



Treinando desenvolvedores a  
programarem de forma segura

Rafael dos Santos rafael.santos@corp.globo.com

# \$ whoami

Rafael dos Santos  @rafasantos5  
[github.com/rafaveira3](https://github.com/rafaveira3)

 Ciência da Computação - UFRJ

 Apaixonado por futebol

 OSCP + OSCE

 Analista de Segurança @ globo.com

 Sec Tools + Desenvolvimento de Exploit



Um dia na vida de um time de  
desenvolvimento...

# Um dia na vida de um time de desenvolvimento...



## Paralamas ensaiam com música de Gil

- Paolla, Grazi e mais famosos

## 'A Dona do Pedaço': Vivi fica sem comida

- Cássia arma vingança



Nova Feat!



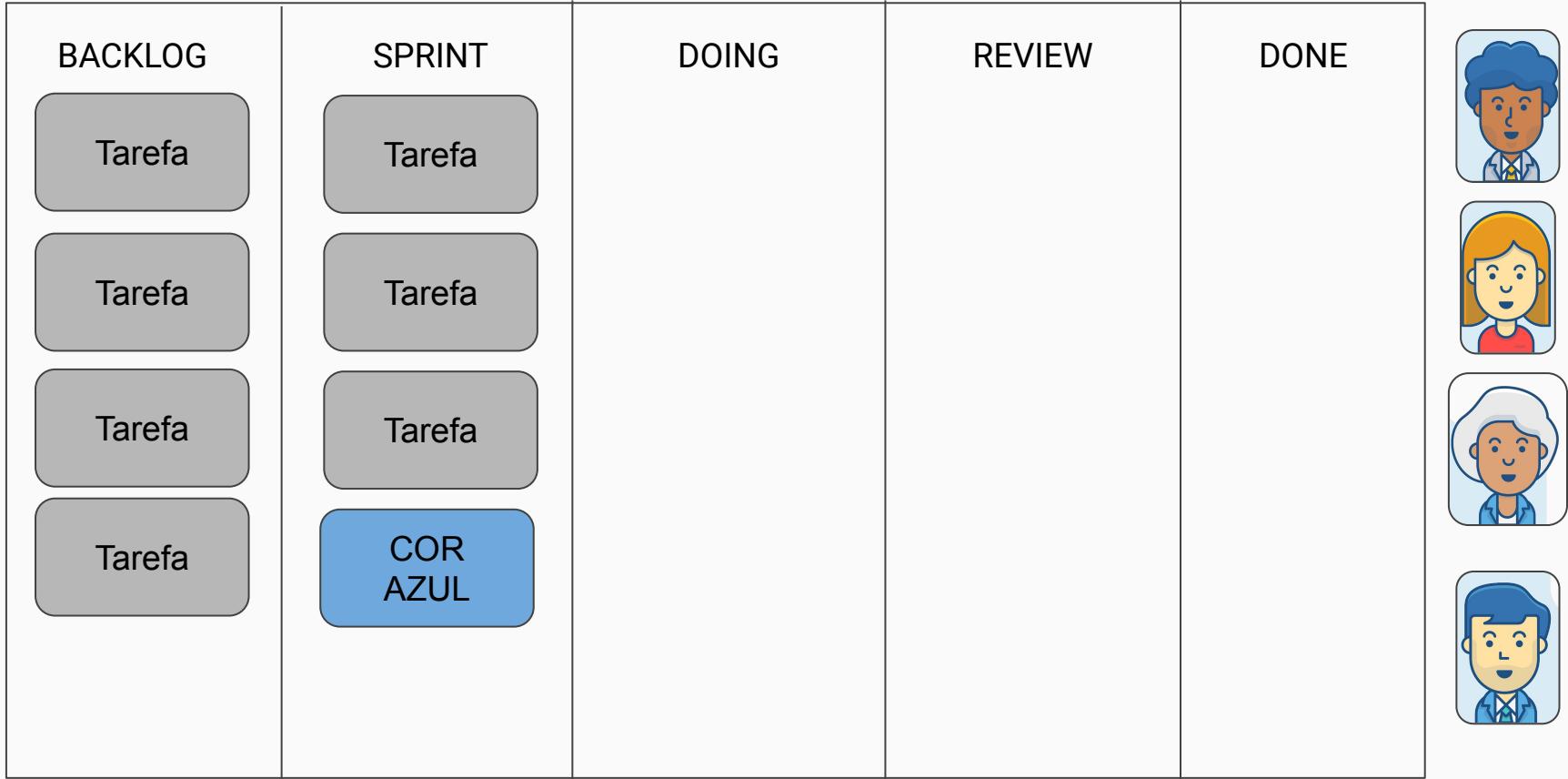
## Paralamas ensaiam com música de Gil

- Paolla, Grazi e mais famosos

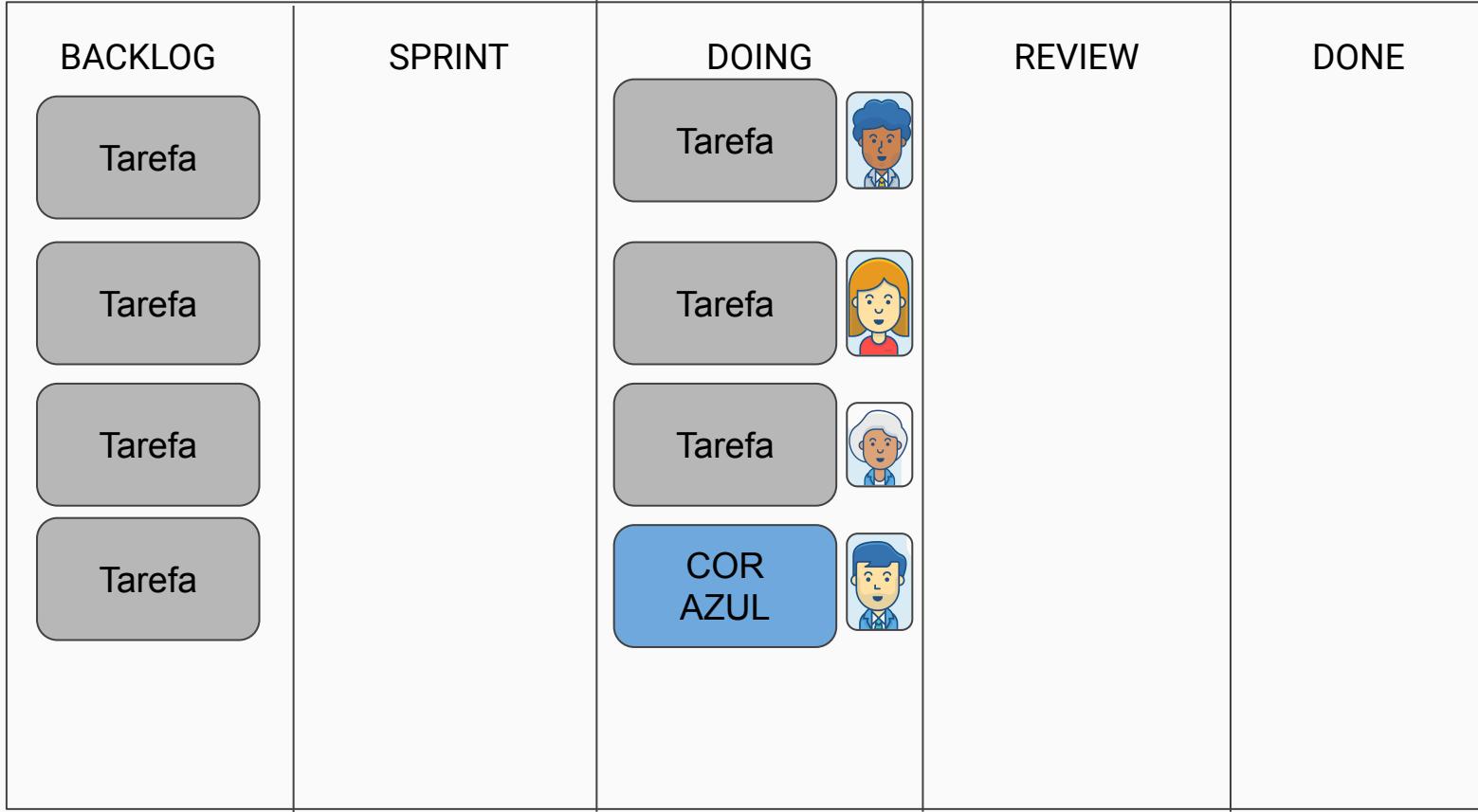
## 'A Dona do Pedaço': Vivi fica sem comida

- Cássia arma vingança

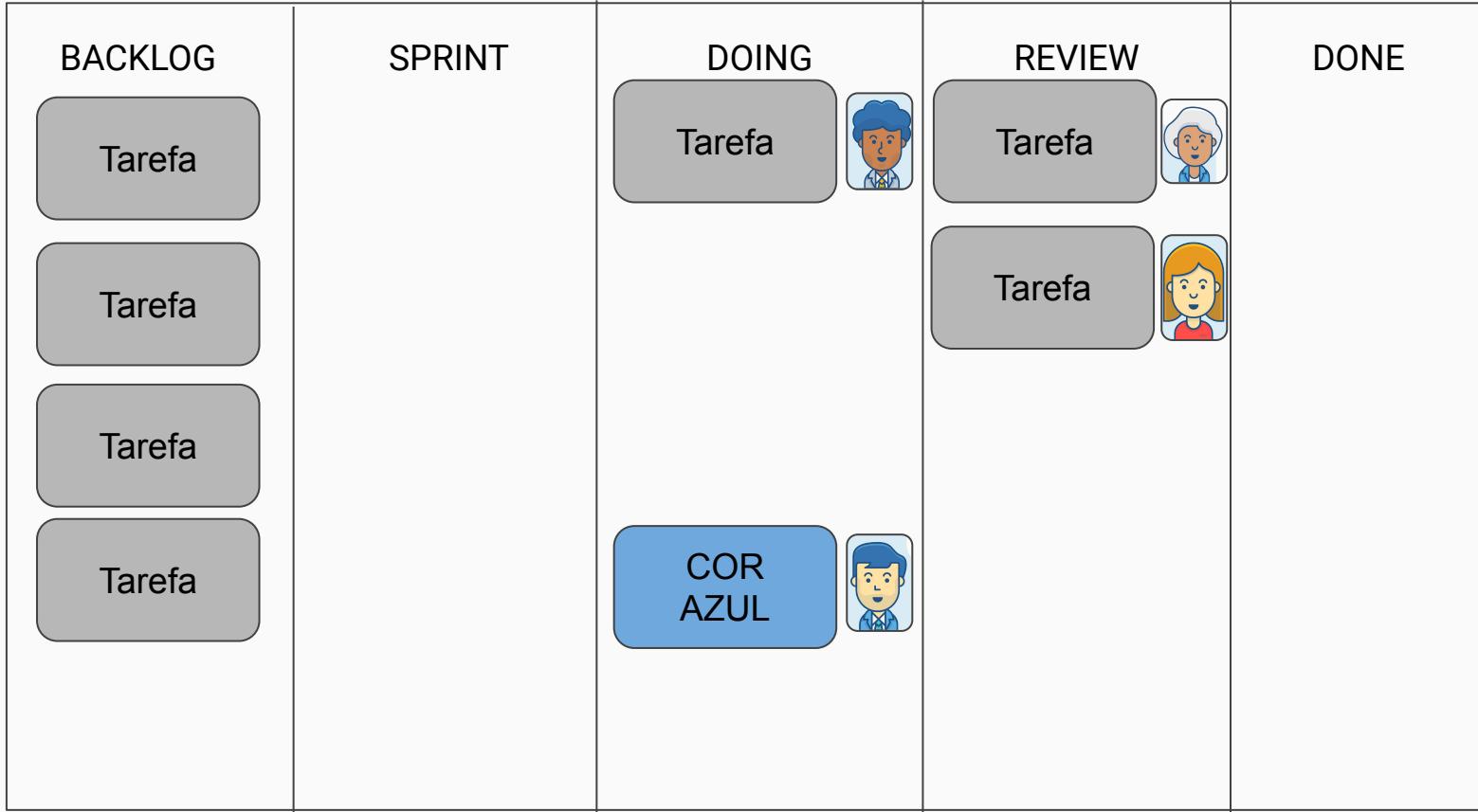
# Um dia na vida de um time de desenvolvimento...



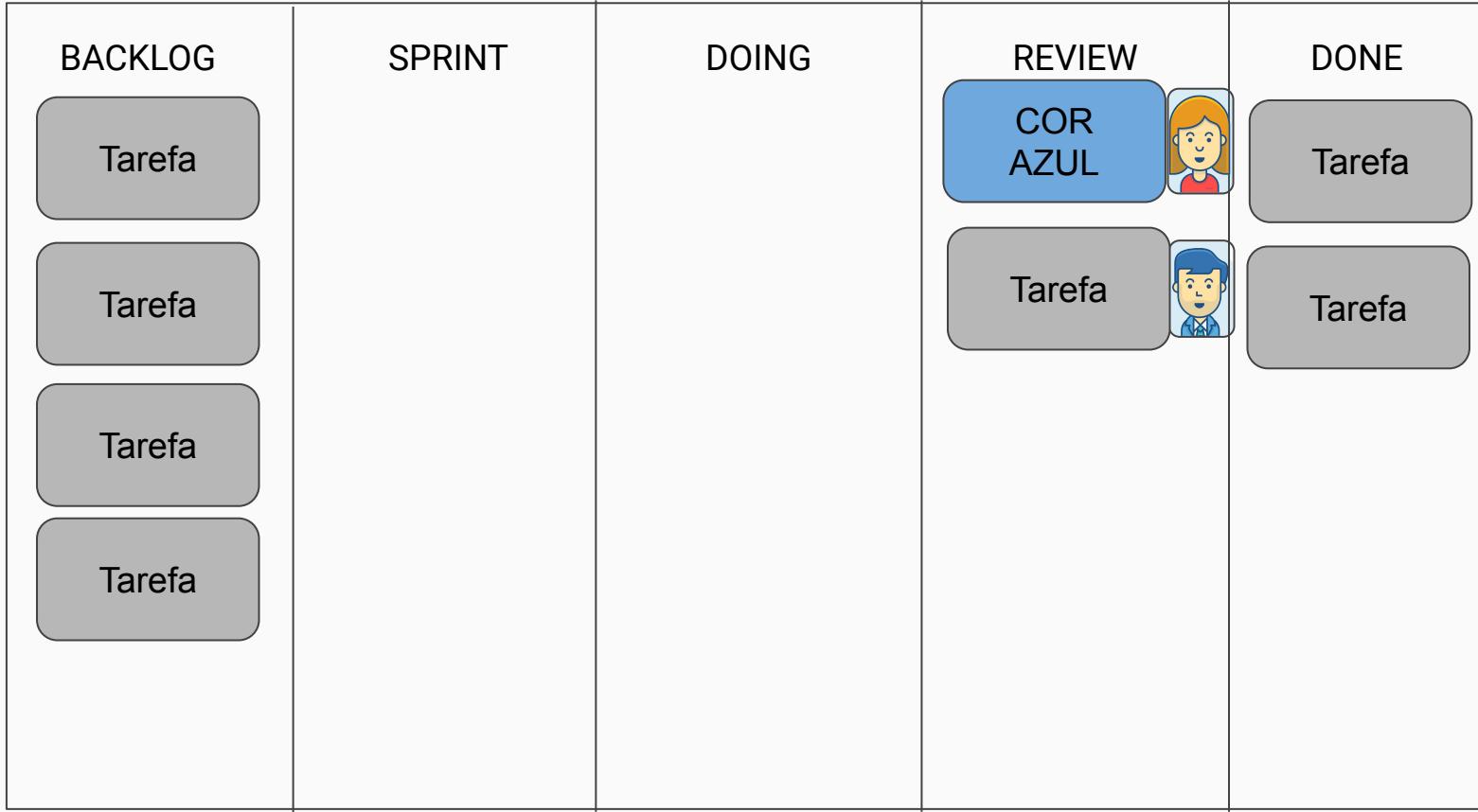
# Um dia na vida de um time de desenvolvimento...



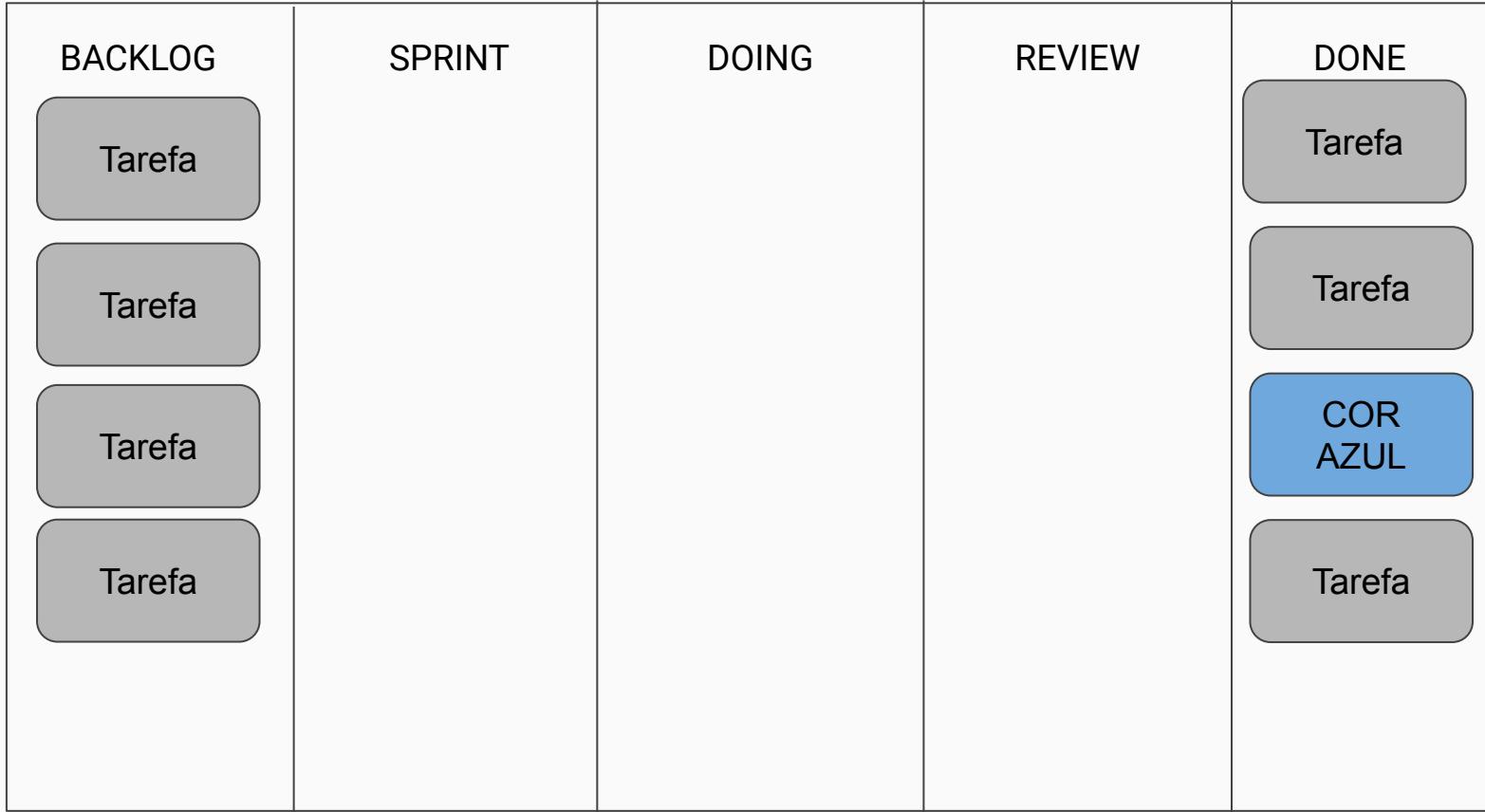
# Um dia na vida de um time de desenvolvimento...



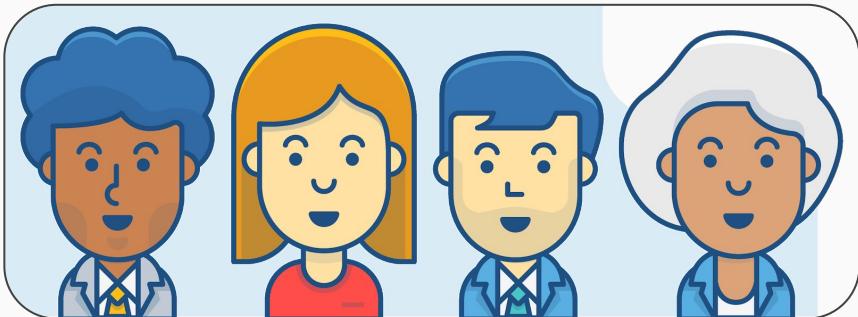
# Um dia na vida de um time de desenvolvimento...



# Um dia na vida de um time de desenvolvimento...



# Um dia na vida de um time de desenvolvimento...



**tsuru** ✓

The word "tsuru" is written in a large, dark blue, lowercase, sans-serif font. A green checkmark symbol is positioned to the right of the letter "u".A horizontal arrangement of news snippets, each consisting of a small thumbnail image and a brief headline. The snippets are organized into three columns separated by thin vertical lines.

Motorista no bate em carro parado e capota veículo no Ceará	CR7 chama Messi para jogar na Itália: 'Como eu, aceite o desafio'	Ex-paquitas se reúnem para amigo secreto na casa de Xuxa
Executiva da Huawei presa pede liberação por motivos de saúde	Após anunciar Carille, Timão tenta acelerar montagem de elenco	Sasha posa decotada e capricha no 'carão' para encantar fãs
Ghosn é acusado formalmente no Japão por violação financeira	Cobiçado por clubes do Brasil, Sassá seguirá no Cruzeiro, diz diretor	Gavassi mostra Bruna 'desesperada' em show de Sandy: vídeo
Operação mira suposto esquema de médicos e empresários para furar fila do SUS		
Santos não tem avanços após 'não' de Abel e inicia semana decisiva por novo técnico		
Alok passa mal durante show no Festival de Verão e relata suspeita de zika		



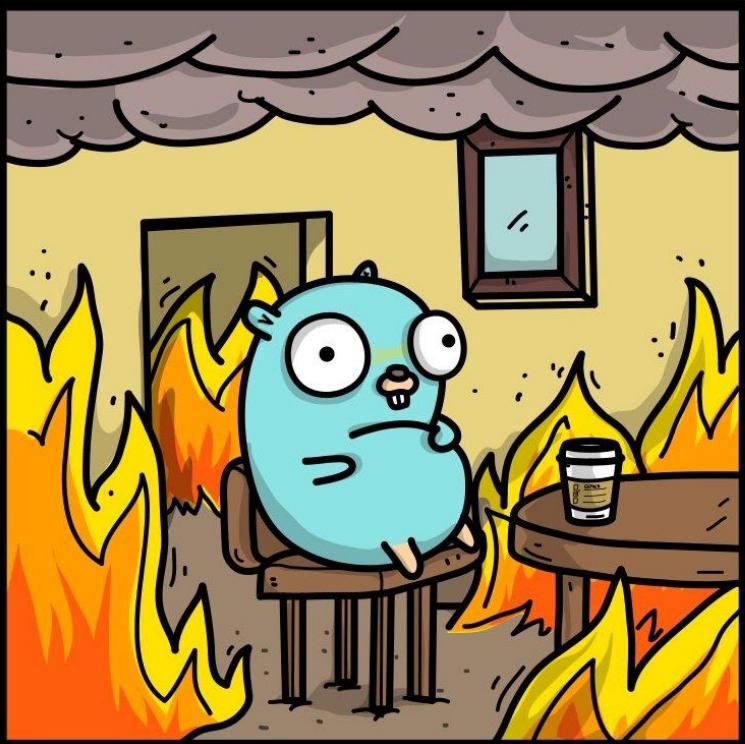
Vamos fazer um treinamento de  
segurança?

# Treinamento de segurança por uma semana

BACKLOG	SPRINT	DOING	REVIEW	DONE
<div>Tarefa</div>		<div> </div>		
<div>Tarefa</div>		<div> </div>		
<div>Tarefa</div>		<div> </div>		
<div>Tarefa</div>		<div> </div>		

# Treinamento de segurança por uma semana

BACKLOG	SPRINT	DOING	REVIEW	DONE	
<div><p>Tarefa</p></div>				<div></div>	
<div><p>Tarefa</p></div>				<div></div>	
<div><p>Tarefa</p></div>				<div></div>	
<div><p>Tarefa</p></div>				<div></div>	



# Treinamento de segurança por uma semana

## 2 - XSS - Refletido



### Reflected XSS



1. Social engineering,  
e.g. a dodgy URL



2. Request includes malicious data



3. Bad app uses  
malicious data  
verbatim

5. Bad thing  
happens



```
http://www.example.com/search.asp?q=<script>a  
lert(1);</script>
```



# Treinamento de segurança por uma semana

## 3 - XSS - Persistente

globo.com



### Stored XSS



5. Bad thing happens



1. Attacker gets malicious data into the database (no social engineering required)

2. Entirely innocent request

4. Response includes malicious data as active content

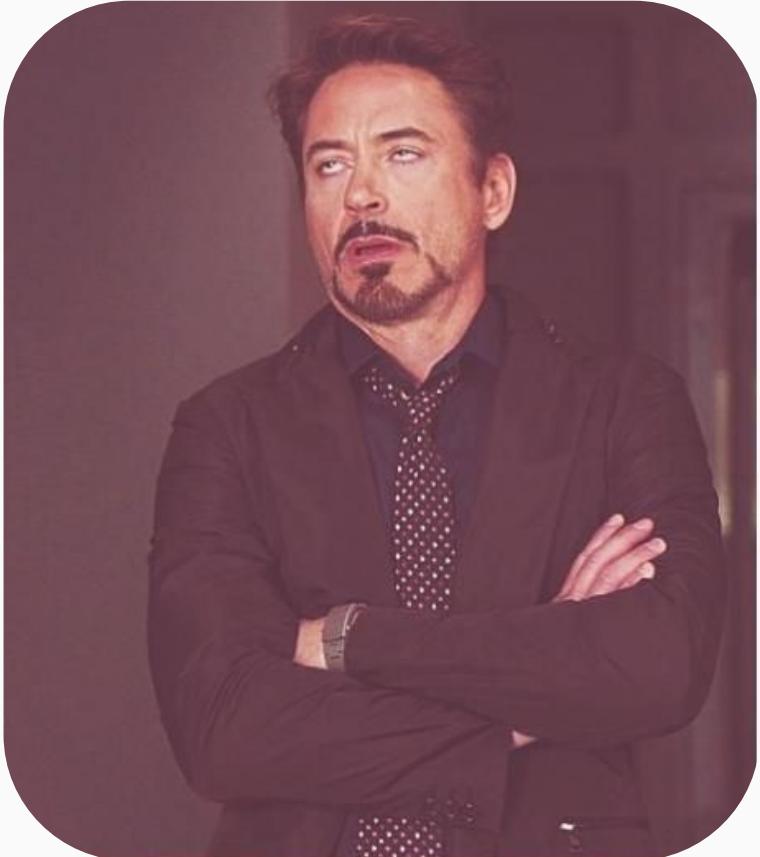


3. Bad app retrieves malicious data and uses it verbatim

<http://www.example.com/profile.asp>



# Treinamento de segurança por uma semana



Let's hack! A white circular icon of a panda's head with black ears, eyes, and a small pink tongue sticking out.

# Let's hack!



# Planejamento: DEVs gostam de... Hands on!



A screenshot of a terminal window titled "index.js". The code is a Python script using the MongoClient module to connect to a MongoDB database named "stego". It adds a default "admin" user to the database and logs a message when successful. It also defines a GET route for "/login" which renders a "login.html" template, and a POST route for "/login" which handles user login credentials. The code uses req.body to access the user input.

```
40
41
42
43
44
45
46
47     // Add "admin" default user to the database
48     MongoClient.connect(url, function(err, db) {
49         if (err) throw err;
50         var dbo = db.db("stego");
51         var myobj = { username: "admin", password: "admin" };
52         dbo.collection("users").insertOne(myobj, function(err, res) {
53             if (err) throw err;
54             console.log("Admin user added to the database");
55             db.close();
56         });
57     });
58
59     // User login route, get webpage
60     router.get("/login", function(req,res) {
61         res.render("login.html");
62     })
63
64     // User login route, submit POST request to server
65     router.post("/login", function(req,res) {
66         var username = req.body.user.name;
67         var password = req.body.user.password;
68
69         // Verifies user credentials
70         function VerifiesUser(callback) {
71             MongoClient.connect(url, function(err, db) {
```

master\* 0▲0 Python 2.7.15 You, 21 days ago Ln 74, Col 52 Spaces: 4 UTF-8 LF JavaScript

# Planejamento: Atacar? Não! Desenvolver



```
database: webpage
Table: users
[3 entries]
+-----+
| id | username | password
+-----+
| 3  | Rafael   | $2a$14$Lt7QRVZfz9XeN2KIVyzLsux0BKQ3uciwd440DS0hfRvVKUyEKMMtW |
| 4  | Bianca    | $2a$14$LkJimyfVsD1WRgqCeX4eeAu0v4fHUNAHN6yj.jA.oLUJ32TZs1cS |
| 5  | C4rl0z    | $2a$14$ObEBiURggHjBDVUUMP6UK.Cnm.t9xTf.Mz9c5Iy4z0uDi7CKSipU6 |
+-----+
```



```
query := fmt.Sprintf("select * from Users where username = '" + user + "'")
rows, err := dbConn.Query(query)
stmt, stmtErr := dbConn.Prepare("SELECT * FROM Users WHERE username = ?")
if stmtErr != nil {
    return false, stmtErr
}
```

# Planejamento: O que eles precisam aprender?



## A1:2017- Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

## A2:2017-Broken Authentication

Application functions related to authentication are implemented incorrectly, allowing attackers to compromise authentication in a variety of ways, such as by session hijacking, password cracking, or privilege escalation. Attackers may use social engineering, such as phishing, to trick users into revealing their credentials. They may also exploit other implementation flaws to assume the identity of legitimate users.

## A3:2017- Sensitive Data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial information, healthcare, and PII. Attackers may steal or alter this information to commit card fraud, identity theft, or other crimes. Even more serious than data theft is data protection, such as encryption at rest and in transit, which is often not properly implemented or exchanged with the browser.

## A4:2017-XML External Entities (XXE)

Many older or poorly configured XML processors accept external entity references within XML documents. External entities can be used to include files from a local file system, internal port scanners, or even remote servers.

## A5:2017-Broken Access Control

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to gain access to other users' accounts, view sensitive information, or change settings.

## A6:2017-Security Misconfiguration

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.

## A7:2017- Cross-Site Scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

## A8:2017- Insecure Deserialization

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

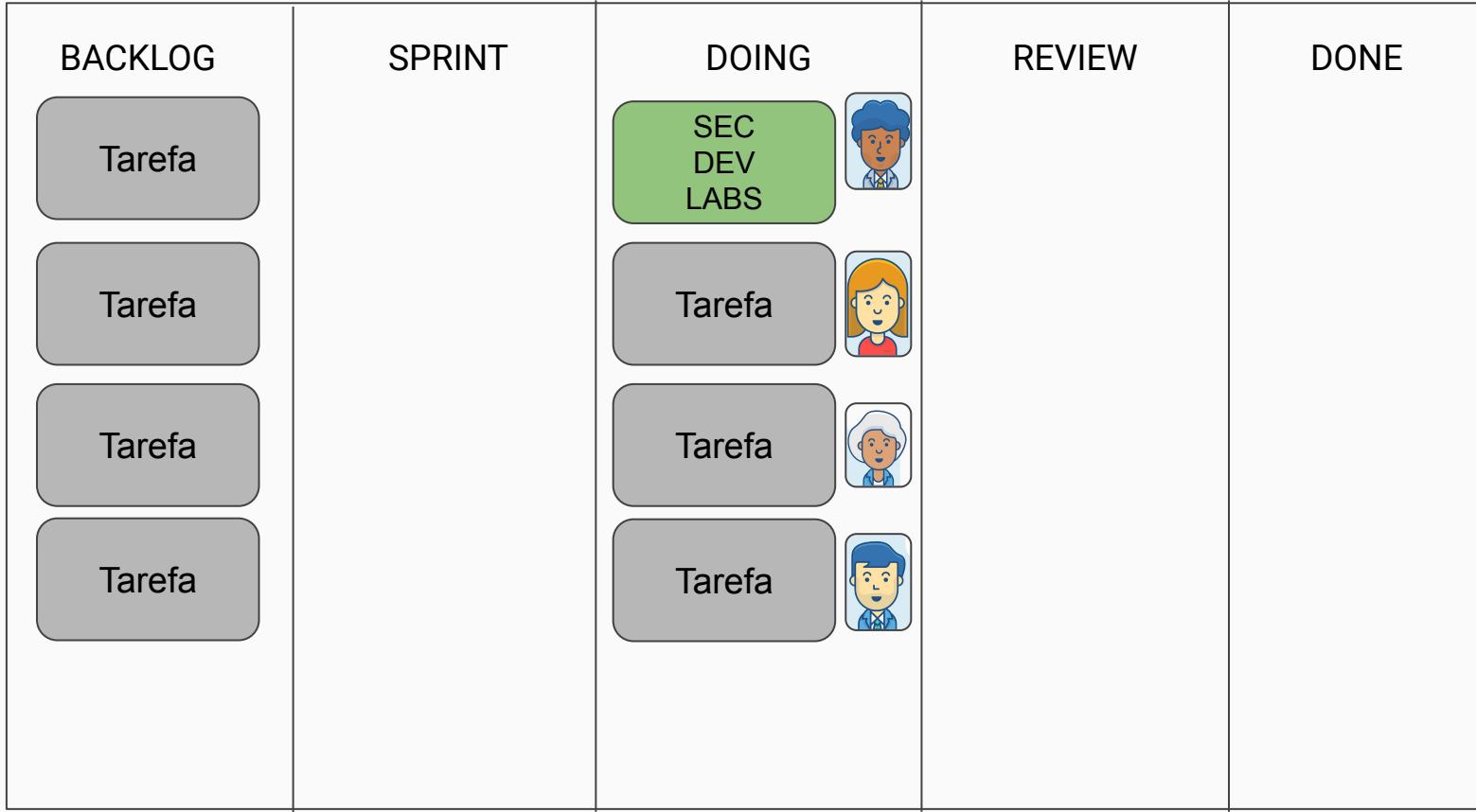
## A9:2017-Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

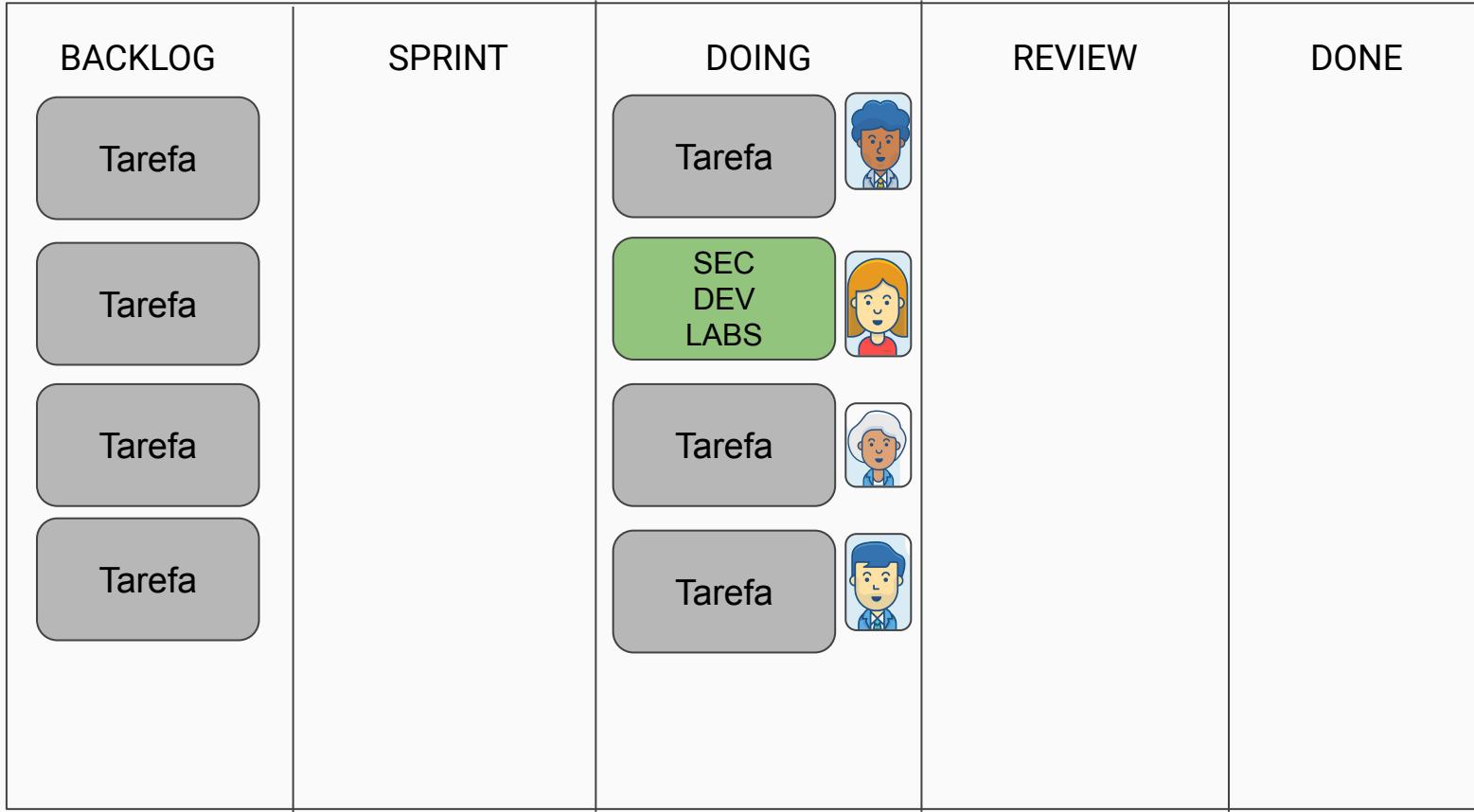
## A10:2017- Insufficient Logging & Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

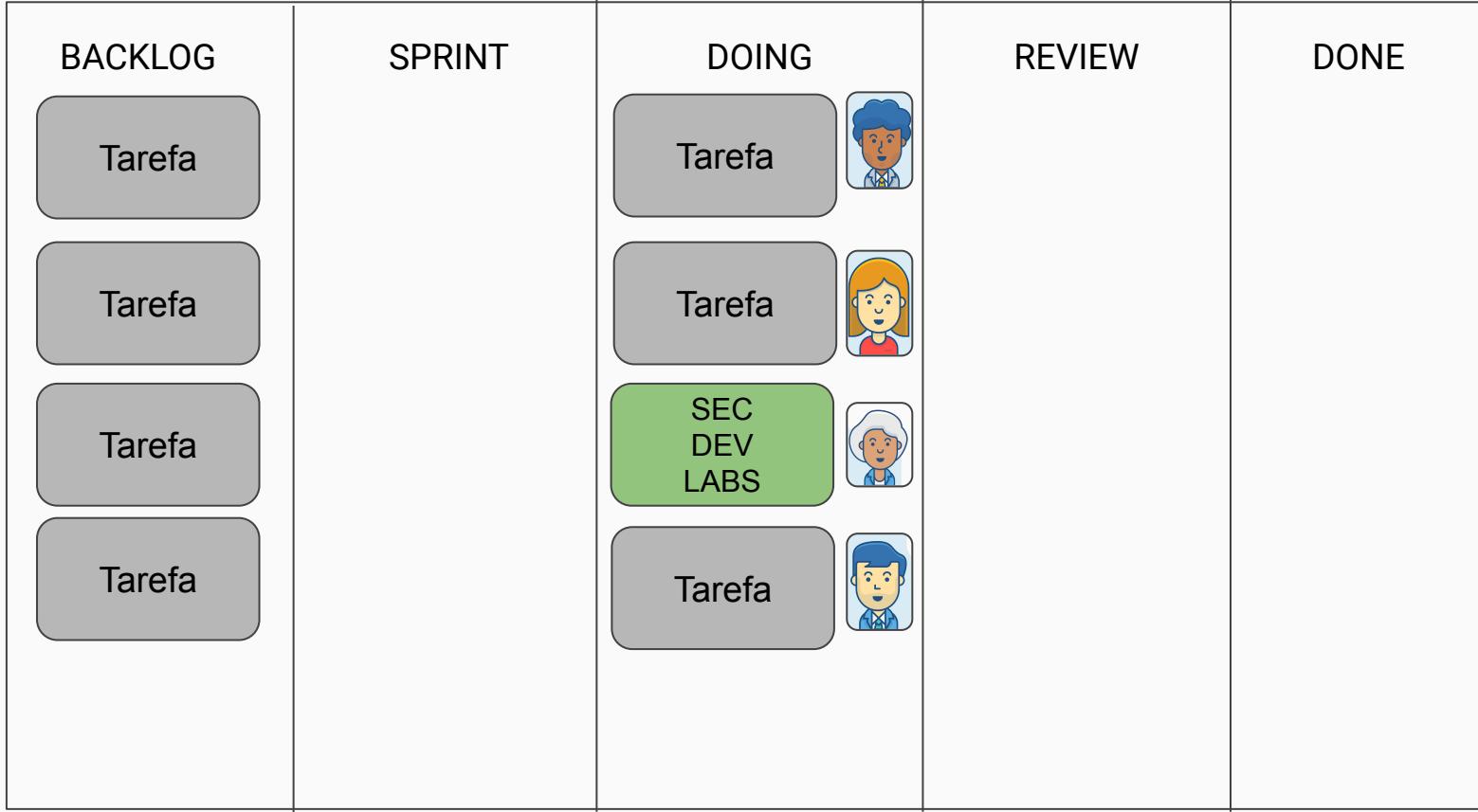
# Planejamento: Não atrapalhar o sprint!



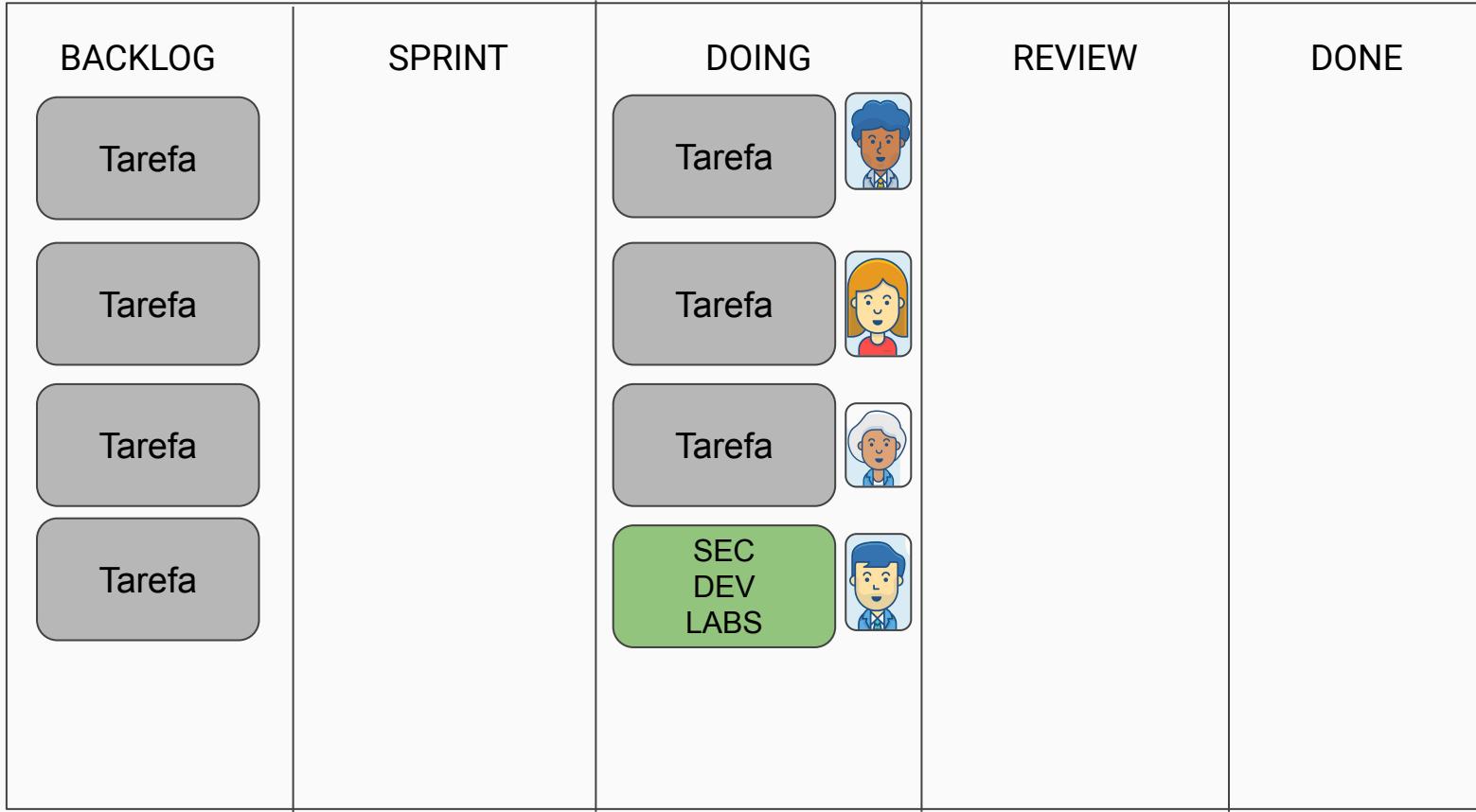
# Planejamento: Não atrapalhar o sprint!



# Planejamento: Não atrapalhar o sprint!



# Planejamento: Não atrapalhar o sprint!





# secDevLabs

# secDevLabs: Apps vulneráveis!

The image consists of two parts. The left side shows a blurred photograph of a soccer player, Isco, wearing a white Real Madrid jersey, pointing upwards with his right index finger. The right side is a screenshot of a web application titled "Gossip World". The header of the app is pink and contains the title "Gossip World" and navigation links "Home", "New gossip", and "Logout". Below the header, a large section is titled "Last gossips" and features a news item about "Chico Buarque buy baguettes for snack". The text of the item reads: "The singer and composer was dressed in shorts, T-shirts and slippers along the streets of Leblon, South Zone of Rio". A blue "Read more →" button is located below the text. To the right of the news item is a search bar containing the code "<script>alert(1);</script>" and a "Go!" button. At the bottom of the app's screenshot, there are navigation links for "Older" and "Newer". The footer of the app is also pink and contains the copyright notice "Copyright © Gossip World 2018".

## Gossip World

Home New gossip Logout

### Last gossips

**Chico Buarque buy baguettes for snack**

The singer and composer was dressed in shorts, T-shirts and slippers along the streets of Leblon, South Zone of Rio

[Read more →](#)

Posted on 2019-01-21 by vitoria

← Older    Newer →

Copyright © Gossip World 2018

# secDevLabs: Apps vulneráveis!

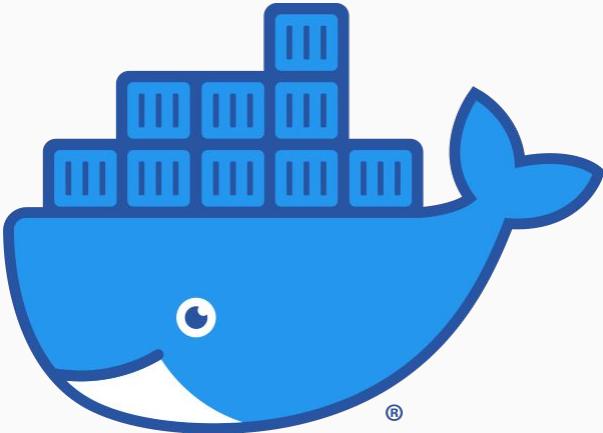
## OWASP Top 10 (2017) apps:

Disclaimer: You are about to install vulnerable apps in your machine! 🔥

Vulnerability	Language	Application
A1 - Injection	Golang	<a href="#">CopyNPaste API</a>
A2 - Broken Authentication	Python	<a href="#">Saidajaula Monster Fit</a>
A2 - Broken Authentication	Golang	<a href="#">Insecure go project</a>
A3 - Sensitive Data Exposure	Golang	<a href="#">SnakePro</a>
A4 - XML External Entities (XXE)	PHP	<a href="#">ViniJr Blog</a>
A5 - Broken Access Control	Golang	<a href="#">Vulnerable Ecommerce API</a>
A6 - Security Misconfiguration	PHP	<a href="#">Vulnerable Wordpress Misconfig</a>
A6 - Security Misconfiguration	NodeJS	<a href="#">Steganography</a>
A7 - Cross-Site Scripting (XSS)	Python	<a href="#">Gossip World</a>
A8 - Insecure Deserialization	Python	<a href="#">Amarelo Designs</a>
A9 - Using Components With Known Vulnerabilities	PHP	<a href="#">Cimentech</a>
A10 - Insufficient Logging & Monitoring	Python	<a href="#">Gameslrados.com</a>

# secDevLabs: Laboratório local

```
1 docker-compose.yml
1 version: '3'
2
3 services:
4
5   api:
6     container_name: a1_api
7     build:
8       context: ../
9       dockerfile: deployments/alinj.Dockerfile
10    ports:
11      - "3000:3000"
12    networks:
13      - a1net
14    command: "go run app/server.go"
15    environment:
16      MYSQL_ROOT_PASSWORD: root
17      MYSQL_USER: user
18      MYSQL_PASSWORD: pass
19      MYSQL_DATABASE: a1db
20    depends_on:
21      - mysqldb
22    external_links:
23      - mysqldb:mysqldb
24    restart: unless-stopped
25
26 mysqldb:
27   container_name: mysqldb
28   image: mysql:5.7
29   ports:
30     - "3307:3307"
31   volumes:
32     - db_data:/var/lib/mysql
33   environment:
34     MYSQL_ROOT_PASSWORD: root
```



← → C ⌂ ⓘ localhost:3000

CopyNPaste

Welcome to CopyNPaste API demo page!

REGISTER

LOGIN

User

# secDevLabs: Narrativa de ataque

As no validation is being used to avoid **ENTITIES** being sent to the PHP file, an attacker could create the following `evilxml.xml` to perform a XXE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE root [
<!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<contact>
<name>&xxe;</name>
<email>RAFAEL@EXAMPLE.com</email>
<subject>YOU ROCK</subject>
<message>I LOVE WATCHING YOUR SKILLS, MAN</message>
</contact>
```

And, as the following picture shows, it is possible to realize that the attack succeeds and sensitive information is retrieved from the server that is hosting the vulnerable app:

```
curl -d @evilxml.xml localhost:10080/contact.php ; echo
```

```
✓ [14:06] rafael.santos@labs:~$ curl -d @evilxml.txt localhost:10080/contact.php ; echo
Thanks for the message, root:x:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

# secDevLabs: Mitigação via Pull Requests

```
168 +     comment = escape(request.form.get('comment'))
169 +     user = escape(session.get('username'))
168 170     date = datetime.datetime.now()
169 171     if comment == '':
170 172         flash('All fields are required', 'danger')
#
@@ -198,10 +200,10 @@ def gossip(id):
198 200     @login_required
199 201     def newgossip():
200 202         if request.method == 'POST':
201 -             text = request.form.get('text')
202 -             subtitle = request.form.get('subtitle')
203 -             title = request.form.get('title')
204 -             author = session.get('username')
203 +             text = escape(request.form.get('text'))
204 +             subtitle = escape(request.form.get('subtitle'))
205 +             title = escape(request.form.get('title'))
206 +             author = escape(session.get('username'))
205 207             date = datetime.datetime.now()
206 208             if author is None or text is None or subtitle is None or title is None:
207 209                 error('gossip', 'Invalid parameters', session.get('username'))
```

# secDevLabs: Mitigação via Pull Requests



## [A5][Ecommerce API] Broken Access Control #275

Closed joserenatosilva wants to merge 1 commit into `globocom:master` from `joserenatosilva:a5-ecommerce-api`

Conversation 1   Commits 1   Checks 0   Files changed 5   +47 -14

joserenatosilva commented 26 days ago Member + ...

This solution refers to which of the apps?  
A5 - Ecommerce API

What did you do to mitigate the vulnerability?  
Added the user ID to the JWT Token (sessionIDa5). Changed the ticket endpoint to get the current user ID from the JWT Token and the login endpoint now redirects to the new ticket endpoint.

Did you test your changes? What commands did you run?  
I couldn't test since the endpoint changed and I can't supply another user's ID to the ticket endpoint.

**Reviewers** Krlier

**Assignees** No one—assign yourself

**Labels** A5-OWASP-2017 Vulnerable Ecommerce API globo.com mitigation solution



Krlier commented on Aug 28

Member

+  ...

Very nice, @joserenatosilva! What you proposed mitigates this vulnerability! 

Although creating a regexp to try and filter out malicious users' inputs works in this case, this might create quite a cat and mouse situation, as they could come up with new queries to bypass the validation. You could try using parameterized queries, also known as prepared statements, which will pre-compile a SQL query so that all you need to do is supply the parameters to it. This way, if an attacker tries to perform a SQL injection, the query is already compiled, limiting the attack vectors.

Would you like to try adding this layer of protection as well?



1

rafaeira3 requested changes on Mar 28

View changes

rafaeira3 left a comment • edited

Member + 😊 ...

Hey, @lousander ! Nice start! 😊

A good strategy is trying to check if your log answers the following security questions:

- Who? "owner" : { "name" : "user", "ip" : "[::1]:52355" }
- What? "action" : {"insert coupon", "response\_status" : 404}
- How? Where? "request" : {"route" : "/coupon", "method" : "POST"}
- Why? "errors" : [{"Invalid parameters" }]
- When? "start\_time" : "2018-10-25T15:38:51.870Z", "end\_time" : 2018-10-25T15:38:51.870Z", "level": "ERROR"}

Would you like to take a deeper look in your logs for this app? I will send you our internal presentation for references.

Just keep in mind that in some cases (as login attempts, for example) you may consider that some users can send their passwords into an username field mistakenly ! Logging this username may cause a data sensitive exposure vulnerability! 😱

1



vitoriaro commented on Feb 5 • edited

Contributor

+  ...

Hi @pedrokiefer, your solution works well! But I noticed that you are using bleach in password fields too. This is not a good idea because you are changing the user password, for example:

If I have the following password: <script>  
bleach changes it to &lt;script&gt

Remember that passwords should not be shown to users so, in this case, we don't have to escape special characters.

# secDevLabs: Projeto Open Source

[globocom / secDevLabs](#)

Code Issues 14 Pull requests 10 Projects 0 Wiki Security Insights Settings

A laboratory for learning secure web development in a practical manner. [Edit](#)

owasp-top-10 labs development training security vulnerability Manage topics

523 commits 1 branch 0 releases 9 contributors BSD-3-Clause

Branch: master New pull request Create new file Upload files Find File Clone or download

rafaveira3 Merge pull request #269 from globocom/fix-README-template-image	Latest commit 34cff73 on Aug 15	
.github	docs FEAT: Add new suggestions to template	2 months ago
docs	docs FIX: Add link to payload image	2 months ago
images	[DOCS] Add new logo and sections to README.md	8 months ago
owasp-top10-2017-apps	[FIX] Add . into localhost link	3 months ago
.gitignore	[FEAT] Add a2 and a10 apps	9 months ago

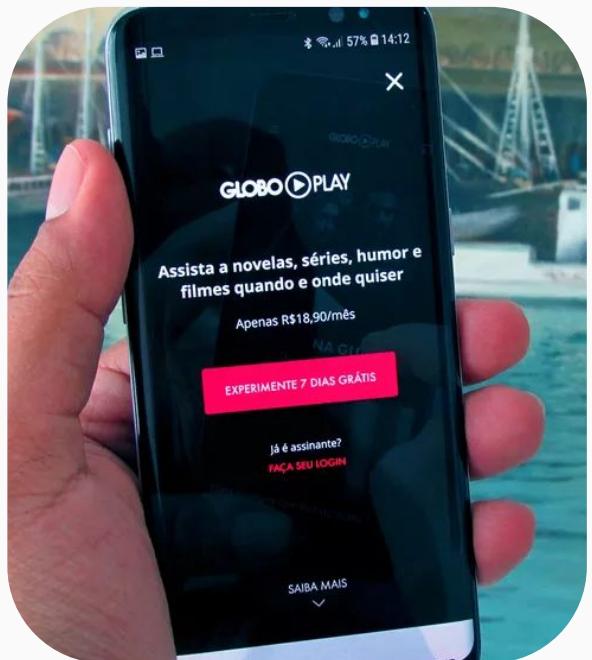
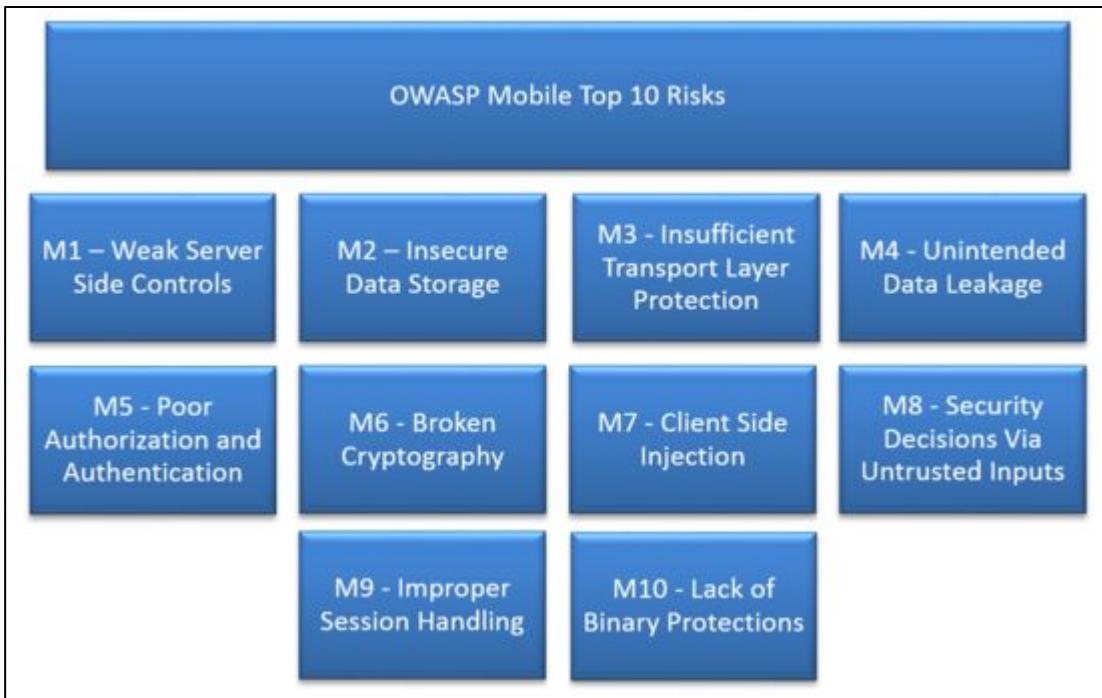
<https://github.com/globocom/secDevLabs>

Demo 🔥

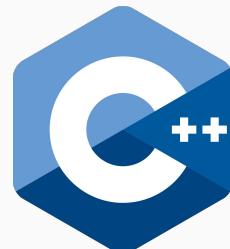
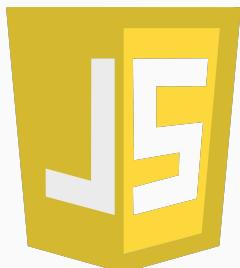
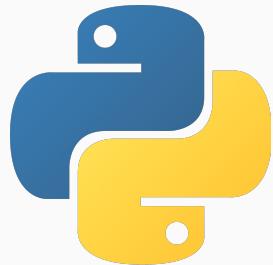
# Próximos passos



# Próximos passos: Apps em Mobile



# Próximos passos: Contemplar mais linguagens



Extra ☕

# Extra: Premiações secDevLabs dentro da Gcom! SECCIM;



# Extra: Hacktoberfest



# Hacktoberfest

**1 a 31 de outubro**  
na [Globo.com](https://Globo.com)

Contribua e ganhe uma camiseta exclusiva.

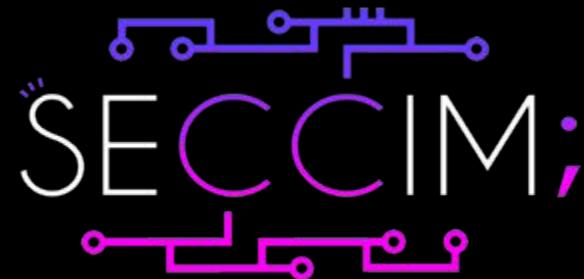


<https://opensource.globo.com>

# Extra: Trabalhe conosco!



<https://talentos.globo.com>



<https://github.com/globocom/secDevLabs>  
Perguntas?



Rafael dos Santos rafael.santos@corp.globo.com



<https://github.com/globocom/secDevLabs>  
Obrigado!



Rafael dos Santos rafael.santos@corp.globo.com