

KAMILA SILVA VALENTINO

**CARACTERIZAÇÃO DA MODULARIZAÇÃO DE
SISTEMAS DE SOFTWARE OPEN SOURCE - UM
ESTUDO DE CASO NO DOMÍNIO JOGOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

LAVRAS
MINAS GERAIS - BRASIL
2014

KAMILA SILVA VALENTINO

**CARACTERIZAÇÃO DA MODULARIZAÇÃO DE
SISTEMAS DE SOFTWARE OPEN SOURCE - UM
ESTUDO DE CASO NO DOMÍNIO JOGOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

Área de Concentração:
Engenharia de Software

Orientador:
Prof. Dr. Heitor Augustus Xavier Costa

Coorientadora:
Bel. Mariana de Azevedo Santos

LAVRAS
MINAS GERAIS - BRASIL
2014

**Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca
Central da UFLA**

Valentino, Kamila Silva

Caracterização da Modularização de Sistemas de Software Open Source - Um Estudo de Caso no Domínio Jogos / Kamila Silva Valentino. Lavras - Minas Gerais, 2014. 64p.

Monografia de Graduação - Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Análise de Correlação. 2. Qualidade de Software. I. VALENTINO, K. S. II. Universidade Federal de Lavras. III. Caracterização da Modularização de Sistemas de Software Open Source - Um Estudo de Caso no Domínio Jogos.

KAMILA SILVA VALENTINO

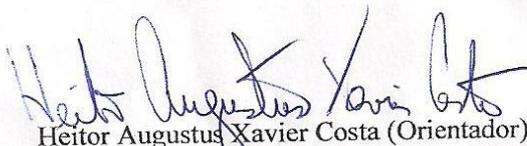
**CARACTERIZAÇÃO DA MODULARIZAÇÃO
ENTRE SISTEMAS DE SOFTWARE - UM
ESTUDO DE CASO DE PROJETOS OPEN
SOURCE DO DOMÍNIO JOGOS**

Monografia de graduação apresentada ao
Colegiado do Curso de Bacharelado em
Sistemas de Informação, para obtenção
do título de Bacharel.

APROVADA em 1 de julho de 2014.

Antônio Maria Pereira de Resende

André Pimenta Freire


Heitor Augustus Xavier Costa (Orientador)
Mariana de Azevedo Santos (Co-Orientador/a)

**LAVRAS-MG
2014**

AGRADECIMENTOS

A Deus por permitir que tudo isso acontecesse.

Ao meu orientador e coorientadora, pela paciência, dedicação e apoio durante o desenvolvimento desta monografia.

A esta universidade, aos professores e demais profissionais que foram tão importantes durante minha vida acadêmica.

Aos meus familiares e amigos, pelo incentivo e apoio em todos os momentos.

Caracterização da Modularização de Sistemas de Software Open Source - Um Estudo de Caso no Domínio Jogos

RESUMO

Os jogos possuem um enorme sucesso econômico e cultural, porém, ainda representam um campo empírico bastante novo. Eles também apresentam necessidades distintas dos demais sistemas, fazendo com que a indústria de Jogos enfrente o desafio de adotar boas práticas de Engenharia de Software para que os sistemas deste domínio sejam de alta qualidade. Com base neste contexto, o presente estudo tem por objetivo realizar uma caracterização da modularização destes sistemas. Para tal, foi realizada uma Análise de Correlação de *Spearman* utilizando seis medidas de acoplamento e três medidas de coesão reconhecidas na literatura de Engenharia de Software. Tais medidas foram coletadas a partir de 100 projetos *open source*, categorizados como *Games* pelo repositório *Sourceforge*. Como resultados, foram encontradas correlações significantes entre as medidas de acoplamento e duas medidas de coesão (LCOM e ILCOM), o que indica que estas medidas são relevantes para a avaliação destas propriedades em Jogos.

Palavras-chave: Qualidade de software, coesão, acoplamento, análise de correlação, jogos.

Characterization of Modularization of Open Source Software Systems - A Case Study in Games Domain

Abstract

Games have a huge economic and cultural success. However, they represent a fairly new empirical area. They also have different needs from other systems, making the games industry face a challenge in adopting good practices of Software Engineering for systems of this domain have high quality. Based on this context, the study aims to perform a characterization of the modularization of these systems. For such a Spearman Correlation Analysis using six measures of coupling and three measurements of cohesion recognized in the Software Engineering literature. Such measures were collected from 100 open source projects, categorized as games at Sourceforge repository. As a result, significant correlations between measures of coupling and two measures of cohesion (LCOM and ILCOM), were found indicating that these measures are relevant to the assessment these properties in games.

Keywords: Software quality, cohesion, coupling, correlation analysis, games.

SUMÁRIO

1. INTRODUÇÃO	1
1.1. Motivação	2
1.2. Objetivo	3
1.3. Metodologia de Desenvolvimento	3
1.3.1. Tipo de Pesquisa	3
1.3.2. Procedimentos Metodológicos	4
1.4. Estrutura do Trabalho	5
2. QUALIDADE DE SOFTWARE.....	6
2.1. Considerações Iniciais	6
2.2. Conceitos.....	6
2.3. Medidas de Software	10
2.3.1. Medidas de Chidamber & Kemerer (CK).....	12
2.3.2. Medidas de Li & Henry.....	13
2.3.3. Outras Medidas.....	13
2.4. Considerações Finais	15
3. MEDIDAS DE CORRELAÇÃO	16
3.1. Considerações Iniciais	16
3.2. Análise de Correlação.....	17
3.3. Análise de Correlação de Produto-Momento <i>Pearson</i> (Correlação Linear de <i>Pearson</i>)	18
3.4. Análise de Correlação de <i>Spearman</i>	22
3.5. Considerações Finais	23
4. TRABALHOS RELACIONADOS.....	25
5. ESTUDO DE CASO.....	27
5.1. Considerações Iniciais	27
5.2. Caracterização dos Sistemas.....	27
5.3. Coleta das Medidas.....	30
5.4. Análise Descritiva	36
5.4.1. Distribuição de Frequência.....	38
5.4.2. Análise Gráfica.....	41
5.5. Análise de Correlação.....	50
5.6. Discussão dos Resultados	53
5.6.1. Correlação Intrapropriedades.....	53
5.6.2. Correlação Interpropriedades.....	54
5.7. Considerações Finais	54
6. CONSIDERAÇÕES FINAIS	56
6.1. Conclusões	56
6.2. Contribuições	56

6.3. Limitações.....	56
6.4. Trabalhos Futuros	57
REFERÊNCIAS BIBLIOGRÁFICAS.....	58
APÊNDICE A	62
APÊNDICE B	63
APÊNDICE C	64

LISTA DE FIGURAS

Figura 2-1 - Atributos de Qualidade do Produto de Software (Fonte: [ISO/IEC 25010, 2010])	10
Figura 3-1 - Diagramas de Dispersão [Lira, 2004]	18
Figura 3-2 - Comparação entre o Coeficiente de Correlação de <i>Pearson</i> e o Diagrama de Dispersão [Larson, Farber; 2010]	19
Figura 5-1 - Janela de Visualização para Escolha do <i>Metrics View</i>	31
Figura 5-2 - Janela de Propriedades do Projeto	31
Figura 5-3 - Janela de Exibição de Resultados do <i>Metrics</i>	32
Figura 5-4 - Janela de Visualização do <i>VizzMaintenance</i>	32
Figura 5-5 - Seleção de Sistemas para ser Analisado pelo <i>VizzMaintenance</i>	33
Figura 5-6 - Resultados Calculados pelo <i>VizzMaintenance</i>	33
Figura 5-7 - Janela de Opções de Correlação Bivariável.....	36
Figura 5-8 - Gráfico <i>Boxplot</i> para a Medida CBO	41
Figura 5-9 - Gráfico <i>Boxplot</i> para a Medida DAC	42
Figura 5-10 - Gráfico <i>Boxplot</i> para a Medida MPC.....	42
Figura 5-11 - Gráfico <i>Boxplot</i> para a Medida RFC.....	43
Figura 5-12 - Gráfico <i>Boxplot</i> para a Medida CA	43
Figura 5-13 - Gráfico <i>Boxplot</i> para a Medida CE	44
Figura 5-14 - Gráfico <i>Boxplot</i> para a Medida ILCOM.....	44
Figura 5-15 - Gráfico <i>Boxplot</i> para a Medida LCOM	44
Figura 5-16 - Gráfico <i>Boxplot</i> para a Medida TCC	45
Figura 5-17 - Diagrama de Dispersão entre as Medidas CBO e DAC.....	46
Figura 5-18 - Diagrama de Dispersão entre as Medidas CBO e MPC	47
Figura 5-19 - Diagrama de Dispersão entre as Medidas CBO e CA.....	47
Figura 5-20 - Diagrama de Dispersão entre as Medidas CBO e CE.....	47
Figura 5-21 - Diagrama de Dispersão entre as Medidas DAC e CA.....	48
Figura 5-22 - Diagrama de Dispersão entre as Medidas DAC e CE.....	48
Figura 5-23 - Diagrama de Dispersão entre as Medidas MPC e RFC.....	49
Figura 5-24 - Diagrama de Dispersão entre as Medidas MPC e LCOM.....	49
Figura 5-25 - Diagrama de Dispersão entre as Medidas RFC e ILCOM.....	49
Figura 5-26 - Diagrama de Dispersão entre as Medidas RFC e LCOM	50

Figura 5-27 - Diagrama de Dispersão entre as Medidas CA e CE.....	50
Figura 5-28 - Diagrama de Dispersão entre as Medidas ILCOM e LCOM.....	51
Figura 5-29 - Janela de Correlação Bivariável	51

LISTA DE TABELAS

Tabela 3-1 - Significância Estatística do Coeficiente de Correlação de <i>Pearson</i> [Meyers, Binkley; 2007]	19
Tabela 3-2 - Gastos com Propaganda X Vendas [Larson, Farber; 2010]	20
Tabela 3-3 - Valor das Variáveis x e y (Correlação Linear de <i>Pearson</i>).....	20
Tabela 3-4 - Valor das Variáveis x e y (Correlação de <i>Spearman</i>).....	23
Tabela 5-1 - Sistemas Selecionados para o Estudo de Caso.....	28
Tabela 5-2 - Medidas Selecionadas para o Estudo de Caso.....	30
Tabela 5-3 - Medidas Calculadas pelo <i>Metrics</i> e pelo <i>VizzMaintenance</i>	34
Tabela 5-4 - Estatísticas Descritivas.....	36
Tabela 5-5 - Distribuição de Frequência da Medida CBO	38
Tabela 5-6 - Distribuição de Frequência da Medida DAC	38
Tabela 5-7 - Distribuição de Frequência da Medida MPC.....	39
Tabela 5-8 - Distribuição de Frequência da Medida RFC.....	39
Tabela 5-9 - Distribuição de Frequência da Medida CA	39
Tabela 5-10 - Distribuição de Frequência da Medida CE	40
Tabela 5-11 - Distribuição de Frequência da Medida ILCOM.....	40
Tabela 5-12 - Distribuição de Frequência da Medida LCOM	40
Tabela 5-13 - Distribuição de Frequência da Medida TCC	40
Tabela 5-14 - Resumo dos Resultados	45
Tabela 5-15 - Resultado da Análise de Correlação de <i>Spearman</i>	52
Tabela A-1 - Valores Críticos para o Coeficiente de Correlação de <i>Pearson</i> [Larson; Farber, 2010].....	62
Tabela B-1 - Valores Críticos para o teste t Coeficiente de Correlação de <i>Pearson</i> [Larson; Farber, 2010].....	63
Tabela C-1 - Valores Críticos para o Coeficiente de Correlação de <i>Spearman</i> [Larson; Farber, 2010].....	64

1. INTRODUÇÃO

Para controlar a qualidade do software e os atributos relacionados a ela, tais como confiabilidade, facilidade de manutenção e facilidade de utilização, é necessário medir até que ponto esses atributos são alcançados pelo software. Tais medições são valiosas na análise do produto acabado e como forma de orientar o processo de desenvolvimento para evitar resultados indesejáveis, tais como, atrasos e sistemas de software de difícil manutenção [Hitz, Montazeri; 1995]. A utilização de medidas de software pode ajudar a prever o progresso do desenvolvimento e a melhorar a qualidade do software [Mäkelä; Leppänen, 2009]. Dentre as medidas que podem ser utilizadas para medir a qualidade, estão as medidas de coesão e de acoplamento.

A estrutura de um sistema de software tem grande impacto na sua manutenção. Para facilitar a manutenção, os sistemas são normalmente organizados em módulos [Bavota *et al.*, 2010]. A modularidade é um atributo de qualidade interno e deve ser aplicada nos níveis de abstração, desde a especificação de requisitos ao nível de código executável [Brito e Abreu, Goulão; 2001]. A modularização possui benefícios gerenciais, pois permite que diferentes equipes trabalhem em cada módulo com pouca necessidade de comunicação, reduzindo o tempo de desenvolvimento. Além disso, aumenta a flexibilidade do software, pois é possível realizar alterações em um módulo sem a necessidade (minimamente) de alterar os outros e facilita a compreensão permitindo a análise individual dos módulos. Assim, o sistema pode ser mais bem projetado, pois é mais bem compreendido [Parnas, 1972].

Baixo acoplamento e alta coesão têm sido considerados importantes atributos para sistemas modularizados [Sarkar *et al.*, 2007]. Tais atributos podem ser capturados durante as fases do desenvolvimento e são utilizadas para avaliar a qualidade interna dos sistemas. Dentre os benefícios relatados na literatura, um sistema de baixo acoplamento e alta coesão é mais fácil de entender, de desenvolver, de testar e de manter [Chowdhury; Zulkernine, 2011]. Pode-se dizer que um módulo tem forte coesão se representar uma tarefa do domínio do problema e seus elementos contribuem para essa tarefa [Kabaili *et al.*, 2001]. Por outro lado, o acoplamento diz respeito ao nível de dependência entre os módulos de um sistema [Chowdhury; Zulkernine, 2011]. Alto acoplamento dificulta manutenção e

gerenciamento, pois qualquer mudança pode afetar as demais classes do sistema [Meyers; Binkley, 2007].

1.1. Motivação

Os jogos de computador alcançaram sucesso econômico e cultural, porém ainda representam um campo empírico novo [Aarseth, 2003]. Em 2009, o desenvolvimento de jogos representava bilhões de dólares na economia dos Estados Unidos (EUA) e as empresas do setor empregavam diretamente 24.000 pessoas, apresentando crescimento rápido [Bond, Beale; 2009]. Os sistemas de software *open source* estão tornando-se fenômeno cultural e econômico e a quantidade desses sistemas está crescendo. Uma quantidade significativa de sistemas de software comerciais está sendo liberada com licença de código aberto [Hippel, Krogh; 2003]. Porém, as comunidades de desenvolvimento de software livre não adotam prontamente os processos da Engenharia de Software [Scacchi, 2004].

Normalmente, o software de código aberto é desenvolvido por uma comunidade de programadores baseada na Internet. A participação é voluntária e o código fonte completo é disponibilizado para o público. A maioria dos direitos de propriedade é destinada ao público, incluindo o direito de usar, de redistribuir e de modificar o software gratuitamente. [Hars, Ou; 2001]. Em comunidades de software livre de jogos de computador, uma pessoa costuma desempenhar várias tarefas. Em geral, os desenvolvedores de software livre criam modificações para jogos de computador (modos de jogos) que possuem aceitação e geram receita de vendas substancial para o fornecedor exclusivo do jogo e, por vezes, uma participação nos lucros para os desenvolvedores [Scacchi, 2004].

Na Engenharia de Software, o desenvolvimento de jogos é único, na medida em que combina o trabalho das equipes abrangendo múltiplas disciplinas (e.g., arte, música e programação) [Kanode, Haddad; 2009]. Como os jogos se tornam cada vez mais complexos e as expectativas do consumidor mais altas, os desenvolvedores de jogos devem enfrentar o desafio da evolução dos métodos da Engenharia de Software para satisfazer as suas necessidades [Kanode, Haddad; 2009]. Esses métodos devem levar ao desenvolvimento de sistemas de software modularizados, com baixo nível de acoplamento e alta coesão, fazendo com que os jogos sejam fáceis de entender, de depurar e de manter. Além disso, o desenvolvimento desses sistemas permite que pessoas diferentes

desenvolvam módulos de um sistema com pouca necessidade de comunicação, o que facilita o desenvolvimento de sistemas *open source*.

1.2. Objetivo

Neste trabalho, o objetivo foi caracterizar sistemas de software do domínio Jogos quanto à sua modularização, visto que saber o quão modularizado é um jogo pode contribuir para desenvolvimento desses sistemas com boa manutenibilidade. Para tanto, foram extraídos valores de medidas de acoplamento e de coesão e utilizados na Análise de Correlação de *Spearman* que visa descobrir como e quão relacionadas são duas variáveis. Espera-se que, com a análise da modularização, possam ser encontradas correlações que mostrem a relevância das medidas estudadas para cálculo de acoplamento e de coesão em sistemas de software do domínio Jogos. Para atingir esse objetivo, foi necessário alcançar os seguintes objetivos específicos:

- Identificar sistemas de software *open source* do domínio Jogos, coletados no repositório de acesso gratuito *Sourceforge*, com avaliação dos usuários acima de 3,5 estrelas;
- Identificar quais são as medidas mais relevantes na caracterização da qualidade de software do domínio Jogos utilizando a Análise de Correlação.

1.3. Metodologia de Desenvolvimento

1.3.1. Tipo de Pesquisa

Quanto à natureza, este trabalho pode ser classificado como **pesquisa aplicada**, pois seu objetivo é permitir a aplicação prática do conhecimento obtido com a realização da Análise de Correlação, contribuindo para o desenvolvimento de sistemas com maior qualidade. Em relação aos objetivos, pode ser caracterizado como **pesquisa exploratória**, visto que se refere à caracterização da modularização de jogos, por meio da interpretação dos resultados da Análise de Correlação. Quanto à sua abordagem, este trabalho pode ser uma **pesquisa quantitativa**, pois os resultados foram obtidos por meio da extração de valores das medidas de software selecionadas. Em relação aos procedimentos, pode ser caracterizado como **pesquisa operacional**, pois os resultados foram obtidos por meio da aplicação de análises estatísticas. A coleta de dados é feita por meio de **observação**, tendo em vista que os dados iniciais foram analisados para extração de conhecimento.

1.3.2. Procedimentos Metodológicos

O presente trabalho foi desenvolvido por meio da coleta de 100 sistemas de software caracterizados como *Games* (Jogos) pelo *Sourceforge*, um repositório de sistemas de código aberto que disponibiliza mais de 324 mil projetos para mais de 46 milhões de usuários. O processo de busca consistiu em selecionar sistemas desenvolvidos em Java, que deveriam possuir classificação superior a 3,5 estrelas - classificação dos usuários do repositório em decorrência de alguma importância dos sistemas para eles, atribuindo valores de 0 a 5 estrelas. O código fonte deveria estar disponibilizado no repositório e ser compilável.

Além disso, foi realizada uma seleção das medidas de acoplamento e de coesão e das ferramentas utilizadas para o cálculo dessas medidas. Foram selecionadas nove medidas de acoplamento e de coesão, implementadas nos *plug-ins* para o Eclipse: *Metrics*¹ e *VizzMaintenance*². Os valores calculados para cada medida nos 100 sistemas selecionados foram utilizados como entrada para a Análise de Correlação de *Spearman*. Também, foram calculadas medidas de tamanho para cada um dos sistemas, utilizando o *Metrics*. Essas medidas foram utilizadas para mostrar que a amostra era diversificada em relação ao tamanho dos projetos.

Para realizar a Análise de Correlação, foi utilizado o software *IBM SPSS Statistics*, um software utilizado para análise estatística nas ciências humanas e sociais. Com os resultados dessa análise, foi possível avaliar a relação entre as medidas selecionadas para o estudo e definir quais são as medidas relevantes para a avaliação do acoplamento e da coesão de sistemas de software no domínio Jogos. Além disso, *IBM SPSS Statistics* foi utilizado para calcular a média e o desvio padrão das medidas, ambos utilizados para realizar Análise Descritiva da amostra. Nessa análise, foi possível definir a amplitude amostral e os valores de referência para as medidas.

Por fim, foram gerados os gráficos *boxplot* e diagramas de dispersão, utilizando *IBM SPSS Statistics*. Com a análise dos gráficos *boxplot*, foi possível visualizar os intervalos dos valores das medidas e identificar valores discrepantes (*outliers*) na amostra. Com a análise dos diagramas de dispersão, pôde ser visualizado o relacionamento entre as

¹ <http://metrics.sourceforge.net/>

² <http://marketplace.eclipse.org/content/vizzmaintenance>

medidas. Os resultados das análises foram discutidos com o objetivo de extrair conhecimento em relação à modularização dos sistemas analisados.

1.4. Estrutura do Trabalho

O restante deste trabalho está organizado da seguinte forma.

Breve descrição dos conceitos de qualidade de software e medidas de software é apresentada no Capítulo 2.

Conceitos de medidas de correlação são tratados no Capítulo 3.

Alguns trabalhos relacionados são brevemente descritos no Capítulo 4.

Os resultados obtidos no estudo de caso são apresentados no Capítulo 5.

Conclusões, contribuições, limitações existentes e sugestões de trabalhos futuros são apresentadas no Capítulo 6.

2. QUALIDADE DE SOFTWARE

2.1. Considerações Iniciais

Uma das metas na Engenharia de Software é produzir sistemas com qualidade. Técnicas e padrões de qualidade são utilizados no desenvolvimento de sistemas de software para atingir esta meta. O acoplamento e a coesão são dois fatores importantes para determinar a qualidade [Atole, Kale; 2007].

Os principais conceitos da qualidade de software são apresentados na Seção 3.2. As medidas para avaliar a qualidade e algumas suítes que definem medidas utilizadas neste trabalho são mostradas na Seção 3.3.

2.2. Conceitos

A qualidade deve ser definida e medida para que melhorias sejam alcançadas. No entanto, o termo qualidade é ambíguo e é comumente mal interpretado [Kan, 2002]. A qualidade pode ser definida como características dos produtos que satisfaçam as necessidades dos clientes; ou como a ausência de deficiências e de erros que geram retrabalho ou que resultam em falhas, insatisfação do cliente, reclamações, entre outros problemas [Juran, Godfrey, 2000]. No software, a qualidade do produto é geralmente reconhecida como a ausência de *bugs*. Além disso, é reconhecida como a conformidade com os requisitos, pois, se o software contém defeitos funcionais, o requisito básico de fornecer a função desejada ao cliente não é atendido [Kan, 2002]. Uma das definições de qualidade de software é [ISO/IEC 25010, 2010]

A qualidade de um sistema é o grau em que o sistema satisfaz as necessidades explícitas e implícitas das diferentes partes interessadas e, portanto, fornece valor.

O gerenciamento da qualidade pode ser estruturado em três atividades principais [Sommerville, 2011; Santiago, 2011]:

- **Garantia da qualidade.** Consiste em definir como a qualidade de sistemas de software pode ser atingida e como a organização de desenvolvimento sabe que o sistema possui o nível de qualidade necessário. Ela está relacionada à definição e à seleção de padrões a serem aplicados ao processo de desenvolvimento de sistemas. Os processos dessa

atividade garantem que os sistemas e os processos no ciclo de vida do projeto estejam em conformidade com seus requisitos de especificação, pelo planejamento, pelo ordenamento e pela execução de atividades para fornecer a confiança adequada de que a qualidade é incorporada ao sistema;

- **Planejamento da qualidade.** Consiste em desenvolver um plano para o projeto, que deve definir quais são os critérios desejados de qualidade para o projeto e a forma como a qualidade deve ser avaliada;
- **Controle da qualidade.** Consiste em monitorar o desenvolvimento de sistemas para assegurar que os procedimentos e os padrões de garantia da qualidade sejam seguidos.

A norma ISO/IEC 25000 categoriza a qualidade do sistema de software de acordo com oito características (Figura 2-1) [ISO/IEC 25010, 2010]:

- **Adequação Funcional:** Capacidade de o software atender as necessidades implícitas e explícitas, quando utilizado em situações específicas:
 - **Compleitude Funcional:** grau em que a funcionalidade do software realiza as tarefas especificadas alcançando os objetivos do utilizador;
 - **Conveniência Funcional:** capacidade da funcionalidade do software facilitar a realização de tarefas específicas e objetivas;
 - **Correção Funcional:** capacidade de o software fornecer resultados corretos com o grau de precisão necessário;
- **Eficiência de Desempenho:** Capacidade de o software atingir bom desempenho em relação à quantidade de recursos utilizados sob condições estabelecidas:
 - **Capacidade:** limite máximo em que um determinado parâmetro do software pode atender os requisitos;
 - **Utilização de Recursos:** quantidade de recursos utilizados pelo software para realizar as suas funções e satisfazer os requisitos;
 - **Comportamento Temporal:** tempo de processamento e de resposta consumidos pelo software para que suas funções atendam aos requisitos;
- **Compatibilidade:** Capacidade de o software trocar informações com outro software e/ou realizar suas funções necessárias, ao compartilhar mesmo hardware ou ambiente de software:
 - **Interoperabilidade:** grau em que dois ou mais sistemas de software podem trocar informações e utilizar as informações trocadas;

- **Coexistência:** Grau em que o software pode desempenhar as suas funções de forma eficiente ao compartilhar de um ambiente e recursos comuns com outro software, sem produzir impacto negativo em qualquer outro produto;
- **Usabilidade:** Grau ao qual um software pode ser utilizado por usuários específicos atingindo os objetivos especificados com eficácia, eficiência e satisfação em um contexto de uso específico:
 - **Operabilidade:** grau ao qual o software possui atributos que o tornam fácil de operar e controlar;
 - **Apreensibilidade:** grau ao qual o software pode ser utilizado por usuários específicos para atingir os objetivos especificados de aprendizagem, permitindo utilizar o software com eficácia, eficiência, sem risco e com satisfação em um determinado contexto de uso;
 - **Proteção Contra Erros do Usuário:** capacidade do software proteger os usuários de cometer erros;
 - **Acessibilidade:** grau ao qual o software pode ser utilizado por pessoas com a maior variedade de características e capacidades para atingir um objetivo específico em um determinado contexto de uso;
 - **Estética da Interface do Usuário:** grau em que a interface de usuário permite a interação de forma agradável e satisfatória;
 - **Reconhecimento:** grau em que os usuários podem reconhecer se o software é adequado para as suas necessidades;
- **Confiabilidade:** Capacidade de o software executar as funções necessárias em condições específicas por um período de tempo determinado:
 - **Maturidade:** grau em que o software satisfaz as necessidades de fiabilidade em operação normal;
 - **Tolerância a Falhas:** grau em que o software funciona como pretendido, apesar da presença de falhas de hardware ou software;
 - **Capacidade de Recuperação:** capacidade de recuperar dados diretamente afetados e restabelecer o estado desejado para o software caso ocorra uma interrupção ou falha;
 - **Disponibilidade:** grau em que o software está operacional e acessível quando for necessário utilizá-lo;

- **Segurança:** Grau ao qual o software protege a informação e os dados, de modo que pessoas ou outro software tenha o grau de acesso aos dados apropriado para seus tipos e níveis de autorização:
 - **Confidencialidade:** grau em que o software garante que os dados são acessíveis somente a usuários ou sistemas autorizados;
 - **Integridade:** grau em que o software impede o acesso ou modificação não autorizados de programas e dados;
 - **Autenticidade:** grau em que a identidade de um indivíduo ou recurso pode ser provada se reivindicado;
 - **Não Repúdio:** grau em que a ocorrência de ações ou eventos pode ser provada, de modo que os eventos ou ações não podem ser repudiados posteriormente;
 - **Responsabilidade:** grau em que as ações de uma entidade podem ser atribuídas exclusivamente à entidade;
- **Manutenibilidade:** Grau de eficácia e de eficiência com que o software pode ser modificado pelos mantenedores:
 - **Modularidade:** grau em que o software é composto de componentes discretos, de modo que a alteração em um componente tenha impacto mínimo sobre outro software;
 - **Reuso:** grau em que o software pode ser utilizado em mais de um sistema ou na construção de outro software;
 - **Modificabilidade:** grau ao qual o software pode ser modificado de forma eficaz e eficiente sem a introdução de defeitos ou degradação da qualidade do software existente;
 - **Testabilidade:** grau de eficácia e de eficiência com que se pode estabelecer critérios de teste para o software e com que os testes podem ser realizados para determinar se esses critérios foram cumpridos;
 - **Analísabilidade:** capacidade de avaliar o impacto de uma alteração no software, diagnosticar as deficiências ou causas de falhas de um software ou identificar partes a serem modificadas de forma eficiente e eficaz;
- **Portabilidade:** Grau de eficácia e de eficiência com o qual o software pode ser transferido para outro hardware, software ou ambiente operacional:
 - **Portabilidade:** grau ao qual o software pode ser substituído por outro software especificado para o mesmo fim;

- **Capacidade de Instalação:** grau de eficácia e de eficiência com que o software pode ser instalado/desinstalado com êxito em um ambiente especificado;
- **Adaptabilidade:** grau em que o software pode ser adaptado de forma eficiente e eficaz para um hardware, software ou ambiente operacional diferente.

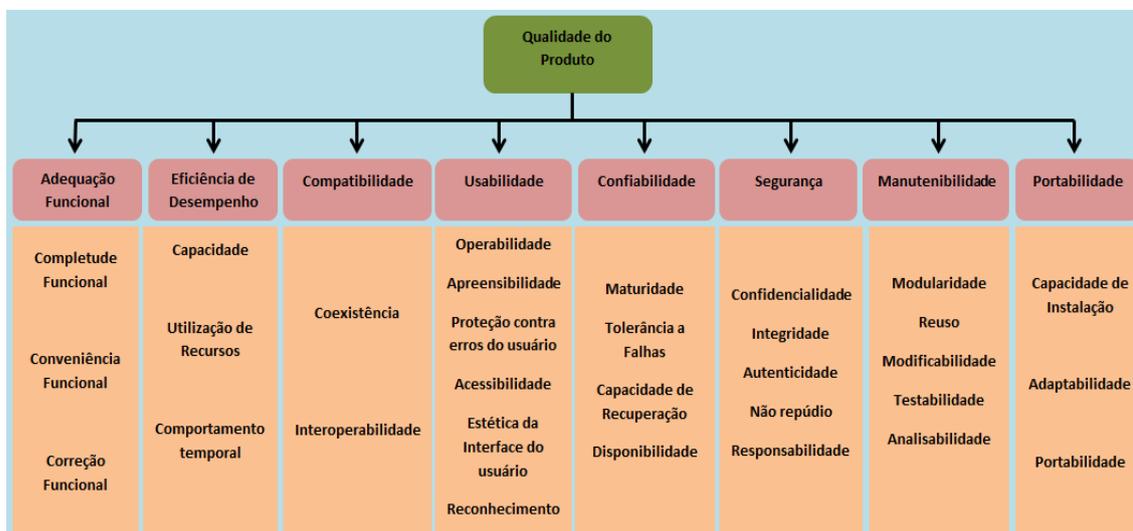


Figura 2-1 - Atributos de Qualidade do Produto de Software (Fonte: [ISO/IEC 25010, 2010])

A manutenibilidade é uma importante característica de qualidade de software, uma vez que a manutenção é a fase que consome mais recursos no ciclo de vida, pois envolve várias atividades para adaptar, migrar e evoluir o software de acordo com as novas exigências e necessidades [Fontana, Maggioni; 2011]. Ela pode ser definida, de uma maneira geral, como o nível de esforço necessário para modificar o software [Pereplechikov *et al.* 2007].

2.3. Medidas de Software

Durante o desenvolvimento de um software, o gerente de projeto deve ser capaz de responder algumas questões. Geralmente, essas questões envolvem previsões como quanto recurso será necessário para realizar o projeto do software ou quanto recurso será utilizado na manutenção [Ince, 1990]. Medir um software consiste em aferir alguma propriedade e avaliar a capacidade do software atingir um objetivo predefinido [Chowdhury; Zulkernine, 2011], tornando possível a previsão do progresso de desenvolvimento de um software e o auxílio para melhorar a sua qualidade [Mäkelä; Leppänen, 2009]. As medidas podem apoiar na melhoria da qualidade de software, por exemplo, por ajudar no gerenciamento do processo, nas tomadas de decisão e na descoberta de áreas problemáticas e de gargalos no processo de software. Além disso, podem ajudar os engenheiros de software a avaliar a

qualidade do seu trabalho para propósitos de garantia e para a melhoria de qualidade de processo em longo prazo [Santiago, 2011]. Assim, medidas de software podem ser organizadas em três categorias [Akingbejin, 2009]:

- **Medidas de produto.** Descrevem características do software, por exemplo, tamanho, complexidade, desempenho e qualidade;
- **Medidas de processo.** Envolvem o desenvolvimento de software e as atividades de manutenção, por exemplo, tempo de desenvolvimento, eficácia na remoção de defeitos e tempo para corrigir um erro;
- **Medidas de projeto.** Descrevem as características do projeto de software, por exemplo, recursos humanos, custo, tempo e produtividade.

Neste trabalho, foi realizada uma caracterização da modularização de Jogos. Portanto, foram utilizadas medidas, que visam avaliar a qualidade dos sistemas de software, relacionadas às propriedades de acoplamento e de coesão. A qualidade de um software está relacionada com a coesão e o acoplamento, sendo importantes propriedades para o reúso de software [Fontana, Maggioni; 2011] e medir a qualidade estrutural do software [Chowdhury; Zulkernine, 2011]. Sistemas com alto acoplamento e baixa coesão são mais difíceis de entender, de desenvolver, de testar e de manter, pois podem apresentar efeito colateral após manutenções, como a propagação de erros. O nível de acoplamento e de coesão é medido durante as fases do ciclo de vida de um software e utilizado para avaliar a sua qualidade.

Acoplamento refere-se ao nível de interligação e dependência entre os componentes do software. Componentes altamente acoplados são aqueles que dependem um do outro de tal forma, que uma mudança em um componente exige alterações nos seus dependentes. Além disso, entidades fortemente acopladas são difíceis de reutilizar e de entender de forma isolada, visto que as entidades dependentes também devem ser consideradas. Coesão refere-se ao grau em que uma determinada entidade fornece uma função única para o sistema de software como um todo. Componentes altamente coesos são aqueles que possuem apenas uma função. Esses componentes são mais desejáveis porque são mais fáceis de manter e de reutilizar.

No contexto de orientação a objetos, sistema de software é uma coleção de classes, sendo cada classe uma coleção de atributos e de métodos. Um sistema de software pode ser

considerado de boa qualidade se suas classes têm coesão máxima e acoplamento mínimo [Gupta; Chhabra, 2011]. Para isso, algumas medidas foram elaboradas, por exemplo, as medidas de Chidamber & Kemerer [Chidamber, Kemerer; 1991], as medidas de Li & Henry [Li, Henry; 1993], as medidas de Bieman & Kang [Bieman, Kang; 1995] e as medidas de Martin [Martin, 1994].

2.3.1. Medidas de Chidamber & Kemerer (CK)

Em um trabalho teórico [Chidamber, Kemerer; 1991], foi proposta uma suíte de medidas cujo foco é sistemas de software orientados a objetos. Nessa suíte, há seis medidas de nível de classe que medem diferentes propriedades estruturais relacionadas ao tamanho, à complexidade, ao acoplamento, à coesão e à herança. Essas medidas são [Elish *et al.*, 2011; Suresh *et al.*, 2012]:

- **Métodos Ponderados por Classe (*Weighted Methods per Class - WMC*)**. Soma da complexidade ciclomática dos métodos definidos em uma classe. Essa medida tem provado ser útil para prever os esforços de manutenção e teste;
- **Acoplamento entre Objetos (*Coupling Between Object - CBO*)**. Quantidade de classes com as quais uma classe é acoplada. É uma medida de acoplamento para determinar o quão complexo é o teste de várias partes do projeto;
- **Resposta para uma Classe (*Response For a Class - RFC*)**. Quantidade de métodos que podem ser invocados em resposta a uma mensagem enviada para um objeto. O valor de RFC para uma classe c é calculado da seguinte forma:

$$RFC(c) = |M(c) \cup R(c)|$$

sendo $M(c)$ o conjunto de métodos da classe c e $R(c)$ o conjunto de métodos chamados pelos métodos contidos na classe c ;

- **Ausência de Coesão em Métodos (*Lack of Cohesion in Methods - LCOM*)**. Quantidade de pares de métodos em uma classe sem utilizar os atributos em comum menos quantidade de pares de métodos que o fazem. Se essa diferença for negativa, LCOM é definida como zero. O valor de LCOM para uma classe c é calculado da seguinte forma:

$$LCOM(c) = |P(c)| - |Q(c)|$$

sendo $P(c)$ o conjunto de pares de métodos que não acessam atributos comuns e $Q(c)$ o conjunto de pares de métodos que acessam atributos comuns;

- **Profundidade da Árvore de Herança (*Depth of the Inheritance Tree - DIT*)**. Distância máxima de uma classe para a classe raiz na árvore de herança. Quanto mais profunda uma classe estiver na hierarquia, maior a possibilidade dela herdar grande quantidade de métodos tornando mais complexo para prever o seu comportamento;
- **Número de Filhos (*Number of Children - NOC*)**. Quantidade de subclasses imediatas de uma classe. É um indicador do potencial de influência que uma classe pode ter sobre o sistema.

2.3.2. Medidas de Li & Henry

Nessa suíte, são propostas cinco medidas para sistemas de software orientados a objetos [Li, Henry; 1993]:

- **Acoplamento por Abstração de Dados (*Data Abstraction Coupling - DAC*)**. Quantidade de tipos abstratos de dados definidos em uma classe e dependentes das definições de outras classes. Quanto mais tipos abstratos de dados uma classe tiver, maior é seu acoplamento com outras classes;
- **Acoplamento por Passagem de Mensagem (*Message Passing Coupling - MPC*)**. Quantidade de indicações de envio definidas em uma classe. É utilizada para medir a complexidade da passagem de mensagens entre classes. Uma vez que o padrão da mensagem é definido por uma classe e utilizado pelos seus objetos, a medida indica quantas mensagens são transmitidas entre os objetos das classes;
- **Número de Métodos (*Number of Methods - NOM*)**. Quantidade de métodos locais definidos em uma classe. Essa quantidade pode indicar a propriedade de operação de uma classe. Quanto mais métodos ela tem, mais complexa ela se torna;
- **SIZE**. Quantidade de pontos e vírgulas em uma classe;
- **SIZE2**. Quantidade de propriedades de uma classe, cujo cálculo é a soma da quantidade de atributos e de métodos locais de uma classe.

2.3.3. Outras Medidas

Na suíte de Bieman & Kang [Bieman, Kang; 1995], foram definidas duas medidas de coesão baseadas nas conexões diretas e indiretas dos pares de métodos:

- **Coesão Firme em Classe (*Tight Class Cohesion* - TCC).** Quantidade relativa de métodos diretamente conectados em uma classe. O valor de TCC para uma classe c é calculado da seguinte forma:

$$TCC(c) = \frac{NDC(c)}{NP(c)}$$

sendo $NDC(c)$ o conjunto de pares de métodos que acessam atributos comuns na classe e $NP(c)$ a quantidade máximo de conexões possíveis entre os métodos visíveis da classe. $NP(c)$ é calculado da seguinte forma:

$$NP(c) = \frac{|M(c)|(|M(c)| - 1)}{2}$$

sendo $M(c)$ a quantidade de métodos públicos da classe c ;

- **Coesão Solta em Classe (*Loose Class Cohesion* - LCC).** Quantidade relativa de métodos indiretamente conectados em uma classe. O valor de LCC para uma classe c é calculado da seguinte forma:

$$LCC(c) = \frac{NDC(c) + NIC(c)}{NP(c)}$$

sendo $NIC(c)$ o conjunto de pares de métodos que não acessam atributos comuns na classe.

Na suíte de Martin [Martin, 1994], é descrito um conjunto de três medidas que podem ser utilizadas para medir a qualidade de sistemas de software orientados a objetos em termos da interdependência entre os pacotes:

- **Acoplamento Aferente (*Afferent Coupling* - CA).** Quantidade de classes externas a um pacote que dependem de classes internas a esse pacote;
- **Acoplamento Eferente (*Efferent Coupling* - CE).** Quantidade de classes internas a um pacote que dependem de classes externas;
- **Instabilidade (*Instability* - I).** O cálculo é

$$I = \frac{CE}{CA + CE}$$

cujo resultado varia no intervalo $[0, 1]$. Se $I = 0$, o pacote é altamente estável; se $I = 1$, o pacote é altamente instável.

Há uma formulação alternativa para a medida LCOM [Hitz, Montazeri; 1996] que consiste na **Melhoria de Ausência de Coesão em Métodos (ILCOM)**, correspondendo à quantidade de componentes ligados em uma classe.

2.4. Considerações Finais

A qualidade de software é um dos focos principais da Engenharia de Software e vem sendo cada vez mais buscada pelos desenvolvedores. A utilização de boas práticas de Engenharia de Software tende a aumentar a qualidade do software e facilitar o processo de manutenção. No entanto, a qualidade é uma característica difícil de ser avaliada, visto que pode depender de diversas informações. As medidas de software são fatores decisivos para medir a qualidade de sistemas de software, sendo utilizadas para avaliar a complexidade do sistema, acompanhar o progresso e avaliar a eficácia do software [Suresh *et al.*, 2012].

Um dos propósitos de medidas de software é medir a qualidade dos sistemas. Os resultados podem ser utilizados para prever os custos de manutenção ou melhorar a qualidade do código [Mäkelä; Leppänen, 2009]. As medidas de acoplamento e de coesão podem ser utilizadas para avaliar propriedades de um sistema, tais como a modularização e a manutenibilidade. Portanto, compreender o relacionamento dessas medidas pode ajudar no desenvolvimento de sistemas mais modularizados e mais fáceis de manter.

3. MEDIDAS DE CORRELAÇÃO

3.1. Considerações Iniciais

Estatística é a ciência que coleta, organiza, analisa e interpreta dados para a tomada de decisões. O seu estudo possui duas ramificações [Larson, Farber, 2010]: Estatística Descritiva e Estatística Inferencial. Na Estatística Descritiva, são resumidas as principais características de um conjunto de dados por meio de tabelas, de gráficos e de resumos numéricos. Na Estatística Inferencial, são utilizadas técnicas e procedimentos que permitem dar ao pesquisador um grau de confiabilidade nas afirmações que faz sobre a população baseando-se em amostras [Guimarães, 2008].

Em um estudo estatístico, o tipo ou a natureza das variáveis podem determinar o método apropriado. As variáveis podem ser classificadas por tipo ou por nível de mensuração. O nível de mensuração determina quais cálculos estatísticos são significativos [Larson, Farber; 2010]. Em relação ao nível de mensuração, as variáveis podem ser classificadas em nominais, em ordinais, em intervalares e em racionais [Lira, 2004]. Há dois tipos de conjuntos de dados utilizados na estatística: população e amostra. Enquanto a população consiste em uma coleção dos resultados, das respostas, das medições ou das contagens de interesse do estudo, a amostra é uma parcela da população utilizada para posterior análise de dados [Larson, Farber; 2010]. Na amostragem, a finalidade é fazer generalizações sobre um grupo sem precisar examinar cada um de seus elementos [Stevenson, 1981].

Os testes estatísticos são regras de decisões, vinculadas a um fenômeno da população, que possibilitam avaliar, com o auxílio de uma amostra, se determinadas hipóteses podem ou não ser rejeitadas. O teste de hipótese é uma técnica da Estatística Inferencial que utiliza estatística amostral para testar a afirmação sobre o valor de um parâmetro populacional [Larson, Farber; 2010]. Uma hipótese estatística é uma afirmação ou conjectura sobre um ou mais parâmetros de uma população (ou populações) [Freund, 2006]. Ao analisar uma amostra, devem ser elaboradas a hipótese nula e a(s) hipótese(s) alternativa(s). A hipótese nula (H_0) é qualquer hipótese estabelecida prioritariamente a ser analisada; a hipótese alternativa (H_a) é a aceita quando a hipótese nula é rejeitada [Guimarães, 2008]. Um teste de hipótese pode ser de bicaudal ou unicaudal. Se H_a tiver um

sinal de \neq , então o teste é unicaudal; se H_a tiver sinal de $<$, então o teste é bicaudal à esquerda e se tiver sinal de $>$, então o teste é bicaudal à direita [Larson, Farber; 2010].

Quando se tenta inferir algo a respeito da população por meio de amostragem, existe uma possibilidade de a inferência estar errada. Quando H_0 é rejeitada, mas a hipótese é verdadeira, ocorre um erro do tipo I. Quando H_0 é aceita e a hipótese é falsa ocorre um erro do tipo II. A probabilidade de cometer um erro do tipo I é conhecida como nível de significância do teste, representado pela letra grega α . A probabilidade de cometer um erro do tipo II é representada pela letra grega β [Guimarães, 2008]. A Análise Estatística pode ser aplicada à coleta e à interpretação das informações [Freund, 2006]. Um tipo popular de análise estatística é a Análise de Correlação. Ela é uma ferramenta importante para as diferentes áreas do conhecimento, por ser um método estatístico amplamente utilizado para estudar o grau de relacionamento entre variáveis [Lira, 2004].

Conceitos básicos de Análise de Correlação são mostrados na Seção 4.2. A técnica de *Pearson* é brevemente tratada na Seção 4.3. A técnica de *Spearman* é brevemente tratada na Seção 4.4.

3.2. Análise de Correlação

A Análise de Correlação visa determinar o quanto estão relacionados os valores de duas variáveis. O seu resultado é o coeficiente de correlação, que quantifica o grau de relacionamento entre as variáveis [Stevenson, 1981]. Esse coeficiente é conhecido como medida de associação, de interdependência, de intercorrelação ou de relação entre as variáveis [Lira, 2004]. As técnicas associadas à Análise de Correlação são frequentemente aplicadas nas Ciências Sociais e do Comportamento, na Engenharia e nas Ciências Naturais, visto que, muitas vezes, é necessário descobrir o quanto uma variável interfere em outra. É importante conhecer os diversos métodos de Análise de Correlação para não utilizar a técnica errada [Guimarães, 2008].

O diagrama de dispersão representa os pares ordenados (x, y) no plano cartesiano. Esse gráfico pode ser utilizado para determinar se existe correlação entre duas variáveis. No diagrama, a variável x é medida sobre o eixo horizontal e a variável y é medida sobre o eixo vertical [Larson, Farber; 2010]. Na Figura 3-1, podem ser observados alguns tipos de correlação representados em diagramas de dispersão. No Gráfico 1, é

representada uma correlação linear positiva entre as variáveis x e y , ou seja, os valores de y crescem à medida que os valores da variável x aumentam. No Gráfico 2, é representada a ausência de correlação entre as variáveis, ou seja, não é possível afirmar nada em relação ao valor de y com base nos valores assumidos pela variável x e vice-versa. No Gráfico 3, é representada uma correlação linear negativa, os valores de y decrescem à medida que os valores da variável x aumentam. No Gráfico 4, é representada correlação não linear, ou seja, os valores de x e y estão relacionados, porém não é possível traçar uma reta entre os pontos que representam os pares ordenados no plano cartesiano.

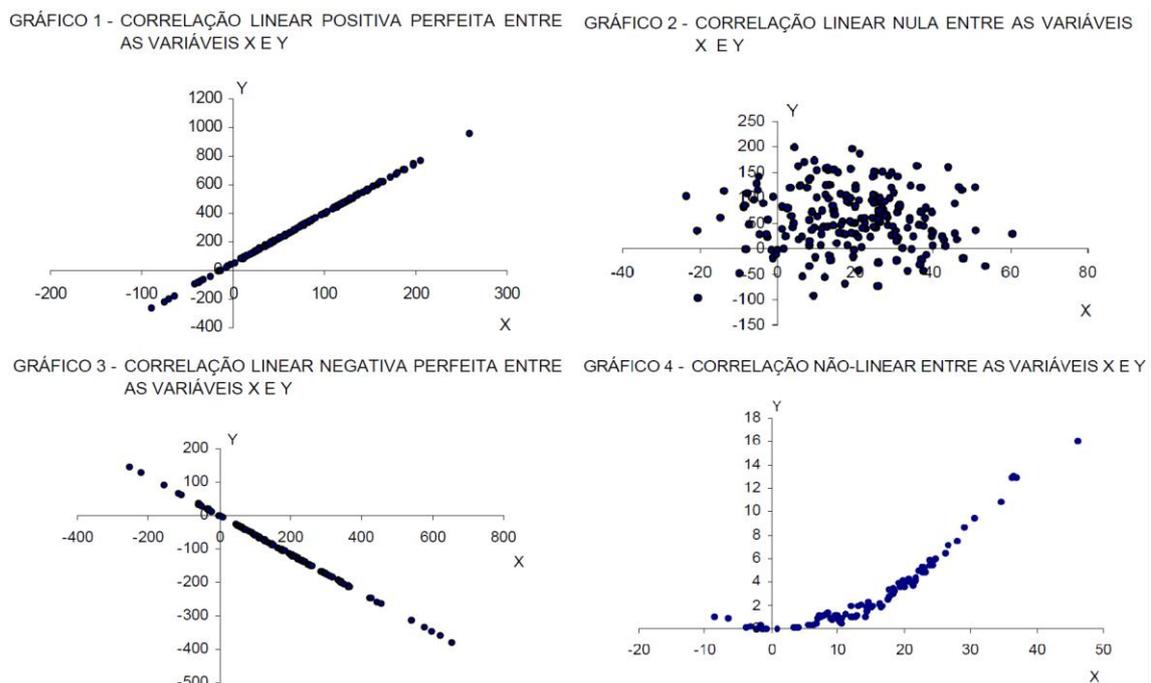


Figura 3-1 - Diagramas de Dispersão [Lira, 2004]

A técnica adequada de análise de correlação a ser utilizada pode ser definida com base no tipo de mensuração dos dados. Nas próximas seções, são apresentados dois tipos de análise de correlação: Análise de *Pearson* recomendada para variáveis intervalares e a Análise de *Spearman* utilizada para variáveis ordinais.

3.3. Análise de Correlação de Produto-Momento *Pearson* (Correlação Linear de *Pearson*)

O método para medir a correlação linear entre duas variáveis é o Coeficiente de Correlação Linear de *Pearson*, conhecido como Coeficiente de Correlação do Momento-Produto [Lira, 2004]. Com esse coeficiente, é medida a probabilidade da existência de uma relação linear entre dois conjuntos de dados. O seu valor varia no intervalo $[-1, 1]$, em

que um valor positivo indica relação linear com correlação positiva e vice-versa [Counsell *et al.*, 2006]. Uma versão conveniente da fórmula para o cálculo do coeficiente de correlação de *Pearson* é [Stevenson, 1981]:

$$r = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{n \sum x^2 - (\sum x)^2} \sqrt{n \sum y^2 - (\sum y)^2}}$$

sendo r o coeficiente de correlação amostral e n a quantidade de pares ordenados. Calcular o coeficiente é uma maneira mais precisa para medir o tipo e a força de uma correlação linear entre duas variáveis. Na Figura 3-2, podem ser observados uma comparação entre os valores obtidos de r e o diagrama de dispersão.

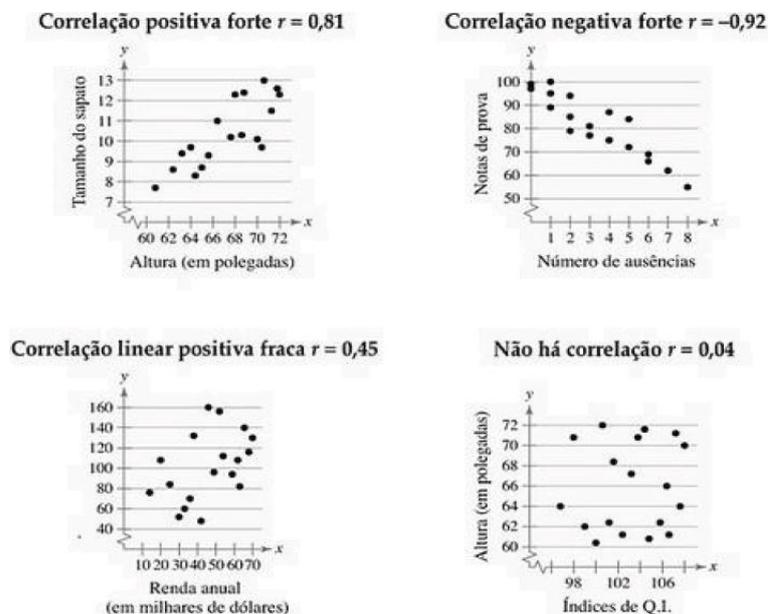


Figura 3-2 - Comparação entre o Coeficiente de Correlação de *Pearson* e o Diagrama de Dispersão [Larson, Farber; 2010]

A significância estatística de r pode ser resumida conforme apresentado na Tabela 3-1. Podem ser observados a relação entre os valores assumidos por r e o grau de correlação linear entre as variáveis. Quanto mais próximo do valor 1 estiver o valor absoluto de r , maior é o grau de correlação linear entre as variáveis.

Tabela 3-1 - Significância Estatística do Coeficiente de Correlação de *Pearson* [Meyers, Binkley; 2007]

Valor de r	Grau de correlação
]0.8, 1.0]	Correlação linear forte
]0.5, 0.8]	Correlação linear moderada
]0.0, 0.5]	Correlação linear fraca ou ausência de correlação

As suposições básicas para a utilização de r é o relacionamento entre as duas variáveis ser linear, que as variáveis envolvidas sejam aleatórias e medidas, no mínimo, em escala intervalar. Além disso, deve-se supor que as duas variáveis tenham distribuição normal bivariada conjunta, o que equivale a dizer que para cada x dado, a variável y é normalmente distribuída [Lira, 2004]. Para exemplificar o uso do coeficiente de correlação de *Pearson*, considera-se que um gerente de *marketing* queira conduzir um estudo para determinar se há uma relação entre o dinheiro gasto com propaganda e as vendas da empresa [Larson, Farber; 2010]. Os dados do estudo são mostrados na Tabela 3-2.

Tabela 3-2 - Gastos com Propaganda X Vendas [Larson, Farber; 2010]

Valor gasto em propaganda (milhares de R\$)	Vendas da empresa (milhares de R\$)
2,4	225
1,6	184
2,0	220
2,6	240
1,4	180
1,6	184
2,0	186
2,2	215

Considera-se que a variável x representa os gastos com propaganda e a variável y representa as vendas da empresa. Então, o valor de r é (Tabela 3-3):

$$r = \frac{8 \cdot (3.289,8) - (15,8)(1.634)}{\sqrt{8(32,44) - (15,8)^2} \sqrt{8(337.558) - (1.634)^2}} = \frac{501,2}{\sqrt{9,88} \sqrt{30.508}} \approx 0,9129$$

Tabela 3-3 - Valor das Variáveis x e y (Correlação Linear de *Pearson*)

x	y	x^2	y^2	$x \cdot y$
2,4	225	5,76	50.625	540
1,6	184	2,56	33.856	294,4
2,0	220	4,00	48.400	440
2,6	240	6,76	57.600	624
1,4	180	1,96	32.400	252
1,6	184	2,56	33.856	294,4
2,0	186	4,00	34.596	372
2,2	215	4,84	46.225	473
$\Sigma x = 15,8$	$\Sigma y = 1.634$	$\Sigma x^2 = 32,44$	$\Sigma y^2 = 337.558$	$\Sigma xy = 3.289,8$

O coeficiente apresentou valor positivo próximo do valor 1, portanto os dados possuem forte correlação linear positiva. Conclui-se que à medida que os valores investidos em propaganda aumentam, aumentam as vendas da empresa. Porém, esse resultado representa apenas o relacionamento dos dados amostrais. Quando se calcula um coeficiente de correlação com base em dados amostrais, o valor obtido para r é uma estimativa do coeficiente de correlação populacional ρ [Freund, 2006]. Após calcular o

valor de r , deve-se determinar se há evidência suficiente para decidir se ρ é significativo. Analisar os valores críticos de r é uma forma de determinar a significância de ρ . Para isso, é necessário determinar o nível de significância da análise [Larson, Farber; 2010].

Utilizando o exemplo anterior com $r = 0,9129$ e determinando um nível de significância de 1% ($\alpha = 0,01$), o valor absoluto de r é maior que o valor crítico apresentado na Tabela A-1 (Apêndice A) para esse tamanho de amostra e nível de significância $|r| > 0,834$, portanto a correlação populacional é significativa. Além disso, pode-se utilizar um teste de hipóteses para determinar se r fornece evidência suficiente para concluir que ρ é significativo. Para o exemplo anterior, é utilizado um teste bicaudal. As hipóteses para o teste são:

$$\begin{aligned} H_0: \rho &= 0 - \text{Não há correlação significativa} \\ H_a: \rho &\neq 0 - \text{Correlação significativa} \end{aligned}$$

Após a definição das hipóteses nula e alternativa, é calculada a estatística de teste para a amostra utilizando [Larson, Farber; 2010]:

$$t = \frac{r}{\sqrt{\frac{1-r^2}{n-2}}}$$

substituindo os dados do exemplo:

$$t = \frac{0,9129}{\sqrt{\frac{1-(0,9129)^2}{8-2}}} \approx 5,478$$

Em seguida, define-se o grau de liberdade do teste ($n - 2$). O exemplo possui 6 graus de liberdade ($8 - 2 = 6$). Além disso, deve-se definir o nível de significância do teste; para o exemplo, é utilizado nível de significância de 1% ($\alpha = 0,01$). Em seguida, compara-se o valor de t com os valores críticos contidos na Tabela B-1 (Apêndice B). Se o valor absoluto de t for maior que o valor crítico então H_0 é rejeitada ($|t| > 3,707$). Com esse valor, a hipótese nula foi rejeitada. Portanto, ρ foi considerado significativo para o exemplo em ambos os testes, utilizando nível de significância de 1%.

3.4. Análise de Correlação de Spearman

A Análise de Correlação de *Spearman* é uma técnica antiga e muito utilizada na estatística não paramétrica em áreas diversas, tais como Psicologia e Ciência Comportamental, Medicina, Biologia, Epidemiologia e Ciências Sociais [Borkowf, 2002]. Ela requer que as variáveis sejam medidas pelo menos em uma escala ordinal [Siegel; Castellan, 2006]. Essa técnica não tenta descobrir uma correlação linear entre as duas variáveis, mas apresenta um coeficiente de correlação, que indica se o aumento em uma variável é acompanhado pelo aumento na outra [Meyers; Binkley, 2007].

O coeficiente de Correlação de *Spearman* é o equivalente não paramétrico ao coeficiente de *Pearson*, sendo utilizado quando as suposições básicas do coeficiente de *Pearson* não são satisfeitas. A diferença é que o coeficiente de *Spearman* pode descrever relações lineares e não lineares e ser utilizado para dados no nível ordinal [Larson, Farber; 2010]. A presença de uma correlação de *Pearson* é normalmente acompanhada pela presença de uma correlação de *Spearman*. No entanto, é possível que os dados apresentem correlação *Spearman* forte e não apresentem correlação de *Pearson*, o que indica a existência de uma correlação não linear entre as variáveis [Meyers, Binkley; 2007]. A fórmula do coeficiente de correlação de *Spearman* é:

$$r_s = 1 - \frac{6 \sum d^2}{n(n^2 - 1)}$$

sendo n a quantidade de pares ordenados e d a diferença entre os valores de x e y [Freund, 2006]. Essa fórmula é obtida por meio do coeficiente de *Pearson* [Siegel, Castellan; 2006]. Para exemplificar, considera-se um grupo de estudantes na ordem de sua classificação no vestibular e de suas pontuações médias no final do último ano escolar. As variáveis x e y representam a classificação no vestibular e a média escolar, respectivamente. A correlação entre x e y é perfeita se as variáveis apresentassem valores iguais para todos os alunos considerados [Siegel, Castellan; 2006]. O valor de r_s é (Tabela 3-4):

$$r_s = 1 - \frac{6 \cdot (52)}{12 \cdot (144 - 1)} = 1 - \frac{312}{1.716} \approx 0,8182$$

Tabela 3-4 - Valor das Variáveis x e y (Correlação de Spearman)

Aluno	x	y	d	d^2
A	2	3	-1	1
B	6	4	2	4
C	5	2	3	9
D	1	1	0	0
E	10	8	2	4
F	9	11	-2	4
G	8	10	-2	4
H	3	6	-3	9
I	4	7	-3	9
J	12	12	0	0
K	7	5	2	4
L	11	9	2	4
				$\Sigma d^2 = 52$

O aluno I ficou em 4º lugar na classificação do vestibular, mas em 7º na classificação da média escolar; a diferença entre as classificações é $d = 4 - 7 = -3$. Para o aluno B, a diferença entre as classificações é $d = 6 - 4 = 2$. O somatório dessas diferenças pode "dar uma ideia" do quão próximo estão as duas variáveis, porém valores positivos e negativos se anulariam; por isso, é utilizado o quadrado da soma das diferenças para o cálculo do coeficiente. O valor de r_s é próximo do valor 1, portanto pode-se concluir que os dados amostrais são fortemente correlacionados. Existe relacionamento entre a classificação no vestibular e a classificação das médias do último ano escolar. Após calcular o valor de r_s , deve-se determinar se há evidência suficiente para decidir se o coeficiente populacional é significativo; para isso, são definidas as hipóteses nula e alternativa e o nível de significância do teste. Para o exemplo anterior, são considerados o valor $\alpha = 0,01$ e as hipóteses:

$H_0: \rho = 0$ - Não há correlação significativa

$H_a: \rho \neq 0$ - Correlação significativa

O valor absoluto de r_s deve ser maior que os valores críticos apresentados na Tabela C-1 (Apêndice C) para que a correlação seja significativa. Portanto, para o exemplo, H_0 deve ser rejeitada e o coeficiente populacional pode ser considerado significativo ($0,8182 > 0,780$).

3.5. Considerações Finais

As análises estatísticas são aplicadas em diversas áreas de conhecimento. Sua utilização é desde a coleta e organização de informações em gráficos, tabelas, entre outros,

a sofisticadas análises numéricas. A Análise de Correlação é um tipo popular de análise utilizada em diversas áreas do conhecimento. Em muitos ramos de pesquisa, devem ser realizados estudos que investiguem o grau de relacionamento entre variáveis. A Análise de Correlação permite esse tipo de investigação. Há diversos tipos de análises correlacionais que podem ser aplicadas aos variados tipos de dados.

A utilização de técnicas estatísticas para investigar relacionamentos entre medidas de software permite a extração de conhecimento a respeito de atributos de qualidade dos sistemas de software estudados. Esse conhecimento pode ser aplicado para permitir o desenvolvimento de sistemas com maior qualidade.

4. TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os trabalhos encontrados na literatura cujos temas assemelham-se ao tema deste estudo. No primeiro trabalho [Gui, Scott; 2007], são apresentadas novas medidas de acoplamento desenvolvidas para avaliar o reúso de componentes desenvolvidos em Java, recuperados da internet por um motor de busca desenvolvido pelo autor do trabalho. Além disso, é realizada uma comparação dessas medidas com seis medidas de acoplamento existentes (CBO, DAC, CF, MPC, RFC e RFC ∞) e uma Análise de Correlação de *Pearson* para avaliar a correlação entre as medidas de acoplamento e o esforço requerido pelos desenvolvedores em adaptar os componentes, na qual, todas as medidas apresentaram correlação significativa.

Em outro trabalho [Gui, Scott; 2009], foi apresentado um estudo semelhante, porém com medidas de coesão e de acoplamento. Foi realizada uma análise de correlação de *Spearman* para identificar o relacionamento entre as medidas e o esforço para adaptar componentes. Com os resultados, foi mostrado que as medidas são boas preditoras do esforço necessário para fazer modificações em componentes desenvolvidos em Java.

No terceiro trabalho [Counsell *et al.*, 2006], duas medidas de coesão orientadas a objeto (CAMC - *Cohesion Among Methods of Class* e NHD - *Normalized Hamming Distance*) passaram por uma análise matemática rigorosa para melhor compreendê-las e interpretá-las. Foi realizada uma Análise de Correlação de *Pearson* com o objetivo de avaliar a relação entre as duas medidas, que indicou que a medida NHD tem forte correlação negativa com a medida CAMC.

Em outro trabalho [Kabaili *et al.*, 2001], foi investigada a relação entre a coesão e a mutabilidade utilizando a Análise de Correlação de *Pearson*. As medidas LCC (*Loose Class Cohesion*) e LCOM - (*Lack of Cohesion in Methods*) foram adotadas para medir o impacto da mudança em três sistemas de tamanho industrial. Conclui-se que essas medidas não devem ser utilizadas como indicadores de mutabilidade.

Uma investigação empírica em larga escala de medidas *slice-based* foi realizada [Meyers; Binkley, 2004]. Nessa investigação, foi relatada uma comparação das medidas por meio da Análise de Correlação de *Pearson*, fornecendo melhor compreensão de suas

relações e indicando quais medidas fornecem informações semelhantes de um software e quais oferecem informações complementares.

Um estudo semelhante ao anterior foi realizado [Meyers; Binkley, 2007], porém utilizando medidas de coesão. Nesse estudo, foi incluída uma análise mostrando que as medidas *slice-based* não são substitutas para medidas simples, baseadas em tamanho, tais como quantidade de linhas de código. A comparação das medidas foi realizada por meio das análises de *Pearson* e *Spearman*.

Este trabalho diferencia-se dos anteriores ao propor a avaliação da modularização dos sistemas orientados a objeto desenvolvidos em Java no domínio Jogos. Quanto à aplicação das técnicas estatísticas, o trabalho segue a mesma linha dos trabalhos anteriores, utilizando a técnica de *Spearman* para analisar a correlação. A contribuição deste trabalho está no mapeamento das características internas de projetos de um domínio específico, não apenas com objetivos descritivos [Aarseth, 2003] e não abordado nos estudos relacionados.

5. ESTUDO DE CASO

5.1. Considerações Iniciais

Os jogos de computador alcançaram sucesso econômico e cultural, porém representam um campo empírico novo [Aarseth, 2003]. Em 2009, o desenvolvimento de jogos representava bilhões de dólares na economia dos EUA e as empresas do setor empregavam diretamente 24.000 pessoas, apresentando um crescimento rápido [Bond, Beale; 2009]. Na Engenharia de Software, o desenvolvimento de jogos é único, na medida em que combina o trabalho das equipes abrangendo múltiplas disciplinas (arte, música, programação, etc.) [Kanode, Haddad; 2009]. Como os jogos se tornam cada vez mais complexos e as expectativas do consumidor mais altas, a indústria de jogos deve enfrentar o desafio da evolução dos métodos da Engenharia de Software para satisfazer as suas necessidades [Kanode, Haddad; 2009].

As características que identificam um bom jogo são difíceis de definir e de reproduzir [Bond, Beale; 2009]. Por isso, neste trabalho, o objetivo foi caracterizar sistemas de software do domínio Jogos quanto à sua modularização. Para isso, as medidas de acoplamento e de coesão conhecidas na literatura foram utilizadas na perspectiva do domínio de jogos.

Caracterização dos sistemas de software utilizados no estudo de caso é mostrada na Seção 5.2. As medidas coletadas desses sistemas de software são apresentadas na Seção 5.3. A Análise Descritiva é tratada na Seção 5.4. A Análise de Correlação é abordada na Seção 5.5. Os resultados são discutidos na Seção 5.6.

5.2. Caracterização dos Sistemas

Os sistemas de software utilizados para a realização deste trabalho foram coletados no *Sourceforge*, no período de 01/03/14 a 30/04/14. O *Sourceforge* é uma ferramenta para o desenvolvimento e distribuição de software de código aberto, possui mais de 430 mil projetos e hospeda mais de 3,7 milhões de usuários registrados. Os sistemas do *Sourceforge* são distribuídos em nove domínios: i) *Audio & Video*; ii) *Business & Enterprise*; iii) *Communications*. iv) *Development*; v) *Home & Education*; vi) *Games*; vii) *Graphics, Science & Engineering*; viii) *Security & Utilities*; e ix) *System Administration*.

Os sistemas selecionados para o estudo foram do domínio *Games* (Jogos). Além da escolha do domínio, foram escolhidos sistemas de software com aceitação mínima de 3,5 estrelas (classificação de aceitação em um intervalo de [0, 5] estrelas atribuídas pelos usuários ao sistema), desenvolvidos em Java e o código fonte disponível e "compilável" no repositório. Com base nesses requisitos, foram escolhidos 100 sistemas de software descritos na Tabela 5-1 que também contém a quantidade de classes, de métodos, de pacotes, de atributos e de linhas de código (calculada pelo *Metrics*). Por exemplo, Braids, cujo identificador é 9, possui aceitação de cinco estrelas, 51 classes, 416 métodos, seis pacotes, 90 atributos e 3.703 linhas de código. Os sistemas escolhidos possuem entre 639 e 169.769 linhas de código e entre 7 e 914 classes o que comprova que a amostra utilizada no estudo é diversificada no que se refere ao tamanho dos sistemas.

Tabela 5-1 - Sistemas Selecionados para o Estudo de Caso

ID	Sistemas de Software	Aceitação	NC	NOM	NOP	NOA	LOC
1	4together	5	20	143	3	117	2.206
2	Ajax RPG	5	33	164	6	95	1.687
3	All-In Hon ModManager	5	162	776	19	458	12.529
4	AOJava	5	466	3.147	14	1.070	28.966
5	Battery	5	97	514	17	422	8.709
6	BehaviorSim	5	177	1.466	13	554	22.335
7	Blackjack	5	11	64	1	63	1.336
8	Blackspirit Graphics	5	106	851	14	405	10.831
9	Braids	5	51	416	6	90	3.703
10	BullsEye	5	13	62	1	74	1.127
11	Cardhouse	5	42	348	1	162	3.765
12	CarDriving	5	97	1.680	20	1.326	25.369
13	CarWars	5	43	187	7	133	3.233
14	Cluedo	5	91	814	11	510	14.348
15	Colossus	5	366	4.550	46	1.889	90.482
16	Coppit	5	17	256	3	114	3.957
17	Crazy Orb	5	35	176	6	113	2.109
18	Crossword Sage	5	34	238	1	158	3.087
19	Death Flight 1.0 Alpha	5	12	71	1	76	712
20	DragonChess	5	145	680	8	418	9.354
21	Earthdawn Character Editor	5	209	1.932	9	923	24.493
22	Echecs	5	37	147	8	103	1.332
23	Elite Comix Reader	5	12	86	1	48	1.602
24	Escoba de 15	5	19	180	2	65	1.826
25	F-4D-CB	5	18	483	6	220	4.354
26	Fabled Lands App	5	181	2.182	2	867	23.422
27	Flesh Snatcher	5	123	1.169	19	652	14.545
28	Fluffware	5	43	427	7	271	5.372
29	Four Row Solitaire	5	21	174	1	133	4.335
30	Frogma	5	57	609	1	639	10.083
31	GoGui	5	227	2.040	26	1.116	38.080
32	Hale	3,7	671	5.068	22	3.127	55.551
33	High-Card	5	26	226	7	89	2.156
34	Improve your English: Listen to the song	4	82	725	9	292	16.659
35	Item Generator	5	151	892	21	335	11.042
36	jagoclient	4,1	328	2.379	28	1.372	32.899
37	Jake2	5	336	1.253	15	1.966	85.592
38	jAMOS	5	61	1.263	7	622	18.399

Tabela 5-1 - Sistemas Selecionados para o Estudo de Caso (cont.)

ID	Sistemas de Software	Aceitação	NC	NOM	NOP	NOA	LOC
39	JaNaG - Java Name Generator	5	14	52	5	18	1.045
40	Java Ploy Boardgame	5	23	305	9	123	3.261
41	javamangband	5	31	414	1	168	5.366
42	jBubbleBreaker	4,1	35	198	3	79	2.104
43	jChecs	5	92	421	4	173	10.733
44	JDiveLog	4,8	57	677	7	360	7.022
45	jDoKo	5	16	32	4	43	971
46	Jin client for chess servers	4,8	353	3.108	51	880	33.871
47	JISO	5	29	144	9	91	1.257
48	Jmines	5	84	504	17	255	9.486
49	jMonopoly	5	32	250	5	113	3.286
50	Joaquim Brazilian Checkers Game	5	24	150	4	45	2.118
51	JPlanarity	5	10	105	5	30	866
52	JRemindX	5	13	32	1	18	848
53	jrpcore	5	23	71	6	44	639
54	Kenshira	5	308	3.111	28	891	42.640
55	Klaverjas Score	4,4	26	176	5	119	3.024
56	Legatus	4	602	7.543	70	3.502	95.381
57	maper	5	73	577	10	209	6.449
58	MazeRunner	5	78	388	8	597	9.779
59	mcSudoku	5	10	109	1	126	1.901
60	MegaAero	5	655	8.077	32	5.208	169.769
61	MekHQ	5	312	4.630	20	2.022	76.520
62	MetaChess	5	84	528	16	289	5.706
63	milero - a liero/molez clone	5	27	116	6	99	1.676
64	Mind Crackers	5	16	118	3	72	1.409
65	Monster-Tris	5	7	146	2	120	2.586
66	Muehle2D - Nine Men's Morris with AI	5	45	382	8	190	5.334
67	Nekodius	5	29	127	6	67	2.367
68	Neoecept	5	36	272	6	126	3.511
69	nullsaf	5	143	961	9	290	7.948
70	Nurikabe	5	44	392	3	331	6.696
71	onlinepoker	5	10	148	1	34	1.338
72	Open Kick-Off	5	52	300	11	134	4.923
73	Open Poker Tournament Director	5	25	177	3	109	1.305
74	OpenRocket	4,6	914	5.308	107	2.262	81.547
75	OpenRPG	5	118	778	20	268	7.578
76	Panzer Combat II	5	560	4.618	67	1.251	50.658
77	PetraGameFrame	5	13	131	4	45	1.498
78	Pine Wood Race Program	5	51	213	1	223	4.740
79	PJTimer - Mobile Speedcubing App	5	19	207	3	218	3.861
80	Pocket Cube J3D	5	21	127	5	79	2.446
81	PokerApp	5	99	546	3	377	14.170
82	Realtime Chaos	5	93	373	10	155	4.675
83	Rolemaster Office	4,9	195	1.445	20	533	18.640
84	Scrabble Dictionary	5	59	506	3	378	7.842
85	scrabbleScore	5	13	99	1	31	2.326
86	Space Butcher 2	5	81	447	14	192	6.022
87	Stigma - The Game	5	551	3.922	120	1.465	51.317
88	Strathclyde Poker Research Environment	5	90	669	16	419	7.691
89	Super Sliding Puzzle	5	28	120	3	66	1.456
90	TCG Warrior	5	118	789	1	456	23.310
91	The Duckpond (project #1)	5	114	688	20	251	6.511
92	The West Fortbattle Player	5	107	757	13	254	8.197
93	Tibetan Sho	5	12	82	3	45	856
94	Tico	4,9	371	1.742	33	931	31.499
95	Tiffanys	5	97	497	32	233	8.743
96	Tower of Zaldagor	5	41	407	11	257	6.946

Tabela 5-1 - Sistemas Selecionados para o Estudo de Caso (cont.)

ID	Sistemas de Software	Aceitação	NC	NOM	NOP	NOA	LOC
97	Two Player Chess	3,7	20	180	6	66	2.048
98	Ultima DeckMaker	5	45	229	14	97	2.426
99	Vtienlen	5	84	1.392	20	909	22.449
100	wizard	5	121	808	25	354	9.860

5.3. Coleta das Medidas

Neste trabalho, o objetivo foi caracterizar sistemas de software no domínio Jogos quanto à sua modularização por meio de medidas de coesão e de acoplamento. Assim, foram selecionadas as medidas listadas na Tabela 5-2. As medidas CA e CE foram aferidas utilizando a ferramenta *Metrics* e as demais foram aferidas utilizando a ferramenta *VizzMaintenance*.

Tabela 5-2 - Medidas Selecionadas para o Estudo de Caso

Sigla	Medida	Propriedade	Ferramenta
CA	<i>Afferent Coupling</i>	Acoplamento	<i>Metrics</i>
CBO	<i>Coupling Between Object Classes</i>	Acoplamento	<i>VizzMaintenance</i>
CE	<i>Efferent Coupling</i>	Acoplamento	<i>Metrics</i>
DAC	<i>Data Abstraction Coupling</i>	Acoplamento	<i>VizzMaintenance</i>
ILCOM	<i>Improvement of LCOM</i>	Coesão	<i>VizzMaintenance</i>
LCOM	<i>Lack of Cohesion in Methods</i>	Coesão	<i>VizzMaintenance</i>
MPC	<i>Message Passing Coupling</i>	Acoplamento	<i>VizzMaintenance</i>
RFC	<i>Response for a Class</i>	Acoplamento	<i>VizzMaintenance</i>
TCC	<i>Tight Class Cohesion</i>	Coesão	<i>VizzMaintenance</i>

O *Metrics* é um *plug-in* para o Eclipse IDE utilizado para calcular medidas de sistemas de software tradicionais e orientados a objeto, bem como analisar dependências entre pacotes de um sistema. Com ele, são calculadas várias medidas, detectados ciclos em pacotes e o tipo de dependência [Lincke *et al.*, 2008]. Para utilizá-lo, deve-se acessar o *menu* do Eclipse pelo caminho Window → Show View → Other. O acesso a esse *menu* permite abrir a janela apresentada na Figura 5-1. O resultado dos cálculos das medidas é exibido em uma janela ao acessar a pasta *Metrics*, selecionar a opção *Metrics View* e pressionar o botão OK.

Para calcular as medidas, o sistema deve ser selecionado com o botão direito do *mouse* e, em seguida, selecionar o *menu* *Properties*. Depois, deve ser selecionada a opção *Metrics* na janela de propriedades, marcar a opção *Enable Metrics* e selecionar o botão OK (Figura 5-2). O *plug-in* inicia o cálculo das medidas e, em seguida, os resultados são exibidos (Figura 5-3). Cada linha na janela de resultados representa uma medida. Para cada medida, é apresentado o valor calculado para o projeto, a média, o desvio padrão, o valor máximo atingido e a classe, o método ou o pacote que atingiu esse

valor. Ao selecionar uma linha da janela, os resultados de cada pacote, de cada classe ou de cada método do sistema são apresentados. Na Figura 5-3, são apresentados os valores aferidos para o sistema de software com ID 70. Por exemplo, na segunda linha da janela de resultados, pode-se observar que esse sistema possui 331 atributos, 7,523 atributos por classe em média e a classe `Help.java` possui 67 atributos, sendo a classe que possui a maior quantidade de atributos.

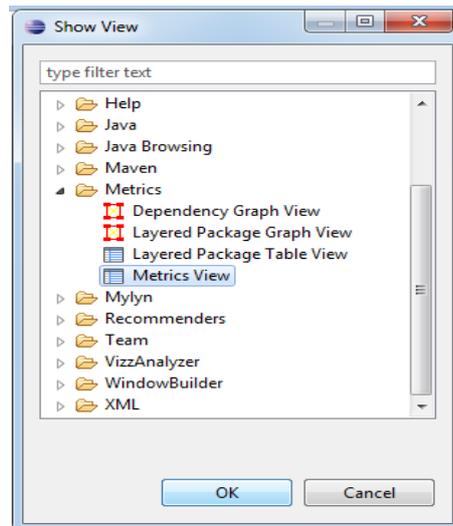


Figura 5-1 - Janela de Visualização para Escolha do *Metrics View*

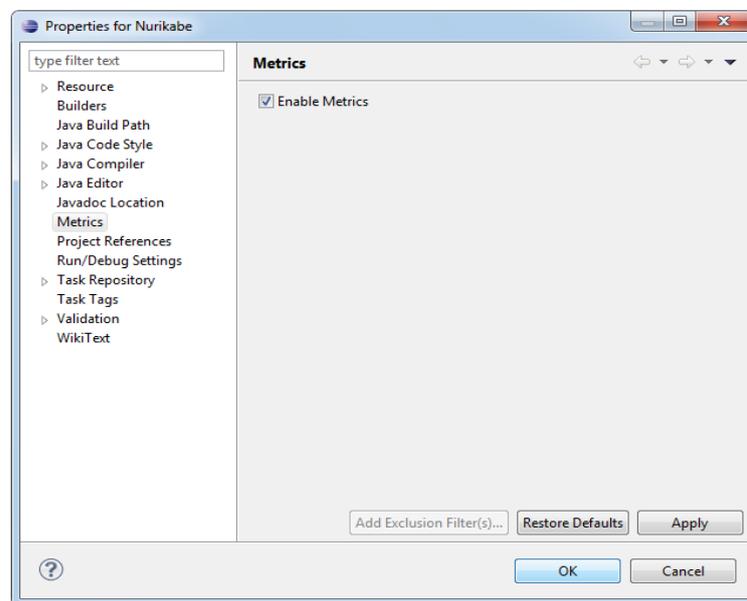


Figura 5-2 - Janela de Propriedades do Projeto

O *Metrics* foi utilizado para obtenção das medidas de tamanho (quantidade de classes, de pacotes, de métodos, de atributos e de linhas de código) e das medidas CA e CE. Na Figura 5-3, nota-se que o *plug-in* não apresenta valor total para algumas medidas, isso ocorre, pois algumas delas são calculadas para classes, métodos ou pacotes. Por isso,

neste trabalho, foi utilizada a média para representar o valor das medidas CA e CE para cada projeto, visto que são calculadas para pacotes.

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Number of Overridden Methods (avg/max per type)	6	0,136	0,404	2	/Nurikabe/src/nurikabe/FontChooser.java	
Number of Attributes (avg/max per type)	331	7,523	12,91	67	/Nurikabe/src/nurikabe/Help.java	
Number of Children (avg/max per type)	0	0	0	0	/Nurikabe/src/boardmvc/BoardModel.java	
Number of Classes (avg/max per packageFragn)	44	14,667	6,128	22	/Nurikabe/src/nurikabe	
Method Lines of Code (avg/max per method)	5081	12,896	29,115	367	/Nurikabe/src/nurikabe/Help.java	initComponents
Number of Methods (avg/max per type)	392	8,909	12,175	75	/Nurikabe/src/nurikabe/Nurikabe.java	
Nested Block Depth (avg/max per method)		1,904	1,257	7	/Nurikabe/src/rules/CellsNotAchievable.java	check
Depth of Inheritance Tree (avg/max per type)		2,477	2,039	6	/Nurikabe/src/nurikabe/Statistics.java	
Number of Packages	3					
Afferent Coupling (avg/max per packageFragn)		7,333	7,542	18	/Nurikabe/src/boardmvc	
Number of Interfaces (avg/max per packageFragn)	1	0,333	0,471	1	/Nurikabe/src/rules	
McCabe Cyclomatic Complexity (avg/max per method)		3,627	3,76	28	/Nurikabe/src/nurikabe/Nurikabe.java	checkMouseClicked
Total Lines of Code	6696					
Instability (avg/max per packageFragment)		0,614	0,335	0,882	/Nurikabe/src/rules	
Number of Parameters (avg/max per method)		0,799	0,805	5	/Nurikabe/src/nurikabe/Statistics.java	saveStatistics
Lack of Cohesion of Methods (avg/max per type)		0,363	0,333	0,992	/Nurikabe/src/nurikabe/FontChooser.java	
Efferent Coupling (avg/max per packageFragn)		9	4,899	15	/Nurikabe/src/rules	
Number of Static Methods (avg/max per type)	2	0,045	0,208	1	/Nurikabe/src/boardmvc/BoardView.java	
Normalized Distance (avg/max per packageFragn)		0,365	0,352	0,857	/Nurikabe/src/boardmvc	
Abstractness (avg/max per packageFragment)		0,021	0,029	0,062	/Nurikabe/src/rules	
Specialization Index (avg/max per type)		0,075	0,257	1,333	/Nurikabe/src/nurikabe/FontChooser.java	
Weighted methods per Class (avg/max per type)	1429	32,477	43,315	252	/Nurikabe/src/nurikabe/Nurikabe.java	
Number of Static Attributes (avg/max per type)	30	0,682	2,391	14	/Nurikabe/src/nurikabe/Nurikabe.java	

Figura 5-3 - Janela de Exibição de Resultados do Metrics

O *VizzMaintenance* é um *plug-in* para o *Eclipse* que fornece análises de medidas para avaliar a manutenibilidade de sistemas de software em Java e calcula 17 medidas. Para calcular essas medidas, deve ser acessado o *menu* do *Eclipse* pelo caminho *Window* → *Show View* → *Other* e escolhida a pasta *VizzAnalyser* e a opção *VizzMaintenance* e selecionado o botão *OK* (Figura 5-4).

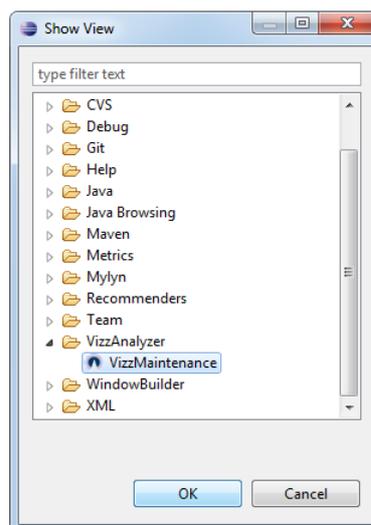


Figura 5-4 - Janela de Visualização do VizzMaintenance

Na janela de visualização aberta, deve ser escolhida a opção para calcular as medidas, representada por uma seta amarela no canto superior direito da janela. Em

seguida, deve ser selecionado o sistema a ser analisado e o botão Finish (Figura 5-5). O *plug-in* inicia o cálculo das medidas e o resultado é exibido (Figura 5-6).

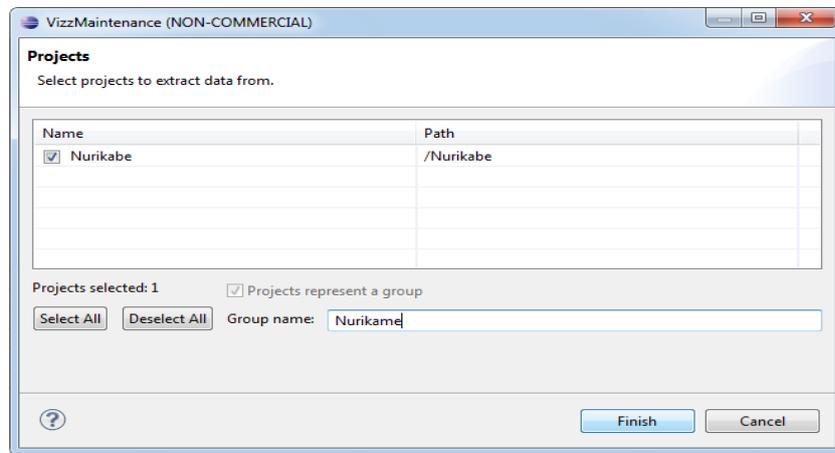


Figura 5-5 - Seleção de Sistemas para ser Analisado pelo VizzMaintenance

FullQualifiedName	name	Maintainab...	CBO	CYC Classes	DAC	DIT	ILCOM	LCOM	LD	LEN	LOC	LOD Class	MPC	NAM	NOC	NOM	RFC	TCC	WMC
nurikabe.Nurikabe.java.Nurikabe	Nurikabe	0,6554	20	20	20	0	1	3307	0,8919	8	2210	0,0132	103	143	0	75	130	0,1439	182
boardmvc.BoardView.java.BoardView.complet...	Completeoff	0,2540	3	20	3	0	1	0	0,1667	11	26	1,0000	7	2	0	1	8	0,0000	6
boardmvc.BoardView.java.BoardView.complet...	Completeon	0,2540	5	20	4	0	1	0	0,1250	10	26	1,0000	8	2	0	1	9	0,0000	6
boardmvc.CellView.java.CellView	CellView	0,2050	1	20	1	0	2	55	0,0000	8	119	0,0000	0	14	0	9	9	0,1111	11
nurikabe.properties.java.properties	properties	0,2050	1	20	1	0	0	1	0,0000	10	38	0,0000	0	1	0	1	1	0,0000	3
boardmvc.BoardModel.java.BoardModel	BoardModel	0,2050	19	20	19	0	2	0	0,5833	10	334	0,0000	64	21	0	14	32	0,5636	51
boardmvc.BoardView.java.BoardView.complet...	Complete	0,1857	2	20	2	0	1	0	1,0000	8	17	1,0000	2	2	0	1	3	0,0000	1
nurikabe.Nurikabe.java.Nurikabe.saveBoard.Save	Save	0,1857	1	20	0	0	1	0	1,0000	4	11	1,0000	3	2	0	1	4	0,0000	1
rules.Rules.java.Rules	Rules	0,1782	1	1	1	0	0	4	0,0000	5	22	0,0000	0	2	14	2	2	0,0000	2
nurikabe.FontChooser.java.FontChooser.Text...	TextFieldKey...	0,1366	0	1	0	0	1	0	1,0000	41	44	0,5000	0	2	0	1	1	0,0000	4
nurikabe.Help.java.Help	Help	0,1366	2	20	2	0	1	7	0,9853	4	634	0,3333	1	72	0	5	5	0,0000	13
nurikabe.SaveAsBoard.java.SaveAsBoard	SaveAsBoard	0,1366	1	20	1	0	1	0	0,5000	11	41	0,5000	1	2	0	1	2	0,0000	1
nurikabe.parserStatistics.java.parserStatistics	parserStatistics	0,1366	2	20	1	0	1	7	0,5000	16	79	0,2500	3	4	0	3	6	0,0000	8
boardmvc.Coordinate.java.Coordinate	Coordinate	0,1366	0	1	0	0	2	48	0,0000	10	78	0,0000	0	10	0	8	8	0,0714	8
nurikabe.FontChooser.java.FontChooser	IFontChooser	0,1366	6	5	6	0	11	546	2,3333	12	914	0,0000	14	49	0	26	26	0,1417	46
nurikabe.skin.java.skin	skin	0,1366	1	20	1	0	1	1	0,8571	4	163	0,2500	3	9	0	3	5	0,0000	6
nurikabe.Preferences.java.Preferences	Preferences	0,1366	1	20	1	0	1	1	0,9286	11	324	0,2500	12	16	0	3	15	0,0000	15
nurikabe.OpenBoard.java.OpenBoard	OpenBoard	0,1366	1	20	1	0	1	0	0,5000	9	40	0,5000	1	2	0	1	2	0,0000	1
nurikabe.Notifications.java.Notifications	Notifications	0,1366	2	20	2	0	1	24	0,8571	13	260	0,2308	8	24	0	12	19	0,1333	18
nurikabe.FontChooser.java.FontChooser.Dial...	DialogCance...	0,1366	1	5	0	0	1	0	0,2500	18	30	0,5000	1	3	0	1	1	0,0000	1
nurikabe.Player.java.Player	Player	0,1366	0	1	0	0	0	0	0,0000	6	27	0,0000	0	5	0	0	0	0,0000	0
nurikabe.FontChooser.java.FontChooser.Dial...	DialogOKAct...	0,1366	1	5	0	0	1	0	0,2500	14	30	0,5000	1	3	0	1	1	0,0000	1
boardmvc.BoardController.java.BoardController	BoardContro...	0,1366	6	20	6	0	2	0	0,3000	15	161	0,0000	18	11	0	8	18	0,4286	18
rules.DiferentBuildingConnected.java.DiferentB...	DiferentBuild...	0,1174	4	1	4	1	1	0	0,5000	25	109	0,0000	12	8	0	4	13	0,0000	17
rules.OneOnlyOutput.java.OneOnlyOutput	OneOnlyOut...	0,1174	4	1	4	1	1	0	0,3636	13	244	0,0000	31	12	0	8	17	0,0000	55
rules.WhiteNotConnected.java.WhiteNotConn...	WhiteNotCo...	0,1174	4	1	4	1	1	0	0,4444	17	125	0,0000	13	9	0	5	14	0,0000	22
rules.AllBlackConnected.java.AllBlackConnected	AllBlackCon...	0,1174	4	1	4	1	1	0	0,5000	17	126	0,0000	15	8	0	4	14	0,0000	23
rules.NumberOfWhiteCellsConnected.java.Nu...	NumberOfW...	0,1174	4	1	4	1	1	0	0,5000	27	139	0,0000	22	9	0	5	15	0,0000	23

Figura 5-6 - Resultados Calculados pelo VizzMaintenance

Cada linha na janela de resultados representa uma classe do projeto. Na Figura 5-6, são apresentados os resultados para o sistema de software com ID 70. Na primeira linha da janela, são exibidos os valores calculados para a classe `Nurikame.java`, do pacote `Nurikame`. As duas primeiras colunas exibem o caminho e o nome da classe, respectivamente; as demais colunas exibem os valores calculados para cada medida. O *VizzMaintenance* foi utilizado para o cálculo das medidas CBO, DAC, ILCOM, LCOM, MPC, RFC e TCC. As demais medidas exibidas na janela de resultados não foram consideradas por não se tratarem de medidas de acoplamento ou de coesão. Como são

exibidos apenas os resultados de cada classe, foi utilizada a média aritmética para definir o valor das medidas para os sistemas de software analisados.

Após a coleta dos sistemas no *Sourceforge*, foi calculado o valor das medidas de acoplamento e de coesão desses sistemas (Tabela 5-3). Por exemplo, o sistema de software com ID 16 possui CBO = 2,294118, DAC = 2,058824, MPC = 16, RFC = 23, CA = 1,667, CE = 1,667, ILCOM = 2,882353, LCOM = 176,4118 e TCC = 0,248430.

Tabela 5-3 - Medidas Calculadas pelo *Metrics* e pelo *VizzMaintenance*

ID	CBO	DAC	MPC	RFC	CA	CE	ILCOM	LCOM	TCC
1	3,263158	2,684211	6,684211	10,210530	7,000000	4,667000	1,210526	27,631580	0,317899
2	2,636364	2,363636	5,969697	8,939394	6,333000	3,500000	1,212121	13,787880	0,299138
3	2,818182	2,139394	6,587879	9,109091	6,105000	4,053000	1,006061	110,121200	0,138926
4	3,501066	3,253731	11,392320	10,479740	35,714000	30,000000	0,769723	326,537300	0,366902
5	5,819820	4,774775	7,558559	9,783784	11,176000	4,000000	0,783784	38,873870	0,167525
6	5,026178	3,910995	13,534030	14,858640	18,923000	8,231000	1,188482	126,905800	0,186613
7	2,727273	2,727273	4,545455	9,000000	0,000000	5,000000	1,181818	27,000000	0,219697
8	4,150000	3,664286	8,807143	12,435710	7,500000	4,643000	0,792857	90,385710	0,152485
9	3,943396	3,660377	7,716981	12,320750	9,333000	7,167000	0,603774	42,037740	0,190657
10	2,461538	2,153846	3,461538	5,923077	0,000000	8,000000	1,230769	11,461540	0,214286
11	2,302326	2,186047	4,906977	9,976744	0,000000	12,000000	1,139535	66,162790	0,166670
12	4,262626	3,929293	21,373740	25,676770	6,250000	3,050000	1,828283	476,939400	0,157701
13	2,386364	2,227273	5,340909	8,068182	2,857000	3,000000	1,431818	18,000000	0,226968
14	4,656250	4,135417	18,635420	16,270830	9,273000	5,000000	1,218750	73,614580	0,192648
15	6,104623	5,150852	20,090020	22,734790	13,957000	5,087000	1,481752	423,328500	0,160486
16	2,294118	2,058824	16,000000	23,000000	1,667000	1,667000	2,882353	176,411800	0,248430
17	3,315789	3,078947	5,894737	8,447368	2,167000	2,333000	0,921053	15,763160	0,265634
18	2,529412	1,794118	8,470588	11,794120	0,000000	0,000000	1,147059	48,147060	0,114202
19	1,666670	1,500000	4,916667	7,583333	0,000000	0,000000	0,833333	5,500000	0,408333
20	3,116883	2,805195	5,688312	6,701299	8,250000	7,625000	0,889610	40,249740	0,138348
21	5,587678	4,938389	17,151660	19,284360	13,000000	7,222000	3,118483	241,848300	0,273361
22	3,465116	3,255814	3,651163	4,953488	8,625000	4,250000	0,860465	11,302330	0,134201
23	2,461538	2,384615	7,307692	13,307690	0,000000	2,000000	0,692308	85,461540	0,000000
24	3,100000	2,600000	11,150000	14,800000	4,000000	4,500000	2,350000	93,900000	0,157684
25	1,222222	1,111111	17,388890	37,611110	2,167000	1,333000	2,333333	4256,056000	0,251135
26	6,395833	5,432292	21,598960	24,630210	1,000000	29,500000	1,343750	237,489600	0,246001
27	5,588235	5,125000	10,397060	16,242650	11,211000	5,526000	1,588235	155,132400	0,291873
28	2,187500	2,062500	6,270833	12,500000	3,714000	3,857000	1,229167	70,125000	0,332112
29	2,238095	2,095238	11,761900	14,142860	0,000000	2,000000	0,857143	36,857140	0,269281
30	3,879310	3,431034	11,465520	17,155170	0,000000	0,000000	1,810345	128,500000	0,254352
31	5,211538	4,861538	15,150000	17,019230	10,769000	4,962000	0,923077	313,500000	0,136384
32	7,077994	5,183844	14,160170	15,013930	52,773000	17,455000	1,228412	124,813400	0,216705
33	1,863636	1,545455	11,477270	22,090910	1,571000	1,714000	2,159091	1769,500000	0,240598
34	3,215517	3,112069	4,362069	9,379310	5,556000	2,889000	1,051724	66,077590	0,204951
35	5,439490	5,235669	6,668790	9,745223	13,714000	4,857000	0,789809	41,082800	0,151443
36	4,430636	4,049133	13,338150	11,471100	11,000000	5,250000	1,040462	94,369940	0,237555
37	7,929825	6,116959	14,514620	14,456140	44,667000	13,067000	1,087719	191,388900	0,104732
38	3,222222	2,603175	17,682540	32,809520	4,571000	3,429000	1,095238	1747,857000	0,215239
39	1,857143	1,785714	6,500000	7,285714	2,000000	1,600000	0,857143	12,500000	0,299943
40	3,960000	3,360000	13,160000	20,000000	3,444000	1,667000	2,080000	120,360000	0,203326
41	2,447368	2,157895	16,131580	17,289470	0,000000	0,000000	1,052632	346,289500	1,188839
42	2,891892	2,216216	6,783784	8,432432	3,667000	4,000000	1,486486	21,216220	0,070077
43	4,550000	3,250000	11,710000	9,740000	13,000000	18,000000	0,990000	19,300000	0,130179
44	2,619048	2,460317	9,507937	16,063490	6,286000	4,000000	2,666667	316,873000	0,270887
45	1,187500	1,125000	1,375000	3,625000	3,250000	2,250000	0,687500	9,312500	0,133333

Tabela 5-3 - Medidas Calculadas pelo Metrics e pelo VizzMaintenance (cont.)

ID	CBO	DAC	MPC	RFC	CA	CE	ILCOM	LCOM	TCC
46	4,606965	4,283582	11,771140	13,703980	8,941000	3,863000	1,097015	177,743800	0,149397
47	0,862069	0,862069	1,724138	5,482759	1,111000	0,778000	0,827586	31,137930	0,126432
48	5,117647	3,858824	18,129410	13,494120	12,882000	4,412000	1,423529	51,352940	0,128680
49	4,000000	3,735294	10,617650	12,529410	9,400000	4,800000	1,558824	58,470590	0,173135
50	3,678571	3,678571	8,821429	9,035714	3,750000	3,500000	1,071429	37,214290	0,097882
51	1,700000	1,700000	9,900000	16,100000	2,200000	1,800000	1,500000	82,100000	0,203962
52	2,000000	1,923077	3,153846	5,153846	0,000000	1,000000	0,923077	2,230769	0,358059
53	2,444444	2,407407	1,333333	3,111111	2,500000	2,000000	0,629630	4,925926	0,115844
54	3,997126	3,577586	11,853450	15,333300	9,179000	3,357000	1,068966	304,922400	0,131399
55	3,035714	2,857143	8,714286	11,214290	3,400000	2,000000	1,035714	93,000000	0,199837
56	6,336735	5,801749	14,253640	18,034990	20,529000	6,971000	2,109329	236,984000	0,162512
57	3,592105	3,302632	11,644740	15,710530	11,900000	4,300000	1,421053	152,657900	0,101908
58	2,282051	2,243590	3,833333	7,615385	1,500000	1,375000	0,794872	23,179490	0,131383
59	3,000000	2,666667	14,583330	17,416670	0,000000	3,000000	1,750000	104,000000	0,107242
60	8,242718	7,011096	36,979200	29,436890	21,656000	10,344000	1,858530	785,729500	0,158747
61	5,334347	4,465046	21,942250	28,109420	0,000000	0,000000	1,027356	690,674800	0,148025
62	3,648352	3,384615	8,714286	11,076920	9,375000	3,938000	1,032967	47,032970	0,283428
63	3,071429	3,000000	5,964286	7,178571	4,167000	3,333000	1,107143	22,250000	0,164447
64	1,750000	1,750000	3,187500	8,125000	1,667000	2,000000	1,062500	31,687500	0,384272
65	1,571429	1,571429	9,142857	30,714290	1,000000	1,500000	2,000000	814,714300	0,134704
66	4,189655	3,517241	10,810340	14,189660	9,125000	5,250000	1,172414	86,603450	0,214268
67	2,500000	2,218750	2,718750	6,781250	3,667000	2,667000	1,281250	65,000000	0,122343
68	3,702703	3,594595	5,864865	9,594595	5,000000	4,833000	1,864865	60,810810	0,293714
69	5,234177	4,829114	8,867089	10,449370	18,444000	10,444000	0,734177	62,569620	0,094988
70	3,180000	3,000000	13,860000	13,080000	7,333000	9,000000	1,380000	92,160000	0,258163
71	4,500000	4,300000	12,000000	20,100000	0,000000	5,000000	1,200000	136,600000	0,343499
72	3,126984	2,968254	6,333333	7,793651	6,364000	3,455000	0,984127	24,698410	0,205892
73	2,560000	2,120000	5,520000	8,560000	4,000000	4,667000	2,840000	204,560000	0,105417
74	6,171371	5,616936	10,777220	10,503020	19,860000	6,047000	1,003024	54,183470	0,160001
75	3,338235	3,073529	4,919118	8,988235	5,500000	2,500000	0,779412	27,066180	0,204843
76	6,085570	5,854027	8,179530	12,204700	13,597000	4,701000	1,055369	120,453000	0,151064
77	2,650000	2,500000	6,950000	11,850000	2,750000	2,750000	1,250000	94,800000	0,155662
78	2,884615	1,269231	3,807692	6,826923	0,000000	0,000000	0,634615	23,942310	0,113217
79	2,105263	1,947368	9,157895	15,368420	2,333000	2,000000	1,368421	125,421100	0,137922
80	2,952381	2,857143	5,714286	9,904762	4,000000	3,600000	1,047619	72,952380	0,109055
81	5,190909	2,781818	20,172730	12,390910	10,000000	9,667000	0,636364	32,863640	0,091268
82	5,118280	4,182796	6,483871	6,612903	17,800000	7,900000	0,548387	34,247310	0,087247
83	4,773399	4,290640	9,866995	12,034480	17,250000	7,800000	1,674877	258,911300	0,136726
84	3,285714	3,031746	9,142857	11,888890	8,000000	9,000000	1,333333	64,539680	0,167814
85	2,846154	2,461538	11,307690	13,000000	0,000000	0,000000	1,384615	44,461540	0,225367
86	3,785714	3,380952	6,833333	9,071429	6,357000	4,571000	1,345238	17,761900	0,190005
87	4,974359	4,580128	7,171474	10,290060	12,367000	3,458000	1,115385	105,937500	0,166232
88	1,901639	1,819672	6,491803	12,934430	5,188000	4,000000	1,622951	81,819670	0,118999
89	3,468750	3,281250	1,343750	4,687500	5,667000	5,667000	0,968750	15,564500	0,147297
90	6,553957	5,165468	14,201440	14,856120	0,000000	0,000000	0,892086	91,064750	0,148171
91	4,637931	3,948276	6,689655	9,362069	8,400000	3,000000	1,129310	56,077590	0,090285
92	3,460938	3,312500	5,820313	9,250000	7,462000	5,231000	0,914063	82,812500	0,151227
93	3,583333	2,166667	10,416670	12,833330	2,333000	2,333000	1,333333	33,500000	0,117569
94	4,721925	4,390374	8,048128	8,227273	26,545000	8,455000	1,106952	42,604280	0,147639
95	4,101010	3,989899	8,444444	9,131313	6,344000	2,500000	0,898990	44,959600	0,157013
96	4,209302	3,023256	15,255810	19,558140	6,273000	2,727000	3,534884	139,534900	0,336360
97	4,500000	4,500000	13,950000	13,950000	6,000000	2,500000	1,200000	55,100000	0,205892
98	2,530303	2,348485	2,454545	6,090909	5,214000	2,714000	1,015152	27,818180	0,142305
99	4,000000	3,913978	20,397850	25,645160	9,350000	3,500000	1,344086	1381,882000	0,109914
100	4,426357	4,178295	10,310080	11,193800	5,920000	2,880000	1,364341	43,914730	0,249593

5.4. Análise Descritiva

Primeiramente, foi analisada a amplitude do conjunto de dados. Ela representa a diferença entre as entradas máxima e mínima [Larson, Farber; 2010]. A amplitude da medida CBO é 7,380649, pois a maior valor apresentado para a medida foi 8,242718 e o menor valor foi 0,862069. A amplitude das medidas foi DAC = 6,149027, MPC = 35,64587, RFC = 34,5, CA = 52,773, CE = 30, ILCOM = 2,986497, LCOM = 4253,825231 e TCC = 1,188839. A medida TCC possui a menor amplitude do conjunto, o que significa que os seus valores variaram menos do que os valores das demais medidas. A medida LCOM possui a maior amplitude, sendo a medida cujos valores obtidos mais variaram.

Em seguida, foram analisados a média e o desvio padrão do conjunto de dados. A média é uma medida de tendência central [Freund, 2006] e o desvio padrão é a medida de quanto uma entrada desvia-se da média. Quando os valores são iguais o desvio padrão é zero; caso contrário, ele tende a ser grande [Larson, Farber; 2010]. O resultado do cálculo da média e do desvio padrão do conjunto de dados é apresentado na Tabela 5-4. Os dados contidos nessa tabela foram calculados pelo *IBM SPSS Statistics*, selecionando a opção Médias e desvio padrão (Figura 5-7).

Tabela 5-4 - Estatísticas Descritivas

Medida	Média	Desvio Padrão
CBO	3,70534002	1,486787294
DAC	3,26191770	1,256877445
MPC	9,94788431	5,697890012
RFC	13,10143657	6,469419684
CA	7,60090000	8,709444568
CE	4,83439000	4,864631430
ILCOM	1,28390560	0,554517039
LCOM	203,29178220	507,894888900
TCC	0,19666005	0,126149405

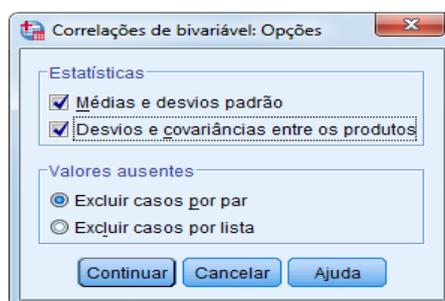


Figura 5-7 - Janela de Opções de Correlação Bivariável

Considerando a média e o desvio padrão da medida CBO, pode ser definido um intervalo de valores para essa medida somando e subtraindo o valor do desvio ao valor da média aritmética. O limite inferior do intervalo é $3,70534002 - 1,486787294 = 2,218552726$ e o limite superior é $3,7053400 + 1,486787294 = 5,192127314$. Considerando os valores da Tabela 5-3, pode-se perceber que, em 71% dos sistemas, os valores da medida CBO encontram-se dentro do intervalo calculado. A porcentagem obtida para a amostra de 100 sistemas de software permite induzir que a maioria dos sistemas no domínio Jogos tende a possuir valores para a medida CBO dentro do intervalo.

Para a medida DAC, o limite inferior do intervalo é igual a $2,005040255$ e o limite superior é $4,518795145$. Pode-se perceber que 70% dos sistemas possuem valor da medida DAC no intervalo, o que permite induzir que o valor da maioria dos sistemas no domínio Jogos tende a estar no intervalo. A medida MPC tem intervalo de valores $[4,249994298; 15,645774322]$ e a medida RFC $[6,632016886; 19,570856254]$. A porcentagem de projetos que apresentam valor para as medidas no intervalo é 74% e 78%, respectivamente. Também, permite induzir que a maioria dos sistemas no domínio Jogos tende a estar contida no intervalo. O intervalo para a medida CA é $[-1,108544568; 16,310344568]$, o de CE é $[-0,03024143; 9,69902143]$ e o de LCOM é $[-304,6031067; 711,1866711]$. Como o valor dessas medidas não pode ser negativo, o limite inferior para esses intervalos é alterado para o valor 0. Assim, 89%, 92% e 94% dos sistemas possuem valor no intervalo, respectivamente. Considerando as altas porcentagens apresentadas na amostra, pode-se induzir que a maioria dos sistemas no domínio Jogos tende a apresentar valor dentro no intervalo. Por fim, as medidas ILCOM e TCC possuem intervalos $[0,729388561; 1,838422639]$ e $[0,070510645; 0,322809455]$, respectivamente. Considerando a amostra, tem-se que 80% e 90% dos sistemas possuem valor para as medidas ILCOM e TCC no intervalo, respectivamente. Assim, pode-se induzir que a maioria dos sistemas no domínio Jogos tende a possuir o valor dessas medidas no intervalo.

Além disso, pode-se perceber, pelo desvio padrão, que os valores das medidas TCC e ILCOM variam pouco, pois são próximos de zero. Por outro lado, os valores da medida

LCOM são os que mais variam no conjunto de dados, como havia sido observado pela análise da amplitude.

5.4.1. Distribuição de Frequência

A distribuição de frequência consiste no agrupamento ou classificação dos dados em um número de categorias ou classes, ela facilita a visualização dos dados apresentando-os de forma compacta [Freund, 2006]. Neste estudo, as medidas foram classificadas com base em intervalos de valores. Na Tabela 5-5, é apresentada a distribuição de frequência para a medida CBO. Na coluna f , é apresentada a frequência simples para cada intervalo de valores, ou seja, a quantidade de sistemas de software que apresentou valores para a medida CBO dentro de cada intervalo. Por exemplo, na terceira linha da tabela, pode ser visualizado que 22 sistemas de software apresentaram valores da medida CBO no intervalo $[0,5; 2,5[$. A coluna fr representa a frequência relativa, porcentagem de sistemas de software que possui valores dentro de cada intervalo, e a coluna fa representa a frequência acumulada, soma das frequências desde o início da distribuição.

Tabela 5-5 - Distribuição de Frequência da Medida CBO

Medida CBO			
Valores	f	fr	fa
$[0,5; 2,5[$	22	22%	22
$[2,5; 4,5[$	49	49%	71
$[4,5; 6,5[$	25	25%	96
$[6,5; 8,5[$	4	4%	100
Total	100	100%	

A distribuição de frequência da medida DAC é apresentada na Tabela 5-6, pode ser observado que o intervalo de valores em que a maioria dos sistemas de software está contida (47%) é $[2,0; 3,5[$. Para a medida MPC, a maioria dos sistemas de software (38%) está contida no intervalo $[1,0; 7,0[$ (Tabela 5-7). Na Tabela 5-8, é apresentada a distribuição de frequência da medida RFC, na qual pode ser observado que 41% dos sistemas de software possuem valores da medida RFC dentro do intervalo $[10,0; 17,0[$. A distribuição de frequência da medida CA é apresentada na Tabela 5-9, em que pode ser observado que 32% dos sistemas de software possuem valores para essa medida dentro do intervalo $[0,0; 3,0[$.

Tabela 5-6 - Distribuição de Frequência da Medida DAC

Medida DAC			
Valores	f	fr	fa
$[0,5; 2,0[$	14	14%	14

Tabela 5-6 - Distribuição de Frequência da Medida DAC (cont.)

Medida DAC			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[2,0; 3,5[47	47%	61
[3,5; 5,0[28	28%	89
[5,0; 7,5]	11	11%	100
Total	100	100%	

Tabela 5-7 - Distribuição de Frequência da Medida MPC

Medida MPC			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[1,0; 7,0[38	38%	38
[7,0; 13,0[36	36%	74
[13,0; 19,0[19	19%	93
[19,0; 25,0[6	6%	99
[25,0; 31,0[0	0%	99
[31,0; 37,0]	1	1%	100
Total	100	100%	

Tabela 5-8 - Distribuição de Frequência da Medida RFC

Medida RFC			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[3,0; 10,0[39	39%	39
[10,0; 17,0[41	41%	80
[17,0; 24,0[12	12%	92
[24,0; 31,0[6	6%	98
[31,0; 38,0]	2	2%	100
Total	100	100%	

Tabela 5-9 - Distribuição de Frequência da Medida CA

Medida CA			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[0,0; 3,0[32	32%	32
[3,0; 6,0[19	19%	51
[6,0; 9,0[18	18%	69
[9,0; 12,0[13	13%	82
[12,0; 15,0[7	7%	89
[15,0; 18,0[2	2%	91
[18,0; 21,0[4	4%	95
[21,0; 24,0[1	1%	96
[24,0; 53,0]	4	4%	100
Total	100	100%	

A maioria dos sistemas de software (43%) apresentou valores para a medida CE no intervalo $[3,0; 6,0[$, conforme distribuição de frequência apresentada na Tabela 5-10. Na Tabela 5-11, é apresentada a distribuição de frequência de ILCOM, em que pode ser observado que metade dos sistemas de software possui valores para a medida ILCOM dentro do intervalo $[1,0; 1,5[$. A distribuição de frequência da medida LCOM é apresentada na Tabela 5-12, em que pode ser observado que 39% dos sistemas de software apresentaram valores para a medida dentro do intervalo $[0,0; 50,0[$. Em relação à medida TCC, a maioria dos projetos apresentou valores no intervalo $[0,1; 0,15[$ conforme distribuição de frequência apresentada na Tabela 5-13.

Tabela 5-10 - Distribuição de Frequência da Medida CE

Medida CE			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[0,0; 3,0[36	36%	36
[3,0; 6,0[43	43%	79
[6,0; 9,0[10	10%	89
[9,0; 12,0[5	5%	94
[12,0; 15,0[2	2%	96
[15,0; 18,0[1	1%	97
[18,0; 21,0[1	1%	98
[21,0; 30,0]	2	2%	100
Total	100	100%	

Tabela 5-11 - Distribuição de Frequência da Medida ILCOM

Medida ILCOM			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[0,5; 1,0[29	29%	29
[1,0; 1,5[50	50%	79
[1,5; 2,0[10	10%	89
[2,0; 2,5[6	6%	95
[2,5; 3,0[3	3%	98
[3,0; 3,5[1	1%	99
[3,5; 4,0]	1	1%	100
Total	100	100%	

Tabela 5-12 - Distribuição de Frequência da Medida LCOM

Medida LCOM			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[0,0; 50,0[39	39%	39
[50,0; 100,0[26	26%	65
[100,0; 150,0[11	11%	76
[150,0; 200,0[5	5%	81
[200,0; 250,0[4	4%	85
[250,0; 300,0[1	1%	86
[300,0; 350,0[5	5%	91
[350,0; 400,0[0	0%	91
[400,0; 650,0[2	2%	93
[650,0; 900,0[3	3%	96
[900,0; 2900,0[3	3%	99
[2900,0; 4900,0]	1	1%	100
Total	100	100%	

Tabela 5-13 - Distribuição de Frequência da Medida TCC

Medida TCC			
Valores	<i>f</i>	<i>fr</i>	<i>fa</i>
[0,0; 0,05[1	1%	1
[0,05; 0,1[6	6%	7
[0,1; 0,15[31	31%	38
[0,15; 0,2[23	23%	61
[0,2; 0,25[18	18%	79
[0,25; 0,3[12	12%	91
[0,3; 0,6[8	8%	99
[0,6; 0,9[0	0%	99
[0,9; 1,2]	1	1%	100
Total	100	100%	

5.4.2. Análise Gráfica

Em gráficos *boxplot*, o retângulo representa o intervalo de valores assumidos por 50% dos dados analisados. Nesse retângulo, a linha central representa o valor da mediana da amostra. Os valores máximo e mínimo para os dados são representados pelas linhas verticais traçadas acima e abaixo do retângulo e *outliers* na amostra podem ser observados. Um *outlier* é uma medida discrepante, ou seja, é uma entrada de dados "muito" afastada das demais entradas do conjunto [Larson, Farber; 2010].

Na Figura 5-8, a mediana da medida CBO é 3,46875. Cerca de 50% dos valores obtidos estão no intervalo [2,53; 4,61]. O valor máximo e mínimo é 7,075 e 0,862069, respectivamente. Na amostra estudada, foram identificados dois *outliers*, representados pelos sistemas de software com ID 37 e ID 60. Na Figura 5-9, é apresentado o gráfico *boxplot* para a medida DAC, cuja mediana é 3,112069. O valor máximo e mínimo é 6,115 e 0,862069, respectivamente, sendo que 50% dos valores estão no intervalo [2,21875; 4,049133]. Foi identificado um *outlier* na amostra representado pelo sistema de software com ID 60.

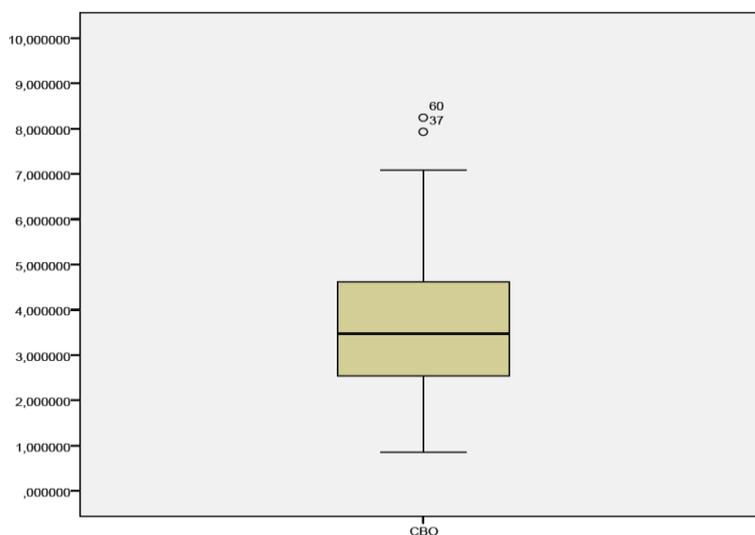


Figura 5-8 - Gráfico *Boxplot* para a Medida CBO

O gráfico *boxplot* para a medida MPC é apresentado na Figura 5-10. Dentre os valores para essa medida, 50% ficaram contidos no intervalo [7,178571; 13,16] e o valor máximo e mínimo é 22,1321435 e 1,333333, respectivamente. A mediana é 8,867089 e o sistema de software com ID 60 é o único *outlier*. Na Figura 5-11, é apresentado o gráfico *boxplot* para a medida RFC, cuja mediana é igual a 11,88889.

Essa medida possui valor máximo e mínimo igual a 24,938697 e 3,111111, respectivamente, sendo que 50% dos valores calculados estão no intervalo [8,988235; 15,368420]. Foram identificados sete *outliers* referentes aos sistemas de software com ID 12, ID 25, ID 38, ID 60, ID 61, ID 65 e ID 99.

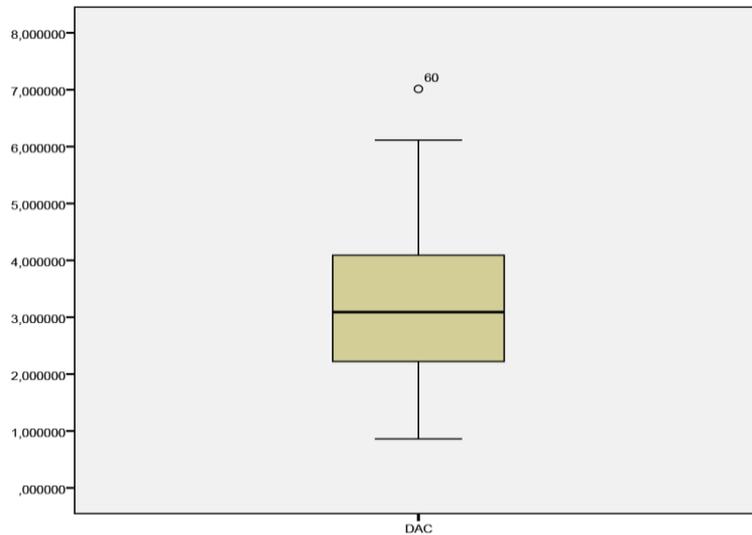


Figura 5-9 - Gráfico *Boxplot* para a Medida DAC

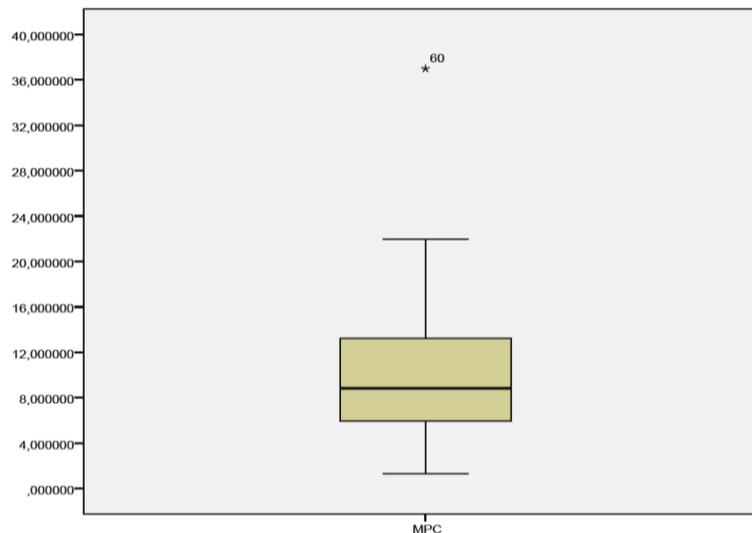


Figura 5-10 - Gráfico *Boxplot* para a Medida MPC

O gráfico *boxplot* para a medida CA é apresentado na Figura 5-12. A mediana é 5,92 e o valor máximo e mínimo é 19,875 e 0, respectivamente. Cerca de 50% dos valores da medida CA está no intervalo [2,167; 9,375]. Para essa medida, foram identificados seis *outliers* referentes aos sistemas de software com ID 4, ID 32, ID 37, ID 56, ID 60 e ID 94. Na Figura 5-13, é apresentado o gráfico *boxplot* da medida CE. Essa medida possui valor máximo e mínimo igual a 9 e 0, respectivamente, sendo que 50% dos valores encontrados estão no intervalo [2,25; 5,231], e mediana igual a 3,863. Para

essa medida, foram identificados nove *outliers* referentes aos sistemas de software com ID 4, ID 11, ID 26, ID 32, ID 37, ID 43, ID 60, ID 61 e ID 69.

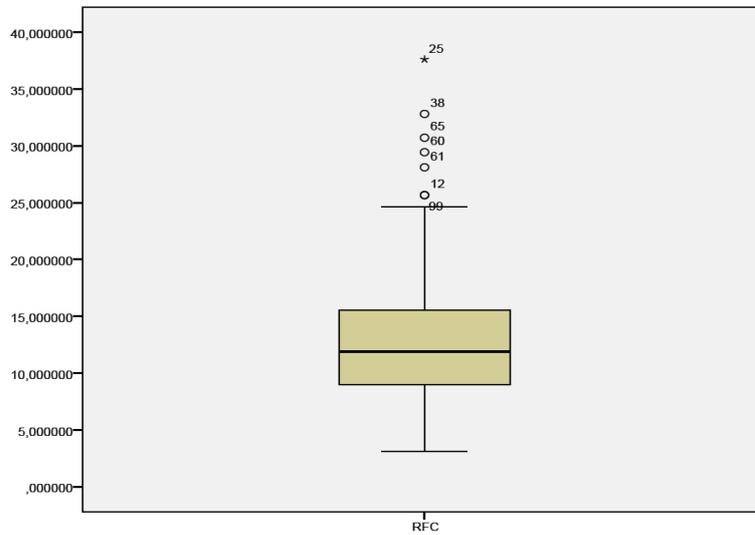


Figura 5-11 - Gráfico *Boxplot* para a Medida RFC

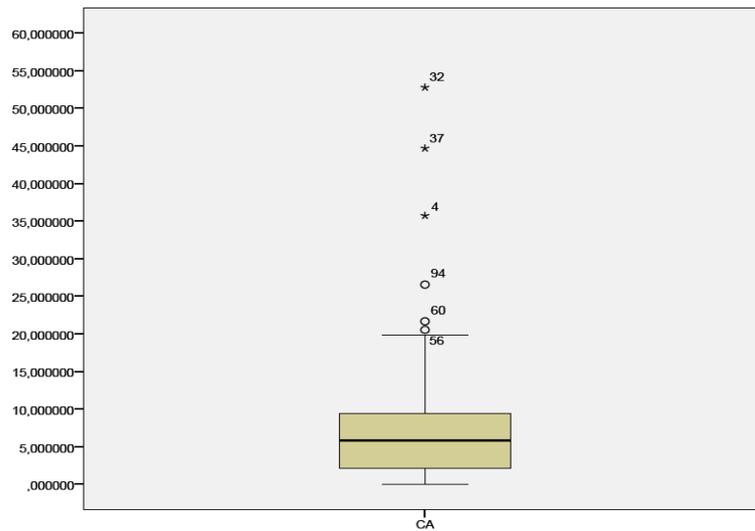


Figura 5-12 - Gráfico *Boxplot* para a Medida CA

Na Figura 5-14, é apresentado o gráfico da medida ILCOM. Metade dos sistemas de software apresentou valor para essa medida no intervalo $[0,923077; 1,42]$. O valor máximo e mínimo é $2,168017$ e $0,548387$, respectivamente. Para essa medida, sete *outliers* foram identificados, referentes aos sistemas de software com ID 16, ID 21, ID 24, ID 25, ID 44, ID 73 e ID 96. A mediana é $1,139535$. No gráfico *boxplot* da Figura 5-15, são apresentados os resultados para a medida LCOM. A mediana é $70,125$ e o valor máximo e mínimo é 260 e $2,230769$, respectivamente. Metade dos valores encontrados está intervalo $[32,86364; 136,6]$. Para essa medida, 14 *outliers* foram identificados, sendo 11 deles apresentados no gráfico, os demais possuem valores superiores a 2.000.

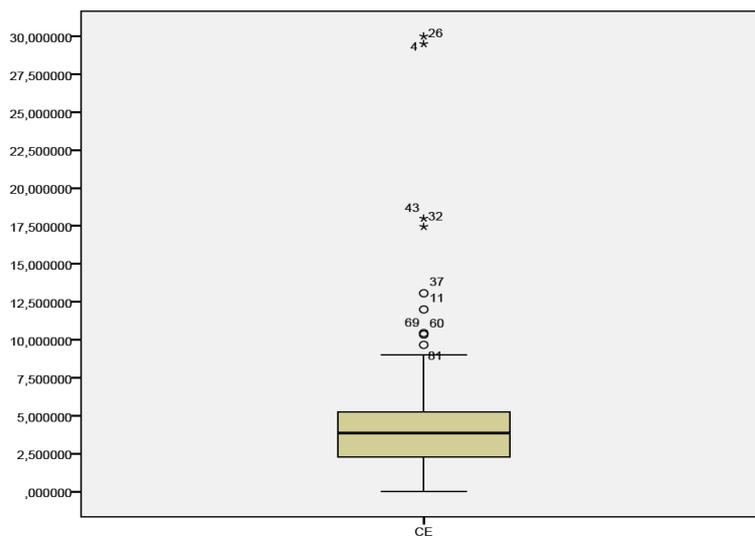


Figura 5-13 - Gráfico *Boxplot* para a Medida CE

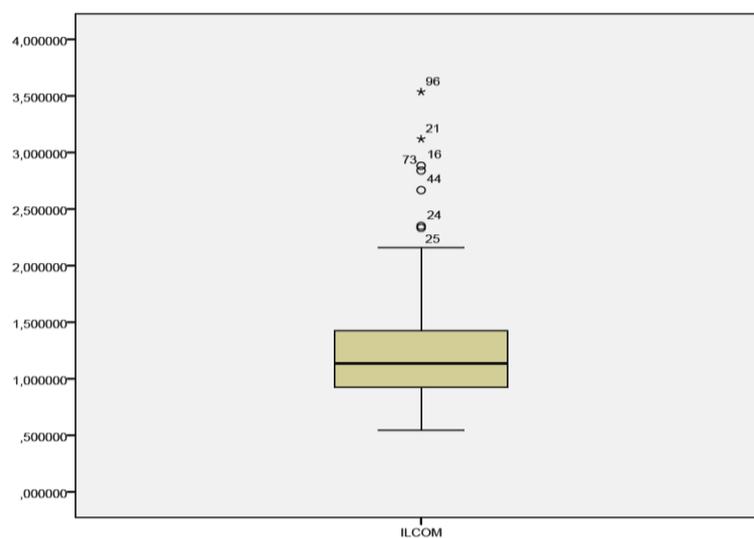


Figura 5-14 - Gráfico *Boxplot* para a Medida ILCOM

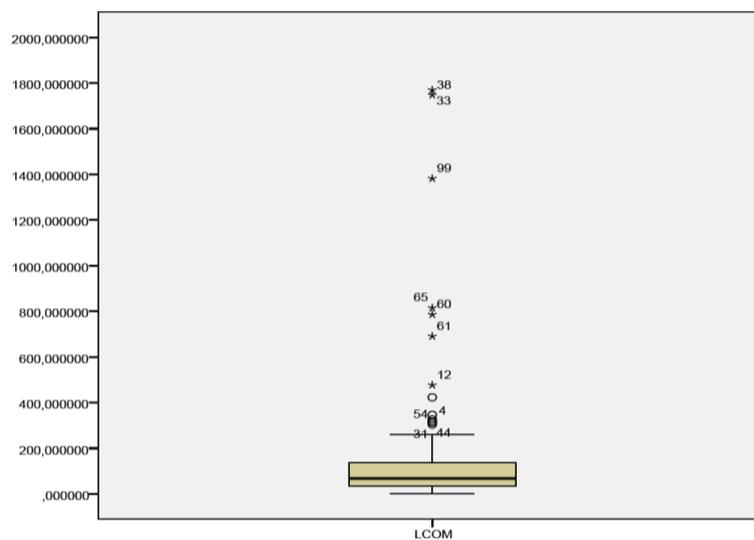


Figura 5-15 - Gráfico *Boxplot* para a Medida LCOM

Na Figura 5-16, é apresentado o gráfico *boxplot* para a medida TCC. Foram identificados dois *outliers* referentes aos sistemas de software com ID 19 e ID 41. O valor máximo e mínimo é 0,393888 e 0, respectivamente, sendo que 50% dos valores estão no intervalo [0,133333; 0,237555]. A mediana é 0,164447.

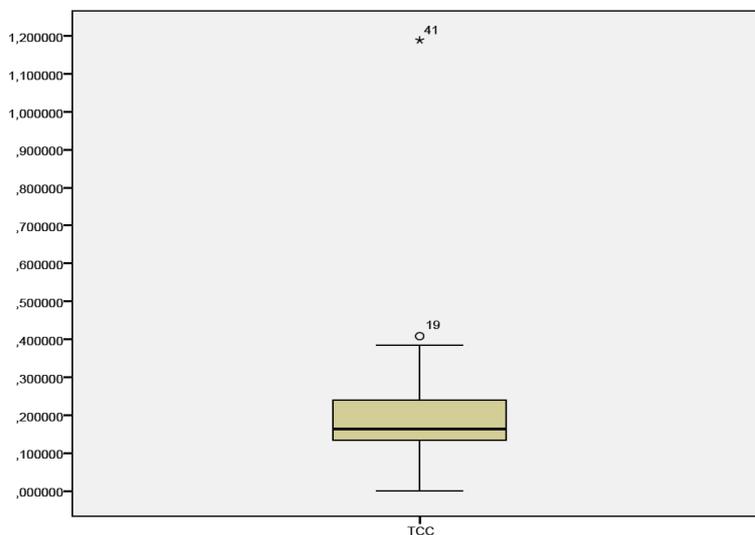


Figura 5-16 - Gráfico *Boxplot* para a Medida TCC

Na Tabela 5-14, estão resumidos os resultados apresentados nessa seção, em que podem ser observados os valores máximo e mínimo, a mediana e a quantidade de *outliers*. Conforme observado nos gráficos apresentados, verifica-se que 99% dos valores encontrados para as medidas DAC e MPC e 98% dos valores das medidas CBO e TCC estão no intervalo formado entre o valor máximo e mínimo. Para as medidas RFC e ILCOM, essa porcentagem é de 93%, para a medida CA é de 94%, para a medida CE é de 92% e para a medida LCOM é de 86%. Esses resultados permitem induzir que 76% dos sistemas de software no domínio Jogos apresentam comportamento padrão de valores para medidas de acoplamento e de coesão, estando no intervalo de máximo e mínimo. Embora a maioria esteja no padrão esperado, 24% tendem a ter valores discrepantes (maiores e menores) em relação ao acoplamento e à coesão. Esse resultado sugere que uma parcela de sistemas de software no domínio Jogos são construídos sem preocupações com qualidade interna de software, pois não é utilizada como referência padrões de bom projeto orientado a objetos, sob o aspecto modularização.

Tabela 5-14 - Resumo dos Resultados

Medida	Mediana	Valor Mínimo	Valor Máximo	Outliers
CBO	3,468750	0,862069	7,075000	2
DAC	3,112069	0,862069	6,115000	1
MPC	8,862069	1,333333	22,132143	1
RFC	11,888890	3,111111	24,938697	7

Tabela 5-14 - Resumo dos Resultados (cont.)

Medida	Mediana	Valor Mínimo	Valor Máximo	Outliers
CA	5,920000	0,000000	19,875000	6
CE	3,863000	0,000000	9,000000	9
ILCOM	1,139535	0,548387	2,168017	7
LCOM	70,125000	2,230769	260,000000	14
TCC	0,164447	0,000000	0,393888	2

Na Figura 5-17, é apresentado um diagrama de dispersão entre as medidas CBO e DAC. Pode ser observado que os pontos estão concentrados ao redor de uma reta crescente. Tal comportamento mostra que essas medidas possuem correlação linear positiva. Assim, se a medida CBO apresentar valor alto, o valor da medida DAC tende a acompanhar tal resultado. O diagrama de dispersão entre as medidas CBO e MPC é apresentado na Figura 5-18. Pode ser observado que os pontos se concentram ao redor de uma reta crescente, porém não estão tão próximos como no diagrama entre as medidas CBO e DAC. Esse comportamento mostra que existe correlação linear positiva entre as medidas CBO e MPF, mas é mais fraca do que entre as medidas CBO e DAC.

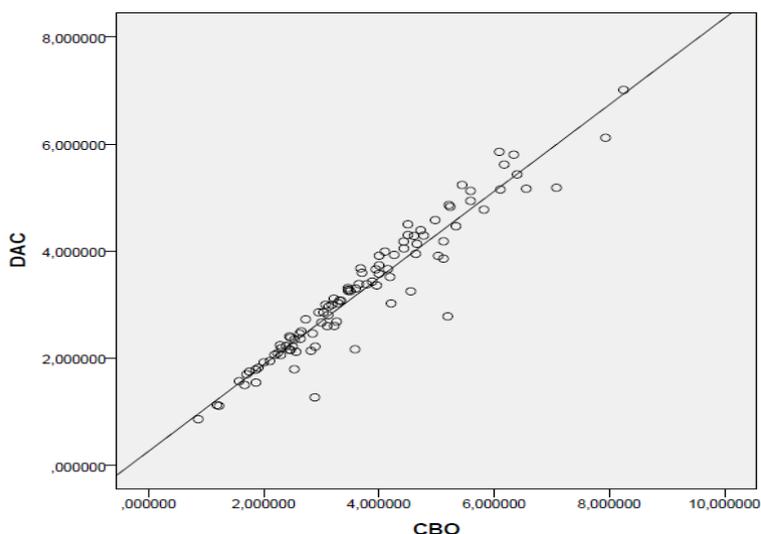


Figura 5-17 - Diagrama de Dispersão entre as Medidas CBO e DAC

A medida CBO apresentou correlação linear positiva com as medidas CA (Figura 5-19) e CE (Figura 5-20). Em ambas, pode ser notado que os pontos concentram-se em torno de uma reta crescente. Além disso, pode ser percebido que a correlação entre as medidas CBO e CA é mais forte do que a correlação entre as medidas CBO e CE, pois os pontos estão mais próximos da reta. No diagrama de dispersão entre as medidas DAC e CA (Figura 5-21), pode ser observado que existe correlação não linear positiva entre essas medidas, pois os pontos concentram-se em torno de uma curva crescente.

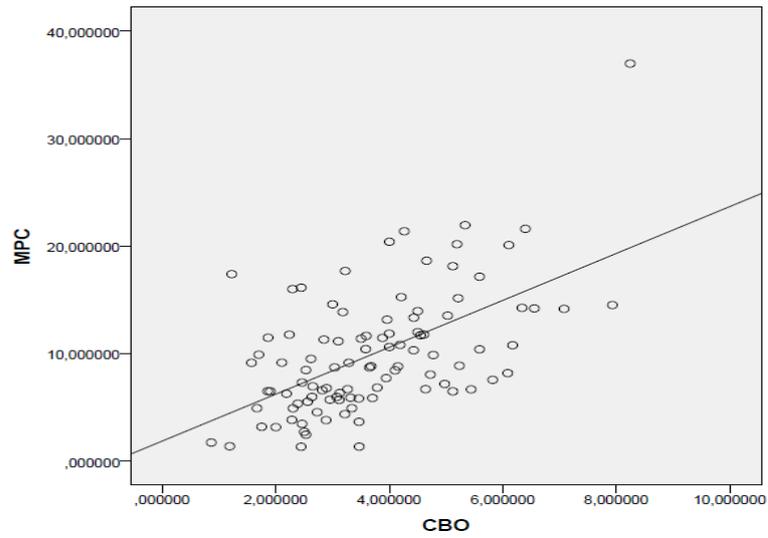


Figura 5-18 - Diagrama de Dispersão entre as Medidas CBO e MPC

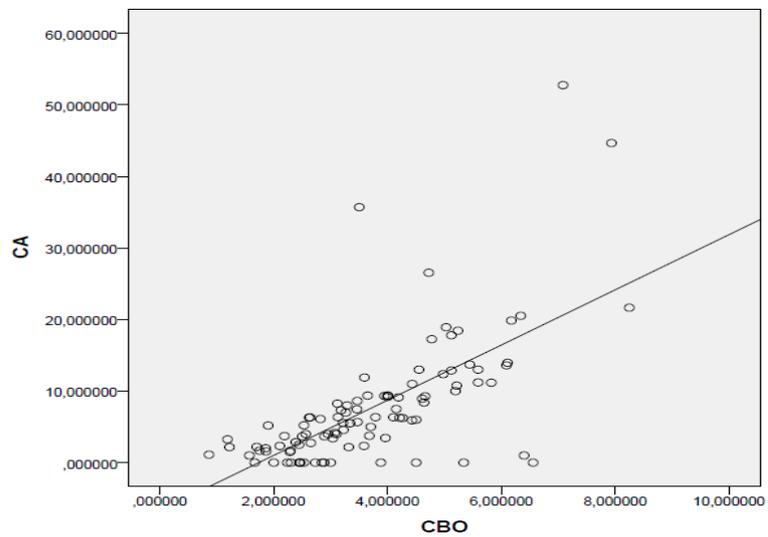


Figura 5-19 - Diagrama de Dispersão entre as Medidas CBO e CA

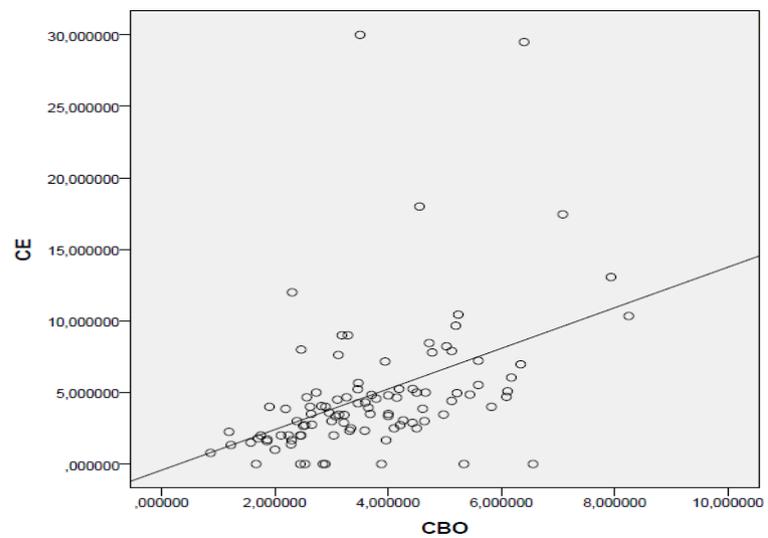


Figura 5-20 - Diagrama de Dispersão entre as Medidas CBO e CE

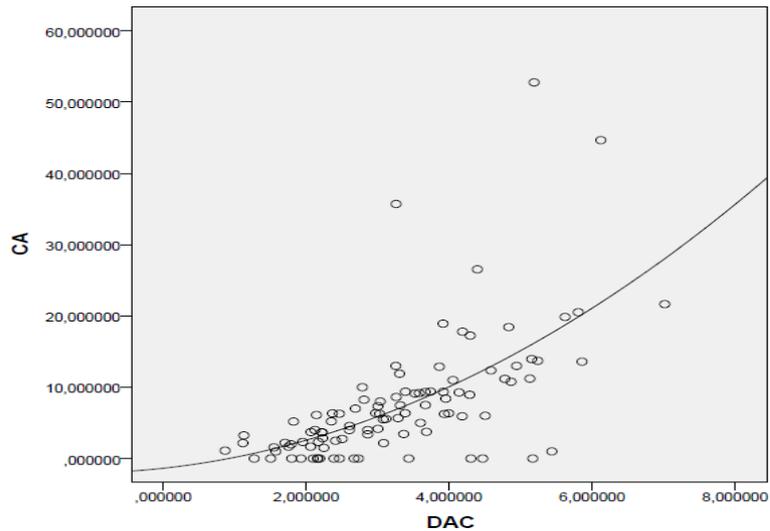


Figura 5-21 - Diagrama de Dispersão entre as Medidas DAC e CA

A medida DAC apresentou correlação não linear positiva com a medida CE (Figura 5-22). Na Figura 5-23, é apresentado o diagrama de dispersão entre as medidas MPC e RFC. Pode ser observado que os pontos concentram-se em torno de uma reta crescente, mostrando a existência de correlação linear positiva entre essas medidas. O diagrama de dispersão entre as medidas MPC e LCOM é apresentado na Figura 5-24, em que os pontos concentram-se em torno de uma curva, indicando a existência de correlação não linear entre as medidas. A correlação entre as medidas RFC e ILCOM pode ser observada no diagrama de dispersão da Figura 5-25. Pode ser notado que os pontos concentram-se ao redor de uma curva, indicando correlação não linear entre essas medidas. A medida RFC apresentou correlação não linear com a medida LCOM (Figura 5-26).

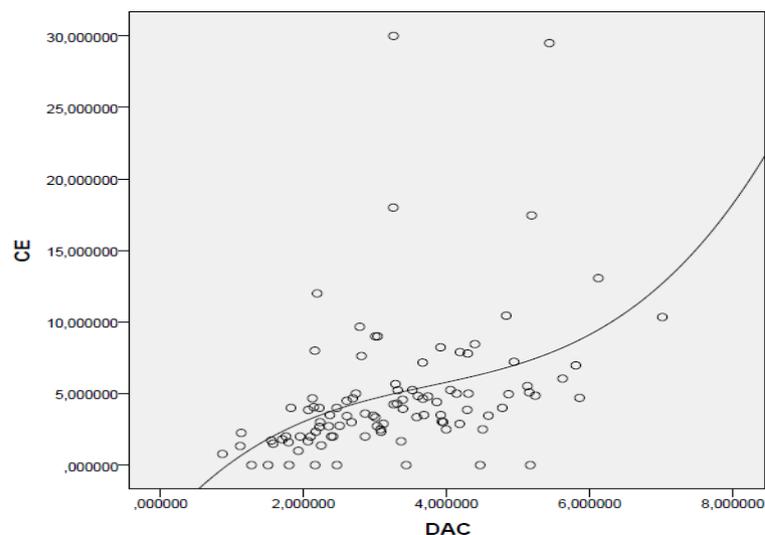


Figura 5-22 - Diagrama de Dispersão entre as Medidas DAC e CE

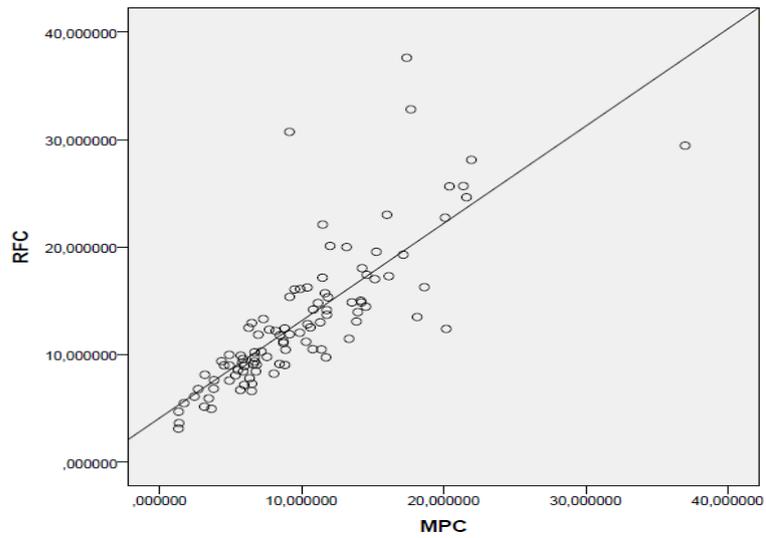


Figura 5-23 - Diagrama de Dispersão entre as Medidas MPC e RFC

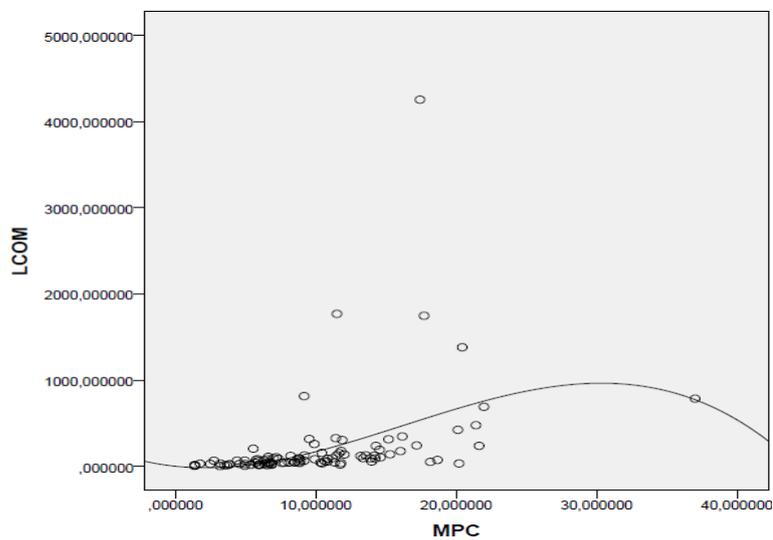


Figura 5-24 - Diagrama de Dispersão entre as Medidas MPC e LCOM

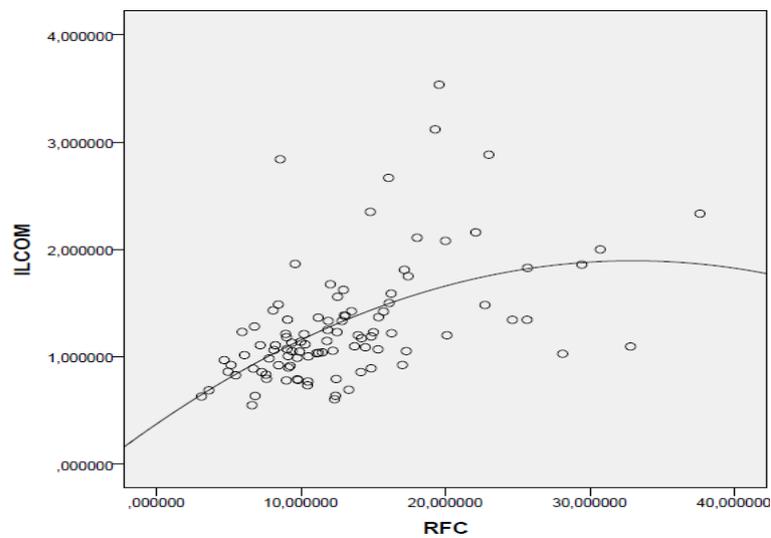


Figura 5-25 - Diagrama de Dispersão entre as Medidas RFC e ILCOM

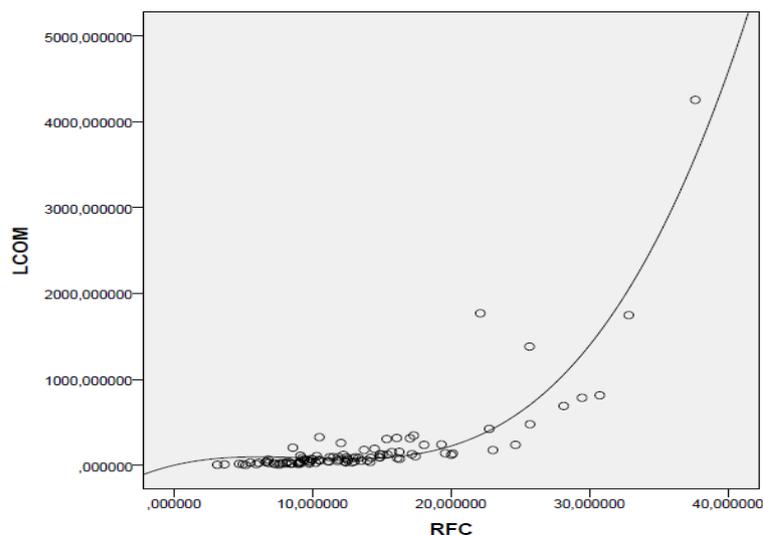


Figura 5-26 - Diagrama de Dispersão entre as Medidas RFC e LCOM

No diagrama de dispersão da Figura 5-27, pode ser observado que as medidas CA e CE possuem correlação não linear, pois os pontos concentram-se em torno de uma curva. Na Figura 5-28, os pontos concentram-se em torno de uma reta crescente, o que indica que há correlação linear positiva entre as medidas ILCOM e LCOM.

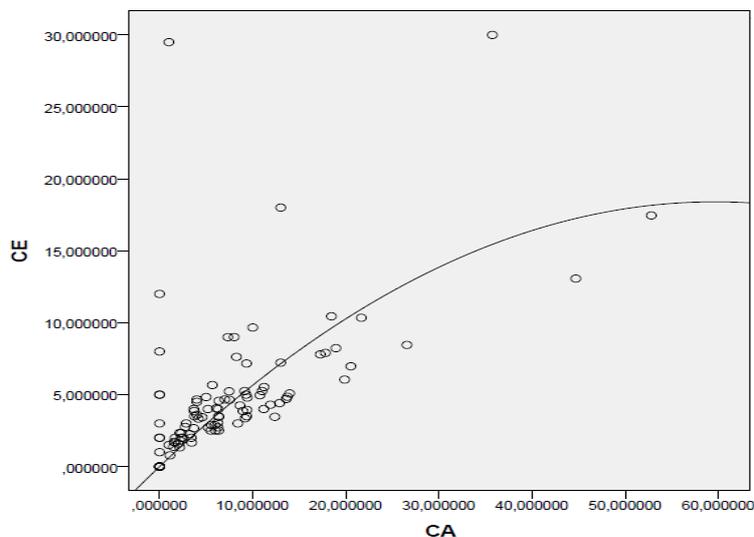


Figura 5-27 - Diagrama de Dispersão entre as Medidas CA e CE

5.5. Análise de Correlação

A amostra utilizada na Análise de Correlação é composta por variáveis intervalares, as técnicas de *Pearson* e *Spearman* podem ser utilizadas para esse tipo de variáveis, embora a técnica *Spearman* também possa ser utilizada para a análise de variáveis ordinais. Neste trabalho, foi utilizada a Análise de Correlação de *Spearman*, pois se pode analisar a existência de relações lineares e não lineares entre duas variáveis, enquanto a técnica de

Pearson investiga apenas relacionamentos lineares. Essa análise foi realizada utilizando o software *IBM SPSS Statistics*.

As medidas apresentadas na Tabela 5-3 foram utilizadas como dados de entrada para a análise, em que cada medida representou uma variável escalar no conjunto de dados. Após a inserção dos dados, foi acessada uma janela pelo *menu* Analisar → Correlacionar → Bivariável (Figura 5-29). Nessa janela, todas as variáveis devem ser selecionadas. No campo Coeficientes de correlação, a opção Spearman e, no campo Teste de significância, a opção Com duas extremidades e a opção Sinalizar correlações significantes devem ser selecionadas.

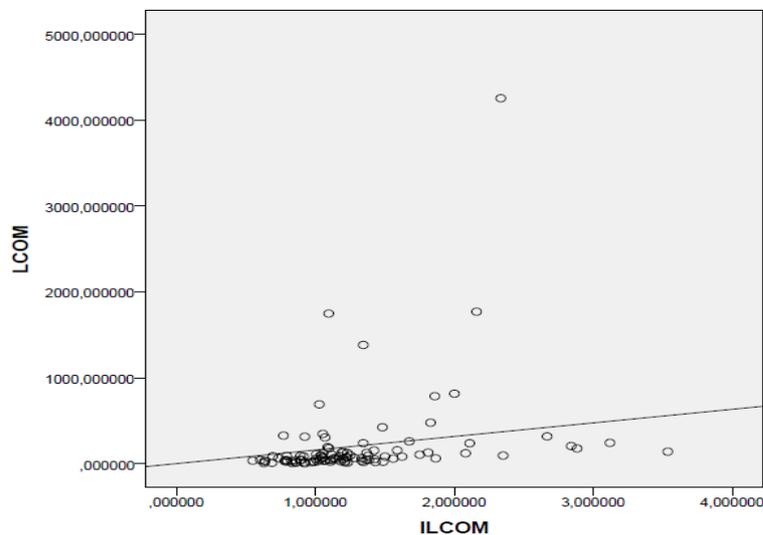


Figura 5-28 - Diagrama de Dispersão entre as Medidas ILCOM e LCOM

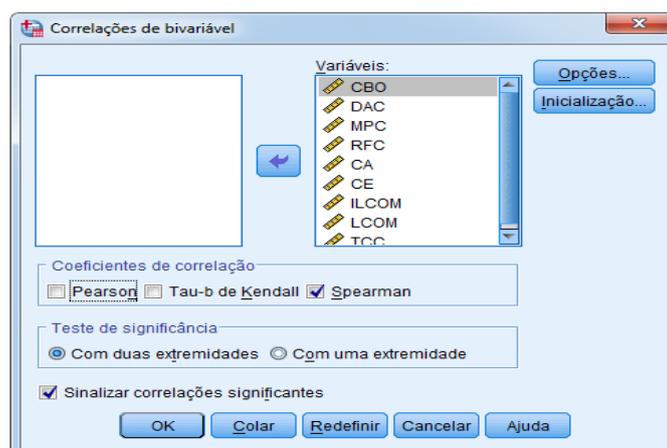


Figura 5-29 - Janela de Correlação Bivariável

Em seguida, foi selecionado o botão *Opções* para abrir a janela apresentada na Figura 5-7. No campo *Estatísticas*, foram selecionadas as opções *Médias* e

desvio padrão e Desvios e covariâncias entre os produtos e, em seguida, são selecionados o botão Continuar e o botão OK (Figura 5-29).

Na Tabela 5-15, é apresentado o resultado da Análise de Correlação de *Spearman*. Os valores em negrito representam correlações significativas, considerando nível de significância de 1%, os valores em vermelho representam correlações fortes e os valores em verde representam correlações moderadas. Apenas um coeficiente apresentou correlação significativa para nível de significância de 5% (esse valor está sublinhado). Para encontrar os valores do coeficiente de correlação entre as variáveis, deve ser realizada a análise do cruzamento entre linhas e colunas. Por exemplo, para descobrir o coeficiente de correlação entre as medidas CBO e RFC, deve-se analisar o valor contido na célula formada pelo cruzamento da linha referente à variável CBO e a coluna referente à variável RFC. Nesse caso, o valor do coeficiente é 0,318.

Tabela 5-15 - Resultado da Análise de Correlação de Spearman

Variáveis		Coeficiente de Correlação de Spearman						Coesão		
		Acoplamento						Coesão		
Acoplamento		CBO	DAC	MPC	RFC	CA	CE	ILCOM	LCOM	TCC
		CBO	1,000	0,955	0,531	0,318	0,699	0,562	-0,005	0,297
	DAC	0,955	1,000	0,457	0,287	0,680	0,544	-0,008	0,304	-0,066
	MPC	0,531	0,457	1,000	0,884	0,268	0,168	0,410	0,708	0,095
	RFC	0,318	0,287	0,884	1,000	0,075	0,021	0,557	0,841	0,163
	CA	0,699	0,680	0,268	0,075	1,000	0,709	-0,027	0,194	-0,155
	CE	0,562	0,544	0,168	0,021	0,709	1,000	0,033	0,119	-0,070
Coesão	ILCOM	-0,005	-0,008	0,410	0,557	-0,027	0,033	1,000	0,503	<u>0,208</u>
	LCOM	0,297	0,304	0,708	0,841	0,194	0,119	0,503	1,000	0,044
	TCC	-0,125	-0,066	0,095	0,163	-0,155	-0,070	<u>0,208</u>	0,044	1,000

Vermelho - correlação forte
Verde - correlação moderada

Negrito - correlação significativa (nível de significância = 1%)
Sublinhado - correlação significativa (nível de significância = 5%)

Há forte correlação positiva entre as medidas CBO e DAC, MPC e RFC e RFC e LCOM, pois seus coeficientes de correlação são 0,955, 0,884 e 0,841, respectivamente. As medidas CBO e DAC possuem correlações significativas com as demais medidas de acoplamento e com a medida LCOM. A medida MPC apresentou correlação significativa com todas as medidas de acoplamento, com exceção da medida CE = 0,168 e com as medidas ILCOM e LCOM, sendo que a correlação com as medidas CBO = 0,531 e LCOM = 0,557 é moderada. A medida RFC apresentou correlação significativa positiva com as medidas CBO, DAC, MPC, ILCOM e LCOM, sendo moderadamente correlacionada com a medida ILCOM = 0,557. A medida CA apresentou correlação significativa com todas as medidas de acoplamento, com exceção da medida RFC = 0,075. A medida CE apresentou correlação significativa com as medidas

CBO = 0,562, DAC = 0,544 e CA = 0,709. A medida ILCOM apresentou correlação moderada positiva com as medidas RFC = 0,557 e LCOM = 0,503 e correlação significativa fraca com a medida MPC = 0,410. A medida LCOM está correlacionada significativamente com todas as medidas de acoplamento, exceto com as medidas CA = 0,194 e CE = 0,119, e com ILCOM = 0,503. A medida TCC apresentou correlação significativa com a medida ILCOM = 0,208, mas com nível de significância de 5%.

5.6. Discussão dos Resultados

Nesta seção, é apresentada uma interpretação qualitativa dos resultados obtidos na Análise de Correlação. Para isso, foi avaliado o relacionamento entre medidas de mesma propriedade para ajudar a definir qual medida pode ser relevante ao determinar a qualidade de sistemas de software.

5.6.1. Correlação Intrapropriedades

As medidas CBO e DAC apresentaram correlação significativa com as demais medidas de acoplamento e forte correlação entre si. A existência de correlação significativa entre essas medidas indica que elas são relevantes para o cálculo do acoplamento de sistemas de software no domínio Jogos. Esse forte relacionamento era esperado, pois, por definição, uma classe que apresenta valores altos para a medida DAC apresenta valores altos para a medida CBO. Isso foi confirmado pela Análise de Correlação, pois elas apresentaram coeficiente de correlação positivo e próximo do valor 1. Essa forte correlação permite decidir por utilizar uma das duas medidas, podendo optar por aquela possui um cálculo mais fácil, por exemplo.

A correlação entre as medidas CBO e DAC e as demais medidas de acoplamento se justifica, pois ambas medem uma característica básica de acoplamento (quantidade de classes com a quais uma classe está acoplada ou das quais depende). Porém, a existência de relacionamento não indica que as demais medidas podem ser substituídas pelas medidas CBO e DAC, pois pode ser necessário medir características de acoplamento mais específicas. As medidas MPC e RFC apresentaram forte relacionamento e mostraram-se relevantes para o cálculo do acoplamento, pois ambas medem o acoplamento por passagem de mensagens. A medida CE apresentou relacionamento com três medidas de acoplamento (CBO, DAC e CA) e mostrou-se relevante para o cálculo de acoplamento. O relacionamento com CA se justifica, pois ambas são medidas relacionadas. A baixa

correlação entre a medida CE e as demais medidas de acoplamento pode ser explicada pelo fato da maioria dos sistemas coletados possuírem pequena quantidade de pacotes e contabilizar a quantidade de classes externas com as quais um pacote está acoplado.

As medidas de coesão não apresentaram correlação forte entre si. As medidas LCOM e ILCOM possuem correlação significativa, o que se justifica pelo fato de a medida ILCOM ser uma variação da medida LCOM. Portanto, essas medidas mostraram-se relevantes para o cálculo da coesão de sistemas de software no domínio Jogos. A medida TCC apresentou correlação significativa com a medida ILCOM, porém com nível de significância de 5%.

5.6.2. Correlação Interpropriedades

Era esperado relacionamento negativo entre as medidas de coesão e de acoplamento, pois um sistema de software com qualidade é caracterizado, entre outras propriedades, por baixo acoplamento e alta coesão. Porém, a Análise de Correlação não apresentou correlação negativa significativa entre essas medidas.

A medida TCC não apresentou relacionamento significativo com medidas de acoplamento, portanto não é possível afirmar que existe relação entre essa medida e o acoplamento de sistemas de software. A medida LCOM apresentou forte relacionamento com a medida RFC e relacionamento significativo com as medidas CBO, DAC e MPC, sendo todos positivos.

É importante ressaltar que a medida LCOM mede a diferença entre a quantidade de pares de métodos que possuem e os que não possuem atributos comuns em uma classe. Assim, quanto maior o valor da medida LCOM, menos coesa é a classe. Se a medida LCOM possui relacionamento positivo com medidas de acoplamento, isso significa que o acoplamento será maior à medida que a coesão em uma classe diminuir.

5.7. Considerações Finais

A indústria de jogos está em constante crescimento. Por isso, é necessário desenvolver métodos para a produção de jogos com qualidade. As medidas de coesão e de acoplamento são utilizadas como indicadores de qualidade no desenvolvimento de sistemas. Uma investigação sobre o relacionamento de tais medidas, considerando sistemas de software no domínio Jogos, pode ajudar no desenvolvimento de jogos com qualidade.

As medidas CBO, DAC, MPC, RFC, CA e CE foram consideradas relevantes para o cálculo do acoplamento em Jogos, enquanto as medidas LCOM e ILCOM foram consideradas relevantes para o cálculo da coesão. Portanto, os resultados obtidos neste trabalho podem auxiliar no entendimento da modularização de jogos, por meio da definição de quais medidas podem ser consideradas no estudo da modularização de sistemas de software no domínio Jogos.

6. CONSIDERAÇÕES FINAIS

6.1. Conclusões

Com base nos resultados da Análise de Correlação de uma amostra de 100 sistemas de software, foi possível inferir que as medidas de acoplamento consideradas no estudo são relevantes para o cálculo do acoplamento em sistemas de software no domínio Jogos. Pois, elas apresentaram correlações significativas. Além disso, as medidas CBO e DAC e MPC e RFC apresentaram forte correlação entre si e, como elas possuem definições semelhantes, é possível optar pela utilização uma dessas medidas.

Dentre as medidas de coesão, as medidas LCOM e ILCOM apresentaram correlações significativas entre si e com algumas medidas de acoplamento. Além disso, essas medidas podem ser consideradas relevantes para o cálculo da coesão de sistemas de software no domínio Jogos. A medida TCC apresentou correlação significativa, com nível de significância de 5%, não se mostrando relevante para o cálculo dessa propriedade.

6.2. Contribuições

Este estudo contribui com a análise da modularização de um domínio de sistemas de software que se torna cada vez mais popular e que representa um campo empírico novo [Aarseth, 2003]. Os resultados deste estudo podem contribuir para o desenvolvimento de sistemas de software no domínio Jogos com maior qualidade. Pois, ao conhecer a modularização desses sistemas e sua relação com as medidas de software, os desenvolvedores podem produzir Jogos com módulos menos acoplados e mais coesos.

6.3. Limitações

Neste estudo, dedicou-se à caracterização da modularização de Jogos, por meio da Análise de Correlação de medidas de acoplamento e de coesão. Essa caracterização foi realizada utilizando uma amostra de 100 sistemas de software, coletados do repositório *Sourceforge* e desenvolvidos na linguagem de programação Java. Uma das limitações é o fato dos sistemas coletados serem desenvolvidos apenas em Java e terem sido coletados apenas em um repositório. A escolha de sistemas desenvolvidos em outras linguagens e de repositórios diferentes permitiria que a caracterização fosse feita de forma mais abrangente, pois seria considerada uma população de projetos maior e mais diversificada. Outra limitação é a escolha de sistemas avaliados pelos usuários com mais de 3,5 estrelas.

As medidas utilizadas também caracterizam uma limitação, pois não foram consideradas "todas" as medidas de acoplamento e de coesão apresentadas na literatura. Além disso, foram consideradas apenas medidas para essas duas propriedades. Um estudo que considerasse maior quantidade de medidas, englobando outras propriedades poderia fornecer mais dados para a avaliação da modularização dos sistemas de software. Outra limitação do estudo foi uso de apenas a Análise de Correlação de *Spearman*. A utilização de outras técnicas estatísticas permitiria que a análise das medidas fosse mais detalhada, possibilitando, por exemplo, o agrupamento das medidas. Pode-se considerar que a escolha de sistemas de software no domínio Jogos caracteriza uma limitação para o estudo. A utilização de projetos de outros domínios permitiria que os resultados obtidos fossem comparados, sendo possível avaliar a relação entre a modularização de sistemas no domínio Jogos e de sistemas de outros domínios.

6.4. Trabalhos Futuros

Entre as sugestões de trabalhos futuros, está a realização de um estudo similar que considere sistemas de software pertencentes a outros domínios. Um estudo desse tipo permitiria que a modularização de sistemas de software no domínio Jogos fosse comparada à modularização de sistemas de software de outros domínios. Assim, seria possível descobrir, por exemplo, se jogos tendem a ser mais acoplados do que sistemas de software em outros domínios.

Um estudo que considere outras medidas de coesão e de acoplamento também é sugerido, visto que, neste trabalho, não foram consideradas "todas" as medidas pertencentes a essas propriedades. Além disso, a utilização de maior quantidade de medidas forneceria mais dados para a avaliação da modularização dos projetos. A utilização de medidas de outras propriedades pode ser considerada em trabalhos futuros, o que permite a caracterização da modularização seja feita com base em outros critérios, por exemplo, o tamanho dos sistemas de software. Sugere-se a utilização de outras técnicas estatísticas, por exemplo, técnicas de clusterização. A utilização deste tipo de técnica permitirá que as medidas estudadas sejam agrupadas e que a análise seja realizada com base em grupos de medidas.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aarseth, E. Playing Research: Methodological Approaches to Game Analysis. In: 5th International Digital Arts and Culture Conference (MelbourneDAC). 2003.
- Akingbehin, K. Taguchi Smaller-the-Best Software Quality Metrics. In: 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel / Distributed Computing. pp. 585-588. 2009.
- Atole, C.S; Kale, K. V. Assessment of Package Cohesion and Coupling Principles for Predicting the Quality of Object Oriented Design. In: 1st International Conference on Digital Information Management. pp. 1-5. 2007.
- Bavota, G.; De Lucia, A.; Marcus, A.; Oliveto, R. Software Re-Modularization Based on Structural and Semantic Metrics. In: 17th Working Conference on Reverse Engineering (WCRE). pp. 195-204. 2010.
- Bieman, J. M.; Kang, B. Cohesion and Reuse in an Object-Oriented System. In: Symposium on Software Reusability. pp. 259-262. 1995.
- Bond, M.; Beale, R. What makes a good game? Using reviews to inform design. In: HCI - People and Computers XXIII. pp. 418-422. 2009.
- Borkowf, C. B. Computing the non-null Asymptotic Variance and the Asymptotic Relative Efficiency of Spearman's Rank Correlation. In: Computational Statistics & Data Analysis. v. 39. is. 3. pp. 271-286. 2002.
- Brito e Abreu, F.; Goulão, M. Coupling and Cohesion as Modularization Drivers: Are We Being Over-Persuaded? In: Fifth European Conference on Software Maintenance and Reengineering. pp. 47-57. 2001.
- Chidamber, S. R.; Kemerer, C. F. Towards a Metrics Suite for Object Oriented Design. In: Conference proceedings on Object oriented programming systems, languages, and applications. pp. 197-211. 1991.
- Chowdhury, I.; Zulkernine, M. Using Complexity, Coupling, and Cohesion Metrics as Early Indicators of Vulnerabilities. In: Journal of Systems Architecture. v. 57. is. 3. pp. 294-313, 2011.
- Counsell, S.; Swift, S.; Crampton, J. The Interpretation and Utility of Three Cohesion Metrics for Object-Oriented Design. In: ACM Transactions on Software Engineering and Methodology. v. 15. is. 2. pp. 123-149. 2006.
- Elish, M. O.; Al-Yafei, A. H.; Al-Mulhem, M. Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems: A case study of Eclipse. In: Advances in Engineering Software. v. 42. is.10. pp. 852-859. 2011.
- Fontana, F.A.; Maggioni, S. Metrics and Antipatterns for Software Quality Evaluation. In: 34th IEEE Software Engineering Workshop (SEW). pp. 48-56. 2011.

- Freund, J. E. Estatística Aplicada: Economia, Administração e Contabilidade. 11ª ed. Porto Alegre: Bookman. 536p. 2006.
- Gui, G.; Scott, P. D. Measuring Software Component Reusability by Coupling and Cohesion Metrics. In: Journal of Computers. v. 4. is. 9. pp. 797-805. 2009.
- Gui, G.; Scott, P. D. Ranking Reusability of Software Components Using Coupling Metrics. In: The Journal of Systems and Software. v. 80. pp. 1450-1459. 2007.
- Guimarães, P. R. B. Métodos Quantitativos Estatísticos. 1ª ed. Curitiba: IESDE Brasil S.A. 245p. 2008.
- Gupta, V.; Chhabra, J. K. Dynamic Cohesion Measures for Object-Oriented Software. In: Journal of Systems Architecture. v. 57. is. 4. pp. 452-462. 2011.
- Hars, A.; Ou, S. Working for Free? - Motivations of Participating in Open Source Projects. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences. pp. 7014-7022. 2001.
- Hippel, E. von; Krogh, G. von. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. In: Organization Science. v. 14. is. 2. pp. 209-223. 2003.
- Hitz, M.; Montazeri, B. Measuring Coupling and Cohesion in Object-Oriented Systems. In: International Symposium on Applied Corporate Computing. pp. 25-27. 1995.
- Hitz, M.; Montazeri, B. Chidamber and Kemerer's Metrics Suite: A Measurement Theory Perspective. In: IEEE Transactions Software Engineering. v. 22. is.4. pp. 267-271. 1996.
- Ince, D. Software Metrics: Introduction. In: Butterworth-Heinemann. v. 32. is. 4. pp. 297-303. 1990.
- ISO/IEC 25010. Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE). 2010.
- Juran, J. M.; Godfrey, A. B. Juran's Quality Handbook. 5ª ed. McGraw-Hill. 1872 p. 2000.
- Kabaili, H.; Keller, R. K.; Lustman, F. Cohesion as Changeability Indicator in Object-Oriented Systems. In: Software Fifth European Conference on Maintenance and Reengineering. pp. 39-46. 2001.
- Kan, S. H. Metrics and Models in Software Quality Engineering. 2ª ed. Addison Wesley. 560 p. 2002.
- Kanode, C. M.; Haddad, H. M. Software Engineering Challenges in Game Development. In: Sixth International Conference on Information Technology: New Generations. pp. 260-265. 2009.
- Larson, R.; Farber, B. Estatística Aplicada. 4ª ed. São Paulo: Pearson Prentice Hall. 638p. 2010.

- Li, W.; Henry, S. Maintenance Metrics for the Object Oriented Paradigm. In: First International Software Metrics Symposium. pp. 52-60. 1993.
- Lincke, R.; Lundberg, J.; Löwe, W. Comparing Software Metrics Tools. In: International Symposium on Software Testing and Analysis. pp. 131-142. 2008.
- Lira, S. A. Análise de Correlação: Abordagem Teórica e de Construção dos Coeficientes com Aplicações. Universidade Federal do Paraná. 196p. 2004.
- Mäkelä, S.; Leppänen, V. Client-Based Cohesion Metrics for Java Programs. In: Science of Computer Programming. v. 74. is. 5- 6. pp. 355-378. 2009.
- Martin, R. OO Design Quality Metrics - An Analysis of Dependencies. In: Workshop Pragmatic and Theoretical Directions in Object-Oriented Software Metrics. 1994.
- Meyers, T. M.; Binkley, D. Slice-based Cohesion Metrics and Software Intervention. In: 11th Working Conference on Reverse Engineering. pp. 256-265. 2004.
- Meyers, T. M.; Binkley, D. An Empirical Study of Slice-Based Cohesion and Coupling Metrics. In: An Empirical Study of Slice-Based Cohesion and Coupling Metrics. v. 17. is. 1. pp. 1-27. 2007.
- Parnas, D. L. On the Criteria to Be Used in Decomposing Systems into Modules. In: Communications of the ACM. v. 15. is. 12. pp. 1053-1058. 1972.
- Perepletchikov, M.; Ryan, C.; Frampton, K. Cohesion Metrics for Predicting Maintainability of Service-Oriented Software. In: Seventh International Conference on Quality Software. pp. 328-335. 2007.
- Presman, R. S. Engenharia de Software - Uma Abordagem Profissional. 7ª ed. Porto Alegre: AMGH. 780p. 2011.
- Santiago, M. R. Ensaio do SWEBoK - Software Engineering Body Of Knowledge. Goiânia: Universidade Gama Filho. 2011.
- Sarkar, S.; Rama, G. M.; Kak, A. C. API-Based and Information-Theoretic Metrics for Measuring the Quality of Software Modularization. In: IEEE Transactions on Software Engineering. v. 33. is. 1. pp. 14-32. 2007.
- Scacchi, W. Free and Open Source Development Practices in the Game Community. In: IEEE Software. v. 21. is. 1. pp. 59-66. 2004.
- Siegel, S.; Castellan Jr., N. J. Estatística Não-Paramétrica para Ciências do Comportamento. 2ª ed. Porto Alegre: ARTMED Editora S.A. 448p. 2006.
- Sommerville, I. Engenharia de Software. São Paulo: Pearson Addison-Wesley. 568p. 2011.
- Stevenson, W. J. Estatística Aplicada à Administração. 1ª ed. São Paulo: Harper & Row do Brasil. 495 p. 1981.

Suresh, Y.;Pati, J.; Rath, S. K. Effectiveness of Software Metrics for Object-Oriented System. In: Procedia Technology. v. 6. is.0 pp. 420-427. 2012.

APÊNDICE A

Tabela A-1 - Valores Críticos para o Coeficiente de Correlação de *Pearson* [Larson; Farber, 2010]

<i>n</i>	$\alpha = 0,05$	$\alpha = 0,01$
4	0,950	0,990
5	0,878	0,959
6	0,811	0,917
7	0,754	0,875
8	0,707	0,834
9	0,666	0,798
10	0,632	0,765
11	0,602	0,735
12	0,576	0,708
13	0,553	0,684
14	0,532	0,661
15	0,514	0,641
16	0,497	0,623
17	0,487	0,606
18	0,468	0,590
19	0,456	0,575
20	0,444	0,561
21	0,433	0,549
22	0,423	0,537
23	0,413	0,526
24	0,404	0,515
25	0,396	0,505
26	0,388	0,496
27	0,381	0,487
28	0,374	0,479
29	0,367	0,471
30	0,361	0,463
35	0,334	0,430
40	0,312	0,403
45	0,294	0,380
50	0,279	0,361
55	0,266	0,345
60	0,254	0,330
65	0,244	0,317
70	0,235	0,306
75	0,227	0,296
80	0,220	0,286
85	0,213	0,278
90	0,207	0,270
95	0,202	0,263
100	0,197	0,256

APÊNDICE B

Tabela B-1 - Valores Críticos para o teste *t* Coeficiente de Correlação de *Pearson* [Larson; Farber, 2010]

	Unicaudal, α	0,25	0,10	0,05	0,025	0,01	0,005
g.1.	Bicaudal, α	0,50	0,20	0,10	0,05	0,02	0,01
1		1,000	3,078	6,314	12,706	31,821	63,657
2		0,816	1,886	2,920	4,303	6,965	9,925
3		0,765	1,638	2,353	3,182	4,541	5,841
4		0,741	1,533	2,132	2,776	3,747	4,604
5		0,727	1,476	2,015	2,571	3,365	4,032
6		0,718	1,440	1,943	2,447	3,143	3,707
7		0,711	1,415	1,895	2,365	2,998	3,499
8		0,706	1,397	1,860	2,306	2,896	3,355
9		0,703	1,383	1,833	2,262	2,821	3,250
10		0,700	1,372	1,812	2,228	2,764	3,169
11		0,697	1,363	1,796	2,201	2,718	3,106
12		0,695	1,356	1,782	2,179	2,681	3,055
13		0,694	1,350	1,771	2,160	2,650	3,012
14		0,692	1,345	1,761	2,145	2,624	2,977
15		0,691	1,341	1,753	2,131	2,602	2,947
16		0,690	1,337	1,746	2,120	2,583	2,921
17		0,689	1,333	1,740	2,110	2,567	2,898
18		0,688	1,330	1,734	2,101	2,552	2,878
19		0,688	1,328	1,729	2,093	2,539	2,861
20		0,687	1,325	1,725	2,086	2,528	2,845
21		0,686	1,323	1,721	2,080	2,518	2,831
22		0,686	1,321	1,717	2,074	2,508	2,819
23		0,685	1,319	1,714	2,069	2,500	2,807
24		0,685	1,318	1,711	2,064	2,492	2,797
25		0,684	1,316	1,708	2,060	2,485	2,787
26		0,684	1,315	1,706	2,056	2,479	2,779
27		0,684	1,314	1,703	2,052	2,473	2,771
28		0,683	1,313	1,701	2,048	2,467	2,763
29		0,683	1,311	1,699	2,045	2,462	2,756
∞		0,674	1,282	1,645	1,960	2,326	2,576

APÊNDICE C

Tabela C-1 - Valores Críticos para o Coeficiente de Correlação de *Spearman* [Larson; Farber, 2010]

<i>n</i>	$\alpha = 0,10$	$\alpha = 0,05$	$\alpha = 0,01$
5	0,900	-	-
6	0,829	0,886	-
7	0,714	0,786	0,929
8	0,643	0,738	0,881
9	0,600	0,700	0,833
10	0,564	0,648	0,794
11	0,536	0,618	0,818
12	0,497	0,591	0,780
13	0,475	0,566	0,745
14	0,457	0,545	0,716
15	0,441	0,525	0,689
16	0,425	0,507	0,666
17	0,412	0,490	0,645
18	0,399	0,476	0,625
19	0,388	0,462	0,608
20	0,377	0,450	0,591
21	0,368	0,438	0,576
22	0,359	0,428	0,562
23	0,351	0,418	0,549
24	0,343	0,409	0,537
25	0,336	0,400	0,526
26	0,329	0,392	0,515
27	0,323	0,385	0,505
28	0,317	0,377	0,496
29	0,311	0,370	0,487
30	0,305	0,364	0,478