**MOOD Metrics Guilherme Abrantes (60971)**

Uma imagem com mesa

Descrição gerada automaticamente

As **mood metrics** são constituídas por 6 métricas diferentes: **MHF**, **AHF**, **MIF**, **AIF**, **PF** e **CF**.

**AHF e MHF são métricas relacionadas com a visibilidade.**

AHF  = 1 - Attributes Visible;

MHF = 1 - Methods Visible;

**MHF** tem em conta o número de métodos visíveis  no nosso projeto podemos ver um MHF de 45,86% o que indica que existem quase tantos métodos públicos como privados o que pode resultar em vários métodos especializados que não podem ser reutilizados (45,86% é um número relativamente alto quando comparado com outros projetos).

**AHF** tem em conta o número de atributos visíveis, sendo que o nosso projeto possui um AHF de 89,63% acho um número bastante aceitável onde a maioria dos atributos são privados.

**MIF e AIF são métricas relacionadas com a herança de métodos.**

MIF = inherited methods / total methods available in classes

AIF = inherited attributes / total attributes available in classes

Podemos então constatar um **MIF** de 0 que pode estar a ser causado por as classes que herdam métodos dos pais estarem a redefinir todos os métodos ou então acrescentar novos, este MIF é considerado bastante mau.

O **AIF** deste projeto é de 68,73% o que sugere que bastantes atributos são herdados pelas classes filhas, o que pode não ser o ideal.

**PF Polymorphism Factor:**

PF = overrides / sum for each class(new methods \* descendants)

A **PF** indica o grau de redefinição de métodos durante a herança de classes, podemos então observar um PF de 100% o que indica que nos estamos a dar override a tudo o que condiz com o facto do MIF ser 0%.

**CF Coupling Factor:**

CF = Actual couplings / Maximum possible couplings

CF mede as copulações que existem, para existir uma população de A para B, A tem de chamar métodos ou variáveis que existem em B, neste projeto este valor é de 0,20% o que indica que quase não existem classes copuladas.

**Pontos problemáticos e code smells:**

Sobre os pontos problemáticos temos o PF e o MIF que se completam e indicam que grande parte dos métodos são redefinidos, o que leva às classes filhas a não aproveitarem os métodos que herdam dos pais, o que pode levar ao code Smell Divergent Change, como não existem métodos comuns sempre que queremos adicionar algo temos de reescrever todos os métodos que já existem.