RC - Maria Eduarda Lemos e Mateus Campos Caçabuena

No trabalha T1 foi utilizado o GitHub Actions para ativar a integração contínua (IC) e revisão de código (RC) automatizados.

Na aula de hoje, faremos um estudo dirigido em grupos sobre esses assuntos.

RC/CR (Exemplos: SonarQube, CodeQL)

Cite ao menos três fontes consultadas. Ao menos um vídeo.

**1 - Sintetizar em um parágrafo o propósito da RC (nas palavras do grupo).**

Processo de revisão de um código desenvolvido a fim de achar erros enviando feedbacks sobre eles para melhorá-lo.

**2 - Descrever o conceito e o uso de heurísticas.**

Heurísticas são atalhos cognitivos que nos auxiliam a resolver problemas que outras pessoas já passaram e mapearam como resolver. Em teste de software temos várias heurísticas sobre como testar diferentes aplicações. Uma delas é a VADER para testes de API Rest.

**3 - Qual a diferença entre code quality e code security?**

Code quality e code security são semelhantes, pois ambos os tipos de problemas podem ser identificados com análise estática. Ao observar os caminhos do fluxo de dados por meio de um aplicativo, as ferramentas de análise estática podem identificar onde um aplicativo manipula dados incorretamente ou o código produz resultados não intencionais.

**4 - Indicar o aspecto/funcionalidade da RC que mais chamou a atenção do grupo e por quê.**

Gostamos do aspecto de compartilhamento de conhecimento, pois beneficia tanto o desenvolvedor do código, quanto o avaliador. Aumentando o conhecimento de ambos, é possível construir o próximo código muito melhor do que o primeiro.

**5 - Pesquisar 3 ferramentas de RC e construir uma tabela comparativa. Ao final, indicar a preferida do grupo e o motivo da escolha.**

| **Ferramentas** | **Custo** | **Vantagens** | **Compatibilidade** |
| --- | --- | --- | --- |
| **GitHub** | Gratuito, planos pagos a partir de 7 reais. | realize download de uma nova versão do software;  faça edições e crie novas versões;  tenha um histórico de todas as alterações feitas em arquivos;  contribua em projetos, de maneira bem simples. | MacOS 10.12 ou posterior  Windows 7 64-bit ou versão posterior. Você deve ter um sistema operacional de 64 bits para executar o GitHub Desktop. |
| **Collaborator** | Um pacote de licença de 5 usuários tem um preço de $535 por ano. | Além da revisão do código fonte, Collaborator permite que as equipes revisem documentos de design também. | Suporta um grande número de sistemas de controle de versões como Subversion, [Git](https://kinsta.com/pt/ajuda/git/), CVS, Mercurial, Perforce, e TFS. |
| **Code Scene** | Uma instalação no local do CodeScene custa 15 euros (cerca de 17 dólares) por desenvolvedor por mês. | Ele realiza análise de código comportamental, incluindo uma dimensão temporal para analisar a evolução da sua base de código. | Suporta o sistema windows e linux, ademais, é comumente usado no GitHub |

O GitHub é o nosso favorito, pois ele é o mais fácil de usar e o mais completo, além de ambos integrantes do trabalho utilizarem muito.

Fontes:

<https://natahouse.com/pt/revisao-de-codigo-descubra-o-que-voce-precisa-saber-sobre-o-assunto>

<https://github.com/pagarme/cafe-com-testes/blob/main/artigos/heuristica-vader.md#:~:text=Heur%C3%ADsticas%20s%C3%A3o%20atalhos%20cognitivos%20que,passaram%20e%20mapearam%20como%20resolver>.

<https://kinsta.com/pt/blog/ferramentas-de-revisao-de-codigo/>

<https://www.google.com/amp/s/www.synopsys.com/blogs/software-security/code-quality-code-security/amp/>

https://www.youtube.com/watch?v=\_7W9pqWPyfc