Desenvolvimento Aberto

Fluxo de trabalho distribuído e Comunidades de Software

Versão 2022/2: Fabrício Barth (fabriciojb@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

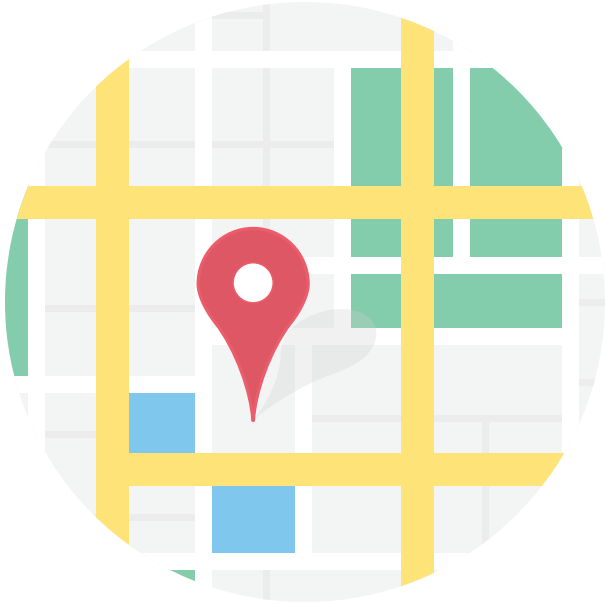
10 minutos cada uma

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`

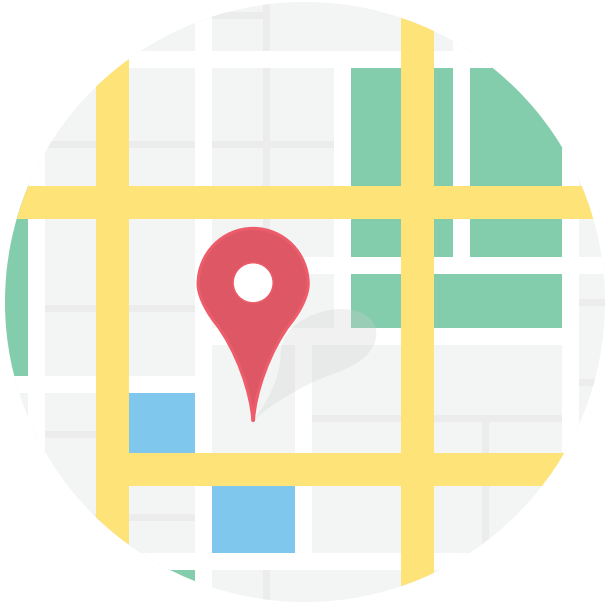
Desenvolvimento Aberto



Documentação de software

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

Desenvolvimento Aberto



(A ausência de d)ocumentação de software

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

Documentação de software

Documentação de usuário:

Documentação de desenvolvimento:

Igor Montagner

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

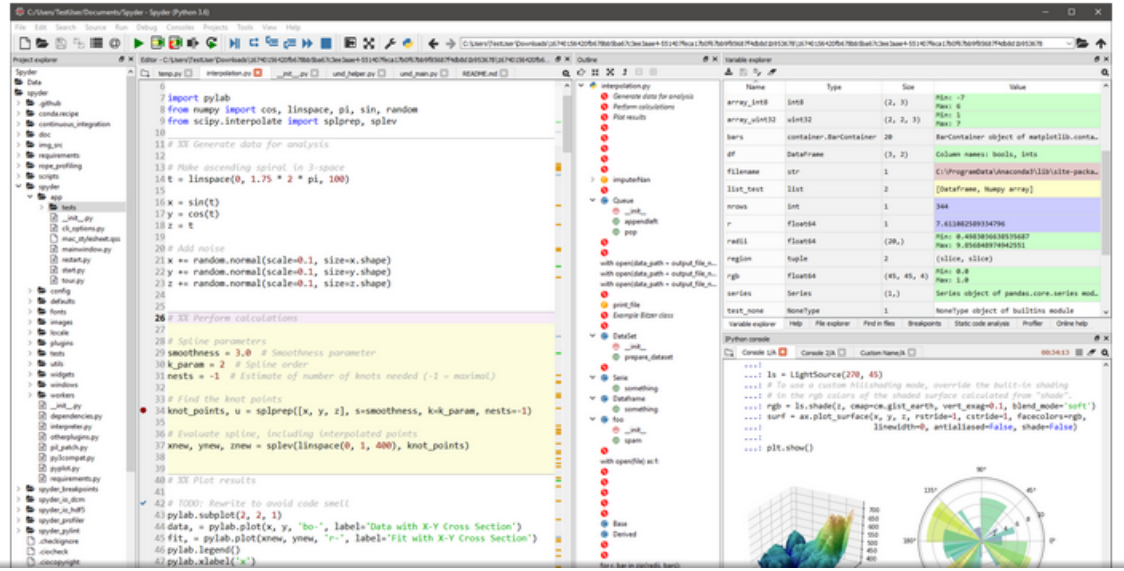
SPYDER | DOCS

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/

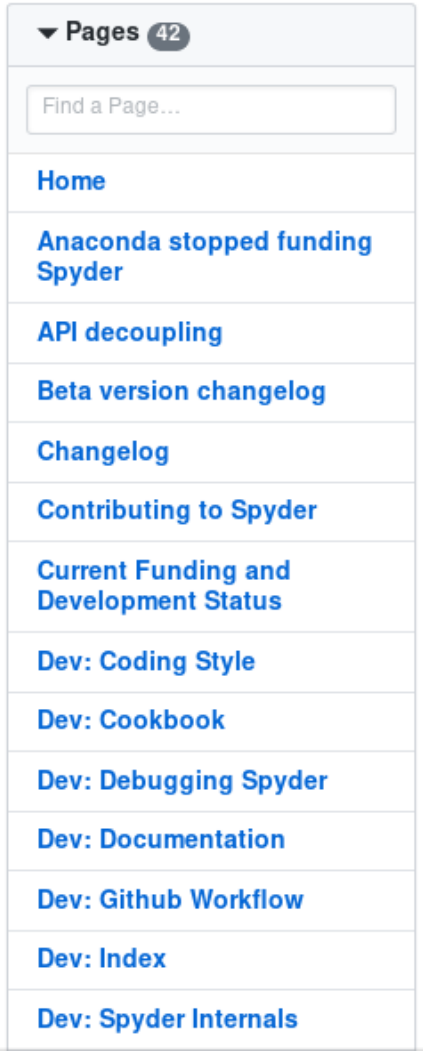
Exemplo: Spyder

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.

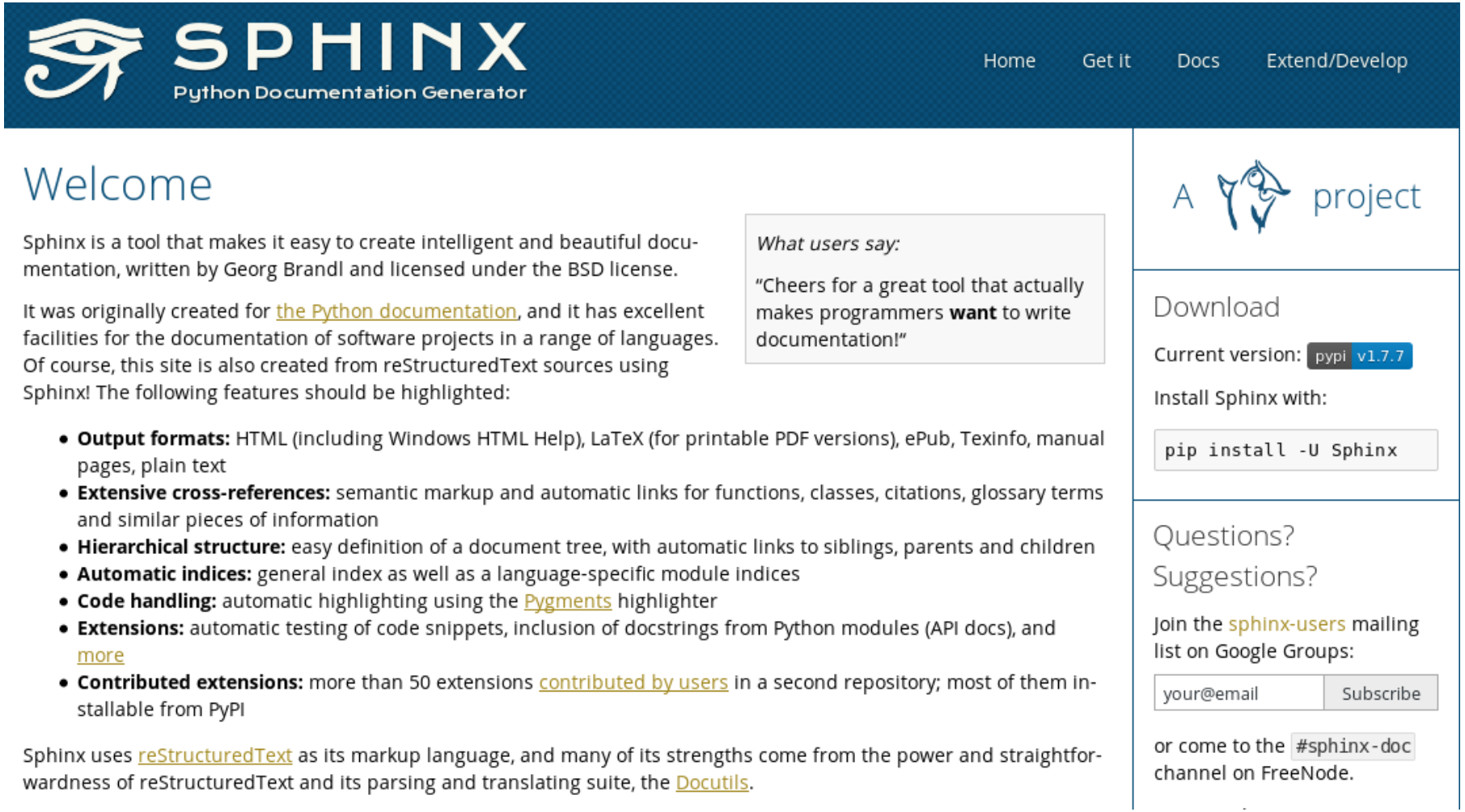
This wiki contains

- 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



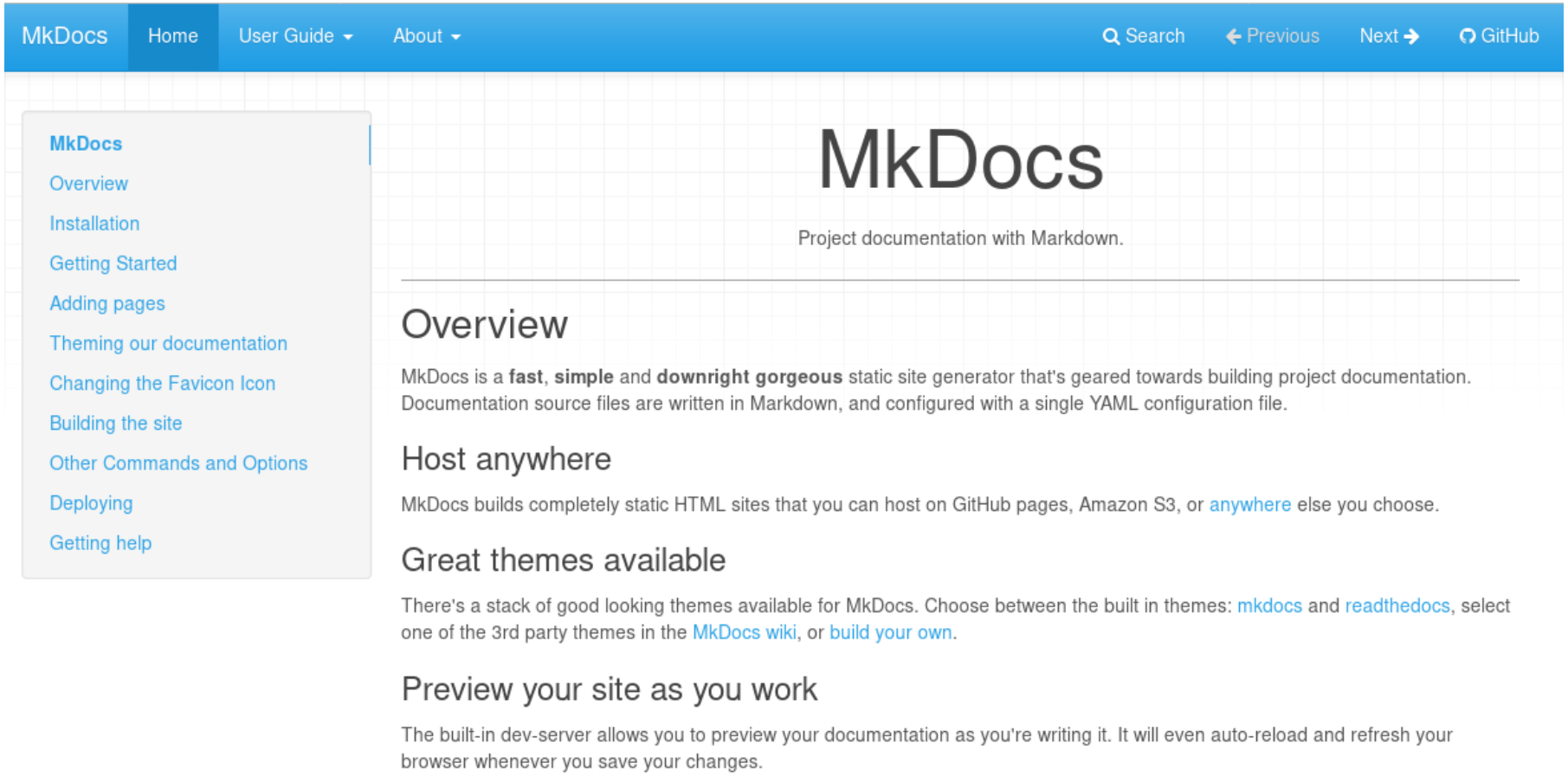
https://github.com/spyder-ide/spyder/wiki

Sistemas de documentação



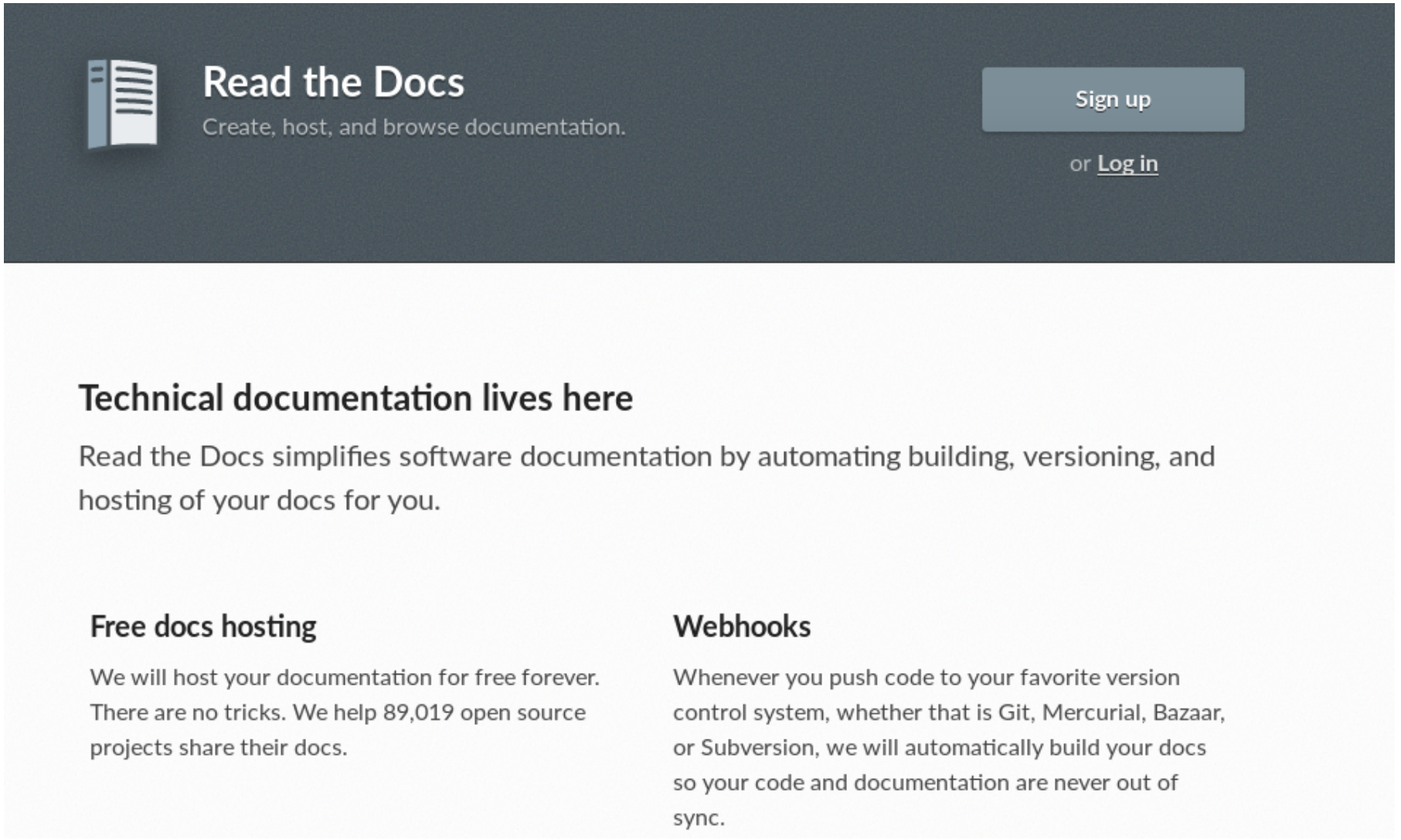
Sphinx

Sistemas de documentação



Mkdocs

Sistemas de documentação (hospedagem)



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

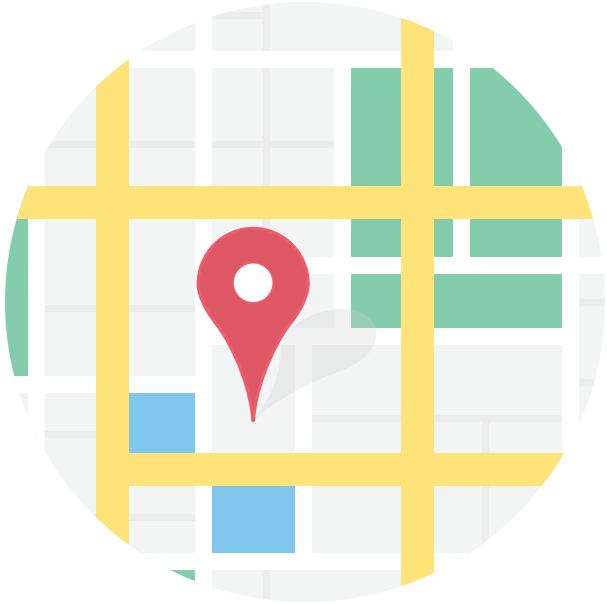
Objetivo: entender como rodar um código sem instruções.

Validação: apresentação do sistema funcionando.

Grupos de até 3 pessoas

Tempo restante de hoje + hora inicial de quinta. Usem handout como guia

Desenvolvimento Aberto



Documentação de software

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