

Inspeção de código para a entrega contínua de software de qualidade

21 de outubro de 2019

Apresentador: Augusto Mariano Pinheiro



ORIENTAÇÕES INICIAIS



Dê preferência ao uso de uma conexão de banda larga

O evento fará uso de vídeo (webcam), avise se houver problemas que alternamos para apenas os slides e áudio

Se for necessário, ajuste o idioma da sala na barra de ferramentas superior

O evento terá cerca de 45 minutos de apresentação e 15 minutos de Q&A

Você pode mandar desde já suas perguntas pelo chat.

Use o chat só para o assunto do webinar

Para quem possui certificação do PMI, como a PMP, o evento vale 1 PDU

Esta sessão será publicada em nosso canal do Youtube: [youtube.com/user/fattoocs](https://www.youtube.com/user/fattoocs)

Certificado de participação será disponibilizado para os assistentes, via e-mail



apoiar nossos clientes no **planejamento e avaliação de desempenho de processos de TI para alavancar o sucesso de seu negócio**



Agenda



- ❑ Necessidades do desenvolvimento moderno de software
 - Ágil, DevOps e suas práticas
- ❑ Integração Contínua
- ❑ Ferramentas para Inspeção Contínua
- ❑ SonarQube
- ❑ Metas de qualidade de código e indicadores
- ❑ Conclusões

Princípios do Manifesto Ágil

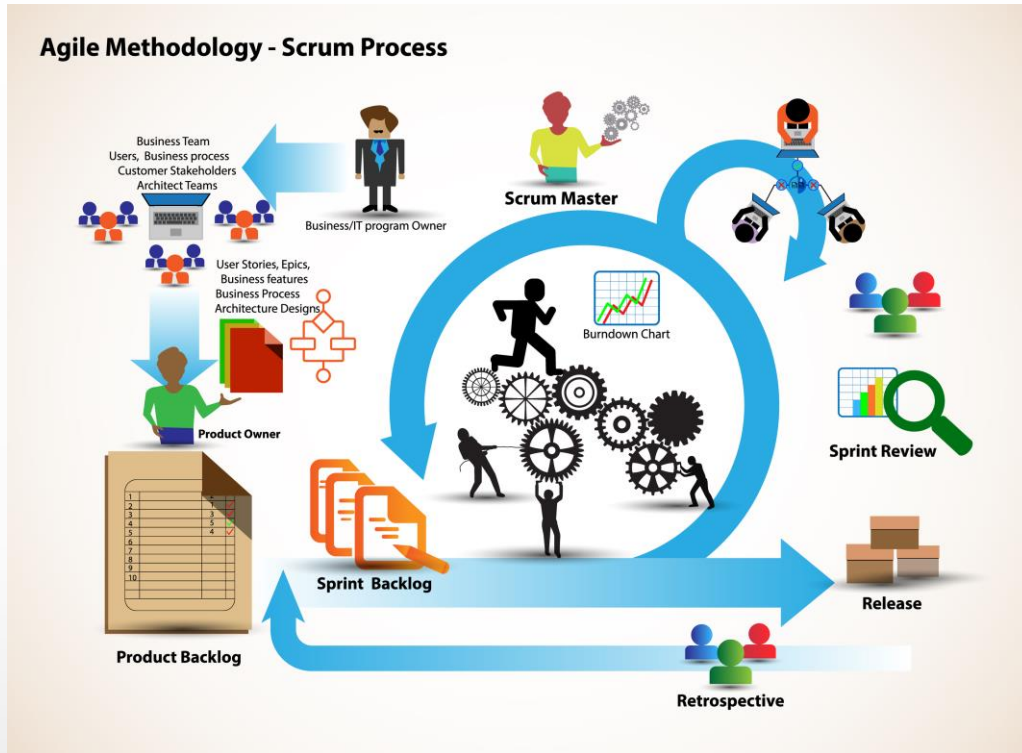
#1 Nossa maior prioridade é **satisfazer o cliente**, através da **entrega adiantada e contínua de software de valor**.

#2 **Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento**. Processos ágeis se adequam a mudanças, **para que o cliente possa tirar vantagens competitivas**.

#3 **Entregar software funcionando com frequência**, na escala de semanas até meses, **com preferência aos períodos mais curtos**.



Necessidades do desenvolvimento



- ❑ #1: Satisfazer o cliente!
 - **Software em funcionamento** mais que documentação abrangente
- ❑ #2: Aceitar mudanças de requisitos
 - **Responder a mudanças** mais que seguir um plano

❑ #9: **Contínua atenção à excelência técnica e bom design**, aumenta a agilidade.

❑ **Inovar??**

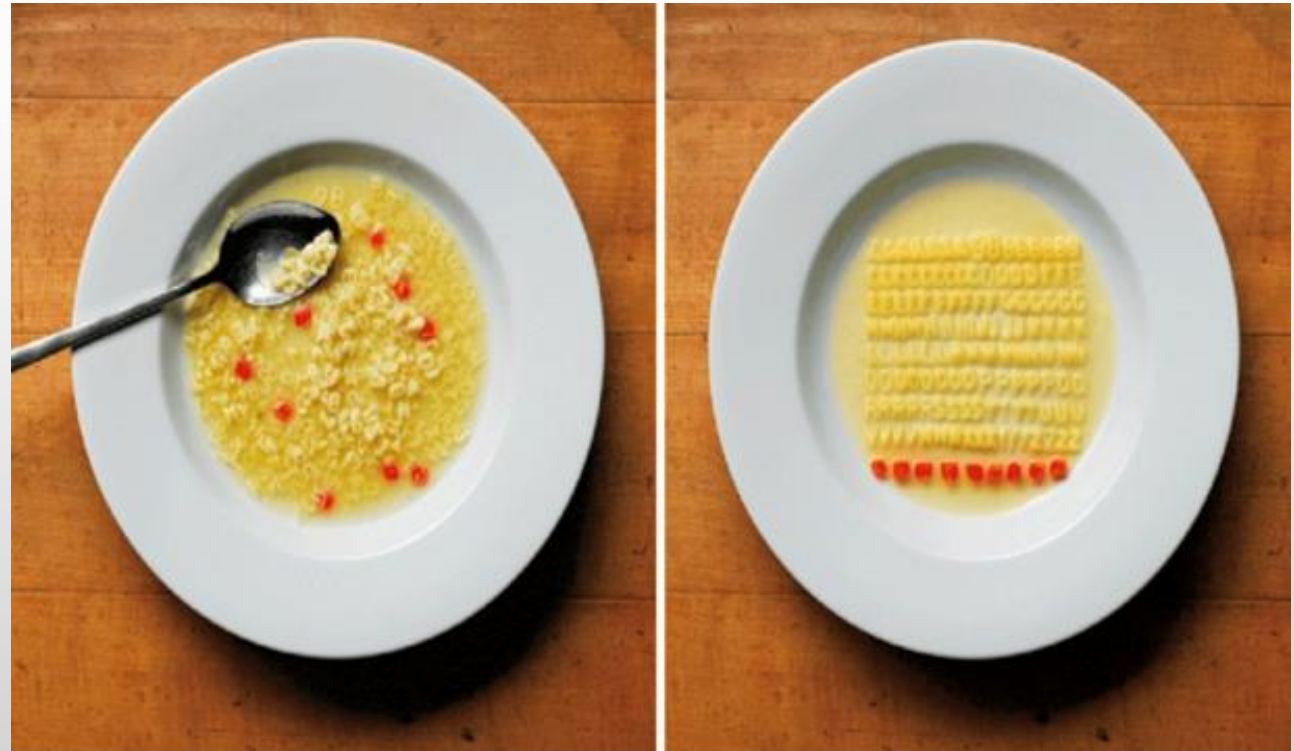


Necessidades do controle de operações



- ❑ Manter o ambiente de produção funcionando e estável

- ❑ Manter o ambiente de produção organizado



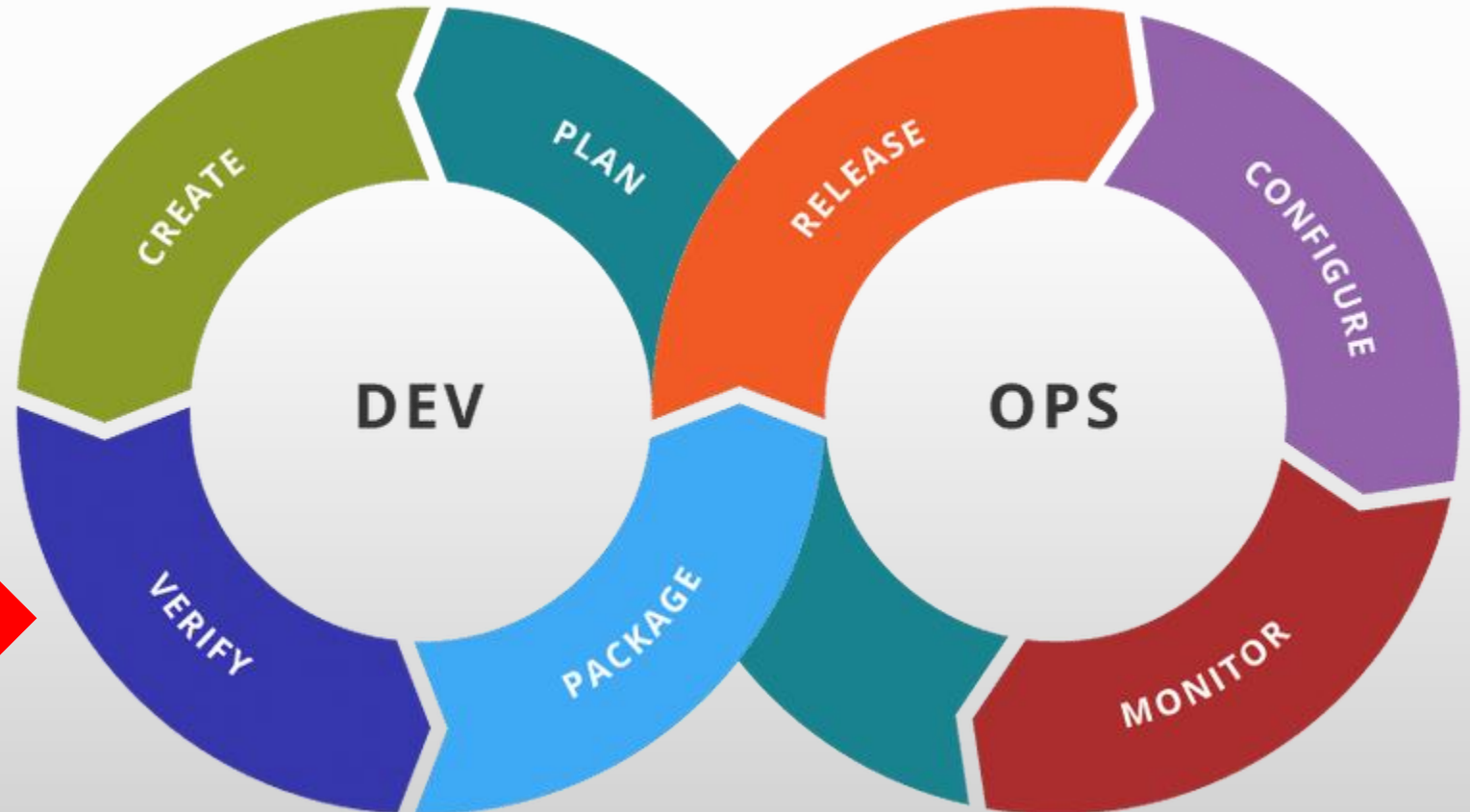
Necessidades de todos!



- ❑ ***#1: Atender aos requisitos do negócio!***
- ❑ **#2: Permitir ao cliente tirar vantagens competitivas através da entrega contínua de software de valor**
- ❑ **#3: Tudo isso, sem comprometer a estabilidade do ambiente de produção**

DevOps como resposta à estas necessidades

- ❑ **Devops** é um termo criado para descrever um conjunto de práticas para integração entre as equipes de **desenvolvimento de softwares, operações** (infraestrutura) e de **apoio** envolvidas (como controle de qualidade) e a adoção de processos automatizados para produção rápida e segura de aplicações e serviços.



**Foco da apresentação
de hoje!**

Práticas do DevOps



- Integração Contínua**
- Entrega Contínua**
- Microserviços**
- Infraestrutura como código**
- Monitoramento e registro em log**
- Comunicação e colaboração**

❑ *O que é*

- é uma prática em que os desenvolvedores, com frequência, juntam suas alterações de código em um repositório central, depois disso, ***compilações e inspeções e testes são executados.***

❑ *Objetivos*

- 1) Investigar e encontrar bugs mais rapidamente***
- 2) Melhorar a qualidade do software***
- 3) Reduzir o tempo que leva para validar e lançar novas atualizações de software***

❑ *Requer, ao mesmo tempo*

- Um (ou mais) componente(s) de automação
- Um componente cultural

Ferramentas para Inspeção

checkstyle

sonarqube

Pmd
DON'T SHOOT THE MESSENGER

sonarlint

A cartoon illustration of a red beetle with large white eyes and black legs.
FindBugs™

Inspeção contínua com o SonarQube



□ O que é:

- **Ferramenta de revisão/inspeção automática de código-fonte** criada para detectar bugs, vulnerabilidades e más práticas de programação.
- Permite a definição dos parâmetros de inspeção, análise dos resultados, consultas e ações de correção e tratamento

□ Provê informações para todos os envolvidos através de seus *dashboards*.

- Equipe de desenvolvimento
- Coordenador do projeto
- Área de qualidade
- Arquiteto

❑ Características que são avaliadas durante as inspeções:

▪ **Confiabilidade**

- Violações (classifica por severidade)

▪ **Segurança**

- Vulnerabilidades

▪ **Manutenibilidade**

- Problemas em potencial

▪ **Cobertura**

- Testes unitários

▪ **Duplicação**

- De linhas, blocos ou arquivos

▪ **Tamanho**

- Linhas de código comentadas (lixo) ou não alcançadas
- Densidade de comentários (anotações)

▪ **Complexidade**

- Ciclomática e Cognitiva

❑ Cálculo do Débito Técnico

- Estimativa de esforço (HH) para corrigir os problemas encontrados

SonarQube – Relação de todos os projetos

sonarqube **Projects** Issues Rules Quality Profiles Quality Gates Administration + A

Perspective: Overall Status Sort by: Name 3 projects

My Favorites **All**

Filters

Quality Gate

Passed 2

Failed 1

Reliability (🐛 Bugs)

A 1

B 0 |

C 0 |

D 0 |

E 2

Security (🔒 Vulnerabilities)

A 1

B 1

C 0 |

D 0 |

E 1

☆ [AppWeb01](#) **Failed** Last analysis: October 16, 2019, 4:35 PM

738 **E** | 6 **E** | 1.9k **A** | 0.0% | 22.6%

🐛 Bugs | 🔒 Vulnerabilities | 🐛 Code Smells | Coverage | Duplications

55k **M** JavaScript, PHP, ...

☆ [ProgramaJava](#) **Passed** Last analysis: September 24, 2019, 2:52 PM

0 **A** | 0 **A** | 0 **A** | 0.0% | 30.2%

🐛 Bugs | 🔒 Vulnerabilities | 🐛 Code Smells | Coverage | Duplications

817 **XS** Java

☆ [RelatorioASP](#) **Passed** Last analysis: October 15, 2019, 5:54 PM

15 **E** | 14 **B** | 113 **A** | 0.0% | 0.2%

🐛 Bugs | 🔒 Vulnerabilities | 🐛 Code Smells | Coverage | Duplications

345k **L** XML, CSS, ...

SonarQube – Visão geral projeto “AppWeb01” (1/2)



AppWeb01 master October 16, 2019, 4:35 PM Version 0.9.0

Overview Issues Measures Code Activity Administration

Quality Gate **Failed**

Reliability Rating on New Code is worse than A

Reliability Measures

738 Bugs **E** started yesterday

New code: since previous version started yesterday

21 New Bugs **E**

Security Measures

6 Vulnerabilities **E**

32 Security Hotspots

0 New Vulnerabilities **A**

0 New Security Hotspots

Maintainability Measures

About This Project

No tags

55k Lines of Code

- JavaScript 24k
- PHP 23k
- HTML 7.4k
- CSS 864
- XML 28

Project Activity

October 16, 2019 **0.9.0** Quality Gate: Red (was Green)

October 15, 2019 **0.8.0**

SonarQube – Visão geral projeto “AppWeb01” (2/2)



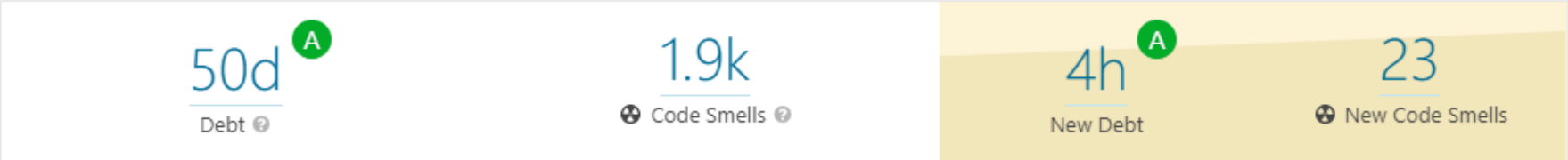
AppWeb01 master

October 16, 2019, 4:35 PM Version 0.9.0

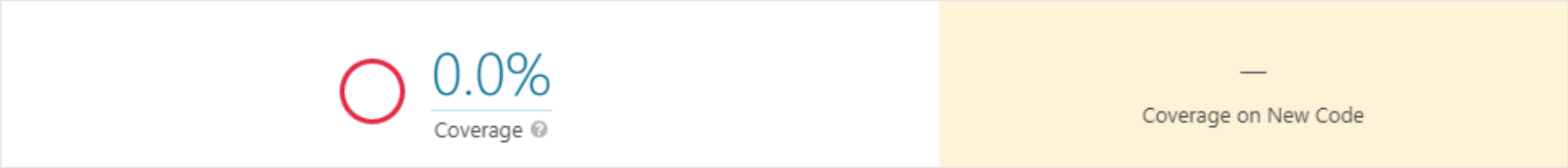
Overview Issues Measures Code Activity Administration



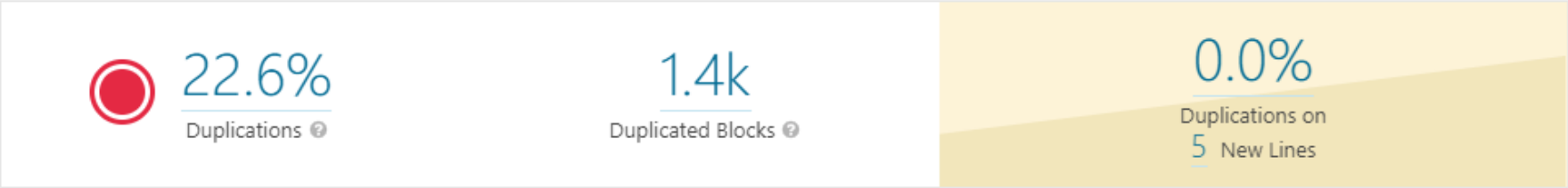
Maintainability Measures



Coverage Measures



Duplications Measures



October 15, 2019

0.8.0

Show More

Quality Gate

(Default) Sonar way

Quality Profiles

- (CSS) Sonar way
- (JavaScript) Sonar way
- (PHP) Sonar way
- (HTML) Sonar way
- (XML) Sonar way

Project Key

AppWeb01

SonarQube – Violações (1)



AppWeb01 master

October 15, 2019, 10:26 AM Version 0.8.0

Overview Issues Measures Code Activity Administration



My Issues All

Bulk Change ↑ ↓ to select issues ← → to navigate 1 / 2,181 issues 44d effort

Filters

Type

- Bug 714
- Vulnerability 6
- Code Smell 1.4k
- Security Hotspot 32

Severity

- Blocker 10
- Critical 782
- Major 881
- Minor 476
- Info 0

Resolution

Status

Security Category

Creation Date

Language

- app/Exceptions/Handler.php**
 - Remove this method "report" to simply inherit it. [See Rule](#) last year L34
 Code Smell Minor Open Not assigned 2min effort Comment
 clumsy, redundant
- app/Exports/EmpresaExport.php**
 - Remove this commented out code. [See Rule](#) last year L8
 Code Smell Major Open Not assigned 5min effort Comment
 unused
- app/Exports/MedicaoExport.php**
 - 1 duplicated blocks of code must be removed. [See Rule](#) 3 months ago
 Code Smell Major Open Not assigned 20min effort Comment
 bad-practice, pitfall
- app/Exports/MinhasMedicoesExport.php**
 - 1 duplicated blocks of code must be removed. [See Rule](#) 3 months ago
 Code Smell Major Open Not assigned 20min effort Comment
 pitfall
- app/Helpers/FMS.php**
 - Immediately return this expression instead of assigning it to the temporary variable "\$perfil". [See Rule](#) 11 months ago L33

SonarQube – Violações (2)



Administration ▾

AppWeb01 / app / Model / EstruturaElementos.php

```
43  egara...
44      public static function getDefElemento($idElemento)
45      {
46  egara...      $el = self::findOrFail($idElemento);
47      $query = "select * from ft_estrutura_elementos({$el->sistema_id},TRUE) where id={$idElemento}";
48      return \DB::select(\DB::raw($query));
49      }
50  egara...
51      public static function checkAjustEstruturaElementos($idSistema)
52      {
53  egara...      $maxOrdem = self::where('sistema_id', $idSistema)->max('ordem');
54      $count = self::where('sistema_id', $idSistema)->count();
55  egara...      if ($maxOrdem != $count || 1 == 1) {
56      $query = "select id from ft_estrutura_elementos({$idSistema},TRUE) order by regexp_split_to_array(caminho_hierarquico
57  egara...      $els = \DB::select(\DB::raw($query));
58      for ($x = 0; $x < count($els); $x++) {
59      $elemento = self::findOrFail($els[$x]->id);
60  egara...      if ($elemento->ordem != $x + 1) {
61  egara...          $elemento->ordem = $x + 1;
62          $elemento->save();
63      }
64      }
```

Identical sub-expressions on both sides of operator "==" [See Rule](#)

2 months ago ▾ L55 🔗

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 2min effort [Comment](#)

No tags ▾

AppWeb01 master October 15, 2019, 10:26 AM Version 0.8.0

Overview **Issues** Measures Code Activity Administration

1 / 6 issues

app/.../Controllers/Auth/LoginController.php

```
97 geyce...
98 egara... $u->name = $user->name;
99          $u->email = $user->email;
100         $u->provider = $provider;
101         $u->provider_id = $user->id;
102         $u->language = 'pt-BR';
103         $u->password = '123456';
```

'password' detected in this variable name, review this potentially hardcoded credential. [See Rule](#) 8 months ago L103

Vulnerability **Blocker** Open Not assigned 30min effort Comment cwe, owasp-a2, sans-top25-porous

Credentials should not be hard-coded

php:S2068

Vulnerability **Blocker** Main sources cwe, owasp-a2, sans-top25-porous Available Since Sep 09, 2019 SonarAnalyzer (PHP) Constant/issue: 30min

Because it is easy to extract strings from a compiled application, credentials should never be hard-coded. Do so, and they're almost guaranteed to end up in the hands of an attacker. This is particularly true for applications that are distributed.

Credentials should be stored outside of the code in a strongly-protected encrypted configuration file or database.

Noncompliant Code Example

```
$uname = "steve";
$password = "blue";
connect($uname, $password);
```

SonarQube – Manutenibilidade (1)



AppWeb01 master

October 16, 2019, 5:14 PM Version 0.11.0

Overview Issues Measures Code Activity Administration



36 / 1,249 issues

Define and throw a dedicated exception instead of using a generic one.

Code Smell

Define a constant instead of duplicating this literal "Permissão Negada" 11 times.

Code Smell +11

alt + ↑ ↓ to navigate issue locations

Define and throw a dedicated exception instead of using a generic one.

Code Smell

Define a constant instead of duplicating this literal "ordem" 15 times.

Code Smell +15

Define a constant instead of duplicating this literal "sistema_id" 8 times.

Code Smell +8

Define a constant instead of duplicating this literal "ordem+1" 3

AppWeb01 app/Http/Controllers/EstruturaElementosController.php

See all issues in this file

```
77 egara... $sistema = Sistema::findOrFail($input->sistema_id);
78          if ($sistema->empresa_id != Auth::user()->empresa_id_ultima_sessao) {
79          throw new Exception( 1 "Permissão Negada");
```

Define a constant instead of duplicating this literal "Permissão Negada" 11 times. [See Rule](#)

2 months ago L79

Code Smell Critical Open Not assigned 24min effort Comment

design

```
80          }
81 egara... if (empty($input->id) || $input->id <= 0) {
```

```
168 egara... // testando permissao acesso elemento=>sistema=>empresa c/ a empresa atual
169          if ($sistema->empresa_id != Auth::user()->empresa_id_ultima_sessao) {
170          throw new Exception( 2 "Permissão Negada");
171          }
172 egara... if ($destino['id'] > 0) {
```

```
330 egara... $sistema = Sistema::findOrFail($elementosAgregar[0]->sistema_id);
331 egara... if ($sistema->empresa_id != Auth::user()->empresa_id_ultima_sessao) {
332          throw new Exception( 3 "Permissão Negada");
333          }
334 egara... $elmtos = [];
335 // conferindo se todos os elementos agregados pertencem ao mesmo sistema
336 for ($x = 1; $x < count($elementosAgregar); $x++) {
337     if ($sistema->id != $elementosAgregar[$x]->sistema_id) {
338     egara... throw new Exception( 4 "Permissão Negada");
```

SonarQube – Manutenibilidade (2)



October 15, 2019, 5:54 PM Version 1.0

RelatorioASP master

Overview Issues Measures Code Activity Administration

Project Overview

- Reliability
- Security
- Maintainability **Code Smells**

Overview

On new code

Code Smells	0
Debt	0
Debt Ratio	0.0%
Rating	A

Overall

Code Smells	43
Debt	43min
Debt Ratio	0.1%
Rating	A
Effort to Reach A	0

Coverage

Duplications

Size

Complexity

C

RelatorioASP / padraovisual/visual_01 / css / MTE_Estilo_Impressao.css

```
1001     COLOR: #4c4436;  
1002     FONT-FAMILY: arial, "trebuchet ms", trebuchet, sans-serif  
1003 }  
1004 DIV  
1005 {  
1006     FONT-WEIGHT: normal;  
1007     COLOR: #4c4436;  
1008     FONT-FAMILY: arial, "trebuchet ms", trebuchet, sans-serif  
1009 }  
1010 P  
1011 {  
1012     FONT-WEIGHT: normal;  
1013     COLOR: #4c4436;  
1014     FONT-FAMILY: arial, "trebuchet ms", trebuchet, sans-serif  
1015 }  
1016 UL  
1017 {  
1018     FONT-WEIGHT: normal;  
1019     COLOR: #4c4436;  
1020     FONT-FAMILY: arial, "trebuchet ms", trebuchet, sans-serif  
1021 }  
1022 OL  
1023 {  
1024     FONT-WEIGHT: normal;  
1025     COLOR: #4c4436;  
1026     FONT-FAMILY: arial, "trebuchet ms", trebuchet, sans-serif  
1027 }
```

Unexpected duplicate selector "DIV", first used at line 37 [See Rule](#) yesterday L1004

Code Smell Major Open Not assigned 1min effort Comment No tags

SonarQube – Manutenibilidade (3)



AppWeb01 master

October 16, 2019, 5:14 PM Version 0.11.0

Overview Issues **Measures** Code Activity Administration

Project Overview

- Reliability
- Security
- Maintainability** Code Smells

Overview

On new code

- Code Smells 16
- Debt 3h 20min
- Debt Ratio 0.0%
- Rating **A**

Overall

- Code Smells 21**
- Debt 3h 45min
- Debt Ratio 1.0%
- Rating **A**
- Effort to Reach A 0

AppWeb01 / public / app / perfis / perfis.js

```
145 function getToolbar() {
146     return toolbar;
147 }
148
149 function getTitulos() {
150
151     return titulos;
152 }
153 function getUser() {
154
155     return seguranca.user;
156 }
157 function getFuncao() {
158
159     return seguranca.funcao;
160 }
```

Remove unused function 'getTitulos'. [See Rule](#) 2 months ago L149

Code Smell Minor Open Not assigned 5min effort Comment unused

Remove unused function 'getUser'. [See Rule](#) 2 months ago L153

Code Smell Minor Open Not assigned 5min effort Comment unused

Remove unused function 'getFuncao'. [See Rule](#) 2 months ago L157

Code Smell Minor Open Not assigned 5min effort Comment unused

SonarQube – Duplicações (1)



AppWeb01 master October 16, 2019, 5:14 PM Version 0.11.0

Overview Issues **Measures** Code Activity Administration

Project Overview

- Reliability
- Security
- Maintainability
- Coverage
- Duplications** Duplicated Lines (%)

Overview 🔗

On new code

- Duplicated Lines 0
- Duplicated Blocks 0

Overall

- Density 21.9%**
- Duplicated Lines 14,785
- Duplicated Blocks 1,364
- Duplicated Files 69

Size

Complexity

Issues

AppWeb01 View as **Tree** to select files to navigate 11 files

Duplicated Lines (%) 21.9% 📈 New code: since 0.10.0

	Duplicated Lines (%)	Duplicated Lines
app	20.3%	3,512
config	5.5%	93
database	13.7%	1,067
public	31.7%	9,466
resources	4.3%	443
routes	53.5%	204
tests	0.0%	0
phpunit.xml	0.0%	0
server.php	0.0%	0
teste.php	0.0%	0
webpack.mix.js	0.0%	0

11 of 11 shown

SonarQube – Duplicações (2)

```
app/Http/Controllers/UsuarioController.php
616
617     public function excluir(Request $request)
618     {
619         $input = (object) $request->input();
620         $ok = true;
621         $mens = "";
622         $dados = null;
623         try {
624             $dados = Usuario::findOrFail($input->id);
625
626             if (! empty($dados->avatar) && $dados->avatar != "user.png" && $dados->avatar != "male.png" &&
627                 $avatar = 'avatar_' . $input->id . '.' . explode('.', $dados->avatar)[1];
628                 $url = Config::get('app.images') . '/avatars/' . $avatar;
629                 if (Storage::disk('interno')->exists($url)) {
630                     Storage::disk('interno')->delete($url);
631                 }
632         }
```

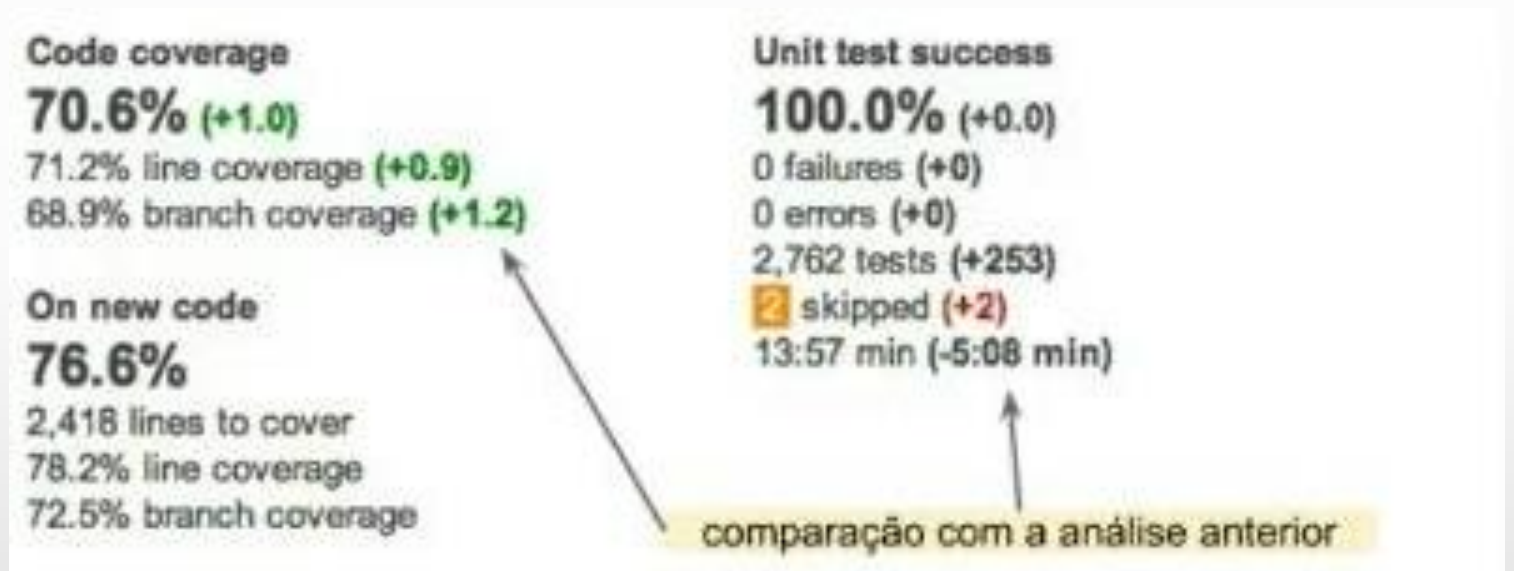
SonarQube – Duplicações (3)



AppWeb01 / app / Http / Controllers / UsuarioController.php

```
706     $mens = $ex->getMessage();
707     }
708     return redirect()->back()->with($ok, $mens);
709 }
710
711 public function avatar_excluir(Request $request)
712 {
713     $input = (object) $request->input();
714     $ok = true;
715     $mens = "";
716     $dados = null;
717     try {
718         $dados = Usuario::findOrFail($input->id);
719         if (! empty($dados->avatar) && $dados->avatar != "user.png" && $dados->avatar != "male.png" && $dados->avatar != "
720             $avatar = 'avatar_' . $input->id . '.' . explode('.', $dados->avatar)[1];
721             $url = Config::get('app.images') . '/avatars/' . $avatar;
722             if (Storage::disk('interno')->exists($url)) {
723                 Storage::disk('interno')->delete($url);
724             }
725
726     Duplicated By
727     Lines: 618 – 632
728     } catch (Exception $ex) {
```

- ❑ Cobertura de código
 - testes unitários



SonarQube – Documentação e Comentários



AppWeb01 master

October 16, 2019, 5:14 PM Version 0.11.0

Overview Issues **Measures** Code Activity Administration



Project Overview

- Reliability
- Security
- Maintainability
- Coverage
- Duplications
- Size **Comments (%)**
 - New Lines 0
 - Lines of Code 9,253
 - Lines 10,655
 - Statements 4,785
 - Functions 252
 - Classes 29
 - Files 30
 - Comment Lines 285
 - Comments (%) 3.0%**
- Complexity
- Issues

AppWeb01 / app / Http / Controllers

View as Tree to select files to navigate 26 files

Comments (%) 3.0%

New code: since 0.10.0

Auth	19.5%
AnexosSistemaController.php	3.3%
Controller.php	0.0%
DashboardController.php	2.6%
EmpresasController.php	0.3%
EstruturaElementosController.php	9.7%
FaseProjetoController.php	3.0%
FatorImpactoController.php	3.0%
FuncoesController.php	0.0%
HomeController.php	8.2%
InmController.php	2.6%
InternacionalizacaoController.php	0.0%
ItemMedicaoController.php	0.0%
LocalizationController.php	0.0%
MedicaoController.php	2.7%

SonarQube – Complexidade Ciclomática e Cognitiva

The screenshot shows the SonarQube interface with a central panel highlighting complexity metrics and a code smell issue. The metrics panel includes a bar chart and a flow diagram.

Complexity Metrics:

- 1.9 /method
- 10.3 /class
- 11.1 /file
- Total: 13,617

Bar Chart: Shows complexity distribution for Methods and Files across complexity levels 1 to 12.

Flow Diagram: A graph with nodes A through H. Node A is at the top, connected to B. B is connected to C and D. C is connected to F. D is connected to E. E is connected to G. F is connected to G. G is connected to H.

Equation: $M = 9 - 8 + 2 \times 1 = 3$

Complexity Formula: $M = A - N + 2 \times C$

Legend:

- M = Cyclomatic Complexity (McCabe)
- A = Arrows
- N = Nodes
- C = Componentes

Code Smell Legend:

- 1 a 4 = Bom
- 5 a 7 = Ok
- 8 a 10 = Considere refatorar
- >10 = Refatorar!

Issue Details:

- Issue: Refactor this function to reduce its Cognitive Complexity from 46 to the 15 allowed.
- Severity: Critical
- Resolution: Open
- Assignee: Harlan Nascimento
- Effort: 36min
- Created: 3 months ago
- Rule: L27
- Tag: brain-overload

Metas de Qualidade de Código como parte de um Acordo de Níveis de Serviço



Métrica	Meta	Severidade
Cobertura de código na camada de negócio	$\geq 50\%$	Média
% de testes automatizados executados com sucesso	$= 100\%$	Média
Densidade de comentários na camada de negócio	$\geq 30\%$	Baixa
Complexidade por método	$\leq 10\%$	Média
Densidade de duplicação	$\leq 5\%$	Baixa
Blocos duplicados na mesma classe	$= 0$	Média
Linhas de código-fonte comentadas	$= 0$	Média
Violações bloqueadoras	$= 0$	Alta
Violações críticas	$= 0$	Alta
Violações maiores	$\leq 5\%$	Média
Violações menores	$\leq 5\%$	Baixa
Débito técnico	$\leq 10\%$	Não se aplica

Metas de Qualidade de Código como parte de um Acordo de Níveis de Serviço



Grupo	Métrica	Unidade	Meta
Projeto	Complexidade / Arquivo	Média total	≤ 10
	Complexidade / Classe	Média total	≤ 10
	Complexidade / Função	Média total	≤ 3
	Duplicações	%	$\leq 4\%$
	Vulnerabilidades (confirmadas ou em potencial)	Unidades	$= 0$
	Taxa de débito técnico	%	$\leq 2,5\%$
	SQALE RATING (Maintainability Rating)	Nota	A
Violações de código (possíveis bugs, estilo de codificação, más práticas de codificação):	Violações Críticas	Unidades	$= 0$
	Violações Bloqueadoras	Unidades	$= 0$
Indicadores relacionados a testes	Cobertura de Testes Unitários (na camada de negócio)	%	$\geq 70\%$
	Testes Unitários executados com sucesso	%	$\geq 100\%$
	Testes Unitários não executados	Unidades	$= 0$

❑ Vantagens

- Muitos defeitos diferentes podem ser descobertos em uma única inspeção
- Versões incompletas do sistema podem ser inspecionadas

❑ Não é adequada

- Para verificar alguns requisitos não funcionais, como desempenho, por exemplo
- Para demonstrar que o software é útil

□ Vantagens

- Permite encontrar inconsistências entre um programa e sua especificação, isto é, **revelar defeitos**
- Permite **demonstrar se o software é (e faz) o que o cliente deseja**

□ Limitações

- Segundo Dijkstra, *“Testes podem somente revelar a presença de defeitos, não a ausência”*
- Alto custo da realização dos testes funcionais
- Só podem ser realizados após a conclusão do desenvolvimento do componente de software

- ❑ **Inspeções e testes não são abordagens concorrentes, mas sim, complementares!**
 - cada uma tem vantagens e desvantagens sobre a outra
 - devem ser usadas em conjunto no processo de Verificação e Validação

- ❑ **Lembrete: Qualidade não é um atributo que se insere no produto ao final do desenvolvimento, portanto deve se ter atenção em todas as atividades do ciclo de vida do software.**

Como a FATTO pode te ajudar?



□ Atividades

- Instalação, configuração e preparação das ferramentas
- Automatização do processo de inspeção
- Mentoria dos envolvidos no processo
- Criação do baseline de erros
- Implantação assistida
- Interpretação e avaliação dos resultados da inspeção
- Ajustes nas regras e perfis de inspeção

□ Contatos

- contato@fattocs.com.br



PRÓXIMOS EVENTOS



☐ WEBINAR:

- SNAP: Medição não funcional de software
- Data: 18/11/2019

13 horas (Horário de Brasília)

Inscrições gratuitas em: <https://bit.ly/35LuyN0>

AVALIAÇÃO





Apresentador

AUGUSTO MARIANO PINHEIRO

- E-mail: augusto.pinheiro@fattocs.com
- WhatsApp: +552799946-2525