

MATURIDADE

Nome: Maturidade

Intenção: Retornar projetos FLOSS que possivelmente estão ou não evoluídos.

Motivação: Considere que o professor deseje proporcionar aos seus alunos de 3º semestre as habilidades profissionais, de comunicação e técnicas necessárias para contribuir para um projeto de software contínuo. Com isso, ele planeja selecionar um projeto FLOSS hospedado no GitHub para que seus alunos realizem contribuições, mas ele está tendo dificuldade de escolher um projeto adequado. Podemos resolver esse problema selecionando os projetos com base na quantidade de releases disponíveis, uma vez que Ellis et al. relatam que para utilização em sala de aula, o projeto deve ter pelo menos uma versão de produção estável. Os projetos que não têm isso podem não ter maturidade suficiente para apoiar a aprendizagem dos alunos. No entanto, para Jaccheri and Osterlie, se um projeto FLOSS estiver maduro, bem testado e perto de um lançamento, muitas das tarefas restantes podem ser complexas. Se o objetivo é contribuir para um projeto, o professor deve, pelo menos, estar ciente destas possíveis dificuldades.

Aplicabilidade: Use o critério Maturidade, quando: i) Possuir interesse em utilizar projetos com um histórico consistente de versões e releases, o que pode indicar que as tarefas restantes são complexas e exige um maior conhecimento dos alunos; ii) Desejar um projeto com um grande número de atividades de desenvolvimento, ou seja, de releases, o que pode indicar que o projeto contenha muitos casos de estudo para serem analisados em sala de aula; iii) Desejar obter projetos FLOSS estáveis, mas com poucas releases para que alunos iniciantes contribuam com o projeto

Consequências: O critério Maturidade tem os seguintes benefícios e desvantagens: i) A maturidade do projeto é verificada, considerando a quantidade de releases, acreditando assim que esse é um indicador adequado. Dessa forma, podem existir outros indicadores úteis que não são abordados neste estudo. ii) A decisão final deve ser tomada pelo professor. O professor é quem irá informar os valores limiares da quantidade de releases que ele deseja obter. iii) Os projetos retornados possuem apenas indícios de maturidade, mas a maturidade de fato, não pode ser garantida. iv) Ele retorna projetos com a quantidade de release específica. Esse critério é utilizado como parâmetro de busca, para localizar, dentre os projetos disponíveis, aqueles que possuem a quantidade de releases informadas

Implementação: Para implementar esse critério, pode ser utilizada a API V3 do GitHub que possui o objeto Release. Dessa forma, é possível observar quantas versões cada projeto possui, e extrair informações relevantes relacionadas a cada release.

Entrada de Dados: Pode ser apresentado um Ranger Slider com controles deslizantes de intervalo personalizado, para que o professor informe o número mínimo e máximo de releases do projeto.