

Desenvolvimento Aberto

```
4  #include "utilities.h"
5
6  Game game;
7
8  int main(int argc, char **argv) {
9
10     game = Game();
11
12     game.initialize();
13 }
```

Explorando seu projeto

Igor dos Santos Montagner (igorsm1@insper.edu.br)

Aulas passadas

- Workflow para projetos distribuídos
- Escolher projeto e issue para primeira contribuição
 - Registre a issue escolhida em [insper/dev-aberto/issues/45](https://insper.dev-aberto/issues/45)

Pandas (Issues)

Formatação

- [\[1\]](#), [\[2\]](#), [\[3\]](#)

Pré-processamento

- [\[1\]](#), [\[2\]](#), [\[3\]](#)

Entrada/saída:

- [\[1\]](#), [\[2\]](#)

Visualização

- [\[1\]](#)

Sphinx

Projeto para documentação de projetos. Usado para documentar o próprio Python.

- Trabalha com textos em diversos formatos
- Faz referência entre documentos e com código fonte.

Exemplos

- [\[1\]](#)

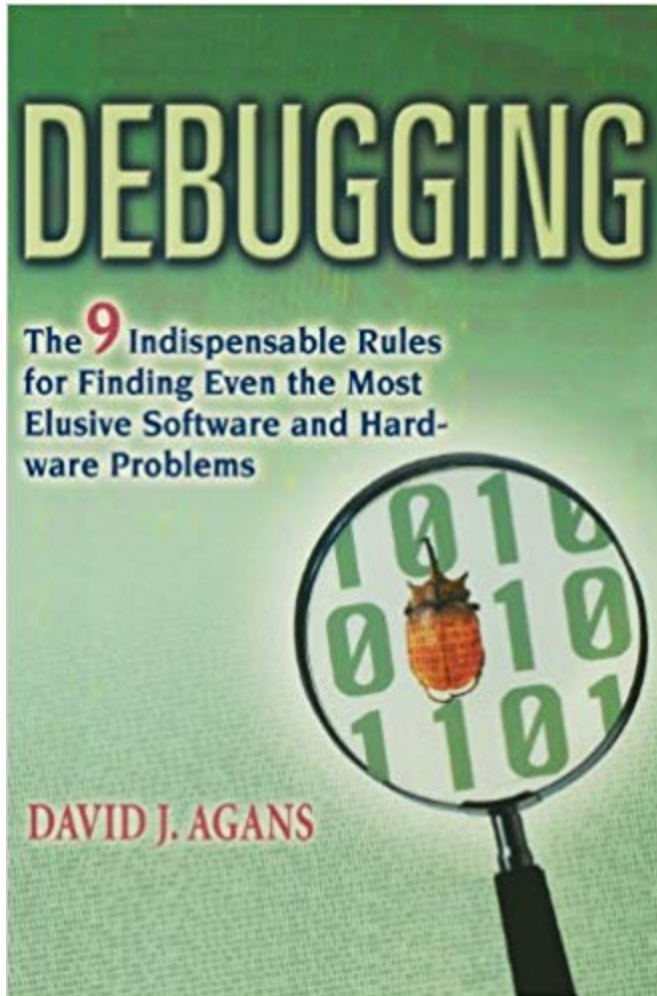
Documentos e formatação

- [\[1\]](#) [\[2\]](#) [\[3\]](#)

"Regras" para debugar qualquer coisa

Ferramentas para compilar e se achar em um projeto

9 regras de debug



9 regras de debug

1. UNDERSTAND THE SYSTEM
2. MAKE IT FAIL
3. QUIT THINKING AND LOOK
4. DIVIDE AND CONQUER
5. CHANGE ONE THING AT A TIME
6. KEEP AN AUDIT TRAIL
7. CHECK THE P LUG
8. GET A FRESH VIEW
9. IF YOU DIDN'T FIX IT, IT AIN'T FIXED

1. Understand the system

Nada acontece se não conseguirmos

1. Baixar a versão de desenvolvimento
2. Instalar todas as dependências
3. Compilar nossa própria versão
4. Rodar testes na versão do `master`

1. Understand the system

Ferramentas importantes:

1. [Virtual environments](#): venv, virtualenv, pipenv, conda
2. [setuptools](#) - *development mode*
3. Debugging
 - *pdb*
 - função `breakpoint()` ([tutorial](#))

2. Make it fail

Reproduza seu bug

- Criar um exemplo mínimo que reproduza o bug desejado ou que usaria a feature que quer implementar
- Possivelmente isto já foi descrito na *issue* escolhida.

3. Quit thinking and look

Encontrar onde está o problema e ler o código com atenção

1. Encontre no código onde o bug pode estar
2. Comece geral (em qual arquivo está a funcionalidade?) e vá restringindo (em qual função o bug "explode"?)
3. Ferramentas de debug são essenciais. Veja o item 1.
 - Não atenderei ninguém que esteja debugando só com `print`

3. Quit thinking and look

Buscando por nomes de arquivos

Comando `find` ([man page](#))

Exemplo: procurar por arquivos cujo nome é aceito por uma certa expressão regular começando no diretório atual.

```
$ find -iname "regexp" .
```

3. Quit thinking and look

Buscando no conteúdo dos arquivos

Comando `egrep` ([man page](#))

```
$ egrep [OPTIONS] PATTERN FILES
```

- `PATTERN` : expressão regular
- `FILES` : lista de diretórios ou arquivos

3. Quit thinking and look - grep

Exemplo 1: buscar todos arquivos nas pasta atual (.) e subpastas com o texto "dialog" ignorando maiúsculas/minúsculas.

```
$ grep -r -i dialog .
```

Exemplo 2: Listas todos os arquivos *.cpp* que fazem algum include

```
$ grep -r --include "*.cpp" "#include" .
```

4. Divide and Conquer

Criar plano de ação

1. Por que o bug ocorre?
 - Está relacionado a qual função?
 - Qual variável tem o valor errado?
2. O que deve ser mudado para que pare de ocorrer?

Pré-requisito: escolher uma linha da função que dá pau e colocar um `breakpoint()`

5. Change one thing at a time

Um bom PR muda o mínimo possível

- use várias branches se quiser testar ideias diferentes

6. KEEP AN AUDIT TRAIL

Registre suas descobertas

- Log de como você encontrou onde mexer
- Log de todos arquivos de interesse e seus usos
- Log de todas as pesquisas feitas
- Log de todas as modificações feitas

Não se esqueça: `git diff` é seu melhor amigo

7. Check the plug

Nunca se esqueça de testar a soluções mais simples primeiro

8. GET A FRESH VIEW

Empacou?

É por isso que vocês trabalham em grupo no primeiro bug.

Empacou mesmo?

Para isto serve seu grupo.

Empacou mesmo?!?!?!?

Me chame!

9. IF YOU DIDN'T FIX IT, IT AIN'T FIXED



Hora de programar

Desenvolvimento Aberto

```
4  #include "utilities.h"
5
6  Game game;
7
8  int main(int argc, char **argv) {
9
10     game = Game();
11
12     game.initialize();
13 }
```

Explorando seu projeto

Igor dos Santos Montagner (igorsm1@insper.edu.br)