

Série: Criando e mantendo um software seguro

Episódio 1:

Oficina – Práticas de Análise Estática de Código

Teoria

- Shift Left
- Systems Development Life Cycle – SDLC
- Security Software Development Lifecycle – SSDLC
- Frameworks
- Automatização de Testes
- Continuous Integration
- DevOps & DevSecOps
- Pipelines
- Integrações

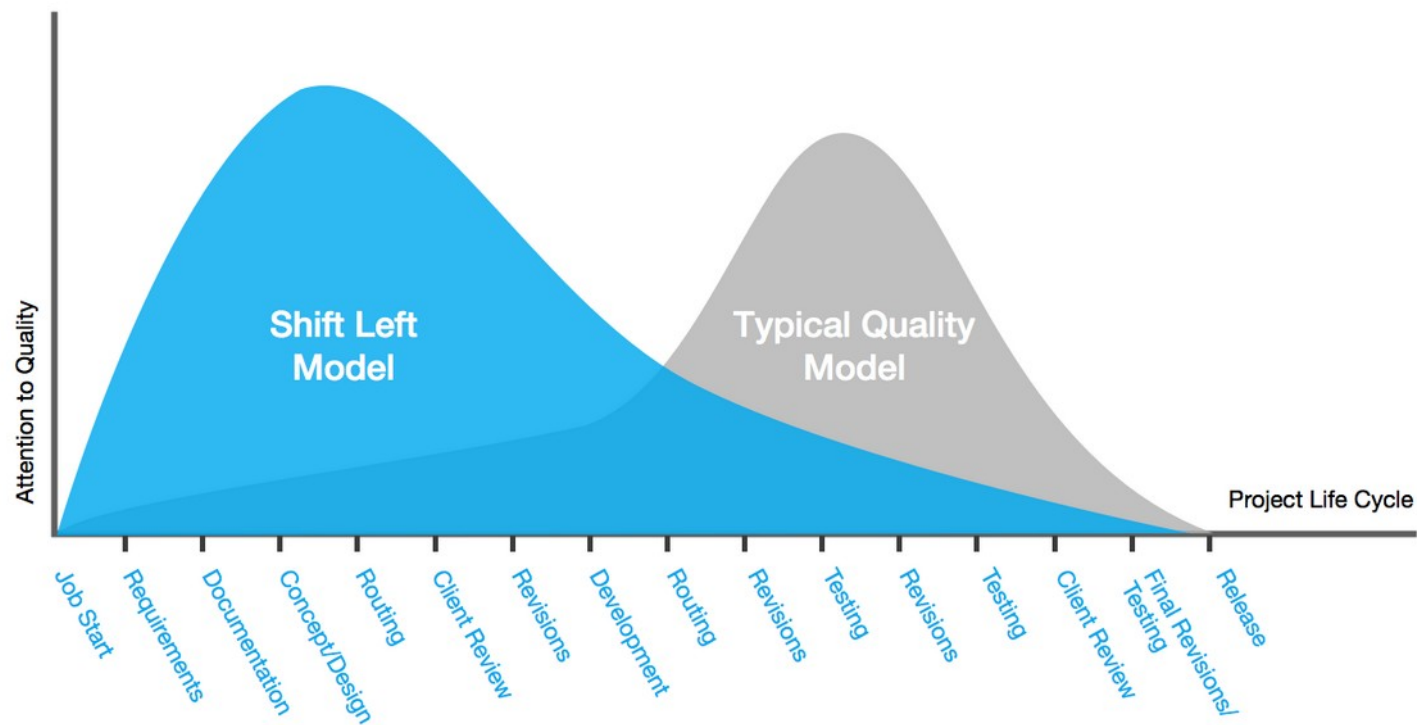
Prática

- Análise estática de código - SAST
- Escopo
- O que é analisado
- Como é analisado
- Escopo de testes
- O que é apresentado após a análise
- Como utilizar os resultados

Shift Left

Série: Criando e mantendo um software seguro

Oficina – Práticas de Análise Estática de Código

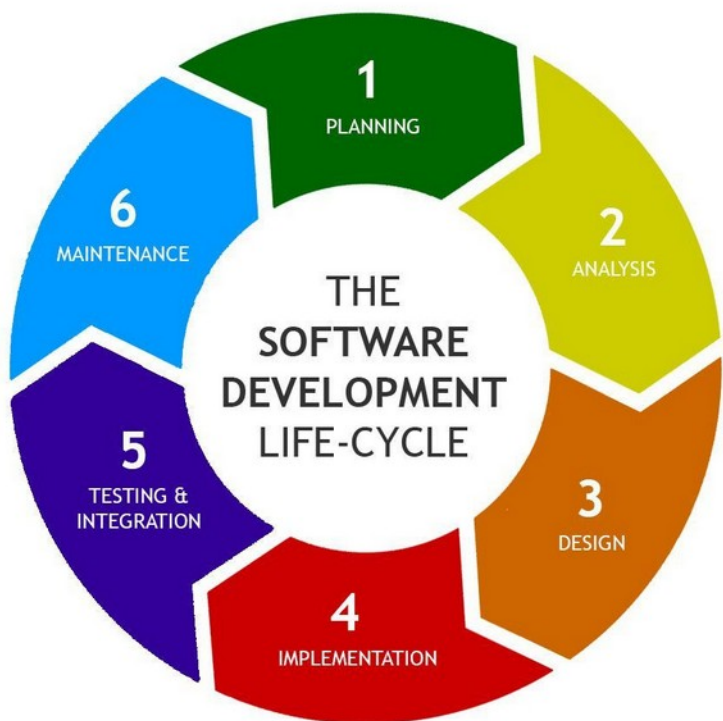


Systems Development Life Cycle – SDLC

Security Software Development Lifecycle – SSDLC

Série: Criando e mantendo um software seguro

Oficina – Práticas de Análise Estática de Código



Frameworks

Referências atuais:

NIST SP 800-160 Vol. 1
NIST SP 800-160 Vol. 2
ISO/IEC/IEEE 15288:2015

Abordagens práticas:

Microsoft Security Development Lifecycle (SDL)
OWASP SAMM

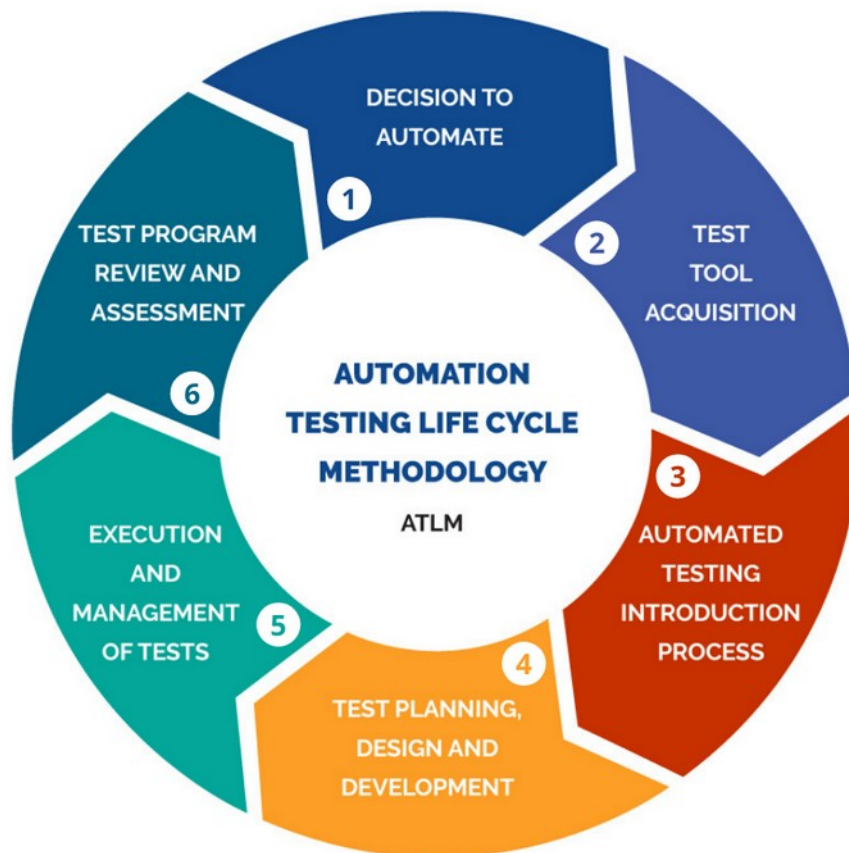
Pelo retrovisor:

NIST SP 800-64 Rev. 2
OWASP CLASP

Automatização de Testes

Série: Criando e mantendo um software seguro

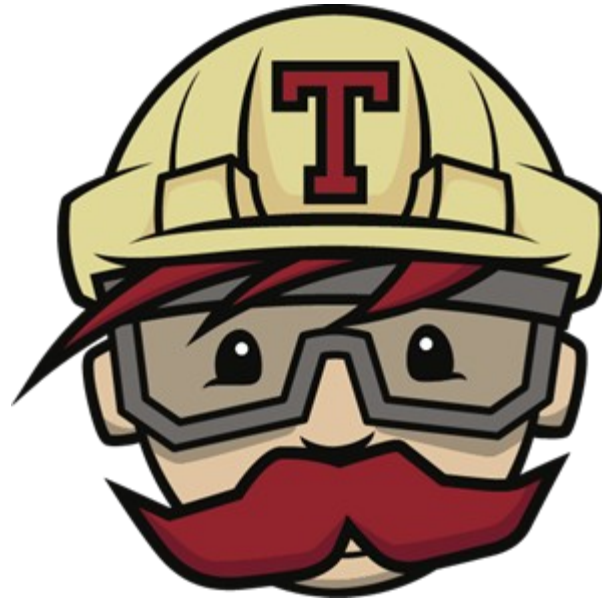
Oficina – Práticas de Análise Estática de Código



Continuous Integration

Série: Criando e mantendo um software seguro

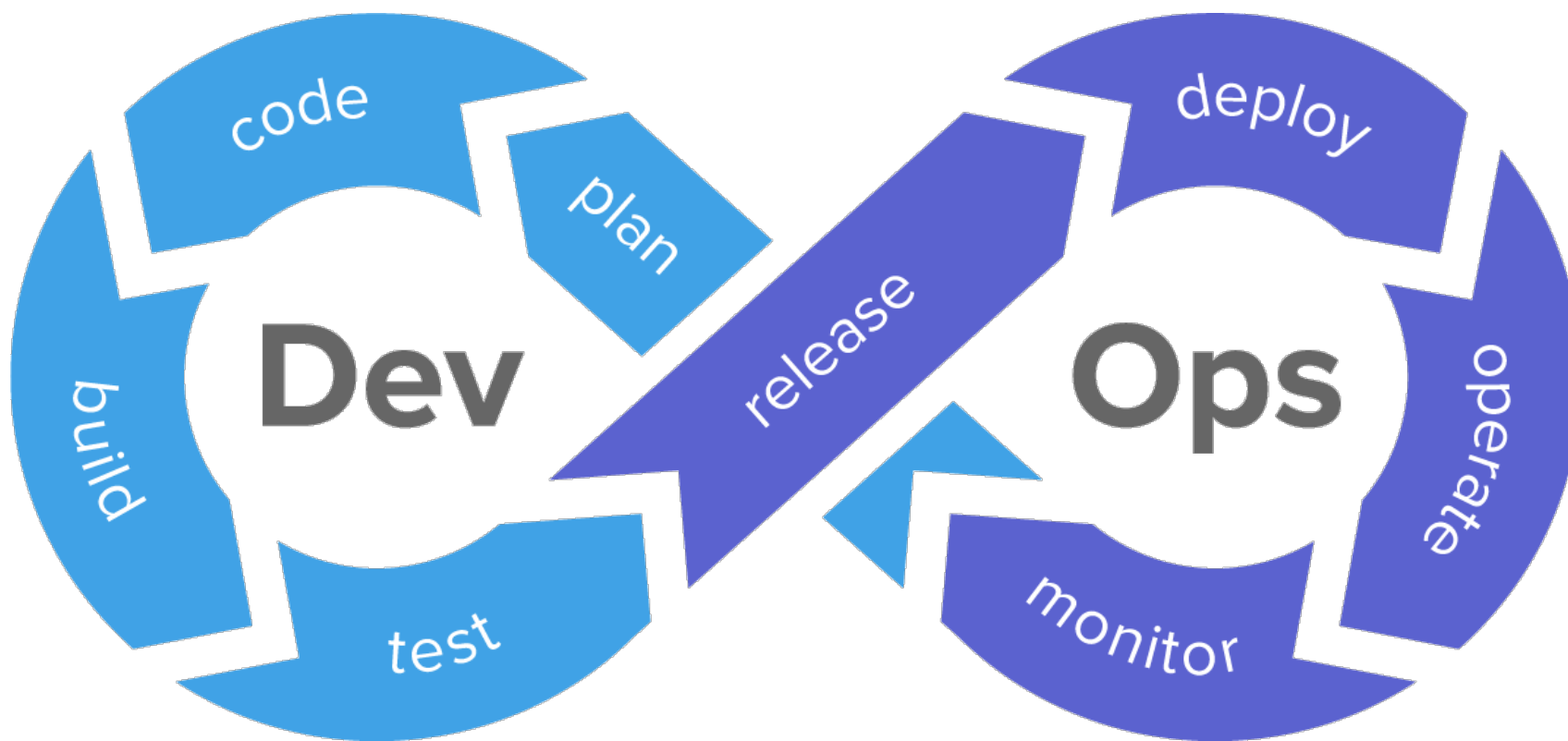
Oficina – Práticas de Análise Estática de Código



DevOps & DevSecOps

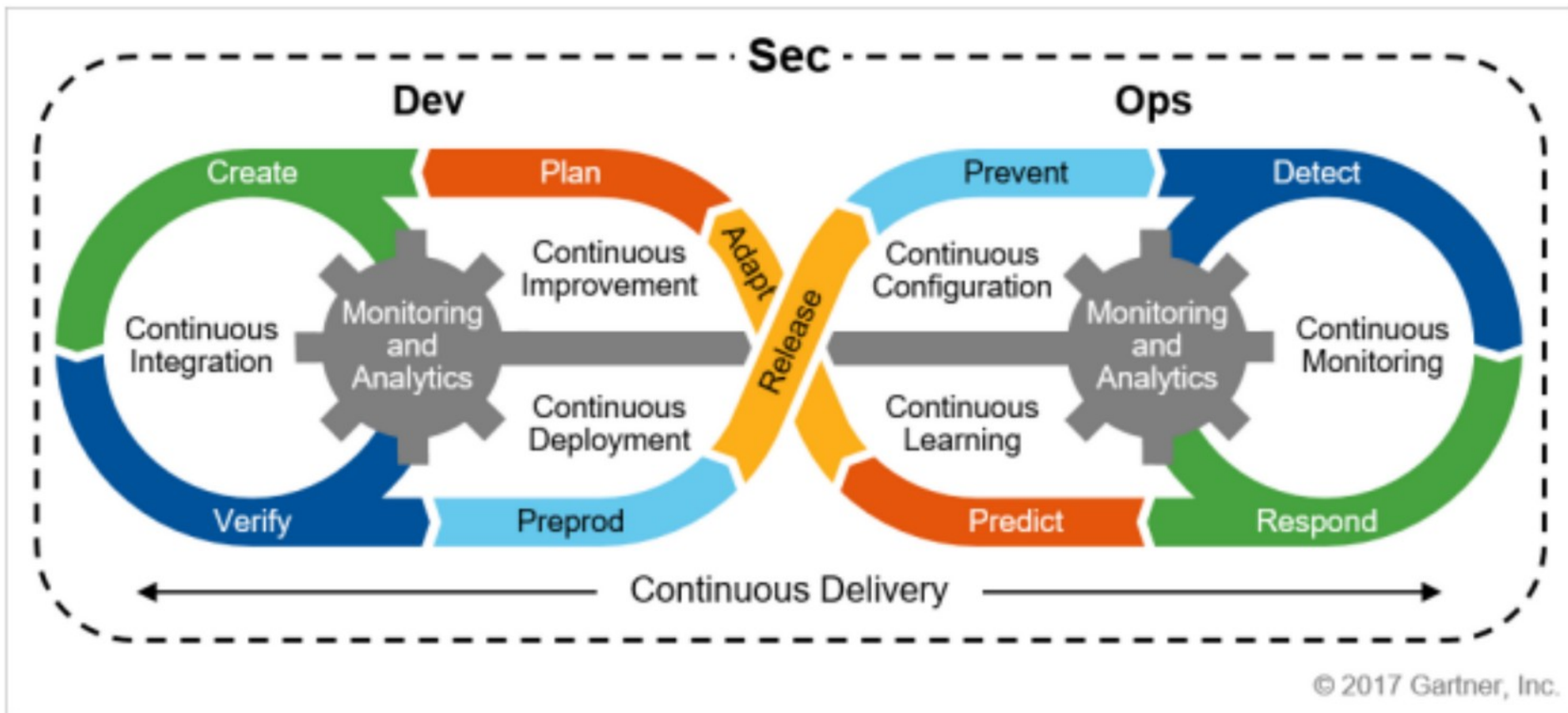
Série: Criando e mantendo um software seguro

Oficina – Práticas de Análise Estática de Código



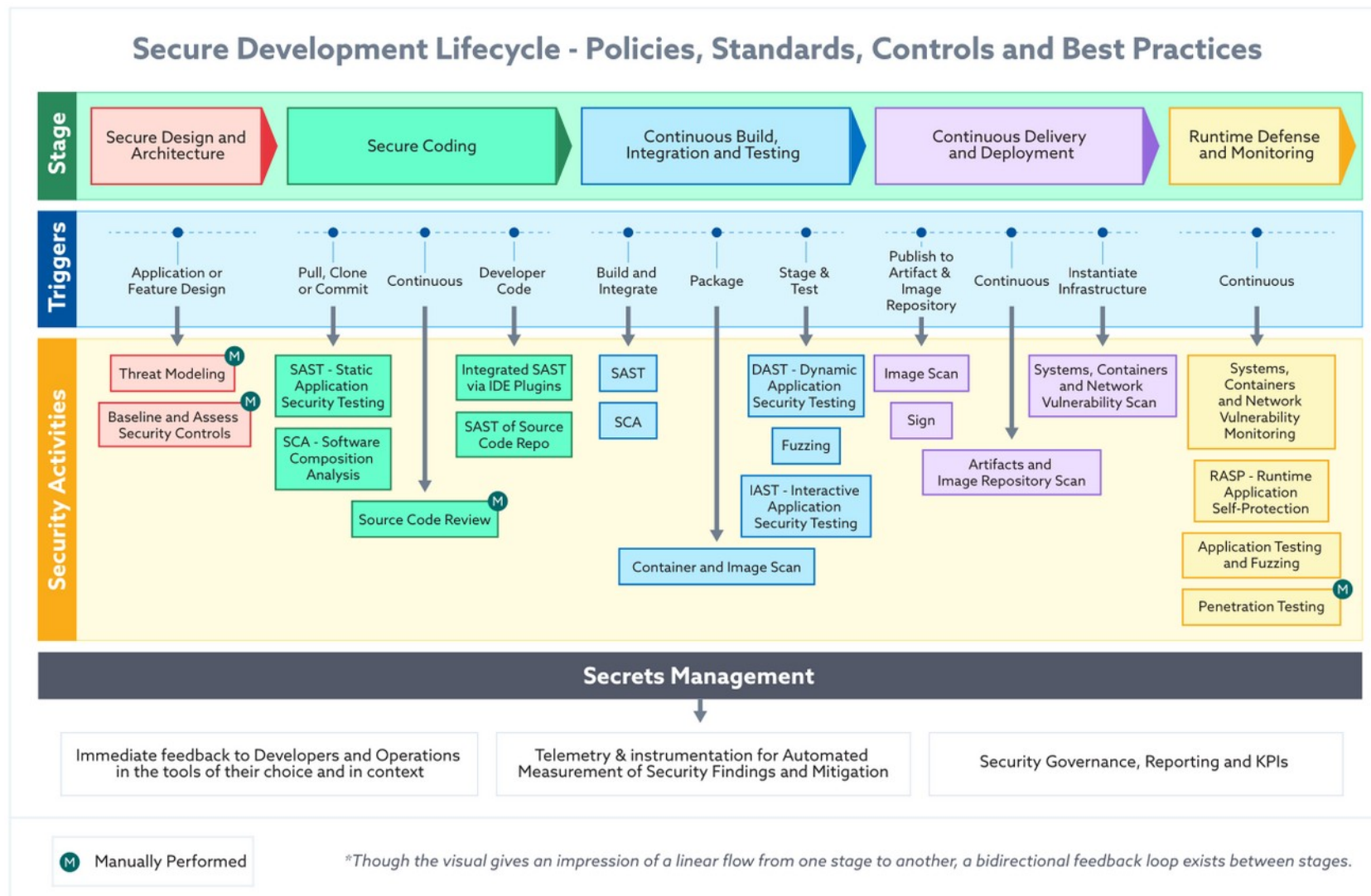
Série: Criando e mantendo um software seguro

Oficina – Práticas de Análise Estática de Código



Série: Criando e mantendo um software seguro

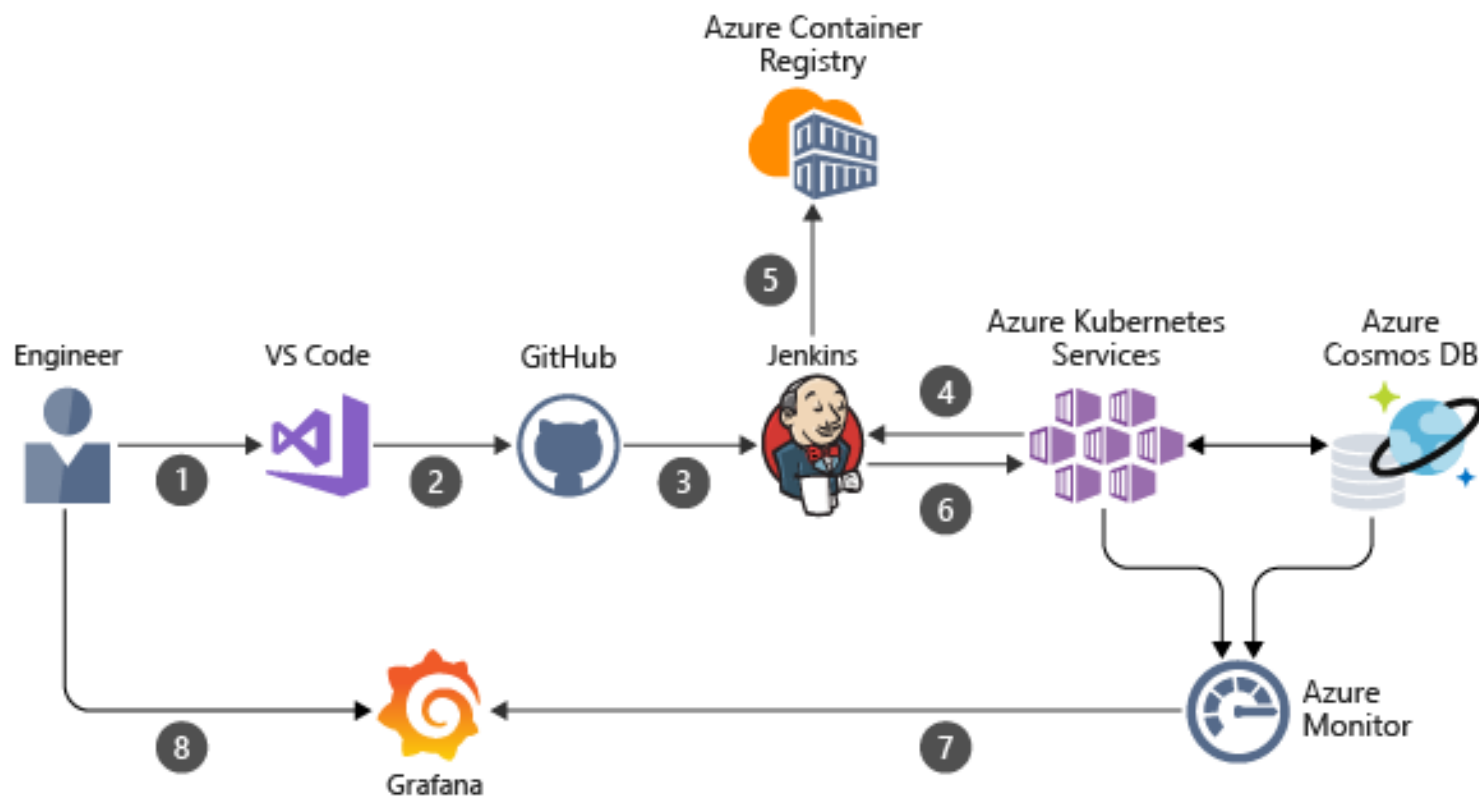
Oficina – Práticas de Análise Estática de Código



Pipelines

Série: Criando e mantendo um software seguro

Oficina – Práticas de Análise Estática de Código



Oficina

Ferramentas



GitHub

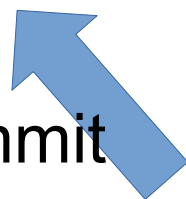


SonarCloud

Ferramentas

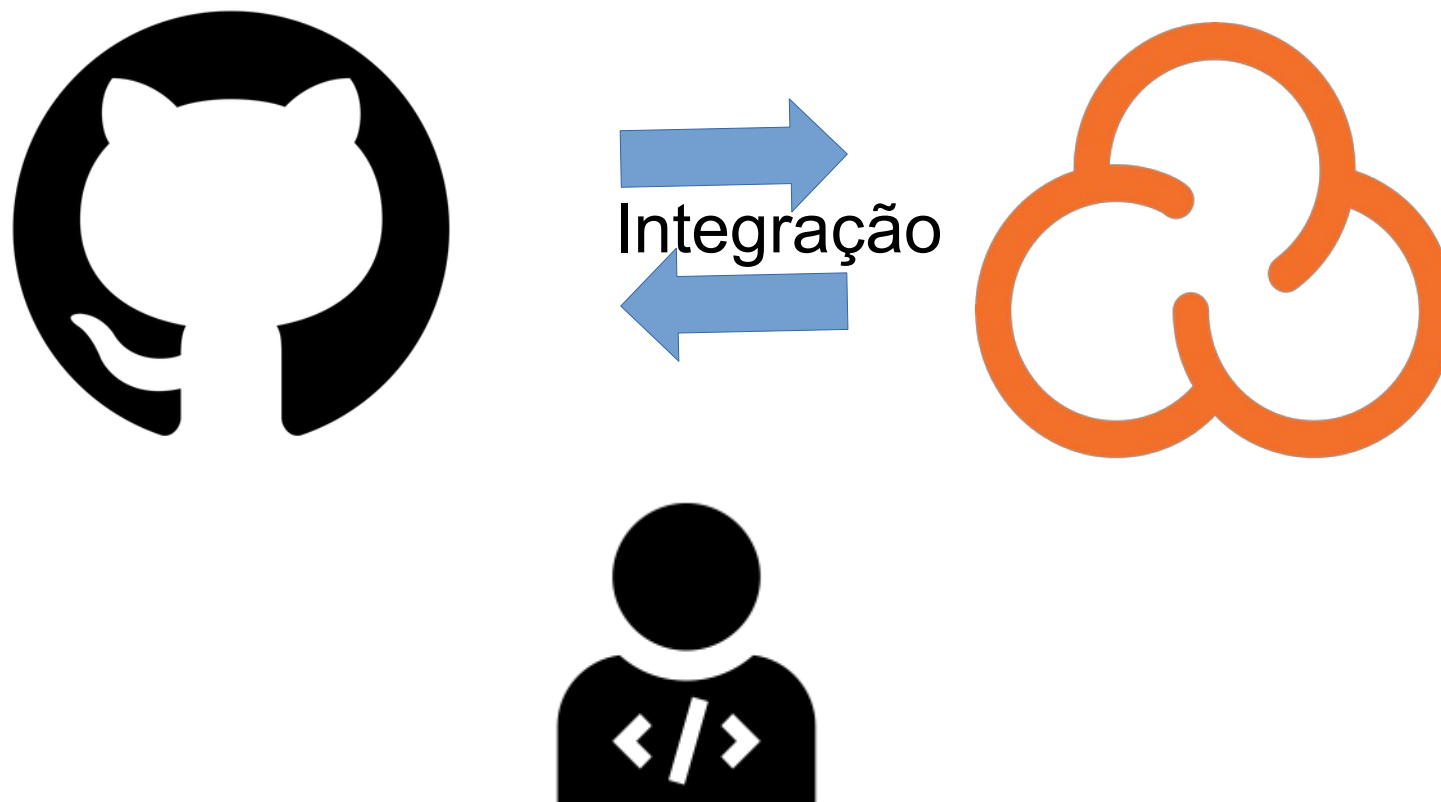


Commit



Dev

Ferramentas



Ferramentas



Dev



Resultados

SAST

Static Application Security Testing

SAST

Static Application Security Testing

código fonte | estática | repositório | IDE

SAST

Static Application Security Testing

código fonte | estática | repositório | IDE
Não testa binários e pacotes. Só fonte!

Guia da Oficina

- **Guia PDF**
 - Arquivo único
 - +/- 60 minutos de leitura total
 - Self-paced
- **Medium**
 - 13 artigos
 - +/- 4 minutos de leitura cada por artigo
 - 61 minutos totais
 - Self-paced
- Não precisa acompanhar a apresentação!

Guia da Oficina

- **Eu não quero guia, só me dê os comandos!**
 - Apresentação ppt

Boa parte do trabalho acontece em SaaS:

- Só comandos de git
- Básico do básico Python/Flask
- Lógica de integração > comandos de console

Obtendo o código

```
$ mkdir lab_fausto
```

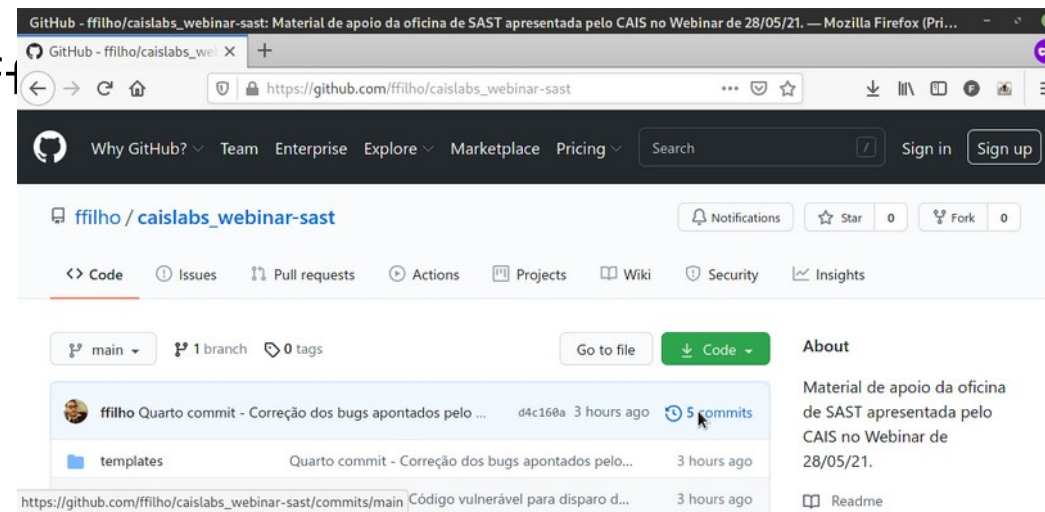
```
$ cd lab_fausto/
```

```
$ git init
```

```
$ git remote add origin https://github.com/ffilho/caislabs_webinar-sast.git
```

```
$ git fetch origin b0546fffb70f500042e9f7439299f1
```

```
$ git reset --hard FETCH_HEAD
```



Executando a aplicação

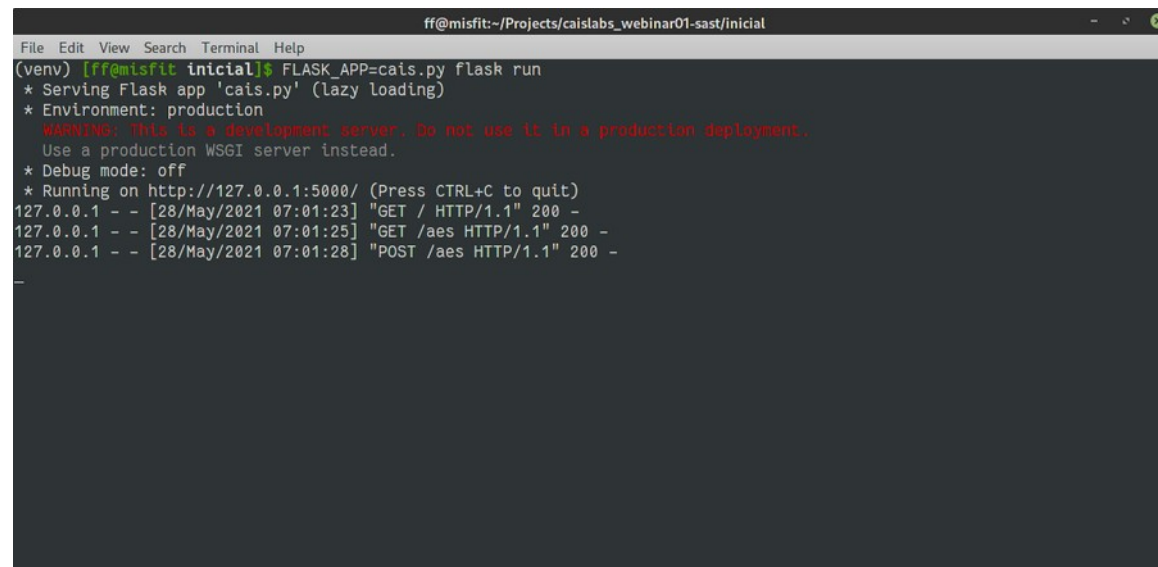
```
$ virtualenv venv
```

```
$ source venv/bin/activate
```

```
(venv) $ pip install -r requirements.txt
```

```
(venv) $ FLASK_APP=cais.py flask run
```

Acesse <http://127.0.0.1:5000/> no seu browser

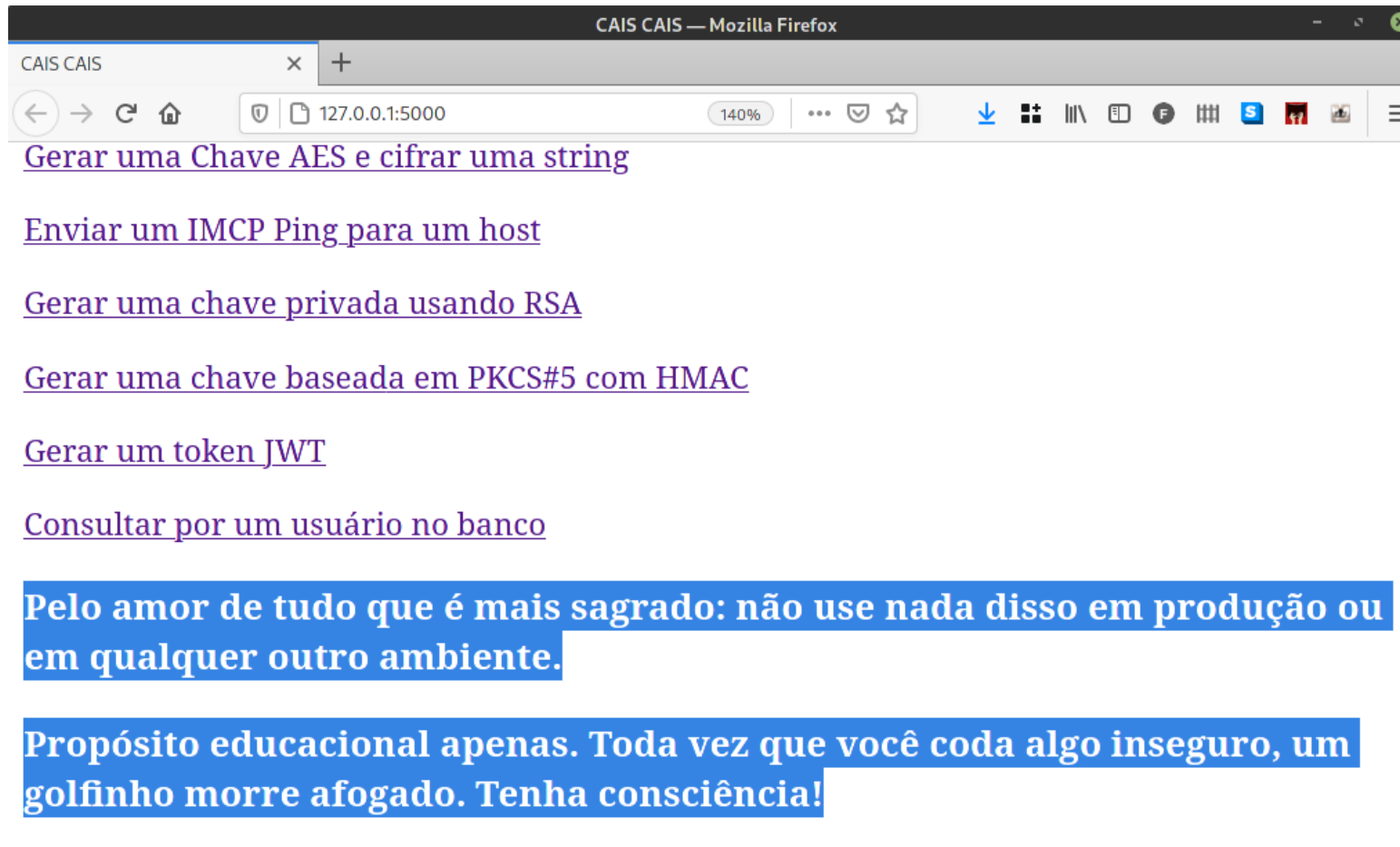


```
ff@misfit:~/Projects/caislabs_webinar01-sast/initial
File Edit View Search Terminal Help
(venv) [ff@misfit: inicial]$ FLASK_APP=cais.py flask run
* Serving Flask app 'cais.py' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [28/May/2021 07:01:23] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/May/2021 07:01:25] "GET /aes HTTP/1.1" 200 -
127.0.0.1 - - [28/May/2021 07:01:28] "POST /aes HTTP/1.1" 200 -
```

Voilà



Atenção!



CAIS CAIS — Mozilla Firefox

CAIS CAIS

127.0.0.1:5000

- [Gerar uma Chave AES e cifrar uma string](#)
- [Enviar um IMCP Ping para um host](#)
- [Gerar uma chave privada usando RSA](#)
- [Gerar uma chave baseada em PKCS#5 com HMAC](#)
- [Gerar um token JWT](#)
- [Consultar por um usuário no banco](#)

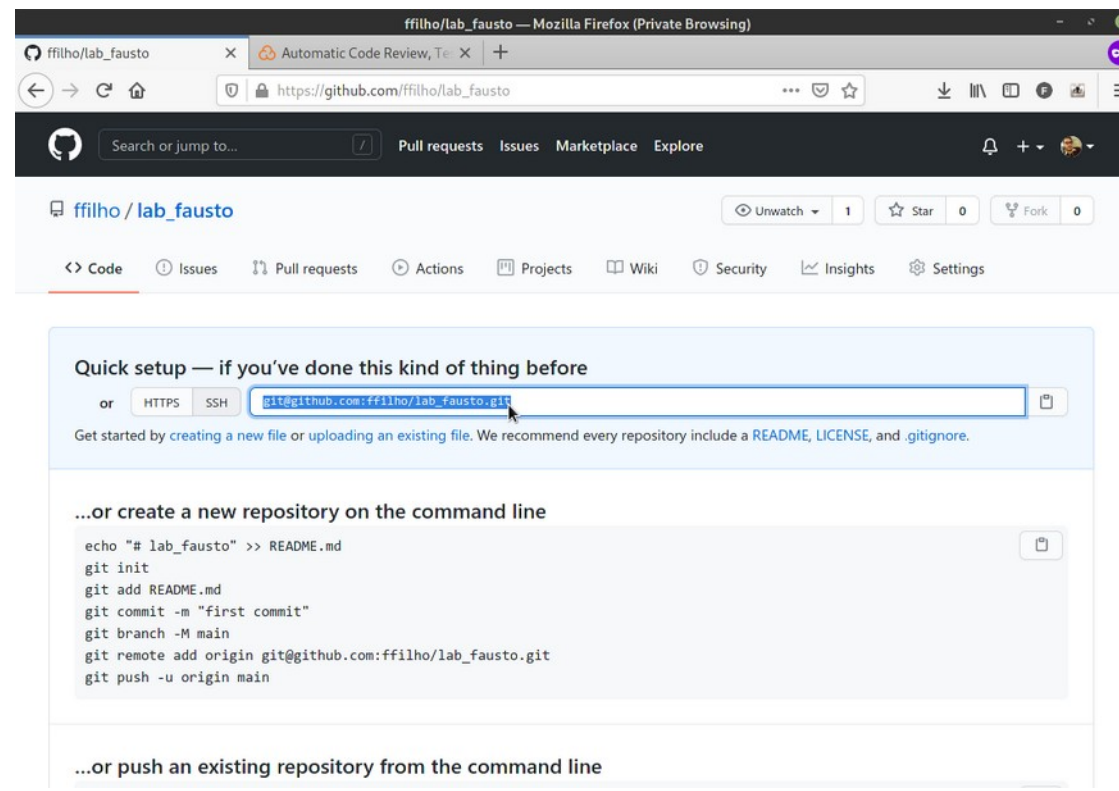
Pelo amor de tudo que é mais sagrado: não use nada disso em produção ou em qualquer outro ambiente.

Propósito educacional apenas. Toda vez que você coda algo inseguro, um golfinho morre afogado. Tenha consciência!

Configurando o remoto

```
$ git remote set-url origin git@github.com:ffilho/lab_fausto.git
```

```
$ git remote -v
```



Integração Github x SonarCloud

\$ sem comandos pra você, hackerman. sry.

\$ sim, você faz tudo isso pela interface gráfica.

\$ na dúvida, olhe o guia.

Primeiro commit!

se você tiver feito algo na mão

```
$ git branch -M main
```

```
$ git add .
```

```
$ git commit -m "Primeiro commit: lab inseguro"
```

se você tiver a cópia do commit do repo é só fazer o push

```
$ git push -u origin main
```

Segundo commit: vulnerabilidades

```
$ git fetch git@github.com:ffilho/caislabs_webinar-sast.git  
afa43ee64ae7c1a308fa9e28c11c235e263975e2
```

```
$ git reset --hard FETCH_HEAD
```

```
$ git push -u origin main
```

Terceiro commit: Security Hotspots

```
$ git fetch git@github.com:ffilho/caislabs_webinar-sast.git  
7aad9a125fb305de4f729208c5fbfdb19fdb051d
```

```
$ git reset --hard FETCH_HEAD
```

```
$ git push -u origin main
```

Quarto commit: Code Smells

```
$ git fetch git@github.com:ffilho/caislabs\_webinar-sast.git
```

```
$ git reset --hard FETCH_HEAD
```

```
$ git push -u origin main
```

Quinto commit: Code Bugs

```
$ git fetch git@github.com:ffilho/caislabs_webinar-sast.git  
26e4b92db4367fa287ce90f95e113d215b678adb
```

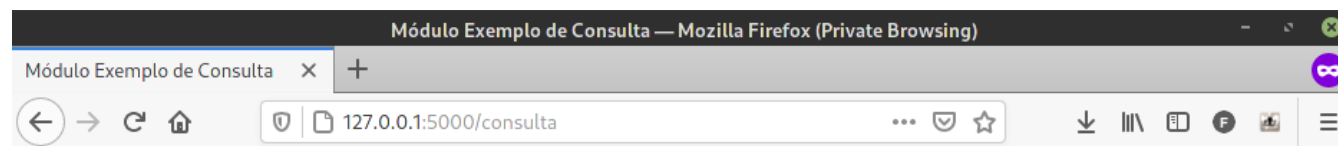
```
$ git reset --hard FETCH_HEAD
```

```
$ git push -u origin main
```


Zerei o mundo! Estou seguro?



Zerei o mundo! Estou seguro?



Módulo Exemplo de Consulta

**[('Fausto', '11999119911'), ('Landim', '11888228822'),
('Nicole', '11777339933')]**

[Voltar ao módulo de Exemplo de Consulta](#)

[Voltar para home](#)

Contatos

- **Fausto Filho**
 - fausto.filho@rnp.br
 - <https://www.linkedin.com/in/faustoafilho/>
 - <https://fausto-filho.medium.com/>

Links

- **Download do guia em PDF:**
 -
- **Artigos do Medium – disponível em breve:**
 - <https://fausto-filho.medium.com/>
- **Repositório do Github:**
 - https://github.com/ffilho/caislabs_webinar-sast

Obrigado!



MINISTÉRIO DO
TURISMO

MINISTÉRIO DA
DEFESA

MINISTÉRIO DA
SAÚDE

MINISTÉRIO DA
EDUCAÇÃO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÕES

