

# Análise do uso da cor no Diagrama de Classes da Linguagem Unificada de Modelagem (UML)

*Analysis of the use of color in the Unified Modeling Language Class Diagram (UML)*

Rodrigo Diego de Oliveira

---

uml, diagrama de classes,  
linguagem gráfica, cor

O uso da cor em diagramas na Engenharia de Software não é novidade, em especial na Linguagem Unificada de Modelagem (UML), que consiste em um conjunto de quinze diagramas utilizados para documentar e especificar a complexidade de um software. Porém, a aplicação da cor muitas vezes é feita de forma aleatória ficando a cargo da subjetividade do projetista ou dos softwares utilizados para o desenho destes diagramas. O presente artigo aborda o uso das cores no diagrama de classes da UML segundo o modelo proposto por Coad, Lefebvre e De Luca (1999), que aplica a cor a partir de uma prática profissional utilizando post-its. O objetivo deste artigo é realizar uma análise descritiva da aplicação da cor para demonstrar a importância desta não só para melhorar a estética, mas também como símbolo, facilitando a leitura, compreensão e o estabelecimento de relações no diagrama de classes.

uml, class diagram,  
graphic language, color

*The use of color in diagrams in Software Engineering is not new, especially in the Unified Modeling Language (UML), which consists of a set of fifteen diagrams used to document and specify the complexity of a software. However, the application of color is often done in a random way, being in charge of the subjectivity of the designer or the software used to design these diagrams. This paper discusses the use of colors in the UML class diagram according to the model proposed by Coad, Lefebvre and De Luca (1999), who apply color from a professional practice using post-its. The purpose of this article is to perform a descriptive analysis of the application of color to demonstrate the importance of this not only to improve aesthetics, but also as a symbol, facilitating reading, understanding and establishing relationships in the class diagram.*

---

## 1 Introdução

O desenvolvimento de *software*, também conhecido como programas de computador, aplicativos ou sistemas, é uma atividade complexa,

exige do projetista uma série de habilidades que vão além da compreensão das linguagens de programação e bancos de dados. Na Engenharia de *Software* uma das etapas fundamentais é chamada de análise de requisitos, onde o projetista vai até o cliente e faz o levantamento das necessidades deste para então iniciar o projeto de um *software*. Durante esta etapa, toda a informação deve ser registrada, pois são elas que vão determinar o quanto o *software* será eficiente e sua qualidade para o cliente final. Uma prática comum nesta fase é utilizar notas autoadesivas (*post-its*) para agrupar as informações, facilitando assim a organização e o entendimento destas. Trata-se de uma prática não só na área de tecnologia conhecida como métodos ágeis, mas também outras áreas ligadas ao planejamento estratégico, inovação, etc.

Após a etapa inicial, o projetista faz uma documentação técnica dos dados coletados utilizando uma série de representações esquemáticas (Diagramas), que auxiliam na compreensão das relações entre as informações, norteando os programadores durante o desenvolvimento do software. Estes diagramas são elaborados quase sempre a partir de uma linguagem conhecida como UML (*Unified Modeling Language*), que não é uma linguagem de programação, mas sim um conjunto de diagramas compostos por uma série de símbolos gráficos com significados bem específicos.

Posto isto, o presente artigo tem como objetivo demonstrar o uso da cor no Diagrama de Classes, o principal diagrama da UML, conforme a proposta de Coad, Lefebvre e De Luca (1999), ilustrando a transposição do uso da técnica do *post-it* para a representação final do diagrama em um software de modelagem, apontando assim a importância da cor como informação e o quanto esta permite uma melhor compreensão deste. Faz-se também na fundamentação, a relação dos diagramas da UML com a perspectiva do Design a partir da visão de Twyman (1979), Horn (1998), das variáveis visuais de Bertin (1983), entre outros.

Este artigo está dividido em quatro seções: Introdução, Fundamentação, Considerações finais e Referências. Na Introdução aborda-se o objetivo e a contextualização. Em Fundamentação fez-se o desenvolvimento a partir da descrição de uma série de conceitos e abordagens como: a cor como informação, diagramas e sua relação com a linguagem gráfica, o que é UML, o que é o Diagrama de Classes, uso da cor na UML, práticas profissionais e sua aplicação conforme o modelo proposto por Coad, Lefebvre e De Luca (1999). Em Considerações finais é feita uma análise da pesquisa demonstrando o uso da cor no Diagrama de classes. Por fim, as Referências listam os principais autores utilizados na pesquisa.

## 2 Fundamentação

### 2.1 Cor como informação

Segundo Quattrer e Gouveia (2019), a cor é uma importante ferramenta de comunicação visual, capaz de organizar e hierarquizar dados. Esta pode atribuir ou receber atribuição de sentido e valor de acordo com o contexto em que é aplicada, ou de acordo com o acorde cromático em que está inserida. No design gráfico as funções da cor vão desde atrair, manter a atenção, transmitir informação ou fazer com que uma informação seja lembrada (Berry e Martin, 1994).

Em outra visão, Pereira (2011) aborda a cor como informação funcionando como um signo. O autor explica com base na classificação peirceana que as conexões que unem as cores àquilo que elas representam se dividem em icônicas, indiciais e simbólicas. Para ela, a associação é icônica quando a cor representa o objeto por semelhança (e.g., o azul representando o céu); a cor é do tipo índice quando antecipa algo, atuando como um indicador (e.g. a coloração verde de uma fruta indicando que ela ainda não está madura); e a relação é simbólica quando a cor se refere ao objeto por convenção (e.g. vermelho denotando perigo ou amor).

Jacques Bertin (1967), um dos pioneiros na busca pela normatização da representação gráfica com foco na comunicação, explana que a leitura de uma tabela de dados por exemplo, exige vários pontos de apreensão de uma informação, contudo se os dados forem representados graficamente a comparação é facilitada e exige apenas um instante de percepção. Na representação gráfica de Bertin (1983), as informações são determinadas por duas variáveis de localização e seis de qualificação (Figura 1), onde a cor é apresentada como uma das variáveis de qualificação, servindo como expressão da relação entre os objetos a serem representados.

Variáveis Visuais		Propriedade Perceptiva Mais Significativa
Tamanho		Quantitativa - manifestação pontual, linear e zonal
Valor		Ordenada - manifestação pontual, linear e zonal
Granulação		Ordenada - somente manifestação zonal com legenda de até quatro classes
Cor		Seletiva - manifestação pontual, linear e zonal
Orientação		Seletiva - manifestação pontual e zonal com legenda de até quatro classes
Forma		Seletiva - manifestação pontual

**Figura 1** Adaptação do quadro de variáveis visuais de Bertin. Fonte: Rosolém (2017).

## 2.2 Diagramas

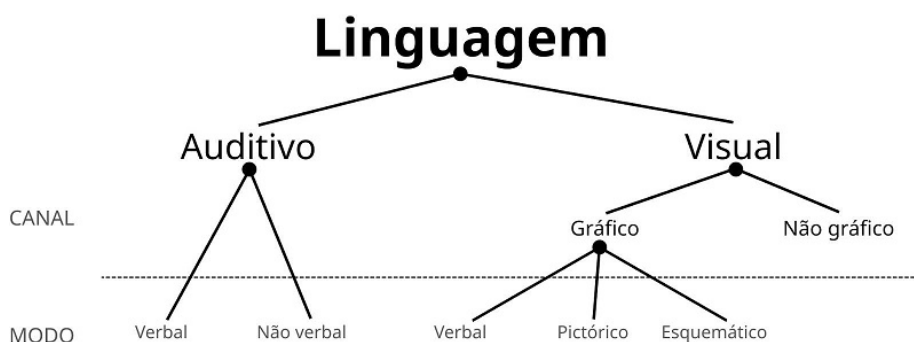
O conceito de diagrama é oscilante, varia muito de autor para autor. No dicionário da língua portuguesa Michaelis (2019), o diagrama é uma “representação gráfica de certos fatos, fenômenos ou relações científicas, sociais, econômicas ou mecânicas por meio de figuras geométricas (pontos, linhas, áreas etc.); gráfico, esquema”. Para Richards (1984), mapas, tabelas e gráficos, entre outras representações gráficas, seriam tipos de diagramas. Em outra perspectiva, Bertin (1983) considera que gráficos e tabelas poderiam ser incluídos no âmbito dos diagramas, enquanto que um mapa seria algo distinto (Engelhardt, 2002, p.146). Por fim, na visão de Gudwin (2010) eles podem ser entendidos como “grafos contendo nós, caminhos entre os nós e textos”.

No Design, a partir dos conceitos da linguagem gráfica (Twyman, 1979) e linguagem visual (Horn, 1998), podemos definir de forma ampla, que um diagrama é uma representação que combina elementos esquemáticos e verbais, e que essa junção permite então entender toda a complexidade de um problema, cenário ou situação e suas relações de forma visual.

Segundo Horn (1998) a linguagem visual é a habilidade de expressar coisas difíceis ou impossíveis de serem ditas na linguagem falada, resultante da interação entre a linguística e os elementos visuais. Ele faz uso de diferentes elementos visuais para demonstrar esta interação: texto (classificado como verbal), imagem e forma (ambos classificados como visuais). Nesta visão, os textos possuem a capacidade de nomear, definir, discutir abstrações, classificar elementos e dão formato ao conceito da comunicação; as imagens são representações gráficas da realidade; e as formas possuem um caráter abstrato em relação à realidade. Na teoria de Horn (1998) a comunicação é constituída de três dimensões funcionais: sintática (estrutural), semântica (significado) e pragmática (uso). É na semântica que este explana sobre o significado da linguagem visual e como estes elementos representam um conteúdo.

Em outra abordagem, Michael Twyman (1979) vê a linguagem como um veículo de comunicação, e que podemos dividir este veículo nos canais auditivo (verbal ou não-verbal) ou visual (gráfico ou não-gráfico). O gráfico, uma ramificação do canal visual, é composto por marcas produzidas com a intenção de comunicar uma mensagem, sendo representado por três modos de simbolização: verbal, pictórico e esquemático (Figura 2). Nessa perspectiva, a linguagem verbal é a representação gráfica da linguagem falada (seja ela tipográfica ou escrita à mão), a linguagem esquemática é constituída por formas gráficas que não incluem palavras, números ou imagens pictóricas (tabelas, diagramas, representações abstratas de estrutura, etc.), e a linguagem pictórica comporta imagens produzidas artificialmente “que remetem por mais remota que seja à aparência ou estrutura de algo real ou imaginado” (Twyman, 1985, p. 249). Semelhante

ao indicado por Horn (1998), o diagrama seria uma combinação de elementos verbais e esquemáticos.



**Figura 2** Modos de simbolização da linguagem. Fonte: do próprio autor (2019), adaptado de Twyman (1979).

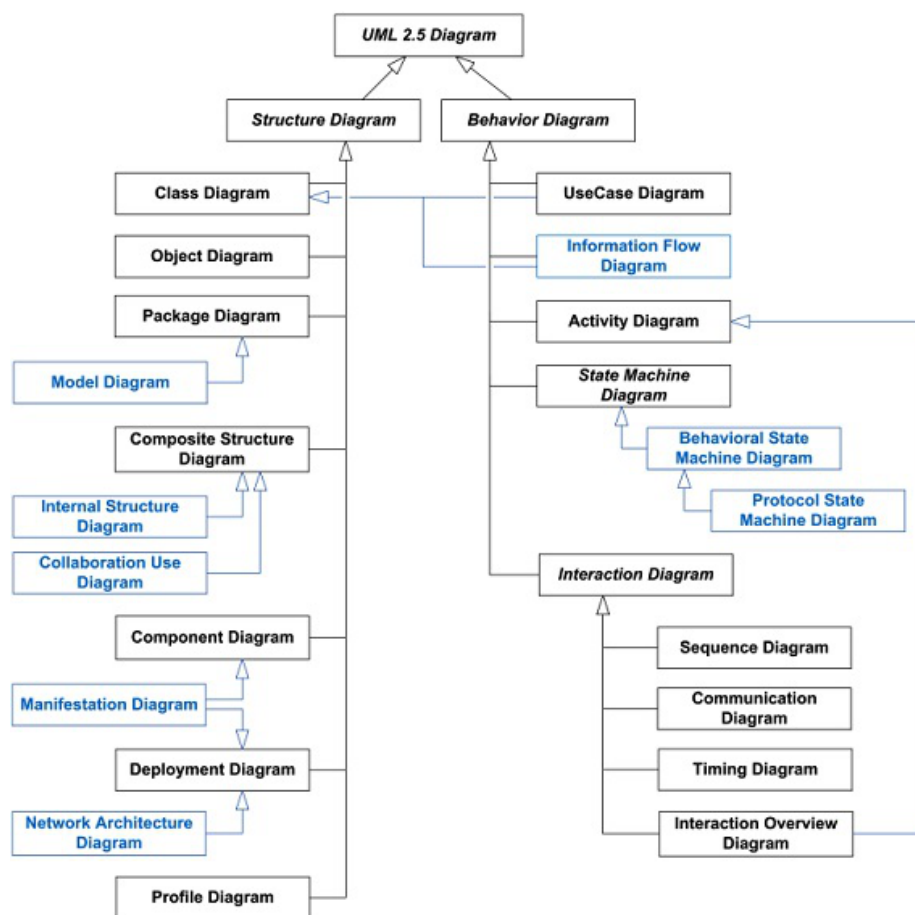
No entender de Carvalho e Aragão (2012, p. 163.), os três elementos que compõem a linguagem visual proposta por Horn (1998) são semelhantes a classificação de linguagem gráfica apontada por Twyman (1979). Podemos interpretar a linguagem verbal como texto, visto que se trata da representação da linguagem falada. As representações pictóricas e esquemáticas equivalem a descrição feita por Horn (1998), sendo a primeira composta por reproduções de imagens e a segunda por formas que não incluem palavras ou imagens. Todavia, a diferença das teorias está na argumentação de Horn (1998) afirmando que textos, imagens ou formas quando sozinhas, não compõem uma linguagem visual. Em outras palavras, um texto apesar de comunicar não é algo visual, as formas não conseguem comunicar sozinhas mesmo sendo visuais e, por fim, uma imagem descontextualizada possui uma comunicação artística (Carvalho e Aragão, 2012, p. 163).

### 2.3 O que é UML?

A UML (*Unified Modeling Language*) é uma linguagem composta de diagramas esquemáticos para especificação, documentação, visualização e desenvolvimento de sistemas complexos de *software*. Para os idealizadores desta linguagem Booch, Rumbaugh e Jacobson (2012), ela é mais que um punhado de símbolos gráficos, cada um possui uma semântica bem definida o que facilita a sua compreensão por profissionais da área. A UML é muito utilizada para construir modelos que expliquem as características ou o comportamento de um sistema de *software*, contribuindo na identificação das características e funcionalidades que este deverá prover, bem como no planejamento de sua construção (Guedes, 2014). No contexto da UML, um modelo é uma representação simplificada da realidade (Barcelar, 2014).

Historicamente, a primeira versão da UML surgiu em junho de 1996 a partir da unificação de três métodos: Booch'93 de Grady Booch, OMT-2 de James Rumbaugh e OOSE de Ivar Jacobson que ficaram conhecidos posteriormente como “Os Três Amigos” (Guedes, 2014, p. 16). Entretanto foi somente em 1997, com a aprovação da *Object Management Group* (OMG) e o apoio de grandes empresas que esta linguagem foi concretizada e oficializou-se como um padrão mundial (Barcelar, 2014).

Atualmente a UML encontra-se na versão 2.5, lançada em 2015, e compreende um conjunto de 15 diagramas classificados em estruturais e comportamentais. O primeiro enfatiza a estrutura a partir da identificação dos objetos e colabora para uma modelagem estática do sistema abstraindo elementos do mundo real. O segundo (comportamental) faz o mapeamento das mudanças ocorridas em um sistema ao longo do tempo (Fakhroutdinov, 2016). Os diagramas da UML 2.5 podem ser categorizados hierarquicamente como demonstrado na (Figura 3) a seguir.







**Figura 3** Diagramas da UML, exceto os elementos em azul que não fazem parte taxonomia da linguagem. Fonte: (Fakhroutdinov, 2016).

## 2.4 Diagrama de Classes









Segundo Booch, Rumbauch e Jacobson (2012), o Diagrama de Classes é considerado um dos mais usados e importantes na Engenharia de *Software*. Na visão de Lobo (2009, p. 53), este “é um dos diagramas mais conhecidos da UML. Sem ele, não será possível definir os outros diagramas do projeto, já que o diagrama de classes é essencial na criação do modelo”.

Através do diagrama de classes é possível visualizar como serão estruturadas as tabelas de um banco de dados e seus relacionamentos. Além disso, este permite visualizar de forma facilitada as operações que devem ser executadas pelo sistema através das entidades, que apresentam atributos e as operações disponíveis (Lobo, 2009, p. 53). Evidentemente que, para um usuário sem conhecimento a respeito de engenharia de *software* este pode não fazer sentido, sendo necessário um repertório mínimo para sua compreensão, da mesma forma que o diagrama não surge do nada, mas sim a partir de uma série de informações levantadas previamente (Análise de requisitos).

De forma resumida, o diagrama de classes oferece três perspectivas de acordo com o observador: conceitual, especificação e implementação. A conceitual representa os conceitos do domínio em estudo, tem uma perspectiva destinada ao cliente com uma quantidade reduzida de informações. A especificação tem foco nas interfaces da arquitetura e métodos, é destinada às pessoas que não precisam saber detalhes de desenvolvimento, tais como gerentes de projeto, também com informações reduzidas. A última, de implementação, é completa e aborda vários detalhes como navegabilidade, tipo dos atributos, etc. (Sampaio e Rocha Neto, 2019). Um diagrama de classes é representado graficamente em duas categorias: entidades (Quadro 1) e relacionamentos (Quadro 2), que somam um conjunto de símbolos verbais e esquemáticos (combinados ou não).

Nome	Descrição	Representação gráfica
Classe concreta	Uma classe é representada na forma de um retângulo, contendo duas linhas que separam 3 partes. A primeira contém no nome da classe, a segunda os atributos da classe e a última os métodos da mesma.	
Classe abstrata	A representação de uma classe abstrata em UML é quase igual à representação de uma classe concreta, a única diferença é o estilo da fonte do nome da classe, que, neste caso, está em itálico.	
Interface <i>Icon</i>	Elemento do modelo que define um conjunto de operações que outros elementos do modelo devem implementar.	
Interface <i>Label</i>	Elemento do modelo que define um conjunto de operações que outros elementos do modelo devem implementar.	

**Quadro 1** Representações gráficas da Entidades do diagrama de classes. Fonte: Sampaio e Rocha Neto (2019).

Nome	Descrição	Representação gráfica
Papel	Descreve o relacionamento de forma textual.	
Multiplicidade	Descrição textual onde: 0..1 significa zero ou uma instância. A notação n..m indica n para m instâncias; 0..* ou * significa que não existe limite para o número de instâncias; 1 é exatamente uma instância; e 1..* significa ao menos uma instância.	
Associação	Na perspectiva conceitual a linha define um relacionamento entre duas entidades. Na perspectiva da especificação a linha representa as responsabilidades entre duas classes, implicando que existem métodos que tratam desta responsabilidade. Na perspectiva da implementação utiliza-se a linha com seta, o que permite saber quem está apontando para quem e em qual lado que está a responsabilidade.	 
Herança	Quando uma entidade herda atributos de outra entidade (classe).	
Agregação	Utilizada apenas na perspectiva de implementação. Agregação é uma associação em que um objeto é parte de outro, de tal forma que a parte pode existir sem o todo.	
Composição	Utilizada apenas na perspectiva de implementação. Em baixo nível, significa passagem por parâmetro ou valor. Enquanto que agregação seria uma passagem por referência.	
Implementação	Utilizado para indicar que uma classe implementa uma interface.	

**Quadro 2** Representações gráficas dos relacionamentos do diagrama de classes.  
Fonte: Sampaio e Rocha Neto (2019).

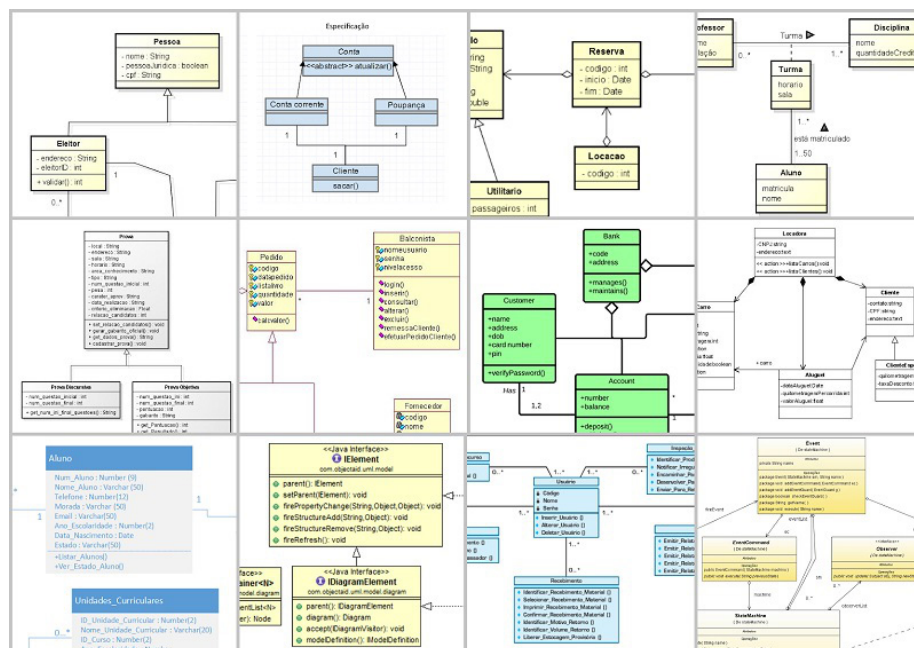
## 2.5 O uso da cor na UML

O uso da cor nos símbolos ou representações da UML não é novidade, mas é, na maioria das vezes, aleatório. Isto é evidenciado por Vieira (2003, p. 30), ao dizer que é preciso “fazer uso de anotações e cores para salientar aspectos semânticos e estruturais importantes”, porém este não informa quais cores ou se existe um padrão a ser seguido. Em concordância Booch, Rumbaugh e Jacobson (2012) abordam o uso de notas e cores como indicação visual para chamar a atenção, porém pedem parcimônia visto que muita gente não vê essas cores e, que elas não devem ser usadas para informações essenciais. Da mesma forma, estes não apresentam um padrão de aplicação de cores. Por fim, no tutorial sobre a ferramenta de modelagem ArgoUML, Nascimento (2010) diz em uma das suas explicações “vamos mudar a cor da classe Aluno, para isso clique na classe e em seguida na aba *Presentation*, nela você pode mudar a cor de fundo e da borda, escolha uma cor de seu gosto”.

Para exemplificar esta disparidade, fez-se uma pesquisa através do Google Imagens pela palavra “diagrama de classes”, coletou-se então uma série de diagramas de forma aleatória para montagem de um *moodboard* (Figura 4) que, segundo Teixeira (2014, p. 26) é “uma coleção de imagens e referências que eventualmente se transformarão no estilo visual do produto em questão”. Neste caso, trata-se de uma representação do estado da arte dos diagramas através



do recorte de diversas imagens para ilustrar um determinado contexto (uso das cores).



**Figura 4** Moodboard com o estado da arte dos diagramas de classe da UML. Fonte: elaborada pelo autor (2019).

Percebe-se que não há uma padronização no uso das variáveis visuais, a aplicação da cor por exemplo, fica a cargo da subjetividade do “projetista” ou pela imposição da ferramenta utilizada na modelagem, aplicou-se até mesmo cores em degradê. Isso é evidenciado também na variação da espessura das linhas e na cor destas, umas coloridas e outras não. A linguagem verbal também apresenta variações de tamanho e cor. Até mesmo os elementos pictóricos foram contemplados em alguns diagramas como substitutos de elementos verbais.

## 2.6 O uso de notas autoadesivas (post-its) na UML

Uma prática comumente utilizada na etapa de análise de requisitos, que é quanto o projetista vai até o cliente levantar as informações sobre a necessidade deste para então modelar o *software*, é utilizar notas autoadesivas para organizar a informação, alguns fazendo agrupamentos por cores e outros apenas utilizando o *post-it* na cor amarela (padrão). Heat (2016) menciona que em métodos ágeis de desenvolvimento de *software*, é comum colocar *post-its* das classes em quadros para que fiquem visíveis para todos e assim, de forma colaborativa, seja fácil alterar o diagrama de classes movendo, adicionando ou removendo as notas. Um exemplo prático desta técnica é demonstrado por Junior (2011), onde este apresenta imagens de um treinamento realizado na empresa de softwares para

negócio ICR, sobre Scrum (método ágil para gestão e planejamento de softwares) e UML. Neste treinamento foram criados vários diagramas da UML utilizando *post-its*, entre eles o diagrama de classes conceitual (Figura 5).



Figura 5 Diagramas da UML a partir de *post-its*. Fonte: Rahal Junior (2011).

Percebe-se que este processo de organização através de *post-its* estimula a discussão e melhora compreensão sobre o problema que precisa ser resolvido. Este fato é relatado pelo engenheiro de *software* Koning (2015, tradução nossa) ao citar que “É ótimo se reunir e ter uma compreensão compartilhada do “fluxo” do software”, após realizar uma sessão para construção de um diagrama a partir do uso de *post-its* (Figura 6).

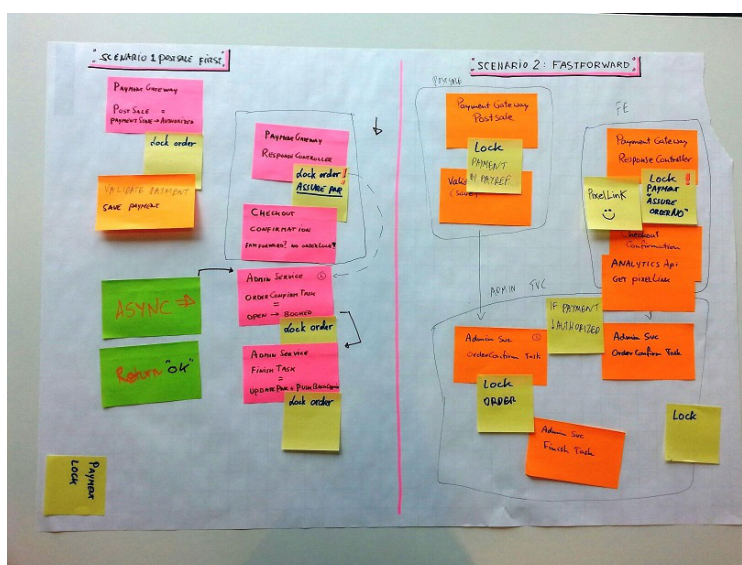


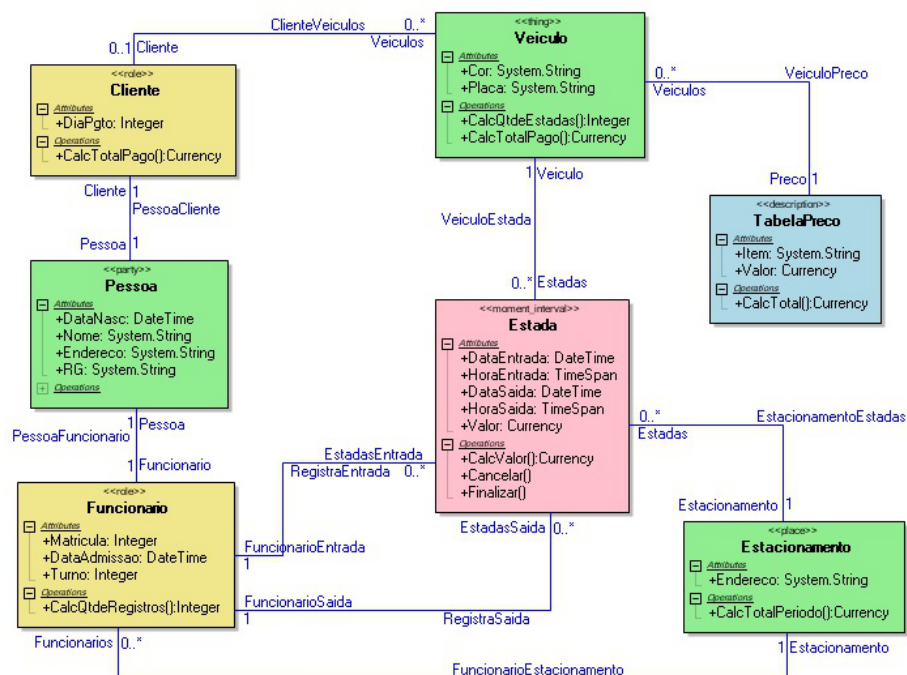
Figura 6 Diagrama antes e depois a partir de *post-its* coloridos. Fonte: Koning (2015).

## 2.7 O modelo proposto por Coad, Lefebvre e De Luca (1999)

Na UML, Coad, Lefebvre e De Luca (1999) foram os primeiros a abordar o uso da cor agregando não só estética, mas atribuindo valor para esta. Em 1997, em um projeto de *software* em Singapura, estes perceberam que as classes seguiam um padrão de relacionamento o qual chamaram de *archetypes* (arquétipos) e, em 1999 lançaram um livro chamado “*Java Modeling in Color with UML*” abordando a técnica desenvolvida. Foram adotadas quatro cores: rosa, amarelo, verde e azul, semelhantes as notas autoadesivas (*post-its*), comumente utilizadas por projetistas para organizar informações ou ideias nas etapas iniciais do projeto de *software* (Seção 2.6). Os autores apontam quatro modelos de arquétipos, um para cada cor, com relação às classes do diagrama:

- Rosa para o arquétipo *Moment-Interval* (momento ou intervalo) que, de forma simplificada, são as atividades, eventos, serviços ou coisas que precisam ser registradas, por exemplo: o registro de uma venda, de que um objeto guarda outro temporariamente, de uma locação, de uma reserva, etc.
- Amarelo para o arquétipo *Role* (papel), que são os papéis envolvidos, por exemplo: funcionário, fornecedor, vendedor, segurança, atendente, etc.
- Verde para o arquétipo *Party, Place, Thing* (pessoa, lugar, coisa ou objeto), que são as classes do modelo que são tangíveis e unicamente identificáveis, atuam em um evento, serviço ou momento. Por exemplo: cadastros diversos e relatórios.
- Azul para o arquétipo *Description* (descrição), que pode ser um catálogo ou um conjunto de rótulos de um objeto, ou seja, são os dados de referências que utilizamos em listas ou componentes de interface. Por exemplo uma lista de opções em um formulário na internet.

Na proposta de Coad, Lefebvre e De Luca (1999), a cor funciona como uma extensão da UML, são camadas extras de informação capaz de expressar mais informações sem ocupar mais espaço (Figura 7).



**Figura 7** Diagrama classes com arquétipos em cores utilizando software específico para modelagem. Fonte: Retamal (2013).

### 3 Considerações finais

O emprego das cores segundo a proposta de Coad, Lefebvre e De Luca (1999), demonstram que estas quando aplicadas nas classes do diagrama da UML vão além do simples colorir, elas apresentam métodos e atributos comuns para cada arquétipo e que, após algumas modelagens a recorrência destes arquétipos deve ficar mais evidente. Percebe-se que as cores funcionam como uma extensão da linguagem da UML, acrescenta-se informações sem ocupar mais espaço. O grande *insight* dos autores a respeito do uso da cor está na apropriação de uma prática comum dos projetistas nas etapas iniciais, onde utilizam-se as notas autoadesivas (*post-its*) para organizar informações e melhor compreender a relação entre elas. As cores propostas em tons pastéis (rosa, amarelo, verde e azul), são as cores destes *post-its* e são facilmente encontradas nas papelarias.

Outra vantagem, é que a cor pode deixar de ser usada de forma aleatória conforme visto na Seção 2.5, criando um padrão visual que apresenta uma estética agradável e facilmente identificada pelos profissionais da área. A leitura do diagrama fica agradável se comparada com o mesmo sem o uso da cor ou com a aplicação de uma mesma cor para todas as classes. A falta de contraste e de grupos por cor exige uma aproximação e mais atenção para identificar os diferentes tipos.

Avaliando o uso da cor na perspectiva de Quattrer e Gouveira (2019), o emprego destas melhoram a organização e hierarquização da informação, sendo que a cor neste caso recebeu a atribuição de valor

através dos arquétipos, e foram escolhidas a partir de um contexto prático. A cor como signo segundo a visão peirceana apontada por Pereira (2011), mostra que as cores rosa, amarela, verde e azul das notas autoadesivas foram escolhidas por convenção, ou seja, de forma simbólica a partir de uma prática profissional comumente utilizada, portanto seria uma atribuição pragmática, baseada no uso. De qualquer forma, este simbolismo permite que o profissional transporte as relações de uma dinâmica física que é a etapa inicial de modelagem e análise de requisitos com fitas autoadesivas, para o ambiente virtual, quando este for desenhar os diagramas utilizando ferramentas específicas com esta finalidade.

Como variável de qualificação (Bertin, 1983), a cor serviu para expressar as relações entre os objetos, pois é nítida a capacidade de relacionar os arquétipos através da cor melhorando a leitura e a compreensão. Nota-se também, que as variáveis visuais podem ser exploradas além da cor, mas também nas linhas e elementos verbais, evidenciado na Seção 2.5 onde constatou-se o emprego aleatório da cor.

Por fim, os diagramas da UML continuam sendo complexos para pessoas que não pertencem a esta área, mas o uso da cor permitiu um ganho não só estético, mas também trouxe significado para cor e uma utilidade prática que irá auxiliar no desenvolvimento e compreensão deste. Porém, para que o emprego da cor de forma aleatória seja eliminado, será preciso que os profissionais da área adotem a prática e façam desta um padrão universal como o pregado pela linguagem UML.

## Referências

- Barcelar, R. R. (2014). Modelagem de Sistemas Orientada a Objetos com UML. Disponível em: <[http://www.ricardobarcelar.com.br/aulas/eng\\_sw/mod3-uml.pdf](http://www.ricardobarcelar.com.br/aulas/eng_sw/mod3-uml.pdf)>. Acesso em: 10 maio de 2019.
- Bertin, J. (1967). *Sémiologie Graphique: Les diagrammes, les réseaux, les cartes*. Paris: Gauthier-Villars.
- Bertin, J. (1983). *Semiology of graphics*. Translated William J. Berg. London: The University of Wisconsin Press Ltd.
- Berry, S.; Martin, J. (1994). *Diseño y color: cómo funciona el lenguaje del color y cómo manipularlo en el diseño gráfico*. Trad. Gloria Prieto Puentes. 1a ed. Barcelona, Espanha: Blume.
- Booch, G; Rumbaugh, J.; Jacobson, I. (2012). Uml - Guia do Usuário, Tradução da Segunda Edição. Elsevier Brasil.
- Carvalho, J.; Aragão, I. (2012) Infografia: Conceito e Prática. São Paulo: InfoDesign.
- Coad, P., Luca, J.D. and Lefebvre, E. (1999). *Java Modeling Color with Uml: Enterprise Components and Process with Cdrom*. Prentice Hall PTR, Upper Saddle River.
- Engelhardt, Y. (2002). *The Language of Graphics: a framework for the analysis of syntax and meaning in maps, charts and diagrams*. Amsterdam: ILLC- Publications.
- Fakhroutdinov, K. (2016). *UML 2.5 Diagrams Overview*. Disponível em: <<https://www.uml-diagrams.org/uml-25-diagrams.html>>. Acesso em: 10 maio de 2019.
- Gudwin, R. R. (2010). Introdução à Linguagem UML. São Paulo: Unicamp. Disponível em: <<http://www.dca.fee.unicamp.br/~gudwin/ftp/ea976/Estruturais2010.pdf>>. Acesso em: 10 mai. 2019.
- Guedes, G. T. A. (2014). UML 2: Guia Prático. São Paulo: Novatec.
- Heat, J. Hermeneutics In Agile Systems Development: Using Foucault, Habermas, and Recueor. UberMann: 2016.
- Horn, R. E. (1998). *Visual Language: Global communication for the 21st Century*. Washington: Macro VU, Inc.
- Koning, M. UML Diagrams are great, Event Storming can be better. 2015. Disponível em: <<http://www.mellekoning.nl/index.php/2015/04/23/uml-diagrams-are-great-event-storming-can-be-better/>>. Acesso em: 10 de maio de 2019.
- Lobo, E. J. R. (2009). Guia Prático de Engenharia de Software: Desenvolva softwares profissionais com uso UML e “best practices” de gestão. São Paulo: Digerati Books.
- Michaelis (2014). Dicionário da língua portuguesa. Disponível em: <[michaelis.uol.com.br](http://michaelis.uol.com.br)>. Acesso em: 15 maio de 2019.
- Nascimento, E. S. (2010). O que é o ArgoUML? Disponível em: <<https://erinaldosn.files.wordpress.com/2010/11/argouml.pdf>>. Acesso em: 10 maio de 2019.
- Pereira, C. P. (2011). A cor como espelho da sociedade e da cultura: um estudo do sistema cromático do design de embalagens de alimentos. Tese (Doutorado) – Universidade de São Paulo. São Paulo.

- Quattrer, M.; Gouveia, A. P. S. O uso das cores em infográficos de divulgação científica. Disponível em: <[https://www.iar.unicamp.br/lis/dcf/uso-das-cores-em-infograficos\\_MilenaQuattrer.pdf](https://www.iar.unicamp.br/lis/dcf/uso-das-cores-em-infograficos_MilenaQuattrer.pdf)>. Acesso em: 10 de maio de 2019.
- Rahal Junior, N. A. S. Scrum e UML - Empresa IRC. 2011. Disponível em: <<http://blogdoabu.blogspot.com/2011/01/scrum-e-uml-empresa-icr.html>>. Acesso em: 10 maio de 2019.
- Retamal, A. Análise e Desenho Orientados por Objetos (2/4). 2013. Disponível em: <<https://edn.embarcadero.com/article/33736>>. Acesso em: 10 de maio de 2019.
- Richards, C. (1984). *Diagrammatics: An investigation aimed at providing a theoretical framework for studying diagrams and for establishing a taxonomy of their fundamental of Art*. Londres.
- Rosolém, N. P. (2017). Um Breve Histórico Sobre os Estudos da Semiologia Gráfica no Brasil. Londrina: Geografia.
- Sampaio, M. C.; Rocha Neto, E. (2019). Material sobre UML: Disciplina Sistemas da Informação II. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/>>. Acesso em: 10 de maio de 2019.
- Teixeira, F. (2014). Introdução e boas práticas em UX Design. São Paulo: Casa do Código.
- Twyman, M. (1979). *A schema for the study of graphic language*. In: Paul A. Kolers, Merald E. Wrolstad; Herman Bouma (Ed.). *Processing of visible language*. Nova York & Londres: Plenum Press.
- Twyman, M. (1985). *Using pictorial language: a discussion of the dimensions*. In: Dufty, Thomas M. & Waller, Robert (Ed.). *Designing usable text*. Orlando, Florida: Academic Press, pp.245-312.
- Vieira, R. (2003). Uso da UML na Especificação do Sistema de Informação da Área de Infra-Estrutura da UFRGS. Porto Alegre: UFRGS.

## Sobre o autor

### Rodrigo Diego de Oliveira

<rddoux@gmail.com>

Universidade Federal do Paraná – UFPR

PPGDesign – Programa de Pós-Graduação em Design

Rua General Carneiro, 460, Centro – Curitiba/PR

Artigo recebido em 06/01/2020,

aprovado em 27/02/2020.