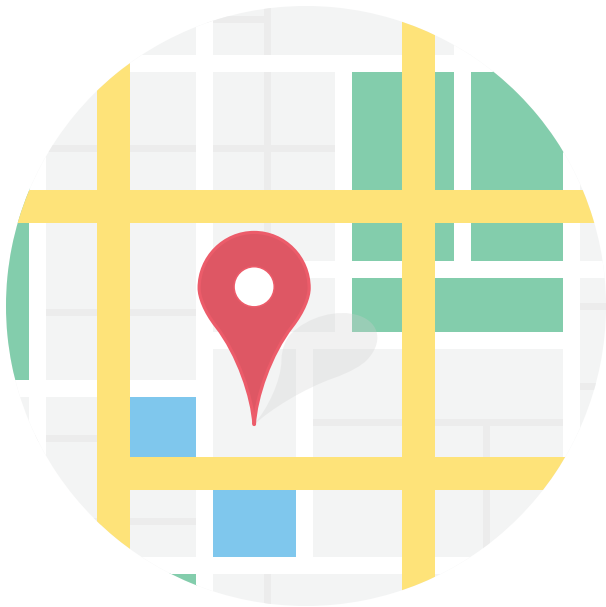
Objetivo: entender como rodar um código sem instruções.

Validação: apresentação do sistema funcionando.

Grupos de até 3 pessoas

Tempo restante de hoje + hora inicial da próxima aula. Usem handout como guia

Desenvolvimento Aberto



Documentação de software

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)