

## Repositório ISCTE-IUL

---

Deposited in *Repositório ISCTE-IUL*:

2018-06-06

Deposited version:

Post-print

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Silva, I., Alturas, B. & Carneiro, A. (2017). Ferramentas de modelação UML: avaliação na perspetiva dos utilizadores. In Álvaro Rocha, Bráulio Alturas, Carlos J. Costa, Luís Paulo Reis e Manuel Pérez Cota (Ed.), 12th Iberian Conference on Information Systems and Technologies (CISTI). (pp. 2262-2267). Lisboa: IEEE.

Further information on publisher's website:

10.23919/CISTI.2017.7975818

Publisher's copyright statement:

This is the peer reviewed version of the following article: Silva, I., Alturas, B. & Carneiro, A. (2017). Ferramentas de modelação UML: avaliação na perspetiva dos utilizadores. In Álvaro Rocha, Bráulio Alturas, Carlos J. Costa, Luís Paulo Reis e Manuel Pérez Cota (Ed.), 12th Iberian Conference on Information Systems and Technologies (CISTI). (pp. 2262-2267). Lisboa: IEEE., which has been published in final form at <https://dx.doi.org/10.23919/CISTI.2017.7975818>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

---

### Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

---

# Ferramentas de Modelação UML:

Avaliação na Perspetiva dos Utilizadores

## *UML Modeling Tools:*

*Assessment in Perspective of Users*

Ivanila Silva  
Instituto Universitário de Lisboa  
(ISCTE-IUL)  
ISTAR-IUL  
Lisboa, Portugal  
ivanila\_395@hotmail.com

Bráulio Alturas  
Instituto Universitário de Lisboa  
(ISCTE-IUL)  
ISTAR-IUL  
Lisboa, Portugal  
braulio.alturas@iscte.pt

Alberto Carneiro  
Universidade Autónoma  
de Lisboa & Universidade  
Europeia  
ISTAR-IUL  
Lisboa, Portugal  
albert.carneiro@netcabo.pt

*Resumo* — As ferramentas de modelação servem para orientar e disciplinar o processo de desenvolvimento de *software* durante a fase de projeto. Entretanto, tais ferramentas não são totalmente exploradas do ponto de vista funcional, seja por complexidade do assunto abordado ou pela usabilidade das mesmas. Este trabalho propõe avaliar o uso de três ferramentas de modelação de sistema, por meio de um estudo que utiliza alunos da Pós-Graduação em Informática Aplicada às Organizações. O objetivo do estudo foi avaliar a usabilidade das três ferramentas, que permitem a modelação de sistema utilizando a notação da UML. São apresentados estudos referentes aos níveis de aceitação e de utilização das ferramentas analisadas. Com base nos resultados preliminares deste estudo, pode-se verificar que a ferramenta Visio possui melhor usabilidade e funcionalidade do que as outras ferramentas.

*Palavras Chave* – UML; usabilidade; funcionalidade; avaliação; ferramentas de modelação.

*Abstract* — Modeling tools serve to guide and discipline the software development process during the design process. However, such tools are not fully explored from a functional point of view, either the complexity of the subject matter or the usability thereof. This study proposes to evaluate the use of three system modeling tools, through a study using students of the Postgraduate Program in Applied Informatics to the Organizations. The objective of the study was to evaluate the usability of three tools that allow system modeling using UML notation. Are presented studies on the levels of acceptance and use of the analyzed tools. Based on the preliminary results of this study, it was observed that the Visio tool has better usability and functionality than the other tools.

*Keywords* – UML; usability; functionality; assessment; modeling tools.

### I. INTRODUÇÃO

A modelação é uma parte central de todas as atividades que levam até a implantação de um bom sistema de engenharia [1].

A linguagem UML (*Unified Modeling Language*) pode ser usada em dois níveis diferentes, embora relacionados entre si:

modelação lógica e desenho de sistemas. A modelação lógica foca-se na compreensão dos problemas a um nível abstrato, enquanto o desenho de sistemas inclui preocupações com questões de implementação [2].

Os fornecedores de *software* têm considerado as necessidades evolutivas de criação de ferramentas de modelação de objetos e componentes de tecnologias cada vez mais potentes e de mais fácil utilização. Interessará sempre verificar como é que as ferramentas de modelação de objetos e seus padrões de usabilidade são considerados na prática das diversas indústrias [3].

Vários algoritmos e ferramentas foram desenvolvidas para gerar diagramas UML automaticamente para fins de facilitar a programação. Muitas ferramentas, no entanto, ignoram frequentemente fatores perceptivos no *layout* desses diagramas. Portanto, os utilizadores ainda têm de gastar muito tempo e esforço a reorganizar as caixas e linhas para fazer o diagrama compreensível [4].

Uma das características mais importantes de ferramentas de modelação é a geração de *outputs*. Um *output* pode ser documentação, código fonte, lista de rede ou qualquer outra representação do sistema a ser construído [5].

Hoje em dia todos os analistas de sistemas usam UML, mas cada um usa diferentes ferramentas, pelo que ainda têm de consumir tempo e esforço reorganizando caixas e linhas para fazer um diagrama compreensível. Pretendendo este trabalho avaliar a usabilidade, é necessário considerar critérios-chave e diretrizes para a disposição efetiva de ferramentas UML a partir da perspetiva dos utilizadores. Três ferramentas UML são avaliadas para ilustrar como os critérios podem ser aplicados para avaliar a capacidade de leitura dos diagramas gerados.

### II. UML

A UML é uma linguagem ou notação de diagramas para especificar, visualizar e documentar modelos de *software* orientados por objetos. A UML não é um método de desenvolvimento, o que significa que não diz o que fazer

primeiro ou o que fazer depois ou como desenhar o seu sistema, mas ajuda a visualizar o desenho e a comunicar com outros analistas. A UML é controlada pelo Object Management Group (OMG) e é a norma da indústria para descrever graficamente o *software*.

Fowler definiu UML como uma família de notações gráficas, apoiada por um meta modelo único, que ajuda na descrição e no projeto de sistemas de *software*, particularmente dos que são construídos utilizando o estilo orientado a objetos [6], já para Larman é uma linguagem visual para especificar, construir e documentar os artefactos dos sistemas [7].

#### A. Breve Historia da UML

Desde os primeiros conceitos de orientação a objetos, diversos métodos foram apresentados à comunidade (chegaram a mais de 50 no período compreendido entre 1989 e 1994). Todavia, a grande maioria pecou por tentar estender os métodos estruturados. Os mais prejudicados eram os utilizadores que não conseguiam encontrar uma completa satisfação no uso do que havia disponível. Essa situação gerou o que se conheceu como a “guerra dos métodos”. Aproximadamente, a partir de meados da década de 90 do século XX, passou-se a conviver com novas abordagens que procuravam trabalhar mais ativamente a ideia do novo paradigma. Nessa época, vários nomes consagrados desta área, por meio de suas publicações e métodos, deram grandes contribuições à orientação a objetos [8].

Em meados da década de 1990, Booch, Rumbaugh e Jacobson uniram forças na Rational Software Corporation e começaram a fundir as suas respetivas metodologias para criar o que seria a primeira versão da UML. Começaram a trabalhar com outros metodólogos e empresas para propor uma linguagem padrão de modelação para o OMG, um consórcio que cria e mantém padrões para a indústria de computadores. Em Novembro de 1997, o OMG adotou a UML e o seu desenvolvimento contínuo [9].

Após a aprovação da UML 1.1, várias versões de UML surgiram. O OMG definiu uma equipa de revisão para aceitar comentários públicos sobre a UML e fazer edições e atualizações técnicas para normas. Diversos fornecedores de produtos e serviços começaram a apoiar e a promover a UML com ferramentas, consultoria e livros. Ainda se utiliza a versão 2.4.1, mas já existe desde 2015 a versão UML 2.5. [10].

#### B. Características da UML

A UML oferece os meios para que seja possível modelar qualquer tipo de sistema independentemente da complexidade da área de negócios. A UML é muito abrangente porque foi concebida de forma a adequar-se à especificação de qualquer tipo de sistemas (sistemas de informação, sistemas de tempo real, sistemas mecânicos, sistemas geográficos e outros) e por outro lado integra grande parte dos aspetos estruturais e comportamentais que habitualmente caracterizam os sistemas [11]. Também auxilia na construção de *software*, pois possibilita um mapeamento de seus modelos para a linguagem de programação ou vice-versa. Assim, é possível gerar código a partir de modelos UML, ou com suporte adequado realizar a

engenharia reversa, a qual também facilita a documentação, pois possui suporte para a criação e documentação de vários artefactos que são criados durante o desenvolvimento de um sistema [8].

### III. FERRAMENTAS UML

Existem diversas ferramentas informáticas para fazer modelação usando a linguagem UML [12], tanto Proprietário como Open Source e todas elas auxiliam na modelação sistemas ou atividades de projetos, algumas com mais funcionalidades do que outras. Neste artigo são abordadas três ferramentas CASE (*Computer-Aided Software Engineering* - Engenharia de Software Auxiliada por Computador).

#### A. ArgoUML

ArgoUML é uma aplicação *open source* que foi criada por Jason Elliot Robbins. A ArgoUML usa UML para modelar o desenho de *software* de computador. A aplicação corre na maior parte das plataformas uma vez que é implementada em Java. Está distribuída sob a licença BSD. Provê suporte para quase todos os tipos de diagrama da UML padrão e contém suporte cognitivo [13].

A ferramenta ArgoUML dá suporte para os seguintes diagramas da UML: diagrama de classe, diagrama de estados, diagrama de atividades, diagrama de casos de uso, diagramas de comunicação (colaboração), diagrama de utilização e componentes e diagrama de sequência [14].

#### B. DiaUML

Dia é um programa de desenho, que pode ser usado para criar diagramas. Dia inclui uma variedade de ferramentas básicas como linhas e caixas e pode carregar "folhas" - coleções de ferramentas que são usadas em um tipo específico de diagrama. A maioria dos objetos do diagrama tem identificadores para que as linhas podem ser conectados de maneira a formar uma estrutura gráfica. Quando os objetos são movidos ou redimensionados, as conexões vão seguir os seus respetivos objetos [15].

Os diagramas UML que Dia suporta são diagramas casos de uso, diagramas de classes, diagramas de sequência, diagrama de estados e diagrama de Instalação.

Dia é um diagrama de ferramentas UML simples, livre e útil. Funciona em plataformas UNIX e MS Windows<sup>1</sup>

#### C. Visio

O Microsoft Visio é uma aplicação para a criação de diagramas para o ambiente Windows e é apresentado pelos seus criadores como uma aplicação que confere a qualquer utilizador a capacidade de visualizar e comunicar ideias, informações e sistemas, podendo utilizar ferramentas simples e flexíveis para criar facilmente gráficos técnicos e comerciais que apresentam as suas ideias com grande impacto [16].

O Visio pode ser utilizado para criar diagramas de diversos tipos, como organogramas, fluxogramas, modelação de dados (usando UML ou outra notação gráfica qualquer), diagramas de

<sup>1</sup> <http://www.csci.csusb.edu/dick/cs202/lab02.html#Goals> consultado em 01/09/15

redes, plantas baixas e cartazes. Segundo o *website* da Microsoft, o Visio oferece suporte para a criação dos seguintes tipos de diagramas UML: diagramas de casos de uso, diagramas de pacote, diagramas de atividade, diagramas de estados, diagramas de sequência, diagrama de comunicação (colaboração), diagramas de componentes e diagramas de implementação.

#### IV. USABILIDADE DE FERRAMENTAS UML

##### A. O conceito de usabilidade

O conceito de usabilidade é diversas vezes confundido com a acessibilidade. No entanto, a norma ISO 9241 define a usabilidade como a capacidade que um sistema interativo oferece ao seu utilizador, em determinado contexto operacional para realizar tarefas de modo eficiente, eficaz e agradável, ou seja, uma medida de acordo com a qual um dado produto pode ser usado pelo utilizador para que este possa alcançar objetivos com eficácia e eficiência, contribuindo assim para um nível pretendido de satisfação. Entenda-se por eficácia a precisão e a rapidez como os utilizadores podem atingir esses objetivos, relacionando-se a eficiência predominantemente com os recursos necessários e com a qualidade e o nível de abrangência do percurso até esses objetivos. Os níveis de satisfação estão estreitamente relacionados com a facilidade e o conforto geral obtido através da utilização de um dado produto.

Sendo a qualidade que caracteriza o uso de programas e aplicações [17], a usabilidade de um produto ou de um processo pode interferir nos modos como o utilizador interage com um sistema, sendo conveniente a sua medição ter em consideração a execução de tarefas. Podem ser apontadas cinco tarefas que auxiliam a medição da usabilidade: aprendizagem, eficiência no processo de aprendizagem, memorização, erros e satisfação do utilizador na sua relação com o sistema [18].

##### B. Usabilidade de ferramentas

A adequação da UML para diversos processos tem de ser avaliada também de um ponto de vista da usabilidade das suas ferramentas. Muitas vezes a fraca usabilidade das ferramentas de um dado processo ou linguagem pode causar a rejeição de um *software*, o qual poderia ser aplicável e benéfico se tivesse sido preparado de modo cuidadoso e sistemático no que se refere às interfaces com os utilizadores. Interessa também considerar que, por exemplo, a avaliação empírica de técnicas de engenharia inversa tem de considerar que em muitos casos, a qualidade dos resultados depende muito da maturidade, da robustez e da usabilidade das ferramentas. As diferenças entre os níveis de usabilidade das ferramentas podem dever-se, por exemplo, ao facto de os utilizadores terem maior capacidade de explorar os sistemas e até de elaborar diagramas em tempos mais reduzidos.

As ferramentas da UML devem ser sujeitas a uma avaliação adequada para se saber a sua eficácia em relação aos objetivos dos utilizadores, sobretudo quando desenvolvem *software* [19]. Por exemplo, um dos benefícios devidos a algumas ferramentas é orientar e disciplinar o processo de desenvolvimento de *software*, desde a análise de requisitos e modelagem até à programação e realização de testes [20]. Além disso, também pode haver benefícios como a qualidade do produto final, uma

menor quantidade de código de programação e a melhoria e redução de custos da manutenção.

Tem interessado considerar a interação entre as ferramentas e o utilizador relacionados com o mapeamento das relações entre modelo e códigos e também aspetos de implementação ([21], [22], [23], [24]), enquanto outros trabalhos focaram na usabilidade das ferramentas, enfatizando a relação de IHC-Interação Humano-Computador ([25], [26]).

#### V. METODOLOGIA

A questão inicial deste estudo é: em que medida as ferramentas UML disponíveis no mercado seguem corretamente a norma UML e satisfazem os seus utilizadores?

A função de pesquisa é compreender em que medida as ferramentas UML disponíveis no mercado seguem a norma UML e satisfazem as necessidades dos utilizadores quanto à usabilidade e funcionalidades.

Este estudo da usabilidade pretende cumprir as seguintes exigências:

- Avaliar as ferramentas de modelação UML: ArgoUML; Dia; Microsoft Visio;
- Verificar quais as ferramentas que seguem mais corretamente a norma UML;
- Comparar as ferramentas entre elas quanto à usabilidade e funcionalidade para um analista.

Para isso foi realizado um estudo de avaliação, com utilizadores que conhecem a linguagem UML, que experimentaram três das cinco ferramentas e avaliaram vários pontos relativos à usabilidade e funcionalidades das mesmas. Para concretização do estudo irá ser apresentado um modelo que possibilitará avaliar as ferramentas.

A população são os Alunos da Pós-Graduação em Informática Aplicada às Organizações e a amostra é composta por 13 alunos, 6 do sexo feminino e 7 do sexo masculino. Para recolher os dados foi aplicado um questionário e realizada análise documental.

Para dar respostas às questões levantadas durante o estudo, efetuou-se um questionário junto de alunos que possuíam conhecimentos e experiência do uso da linguagem UML. Os dados obtidos no estudo foram tratados de forma a apurar quais as melhores ferramentas UML, na opinião dos inquiridos.

Optou-se por uma investigação mista, isto é, qualitativa e quantitativa, por também se apresentar um questionário com perguntas fechadas e abertas.

Este estudo encontra-se assim estruturado em três fases:

- Consulta bibliográfica sobre a UML e as suas ferramentas. Com base nessa informação formulou-se um questionário a ser colocado numa segunda fase;
- Na segunda fase aplicaram-se os questionários, após os alunos terem experimentado três ferramentas a partir de um modelo elaborado essencialmente para este estudo, o propósito deste questionário foi captar a informação adquirida pelos alunos;

- Após a realização das fases anteriores foi possível verificar quais as ferramentas que seguem mais corretamente as normas e também permitiu compará-las entre si.

## VI. RESULTADO

### A. Comparação das ferramentas quanto a Usabilidade

#### Produtividade do utilizador

Neste tópico foi feita a comparação quanto à interface, ou seja, a opinião do utilizador sobre a interface das ferramentas. A interface deve permitir um bom desempenho do utilizador na realização de suas tarefas. Não se está falando de desempenho do *software*, que é um atributo de qualidade utilizado na engenharia de software, mas do desempenho do utilizador em sua interação com um sistema de software. Como se pode observar na Figura 1.

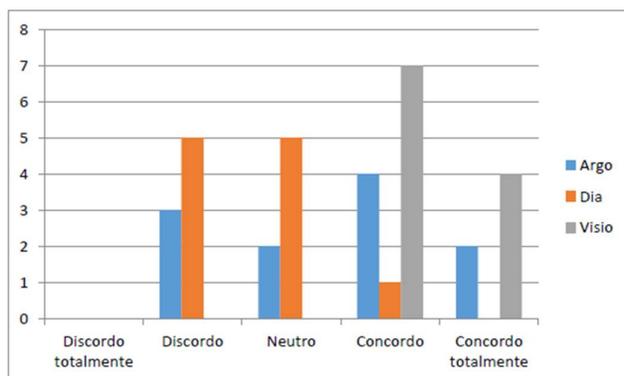


Figura 1. Preferência dos Participantes quanto às interfaces das ferramentas

#### Dificuldade de aprendizagem com as ferramentas

Foi avaliado em quais ferramentas os utilizadores sentiram mais dificuldades.

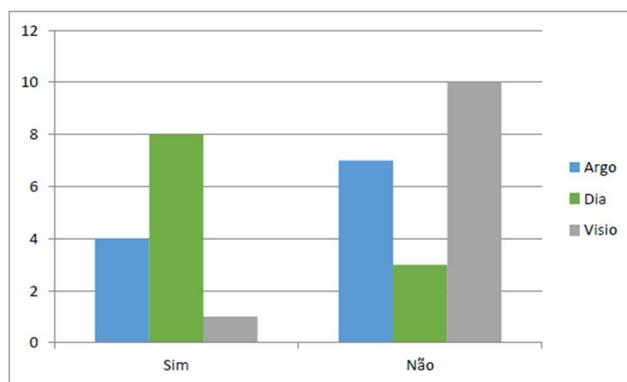


Figura 2. Participantes que sentiram dificuldades de aprendizagem com as ferramentas

Através do relato dos participantes no questionário aplicado, foi analisada a métrica grau de dificuldade de utilização das ferramentas. A verificação foi realizada de duas

alternativas, sim ou não, que podem ser visualizadas no gráfico da Figura 2.

De acordo com os dados recolhidos, quatro participantes sentiram dificuldades de aprendizagem ao utilizar a ferramenta Argo e sete não sentiram dificuldade de aprendizagem ao utilizar a ferramenta Dia, oito participantes sentiram dificuldades de aprendizagem e três participantes não sentiram dificuldades de aprendizagem. Quanto ao Visio apenas um participante sentiu dificuldades de aprendizagem e dez participantes não sentiram dificuldades de aprendizagem. Pelo que se pode concluir que, para a amostra utilizada, a ferramenta Visio é a mais fácil de aprender a utilizar, enquanto a Dia é a mais difícil de aprender a utilizar.

#### Facilidade de utilização

A facilidade de utilização será um complemento para a avaliação anterior dificuldade de aprendizagem. Se foi fácil para o utilizador aprender a utilizar o *software*.

Os participantes envolvidos na experiência foram também questionados sobre a facilidade de utilização das ferramentas.

A questão era classificada pela escala: “discordo totalmente”, “discordo”, “neutro”, “concordo”, “concordo totalmente”. Como pode ser observado na Figura 3, nenhuma das ferramentas foi classificada como “discordo totalmente”. A ferramenta Argo foi classificada como “discordo” por 3 participantes, “neutro” por 2 participantes, “concordo” por 4 participantes, “concordo totalmente” por 2 participantes. Quanto à ferramenta Dia foi classificada com “discordo” por 2 participantes, “neutro” por 4 participante, “concordo” por 4 e “concordo totalmente” por 1. Já no caso do Visio não obteve classificação “discordo” foi classificado como “neutro” por 1 participante, “concordo” por 7 participantes e “concordo totalmente” por 3 participantes.

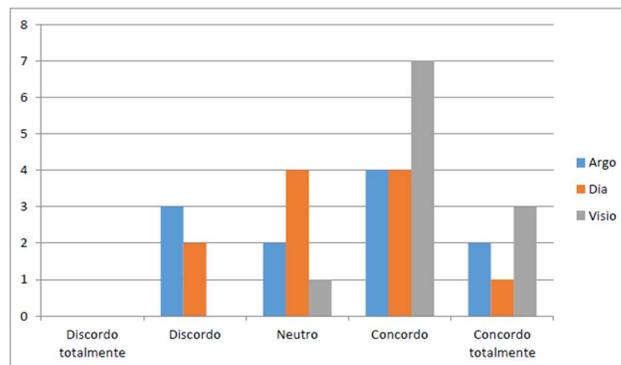


Figura 3. Classificação de facilidade de utilização das ferramentas na experiência

#### Satisfação com as ferramentas

O utilizador deve gostar de utilizar o sistema. Observe-se que a satisfação é um aspeto subjetivo, pessoal, mas ainda assim importante e que deve ser procurado no desenvolvimento de um produto de *software*.

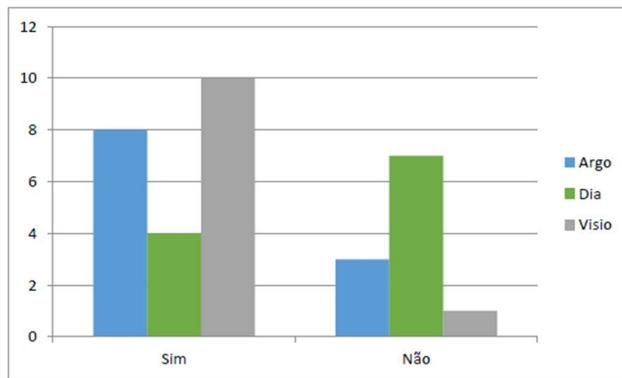


Figura 4. Participantes que usariam novamente a ferramenta ou recomendariam

A satisfação em relação à ferramenta foi analisada através de questões aos participantes quanto à recomendação ou utilização da ferramenta. A verificação foi realizada de duas alternativas, “sim” ou “não”, que podem ser visualizadas no gráfico da Figura 4.

Como mostra o gráfico, no caso de Argo pode-se ver que 8 participantes informaram que “sim” usariam novamente a ferramenta ou recomendariam e 3 participantes informaram que “não usariam novamente a ferramenta ou recomendariam”, já com o Dia foi diferente: 4 participantes informaram que “sim usariam novamente a ferramenta ou recomendariam” e 7 participantes informaram que “não usariam novamente a ferramenta ou recomendariam”, no caso do Visio, 10 participantes da ferramenta Visio informaram que “usariam novamente ou recomendariam a ferramenta”, apenas 1 participante informou que “não usaria novamente ou recomendaria”.

#### B. Comparação das ferramentas quando a Funcionalidade

De acordo com o que foi apurado no questionário, os inquiridos acham que as ferramentas apesar de terem formas de trabalhar diferentes, ajudam a modelar os projetos e a atingir os principais objetivos. A maioria dos inquiridos achou o Dia mais complexo do que o Argo ou Visio; o Dia aparenta ser simples no que respeita à sua interface e representação gráfica, mas para trabalhar é mais complexo do que Argo ou Visio.

Alguns respondentes sentiram-se confusos durante a realização do teste, por exemplo na identificação de algumas representações, como classes associativas e também algumas características/propriedades que a aplicação ia abrindo não eram muito intuitivas, não possibilitando ao utilizador compreender logo o seu objetivo.

Os inquiridos sugeriram algumas funções de que sentiram a necessidade no momento do teste:

- Mensagens de erro quando se efetua ligação incorreta entre as classes;
- O Argo deveria ter Copy + Paste, através do teclado e não exclusivamente através do ícone;
- O Dia deveria possibilitar fazer classes associativas e outros diagramas UML.

## VII. CONCLUSÕES

A UML tem uma ampla visão de utilização, a sua principal função é a modelação de regras de negócio e a especificação de sistemas, compreendendo tanto os aspetos estruturais do *software* como os dinâmicos. Para fornecer esta ampla gama de aplicação, a linguagem foi definida de modo a poder ser suficientemente estendida e genérica para lidar com diferentes tipos de sistemas, evitando assim demasiadas especializações e complexidade.

Também a UML é usada para modelar os mais variados tipos de *softwares*, fornecendo diferentes tipos de visão do sistema, podendo ser aplicada em todas as fases do processo de desenvolvimento, desde o levantamento de requisitos até aos testes finais, passando pela especificação técnica e a implementação.

Com base no que foi apurado nas respostas ao questionário pode-se verificar que a maioria dos inquiridos tem preferência pelo Visio, isso pelo facto de conhecerem o Visio há mais tempo do que as outras duas ferramentas, ArgoUML e Dia.

É possível concluir que os inquiridos ficaram mais satisfeitos, com base nos resultados preliminares deste estudo, que a ferramenta Visio possui melhor usabilidade que a ferramenta ArgoUML e Dia como mostram os gráficos anteriormente apresentados, visto que ela apresentou uma maior facilidade de aprendizado maior produtividade. Ou seja, os participantes rapidamente conseguiram explorar o sistema e elaborar os diagramas num tempo menor do que com as ferramentas ArgoUML e Dia, e tiveram menos dificuldade.

Recorde-se que este estudo da usabilidade pretendeu cumprir as seguintes exigências:

- Avaliar as ferramentas de modelação UML: ArgoUML; Dia; Microsoft Visio;
- Verificar quais as ferramentas que seguem mais corretamente a norma UML;
- Comparar as ferramentas entre elas quanto à usabilidade e funcionalidade para um analista.

O estudo incidiu apenas na avaliação de três ferramentas, mas revelou resultados interessantes e úteis. A nossa recomendação é de que este estudo deva ser realizado novamente, com um quantitativo maior de participantes e/ou de ferramentas UML para que se possa obter um resultado mais significativo e seguro. Para futuros trabalhos apresentam-se as seguintes propostas:

- Ter uma amostra com maior número de participantes;
- Avaliar outras ferramentas;
- Utilizar um formulário de caracterização dos participantes, este formulário visa a avaliação do perfil académico e profissional do inquirido e seu nível de conhecimento em UML;
- Fazer um estudo experimental, com um espaço próprio para elaboração do estudo, com salas preparadas com computadores e ferramentas instaladas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] G. Booch, J. Rumbaugh e I. Jacobson, *The Unified Modeling Language User Guide*, 2nd edition ed., Addison-Wesley, 2005.
- [2] J. Rumbaugh, "ER Is UML," *Journal of Information Systems Education*, vol. 17, nº 1, pp. 21-25, 2006.
- [3] C. Kobryn, "Modeling Components and Frameworks with UML," *Communications of the ACM*, vol. 43, nº 10, pp. 31-38, 2000.
- [4] K. Wong e D. Sun, "On evaluating the layout of UML diagrams for program comprehension," *Software Quality Journal*, vol. 14, nº 3, pp. 233-259, 2006.
- [5] D. Milicev, "Automatic model transformations using extended UML object diagrams in modeling environments," *IEEE Transactions on Software Engineering*, vol. 28, nº 4, pp. 413-431, 2002.
- [6] M. Fowler, *UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos*, 3ª edição ed., Bookman, 2005.
- [7] C. Larman, *Utilizando UML e Padrões*, 3ª edição ed., Bookman, 2005.
- [8] A. C. Melo, *Desenvolvendo aplicações com UML 2.2 – Do conceitual à implementação*, 3ª edição ed., Brasport, 2010.
- [9] G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, J. Conallen e K. A. Houston, *Object-Oriented: Analysis and design with applications*, 3rd edition ed., Person Education, 2007.
- [10] S. Si Alhir, *Learning UML*, O'Reilly & Associates, Inc, 2003.
- [11] P. Ramos, *Desenhar Bases de Dados com UML*, Lisboa: Silabo, 2006.
- [12] B. Alturas, *Introdução aos sistemas de informação organizacionais*, Lisboa: Edições Silabo, 2013.
- [13] A. Ramirez, P. Vanpeperstraete, A. Rueckert, K. Odutola, J. Bennett, L. Tolke e M. van der Wulp, *ArgoUML User Manual : A tutorial and reference description*, Michiel van der Wulp, 2009.
- [14] M. J. Vieira, "Modelagem UML com Software Livre," em *5º Fórum Internacional de Software Livre*, Porto Alegre, Brasil, 2004.
- [15] K. Heise, *Introduction to UML Diagrams in DIA*, Lewisburg, USA: Bucknell University, 2015.
- [16] P. J. Silva, *Microsoft Visio 2003 - Curso Completo*, Lisboa: FCA, 2004.
- [17] W. Cybis, A. H. Betiol e R. Faust, *Ergonomia e usabilidade: Conhecimentos, métodos e aplicações*, 2ª edição ed., São Paulo, Brasil: Novatec Editora, 2010.
- [18] J. Nielsen e R. L. Mack, *Usability Inspection Methods*, New York, USA: Wiley, 1994.
- [19] M. Sensalire, P. Ogao e A. Telea, "Evaluation of software visualization tools: Lessons learned," em *5th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2009)*, Edmonton, Alberta, Canada, 2009.
- [20] L. Issa, C. I. P. S. Pádua, R. F. Resende, S. Viveiros e P. S. Neto, "Desenvolvimento de interface com usuário dirigida por modelos com geração automática de código," em *X Conferencia Iberoamericana de Software Engineering (CIBSE 2007)*, Isla de Margarita, Venezuela, 2007.
- [21] D. Akehurst, G. Howells e K. McDonald-Maier, "Implementing associations: UML 2.0 to Java 5," *Software and Systems Modeling*, vol. 6, nº 1, pp. 3-35, 2007.
- [22] D. Gessenharter, "Mapping the UML2 semantics of associations to a Java code generation model," em *11th international conference on Model Driven Engineering Languages and Systems (MoDELS '08)*, Berlin, Heidelberg, Germany, 2008.
- [23] D. Gessenharter, "Implementing UML associations in Java: a slim code pattern for a complex modeling concept," em *Workshop on Relationships and Associations in Object-Oriented Languages (RAOOL '09)*, New York, USA, 2009.
- [24] Z. Diskin, S. M. Easterbrook e J. Dingel, "Engineering associations: From models to code and back through semantics," em *Technology of Object-Oriented Languages and Systems (TOOLS 2008)*, Zurich, Switzerland, 2008.
- [25] C. S. d. Souza, *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*, The MIT Press, 2005.
- [26] C. S. d. Souza, C. F. Leitão, R. O. Prates, S. A. Bim e E. J. da Silva, "Can inspection methods generate valid new knowledge in HCI? the case of semiotic inspection," *International Journal of Human-Computer Studies*, vol. 68, nº 1-2, pp. 22-40, 2010.