

Chidamber and Kemerer metrics

O conjunto de métrica Chidamber and Kemerer usa as seguintes estatísticas para avaliar a complexidade do código: WMC - *weighted method complexity*; DIT - *depth inheritance tree*; NOC - *number of children*; CBO - *coupling between objects*; RFC - *response for a Class*; LCOM - *Lack of cohesion of methods*.

WMC - *Weighted Method Complexity*

Esta métrica informa o número de métodos implementados numa classe. Números elevados numa classe indicam que as funções, provavelmente, são específicas para a mesma classe, limitando a possibilidade de reutilização. Assim sendo, o WMC é um bom indicador de quanto tempo e esforço é necessário para desenvolver e manter a classe.

Um valor limite para o WMC é dizer que 10% das classes do projeto podem assumir valores superiores a 25.

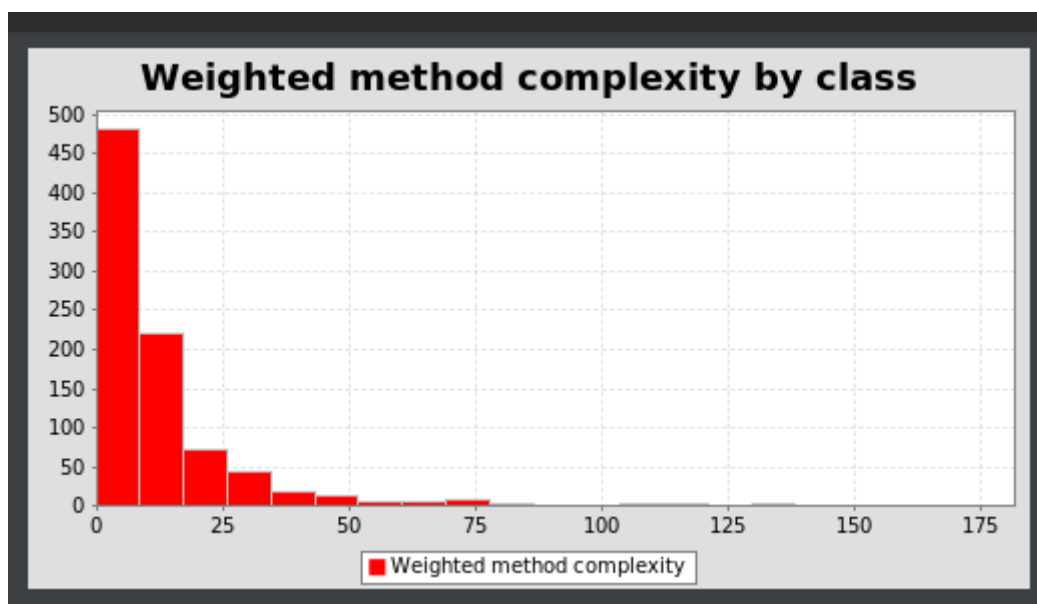


Figura 1: Histograma com número de classes em função do número de métodos implementados

É possível observar através do gráfico que existem cerca de 775 classes com valor inferior a 25. 10% deste número é igual a 77,5. Assim sendo, com o histograma, garantimos que existem mais de 10% de classes com valor WMC superior a 25. Isto significa que o projeto, num modo geral, encontra-se num estado onde existirá uma maior densidade de *bugs* e um decréscimo na qualidade.

DIT - *Depth Inheritance Tree*

Esta métrica apresenta a profundidade da árvore de herança de um projeto. Quanto maior o valor para uma classe, maior a probabilidade da mesma herdar mais métodos e mais variáveis, tornando-a mais complexa. Valores altos demonstram uma maior probabilidade de existirem falhas no projeto, especialmente nas classes que se encontram a meio na árvore. O valor recomendado é igual ou inferior a 5.

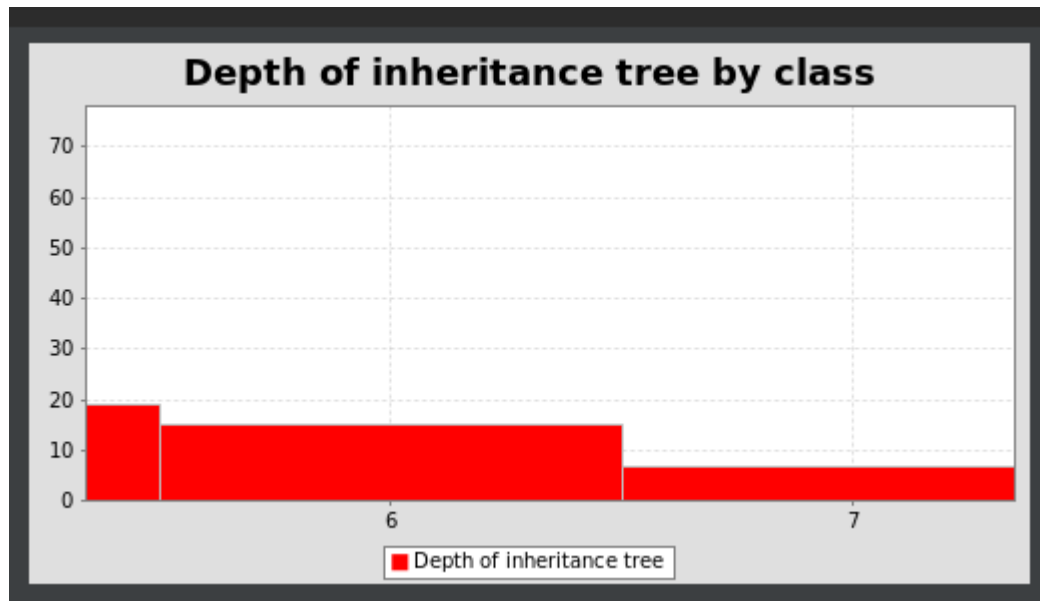


Figura 2: Histograma com número de classes em função da profundidade da árvore de herança

Com o seguinte gráfico é possível observar que existe ainda um valor relativamente alto desta métrica, indicando que provavelmente existiram falhas durante a execução.

NOC - *Number of Children*

Esta métrica indica o número de subclasses derivadas da estudada. Mede a largura da hierarquia de uma classe e pode indicar os seguintes fatores: alta reutilização da classe base (desejado); a classe base necessita de mais testes; abstração indevida da classe; criação errada de subclasses. Classes numa profundidade mais baixa devem ter um maior número de subclasses.

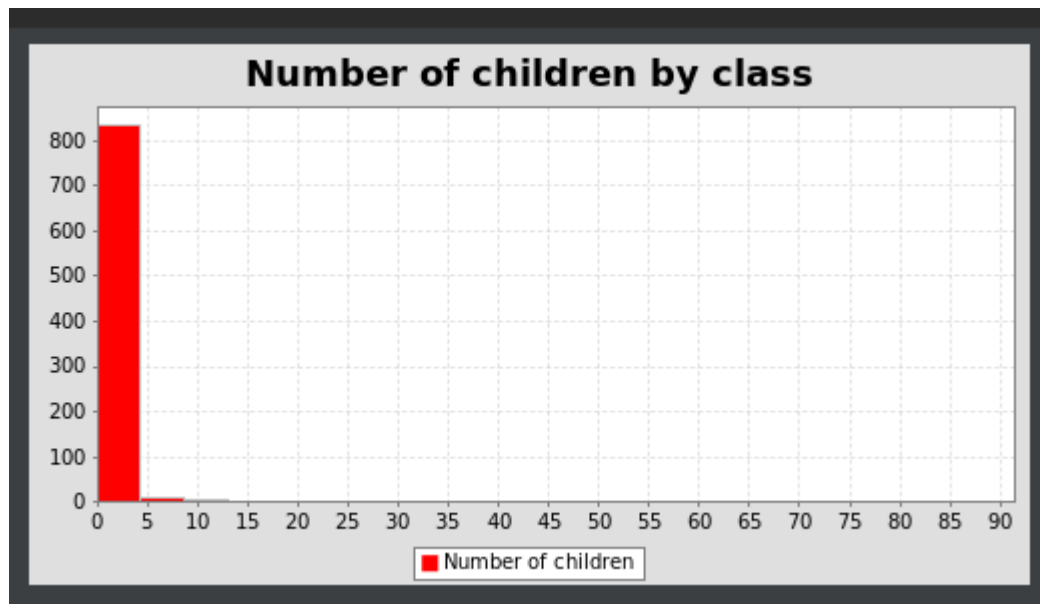


Figura 2: Histograma com número de classes em função do número de subclasses

Com este gráfico, é possível observar que regra geral o projeto não contém classes com muitas subclasses, podendo indicar pouca reutilização de código ao longo do projeto.

LCOM - *Lack of Cohesion Methods*

Esta métrica tem como intuito verificar se é necessário separar uma classe em duas ou mais classes. Caso o valor obtido seja igual a 0, significa que a classe é coesa, no caso de ser superior ou igual a 1, significa que a classe pode ser repartida em duas ou mais classes.

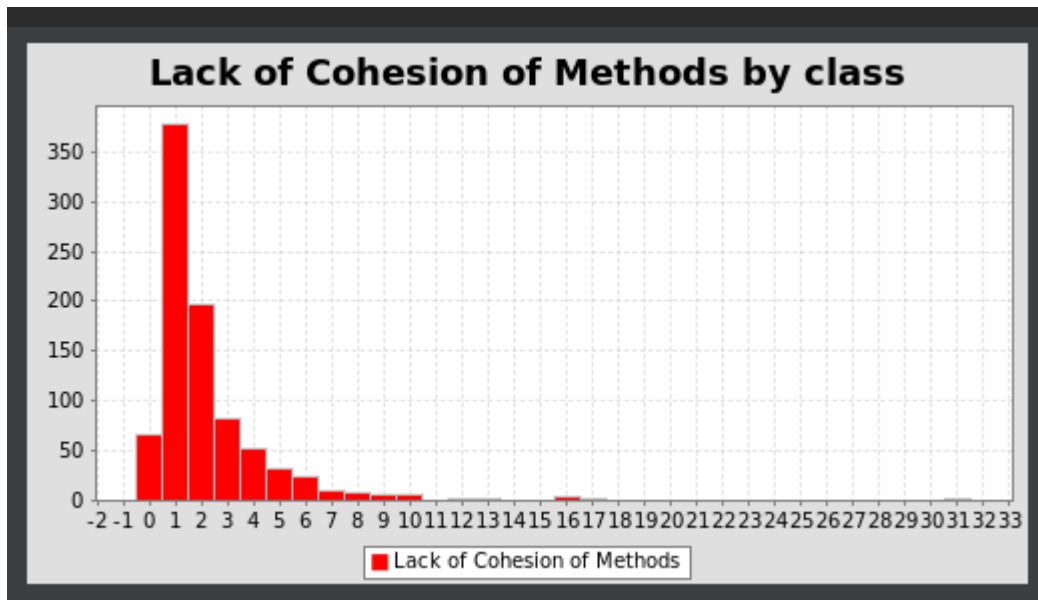


Figura 3: Histograma com número de classes em função da coesão das classes

Através do gráfico é possível observar que a grande maioria das classes no projeto não são coesas e poderiam ser repartidas em duas ou mais classes, demonstrando baixa qualidade no código.

CBO - *Coupling Between Objects*

Esta métrica é importante para avaliar a modularidade do projeto e serve para avaliar se duas classes estão acopladas, isto é, se uma classe usa métodos ou instâncias de variáveis criadas noutra classe. Um valor alto é indesejado, sendo um valor superior a 14 considerado elevado. Valores elevados significam que mudanças na classe que implementa os métodos e as variáveis são muito sensíveis e tornam o desenvolvimento e a testagem mais complicados. O número de falhas e bugs também aumentam.

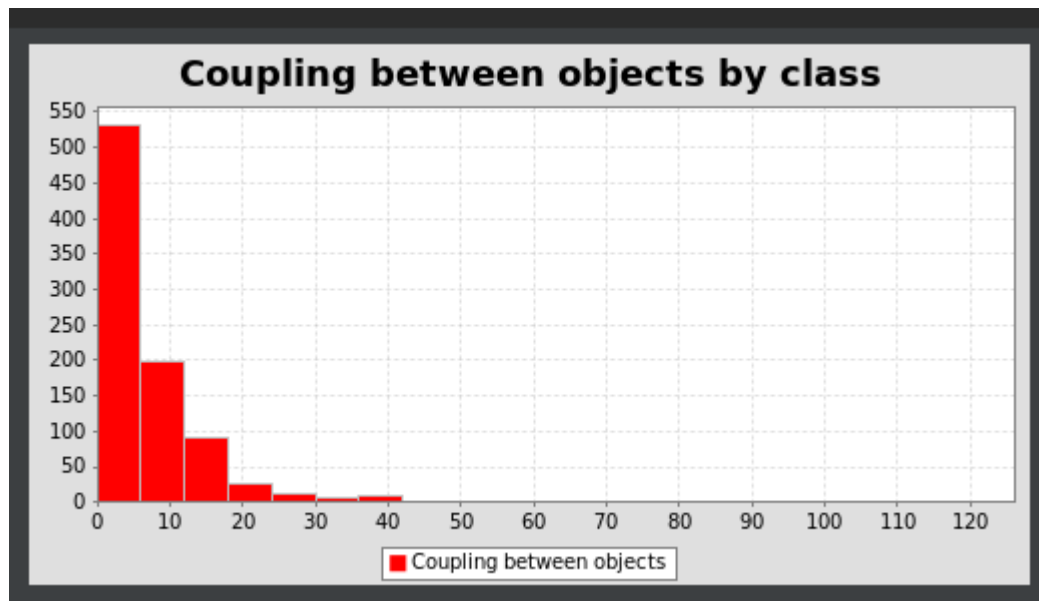


Figura 4: Histograma com número de classes em função do acoplamento de classes

Através do histograma é possível observar que existe um número suficientemente elevado de classes com elevado acoplamento. Existem classes com nível igual a 40, que é extremamente elevado, demonstrando falhas na estrutura do código.

RFC - *Response For a Class*

Esta métrica representa o conjunto de métodos que podem ser chamados numa resposta a uma mensagem recebida por um objeto. Um número elevado significa maior complexidade de manutenção e de testagem, assim como um maior número de falhas e dificuldade de compreensão.

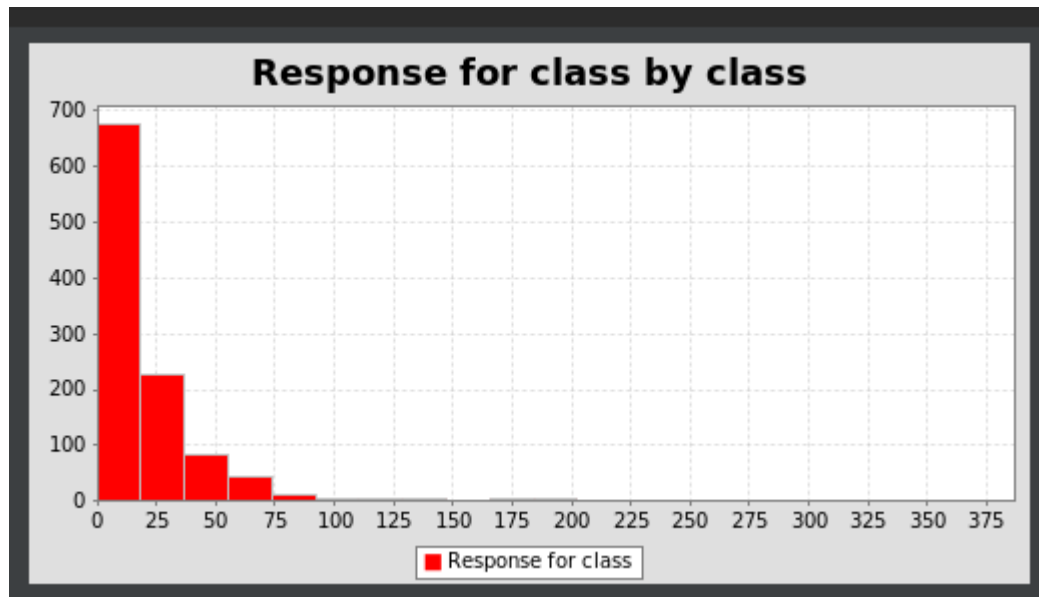


Figura 5: Histograma com número de classes em função do número de elementos no conjunto de métodos de resposta a uma mensagem

Através do histograma é possível observar que cerca de 40% de todo o projeto apresenta valores RFC superiores a 25, significando uma maior complexidade na manutenção e dificuldade da compreensão do código da estrutura da aplicação.

Relação com code smells encontrados

É possível relacionar as seguintes métricas com alguns code smells encontrados, especialmente com a métrica CBO. Com a CBO é possível relacionar o code smell *feature envy*, uma vez que existem várias classes a usar um número excessivo de métodos de outras classes.