

# Os meus, os seus, os nossos: reuso de exemplo e a autoexpressão de alunos de programação

**Luana Müller**  
PUCRS  
Porto Alegre, Brasil  
luana.muller@pucrs.br

**Milene Selbach Silveira**  
PUCRS  
Porto Alegre, Brasil  
milene.silveira@pucrs.br

**Clarisse Sieckenius de Souza**  
PUC-Rio  
Rio de Janeiro, Brasil  
clarisse@inf.puc-rio.br

## RESUMO

Exemplos são comumente utilizados como base para ensino e aprendizado na área de programação de computadores. No entanto, vê-se que estudantes de programação (e até mesmo programadores experientes) negligenciam a etapa de compreensão do código e da mensagem que este código está passando, optando por um reuso e readaptação do mesmo para rápida e facilmente atingir seus objetivos. O presente trabalho investiga como alunos de programação usam exemplos em suas atividades e, em caso de reuso de código, como ocorre a apropriação da mensagem destes exemplos e como isto impacta em sua autoexpressão e na mensagem final que estão passando adiante através do programa.

## Palavras-chave

Autoexpressão, programação, engenharia semiótica, uso de exemplos, apropriação.

## ABSTRACT

Examples are commonly used as support to teach and learn programming. However, programming students (even the most experienced ones) neglect the study of code and the understanding of the message conveyed by it, preferring to reuse and adapt the code to achieve their goals. This work investigates how students use examples in their activities and, in cases where it is reused, how the appropriation of the code message happens and how it impacts in programmer's self-expression and his/her final message communicated with the program.

## Author Keywords

Self-expression, programming, semiotic engineering, examples reuse, appropriation.

## ACM Classification Keywords

H.1.2. Information Systems: User/Machine Systems, D.1.m. Programming Techniques: Miscellaneous.

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.

Each submission will be assigned a DOI string to be included here.

## INTRODUÇÃO

“Nada se cria, tudo se copia”. A frase, comumente citada por programadores com os mais diversos níveis de *expertise* refere-se a um hábito comum na programação: o reuso de código. Na programação, em geral, há uma mudança de domínios para os quais as técnicas são usadas, mas as técnicas em si não mudam. Um algoritmo de ordenação, por exemplo, irá funcionar da mesma forma em softwares totalmente diferentes. No entanto, pesquisas [6, 4] mostram que programadores, até mesmo os mais experientes, ignoram algo fundamental a ser feito antes de usar um exemplo: *entender* o exemplo.

Há ainda outro fator a considerar na prática de programação baseada em extenso reuso de código feito por outros: os traços autorais pessoais do programador que emergem, implícita ou explicitamente, no programa. Perspectivas semióticas sobre software e programação em geral [2] e interação humano-computador em particular [12] mostram que estamos diante de uma nova mídia, onde, através de interfaces e comportamentos de sistemas e tecnologias digitais, os produtores revelam-se a e comunicam-se com os usuários [8]. Desta forma, é interessante investigar que tipo de imagem projetada de autor resulta, para os usuários, quando o artefato é construído como um "mosaico de muitas falas", copiadas e reusadas no processo de composição de uma mensagem própria. Ou seja, estamos diante de uma investigação das formas de apropriação do discurso de terceiros na comunicação entre produtores e consumidores de tecnologia mediada pelas interfaces de usuários, e de suas consequências para o andamento deste processo.

Apropriação pode ser definida, da perspectiva sociocultural, como o processo de pegar algo que pertence a outro(s) e fazer disso seu [15]. Mais que isso, identidades são construídas não somente através da apropriação de um discurso, mas também através da forma como este discurso é incorporado no texto do escritor (Coastley e Doncaster (2001) apud [1]).

Neste contexto surgiu a seguinte questão: Como exemplos impactam nos traços de autoexpressão presentes em programas criados por alunos de programação? Para tentar responder à pergunta, iniciou-se uma pesquisa a fim de entender, inicialmente:

- Como alunos de programação, muitos em processo inicial de construção do aprendizado de programação, utilizam códigos de exemplo?

- Nos casos em que exemplos são usados, como os alunos interpretam esses programas?

Com as perguntas acima queremos poder descobrir se os programadores investigados dão sinais implícitos ou explícitos de “apropriação pessoal” do código produzido por terceiros. Algumas das possibilidades seriam: eles não se apropriarem, ou seja, perceberem que o código é a “fala de outro”, encaixada em um contexto próprio (i.e. do programador investigado); eles se apropriarem da fala do outro e perceberem o código resultante como uma “composição própria” única; eles sequer perceberem que a questão do “próprio vs. alheio” se coloca; e assim por diante.

Desta forma, definimos *o meu*, como sendo o discurso do aluno, *os seus*, como o discurso do designer do exemplo e *os nossos*, como o discurso do designer do exemplo que foi compreendido pelo aluno e do qual ele se apropria.

Nas seções que seguem serão abordados: o referencial teórico que embasa essa pesquisa; a metodologia usada para se investigar as questões elencadas; o desenvolvimento da pesquisa e informações encontradas no seu decorrer; a discussão dos resultados, a partir de sua consolidação; e, as conclusões e prospectos para esta pesquisa, seguidos das referências bibliográficas usadas.

## REFERENCIAL TEÓRICO

Nas seções que seguem serão apresentados conceitos fundamentais para a compreensão da pesquisa apresentada.

### Semiótica, Engenharia Semiótica e Comunicação

Semiótica é a ciência dos signos [13], sendo signos definidos por Peirce [11] como “algo que em determinado aspecto ou modo, representa algo para alguém”.

A semiótica estuda os signos e o processo de significação (i.e. como os signos se constituem e ocorrem na realidade). Na tradição seguida por Umberto Eco [3], em particular, a semiótica estuda também como os signos participam do processo de comunicação. Os sistemas de significação são o resultado de uma convenção cultural, ou seja, um certo conjunto de elementos expressivos que são sistematicamente correlacionados, por uma coletividade social, a um determinado conjunto de conteúdos. Tais sistemas de significação constituem a maioria dos códigos usados no processo de comunicação, servindo também de base interpretativa para signos “inventados” por interlocutores em processo de comunicação.

É nesta definição de semiótica baseada em significação e comunicação que a teoria da Engenharia Semiótica foi construída, para tratar do fato e da forma como artefatos computacionais introduzem novos signos ou sistemas de signos no universo do usuário, e como passam em seguida a determinar um processo de comunicação especial, mediado por computação. Com esta teoria é possível caracterizar e entender os fenômenos envolvidos no design, uso e avaliação de sistemas interativos. A interação de pessoas com artefatos digitais é, como mencionado anteriormente, um processo de

comunicação dos produtores destes artefatos (designers e desenvolvedores) com os usuários finais, a respeito de como estes usuários podem ou devem interagir com o artefato a fim de realizar uma série objetivos ou causar uma série de efeitos possíveis, inscritos no design do artefato e no seu comportamento computacionalmente codificado. Este processo de comunicação dá-se através da interface do sistema que, durante interação, representa os *designers*, repassando aos usuários a mensagem que estes elaboraram, atuando assim como o seu “preposto” [12].

Para a Engenharia Semiótica todos os artefatos baseados em computadores são criados para expressar a visão do designer a respeito do sistema. Com isso, a interação humano-computador (IHC) é um caso particular de comunicação humana mediada por computadores (CMC).

A Engenharia Semiótica [12] postula então que há uma *metacomunicação* entre designer e usuários, ou seja, tecnicamente há comunicação sobre a comunicação, cujas chaves devem ser interpretadas/decodificadas pelo usuário para chegar ao significado/mensagem que está sendo expresso. Cabe a este usuário aprender a linguagem da interface, para conseguir assim comunicar ao artefato suas intenções. No caso da IHC, esta mensagem comunica ao usuário como ele pode realizar determinadas tarefas e, assim, se apropriar do sistema. A mensagem é elaborada para “dizer” o seguinte ao usuário: **“Este é o meu entendimento de quem você é, do que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê. Este, portanto, é o sistema que projetei para você, e esta é a forma como você pode ou deve utilizá-lo para alcançar uma gama de objetivos que se encaixam nesta visão”**.

A fim de compreender melhor os fenômenos envolvidos nessa metacomunicação e como é possível expressar-se através de um metadiscurso, na seção que segue abordamos o conceito de Autoexpressão.

### Autoexpressão

Nos expressamos de muitas maneiras: através do tom da voz, da postura, do rosto e, em casos mais sutis, pintura, música, escultura ou outras formas de arte. Quando nos expressamos, manifestamos uma parte do nosso ponto de vista. Autoexpressão envolve mostrar o estado emocional, cognitivo ou empírico do indivíduo, e está presente em tudo que fazemos [5]; é um fenômeno ligado à comunicação humana que se relaciona à personalidade, individualidade e idiossincrasias do indivíduo [9]

Baseada na perspectiva de computador como mídia e meio de comunicação, todos os tipos de softwares, indiferente de quem os tenha criado, carregam representações de seus criadores [7]. Cada vez mais, crescem as tecnologias nas quais humanos podem “fazer-se presentes” durante interações mediadas pelo computador [8]. Segundo Monteiro [9] “cada mínima alteração realizada é um signo de autoexpressão de seus autores, estes signos são recursos constituintes de suas vozes individuais”.

### Uso de exemplos em programação

Exemplos são importante (senão essenciais) no processo de aprender a programar. No processo de ensino de programação, os exemplos podem ser usados com diversos propósitos: mostrar como usar uma estrutura da linguagem, implementar um algoritmo para solucionar um problema, ou demonstrar um padrão/estilo de programação [10].

Além de aprendizes de programação, programadores em todos os níveis de experiência programam estudando outros programas em livros e manuais ou olhando, reusando ou revisando códigos que outros programadores escreveram previamente [10]. Infelizmente, muitas vezes esse reuso de código ou partes de código é feito sem total compreensão do que o mesmo faz [6].

Gaspar e Langevin [4] reportam que, quando confrontados com novas tarefas, alguns estudantes iniciam a solução do problema identificando palavras chaves na especificação do programa que possam ser relacionadas com exercícios previamente feitos, para os quais eles já têm as devidas resoluções. Além disso, com a difusão da internet e a facilidade de acesso a ela, a cada dia se torna mais fácil obter, via Web, exemplos que correspondam muito similarmente (quando não iguais) as intenções do programador.

### METODOLOGIA

A fim de se investigar as questões levantadas neste trabalho, foi desenvolvida uma pesquisa qualitativa a partir da análise de códigos fontes desenvolvidos por alunos de programação, em disciplina oferecida ao primeiro semestre dos cursos de Ciência da Computação e Sistemas de Informação<sup>1</sup>. Tais códigos correspondem ao trabalho final da primeira disciplina de programação que os alunos cursaram. A seguir apresentaremos os contextos envolvidos, e como a pesquisa foi conduzida.

### Contexto da aplicação da pesquisa

O trabalho final da disciplina em questão envolveu a construção de um sistema, em Java, para avaliação de jogos educacionais, sendo que deveriam ser criados cadastros de Respondentes, Jogos e Questionários (a partir de agora denominado QuestJogos). Além disso, relatórios com informações gerais e estatísticas também foram solicitados.

Para estas mesmas turmas, antes da data de entrega do trabalho final, foi feito em sala de aula um exercício no qual era implementado o sistema de uma Livraria, sendo que foram criados cadastros de Clientes, Livros e Vendas. O código deste exercício foi disponibilizado aos alunos por meio do ambiente virtual utilizado como apoio à disciplina.

Em uma análise inicial, observamos que muitos alunos usaram o código do projeto Livraria como base para seus trabalhos finais. Houve, por exemplo, o uso do código da

Livraria como base para construção do código do QuestJogos: onde na Livraria tínhamos Cliente, no QuestJogos tínhamos Respondente; onde na Livraria tínhamos Livro, no QuestJogos tínhamos Jogo, e onde havia correlação de Cliente e Livro, que no projeto Livraria era feita na Venda, havia no QuestJogos seu equivalente nas classes Respostas/Questionários que correlacionava Respondente e Jogo.

### Etapa 1

Primeiramente, os alunos foram contatados e consultados sobre se gostariam de ceder os códigos dos seus trabalhos para que estes fossem analisados. Aqueles que aceitaram receberam para análise e assinatura o termo de consentimento livre e esclarecido pelo qual davam permissão para o uso dos códigos e publicação dos resultados encontrados através da análise realizada.

Após a concordância formal dos participantes, com o auxílio da ferramenta Moss<sup>2</sup>, verificamos a similaridade entre os códigos fontes desenvolvidos pelos alunos (QuestJogos), e o código feito em sala de aula, como exercício (Livraria). Apesar de o domínio de aplicação do exemplo ser totalmente diferente do domínio de aplicação solicitado, foram encontrados em diversos trabalhos altos índices de similaridade. Ressaltamos aqui, que a intenção pedagógica do professor ao elaborar o projeto Livraria com os alunos, era construir o conhecimento necessário para desenvolverem seus trabalhos finais, não sendo estimulado que os mesmos somente fizessem uma cópia, e alterassem somente o necessário, como foi visto em alguns casos.

Após esta verificação pelo Moss, alguns dos códigos foram comparados manualmente de forma a não somente identificar as similaridades, mas, principalmente, identificar as mudanças, mesmo que sutis, que os alunos tenham feito em suas versões dos programas.

### Etapa 2

Os alunos que tiveram seus códigos analisados ao final da etapa 1, foram convidados a participar de uma entrevista. Àqueles que concordaram em participar, foram apresentados aos objetivos e procedimentos da pesquisa e ao termo de consentimento livre e esclarecido. Na segunda etapa da pesquisa, parte dos alunos foi convidada a participar de uma entrevista, na qual, primeiramente, era solicitado que:

1. Identificassem, dentre outros códigos, aquele que julgassem que era o seu.  
Objetivo: verificar se o aluno consegue identificar sua própria autoexpressão (e o quão fácil ou dificilmente isso ocorrerá).
2. Justificassem sua escolha, destacando ou comentando os itens do código que o levaram a acreditar que aquele era seu código.

<sup>1</sup> Observa-se que nem todos os alunos estão na seriação aconselhada de seu respectivo currículo de curso. Alguns alunos estavam fazendo a disciplina pela segunda/terceira vez, ou já possuíam algum aprendizado prévio sobre programação.

<sup>2</sup> <https://theory.stanford.edu/~aiken/moss/>

Objetivo: compreender os traços de autoexpressão relevantes a estes alunos.

Após esta atividade, o participante era convidado a fazer uma explicação do código que ele selecionou como seu. Para esta etapa, devido ao tamanho do código, foram selecionados alguns trechos representativos. A maioria dos códigos era composta de uma classe para interface, essa classe se comunicava com as três classes de cadastro, e, cada classe de cadastro, era responsável pelo cadastro, pesquisa e remoção de objetos de uma classe de modelo. Por exemplo, a classe de Cadastro de Pessoas, é responsável por gerenciar a adição, remoção e pesquisa de objetos instanciados da classe Pessoa. Desta forma, foram selecionados para esta etapa:

1. Métodos de cadastro: um da classe de interface, um de uma das classes de cadastro.
2. Métodos de remoção: um da classe de interface, um de uma das classes de cadastro.
3. Métodos de pesquisa: um da classe de interface, um de uma das classes de cadastro.
4. Método "executa": o método da classe de interface responsável por criar um menu para o usuário e fazer as chamadas aos demais métodos e classes.
5. Métodos inéditos: se o participante desenvolveu em sua pesquisa métodos ou classes originais (aqueles que não se assemelhavam aos recursos do projeto de exemplo).

Observa-se nesta etapa que, caso o aluno selecionasse na etapa anterior um código que não era seu, ele seria convidado a realizar a explicação sobre o código selecionado.

Além disso, os métodos listados para serem explicados não possuíam informações diretas sobre qual classe do código eles eram provenientes, e os participantes eram convidados a tentar identificar de qual classe o método havia sido retirado.

E, por fim, o participante era convidado a responder um questionário no qual perguntas a respeito de uso de exemplos, comunicação e autoria eram abordadas. As principais questões abordadas foram:

1. Uma determinada especificação de sistema é dada a você. Quais seus primeiros passos na construção da aplicação?
2. Você costuma usar exemplos? Se sim: Em que situações os exemplos são usados? Como você escolhe um exemplo para ser usado?
3. Pensando em seu programa como uma forma de comunicação; com quem você está se comunicando?
4. Você percebe se características suas (de escrita/vocabulário/humor/personalidade) transcendem o software, chegando até aqueles com quem está se comunicando? Dê exemplos.

O questionário encerrava com um convite ao participante a preencher o *template* de metacomunicação da Engenharia Semiótica, completando as sentenças do mesmo (Este é o

meu entendimento de quem você é..., O que aprendi..., etc) de acordo com o que seu código fazia.

#### ACHADOS: ETAPA 1

Como descrito anteriormente, a pesquisa foi conduzida em duas etapas: a primeira conduzida somente pela equipe de pesquisa e a segunda que contava com a participação dos alunos. Nesta seção será descrita a primeira etapa.

Com o uso da ferramenta Moss, foi identificado o percentual de similaridade dos projetos dos alunos (QuestJogos) com o projeto Livraria.

A ferramenta Moss, usa algoritmos de detecção de "impressões digitais" para encontrar pequenas partes iguais, dentro de grandes documentos. Logo, os algoritmos usados no Moss não apenas detectam trechos inteiramente iguais, mas sim padrões [14]. Para os casos analisados nesta pesquisa, mesmo o aluno trocando nomes de variáveis, métodos e a ordem como os mesmos são criados ou chamados, ela ainda assim foi capaz de identificar a replicação de um padrão existente no projeto Livraria.

#### Resultados do Moss

Os percentuais de similaridade, assim como o número de linhas similares identificadas, são apresentados na Tabela 1.

Projeto do participante <sup>3</sup>	Linhas similares
Participante1 (44%) *	323
Participante2 (35%)	194
Participante3 (33%) *	267
Participante4 (31%) *	182
Participante5 (27%)	209
Participante6 (21%)	161
Participante7 (20%)	144
Participante8 (18%)	123
Participante9 (18%)	118
Participante10 (13%)	156
Participante11 (12%) *	99
Participante12 (9%) *	62
Participante13 (6%)	130
Participante14 (3%) *	72
Participante15 (1%) *	41

Tabela 1: Similaridade dos códigos com o projeto de exemplo

A partir deste levantamento removemos da análise aqueles que fizeram o trabalho em dupla, a fim de focar na autoexpressão de um único programador, restando os participantes assinalados na Tabela 1 com um asterisco (\*) ao lado do nome.

<sup>3</sup> O termo Participante pode representar tanto um trabalho individual quanto um trabalho feito em dupla.

### Análise

Para aprofundar a pesquisa, foram selecionados os códigos de dois extremos: **participantes 1 (P1) e 3 (P3)**, sendo estes os mais próximos ao exemplo, e **participantes 14 (P14) e 15 (P15)**, sendo estes os mais diferentes do exemplo.

#### Perfil dos participantes analisados

Conforme citado anteriormente, todos eles eram alunos de uma disciplina de programação oferecida aos primeiros semestres dos cursos de Ciência da Computação e Sistemas de Informação. Os participantes P3 e P15 já haviam cursado a disciplina em semestres anteriores. O participante P14 já possuía conhecimentos em programação oriundos de curso previamente iniciado. Já o participante P1 não possuía conhecimentos em programação. Até o momento da pesquisa nenhum deles tinha conhecimento a respeito de Engenharia Semiótica.

#### Resultados da análise

Cada projeto destes quatro participantes foi minuciosamente analisado à procura não somente das similaridades, mas mais importante que isso, as diferenças que o código do participante possuía em relação ao exemplo dado.

Observamos que, em muitos dos casos, a modelagem dos projetos dos participantes era igual ou muito semelhante à do projeto Livraria, o que facilitou a comparação dos códigos.

Nos quatro projetos analisados, foram procurados itens relacionados às mudanças que esses alunos fizeram em seu código em relação ao projeto Livraria. Cada projeto foi analisado item a item, verificando:

1. Padrão comparado ao do projeto Livraria: a estrutura de classes foi mantida a mesma do projeto Livraria?
2. Mudança da estrutura da interface (código): a estrutura como o aluno codificou sua interface é a mesma do projeto Livraria?
3. Mudança da estrutura da interface (visível ao usuário): a forma como a interface se apresenta ao usuário é a mesma do projeto Livraria? Mudanças de substantivos relacionados ao domínio não são consideradas.
4. Padrão de nomenclatura de classe: as classes, que no projeto Livraria, se iniciavam com a palavra Cad, mantiveram o mesmo padrão nas classes equivalentes do projeto dos alunos?
5. Estrutura da classe: a estrutura da classe (ordem dos métodos) é a mesma do projeto Livraria? Adição e remoção de métodos não estão sendo consideradas.
6. Estrutura dos métodos: a estrutura dos métodos, tanto em funcionamento, quanto em similaridade da nomenclatura usada, é a mesma do projeto Livraria?

Os resultados das análises podem ser vistos na Tabela 2, sendo I se Igual e D se Diferente.

		P1	P3	P14	P15
1	Projeto como um todo	I	I	D	I
2	App	I	D	D	D
3		I	I	D	D
4	CadLivro	D	I	D	D
	CadCliente	D	I	D	D
	CadVenda	D	I	D	D
5	App	I	I	D	D
	CadLivro	I	I	D	D
	CadCliente	I	I	D	D
	CadVenda	I	D	D	D
6	App	I	I	D	D
	CadLivro	I	D	D	D
	CadCliente	I	D	D	D
	CadVenda	I	I	D	D

**Tabela 2: Análise das mudanças nos projetos dos participantes**

Além do relatado na Tabela 3, também foi observado se os participantes adicionaram ou removeram métodos em relação ao projeto Livraria.

	P1	P3	P14	P15
App	Sim	Sim	Sim	Sim
Livro	Não	Não	Não	Não
Cliente	Sim	Sim	Sim	Não
Venda	Não	Não	Não	Não
CadLivro	Não	Não	Sim	Sim
CadCliente	Sim	Sim	Sim	Sim
CadVenda	Sim	Sim	Sim	Sim

**Tabela 3: Análise das inserções e remoções nos projetos dos participantes**

Tais análises nos permitiram perceber quais eram os traços autorais dos participantes em seus códigos e como tais traços carregavam aspectos de autoexpressão. A etapa 2, nos permitiu validar os traços e identificar novos aspectos. Os resultados obtidos nesta etapa posterior são relatados na seção a seguir.

### ACHADOS: ETAPA 2

Nas subseções a seguir apresentaremos os resultados obtidos na segunda etapa da pesquisa, na qual os participantes foram convidados para entrevistas, que foram conduzidas individualmente. Eles estão divididos entre o que os participantes afirmaram e demonstraram ser sua autoexpressão, o que eles falaram sobre o uso de exemplos e o que eles descreveram como a mensagem de metacomunicação que estavam passando aos usuários de seus programas.

#### Este sou eu... isto é meu!

Esta seção apresentará os achados obtidos através da observação do processo de busca pelo seu código feito pelos participantes e os relatos feitos por eles das características dos códigos que os guiaram na busca. Tais achados serão confrontados com a explicação que eles deram sobre trechos de códigos selecionados.

*Participante 1 (P1)*

A primeira tarefa era identificar em meio a outros códigos, aquele que ele julgava corresponder ao seu trabalho. Foram entregues ao participante 7 códigos: o seu próprio código, dois muito semelhantes ao exemplo (P2 e P3), e outros quatro, selecionados aleatoriamente, que gradativamente se tornavam mais diferentes do exemplo (P7, P11, P12, P15). Destaca-se que o trabalho do participante 1 também era fortemente semelhante ao exemplo (44% - 323 linhas).

O participante foi motivado a falar aquilo que estava pensando durante a busca pelo seu código. Os relatos<sup>4</sup> foram anotados pelo pesquisador, que se manteve junto aos participantes durante toda a entrevista. Logo no início da busca, o participante relata que está com dúvida em relação ao código de P2, acreditando que talvez este seja o seu código.

Relata também que procura no menu, presente na classe App, por indícios autorais que o ajudem na busca. O primeiro código descartado é o de P15 (aquele que mais se afasta do exemplo Livraria), e de acordo com o participante, resta escolher entre os códigos pertencentes a P2 e P12, pois ele acredita que um destes seja o seu. Antes de proceder com a escolha, descartou também os códigos dos participantes P7 e P11. Por fim, o aluno agrupa os códigos dos participantes P1, P2, P3 e P12 alegando que estes eram muito parecidos. E em uma última observação, feita muito rapidamente, muda de opinião e assinala como seu, o código que realmente lhe pertencia.

Quando questionado sobre os motivos que o levaram a mudar de opinião e decidir que aquele código era, de fato seu, o participante reporta que, pouco antes da entrega do trabalho, havia participado de uma palestra na qual foi-lhe ensinado que, mesmo que sejam grandes, os nomes de variáveis sempre deveriam ser os mais mnemônicos possível. Tal característica estava presente no código escolhido. Um ponto de destaque, relatado por ele, era a forma como uma data de nascimento estava sendo tratada em uma das classes (com atributos separados para dia, mês e ano) estando parecida com o que ele recordava ter feito.

Durante a explicação dada a trechos do código o participante demonstra pouco conhecimento sobre seu código, tanto em relação ao funcionamento dos métodos quanto em relação à possível localização dos mesmo no código. A explicação do participante dá-se de forma muito pontual, fazendo uma leitura do programa linha a linha, focando em pequenos itens da sintaxe/semântica, ao invés de a forma como tais conteúdos estão aplicados ao trabalho.

Sua explicação nos leva a acreditar que ele reusou o código de exemplo, no entanto, não se apropriou da mensagem que o *designer* passava, demonstrando dúvidas em relação à

mensagem que por fim, o próprio participante estava passando em sua aplicação. Nas seções seguintes, iremos aprofundar a discussão sobre a (não) apropriação do código.

*Participante 3(P3)*

Na primeira tarefa, o participante P3 recebeu o mesmo conjunto de códigos usados pelo participante P1. P3 possuía um projeto com 33% de similaridade (267 linhas) em relação ao exemplo Livraria.

Logo após iniciar, o participante já descarta os projetos dos participantes P14 e P15. Após, descarta quase todos os projetos, mantendo apenas o seu próprio e o projeto de P2. Na sequência assinala seu próprio código, encontrando o mesmo e alega ter certeza que aquele código fora escrito por ele.

Observa-se que o participante conseguiu encontrar seu código em cerca de 2 minutos. O mesmo relata que, devido à forma como uma palavra estava escrita, logo no início do programa (usando caracteres especiais para compor a palavra), ele tem certeza que aquele era seu programa pois se recordava de quando “montou” a palavra.

A tarefa sequente consistia na realização de uma descrição sobre alguns trechos retirados de seu código fonte. Durante a explicação o participante demonstra alta apropriação sobre os trechos de códigos provenientes do projeto Livraria, sabendo dizer seu funcionamento (em totalidade) e sua localização em quase todos os casos. Destacamos aqui que a explicação dada pelo participante era detalhada, precisa e feita em alto nível, diferente da explicação dada pelo participante P1, que em uma leitura pontual dos códigos, conseguia ao fim chegar a uma conclusão (que por vezes era errônea) do objetivo do trecho. Devido ao detalhamento e certezas demonstradas pelo participante durante as explicações, consideramos que ele se apropriou da mensagem passada pelo designer do exemplo, fazendo dela sua própria mensagem. Além disso, não somente se apropriou desta mensagem como também adicionou diversos itens autorais em seu sistema, que eram fruto de sua autoexpressão, como, por exemplo, os diversos métodos feitos para gerar informações estatísticas sobre os dados cadastrados em seu programa. No total, este participante acrescentou 7 novos métodos para gerar esses relatórios, métodos que não possuíam equivalentes no projeto Livraria.

*Participante 14 (P14)*

O participante P14 faz parte do outro extremo que observamos nos códigos, sendo seu sistema somente 3% semelhante ao projeto de exemplo, com diversos itens autorais, incluindo itens e práticas que não comumente eram usadas como abordagem em aula. Ele recebeu seu próprio código, e juntamente a este, os códigos de P13 e P15 (os mais

<sup>4</sup> Na etapa de identificação do código, os relatos dos participantes foram capturados por meio de anotações. Portanto, o que é descrito nesta seção não é a fala literal do participante.

diferentes do exemplo). Os códigos dos participantes P1, P2, P3 e P7 também foram dados ao participante.

O primeiro código excluído pelo participante é aquele pertencente a P15. Logo em seguida, já visualiza seu código e aponta que aquele *podia* ser o seu. No entanto confere todos os demais, e finaliza a procura tendo encontrado o seu código. Segundo o participante o código era seu devido à forma como os menus estavam organizados e as mensagens que cada menu apresentava, indicando ao usuário a ramificação no qual o ele se encontrava (Ex.: “Menu >> Respondente >> Remover”).

O participante, durante a explicação dos trechos de código demonstra total conhecimento sobre os mesmos, sabendo com exatidão o que o mesmo fazia e aonde estava localizado.

#### **Participante 15 (P15)**

O participante P15 recebeu o mesmo conjunto de códigos dado a P14. Este participante, de todos os analisados, era o que possuía o trabalho mais distante do projeto de exemplo, tendo apenas 1% de similaridade. Inicialmente o participante aponta que o código do participante 1 poderia ser o seu. Acreditamos que o participante tenha feito tal confusão, devido ao fato de o projeto Livraria ter sido feito como exercício durante a disciplina. No entanto, recorda-se que seu menu era diferente e tinha áreas diferentes para administradores e participante. Então, olha os demais códigos já tendo tal fato em mente. Logo em seguida identifica seu código, justamente devido a tal características de menu.

Além disso, o participante aponta que a formatação das linhas de comentários de seu código, a formatação do menu e a presença de um trecho de gerador de código, davam a ele a certeza de que aquele era o seu código.

Assim como o participante anterior, este participante demonstra total conhecimento sobre seu código tanto em relação ao funcionamento, como um todo, quanto em relação à localização dos trechos. Também questionamos o participante sobre uma característica única encontrada no seu código, podendo ver com mais amplitude a apropriação que o mesmo tinha sobre seu trabalho.

#### **Como eu uso exemplos...**

Na subseção “Como eu uso exemplos”, iniciaremos um processo de triangulação, comparando aquilo que foi dito pelos participantes em suas respostas dos questionários com o que foi encontrado e relatado na seção “Este sou eu... isto é meu!”.

Como relatamos, os entrevistados P1 e P3 fizeram um alto uso do exemplo Livraria na construção de seu código. As respostas dadas por eles em seu questionário apoiam o fato: ambos responderam que usam exemplos para fazer reuso do código, ou seja, alterar no exemplo as características necessárias para que o mesmo se adeque ao problema que estão tentando solucionar.

O participante P1 diz “*Códigos mais curtos, com poucas mudanças, reutilizo o código mudando o que precisa. Em códigos mais complexos, geralmente mais longos, utilizo como consulta. Embora mesmo assim, utilizo pequenas partes do exemplo (copio)*”. O participante P3, por sua vez relata que “*Utilizo um código de exemplo como base para aprimoramento/aperfeiçoamento até chegar ao objetivo*”.

É possível observar, principalmente pelos dados da etapa 1 que isto foi feito por estes dois participantes no que diz respeito ao QuestJogos. Complementando, P3 fala, durante sua entrevista, que “*Quando fizemos a Livraria em sala de aula tudo se esclareceu. Então percebi que para fazer o Trabalho final bastava seguir aquela mesma modelagem, e deu certo*”.

O participante P14 também relata fazer reuso de código, no entanto, observamos pela sua resposta que tal alternativa só é usada quando as alternativas foram exauridas: “*Geralmente quando estou com dificuldades para implementar algum método específico, por exemplo, busco um código e tento aplica-lo àquela situação específica. Na maioria das vezes quando ‘manipulo’ classes que não estou acostumada a utilizar, para compreender o funcionamento*”.

Os participantes P1 e P14 também relatam que usam código de exemplo como modelo, ou seja, apenas como consulta para sanar dúvidas. Uma observação interessante é que P15, o participante cujo trabalho mais diferia do projeto Livraria, relata que somente usa exemplos desta forma. Já P3, que observamos que fez o reuso de código, relata que somente usa exemplos para reuso.

Também perguntamos aos participantes como eles escolhiam um exemplo. Todos os participantes relataram que na hora de escolher um exemplo, procuravam algo que tivesse o mesmo funcionamento, independente do domínio ao qual o exemplo era aplicado. P3 diz que “*Um bom exemplo serve como ‘esqueleto’ para a modelagem. Muitas vezes permite apenas substituição por objetos que sejam pertinentes ao assunto solicitado*”. Já o Participante 14 diz que “*Se for alguma questão lógica, tento localizar exemplos que possam ser aplicados na situação, independente da linguagem de programação ou de assunto*”. Somente P1 diz que procura por exemplos que já sejam de um mesmo domínio que o problema que está resolvendo, relatando que “*Procuro pelo código mais parecido possível com o assunto, caso não encontre, procuro pelos de mesmo funcionamento*”.

As respostas dos participantes são condizentes com os achados da etapa 1: vemos que os participantes que fizeram o maior reuso de código estão cientes do fato, e tem disso um hábito. Já os outros dois, acreditam que essa é uma última alternativa, usando em casos mais extremos.

#### **E este é você...**

Esta seção nos mostra a visão que os participantes têm sobre o que seu sistema é, sobre quem seus usuários são, e sobre qual mensagem eles acreditam estar passando para estes usuários através de suas aplicações.

Antes de responderem ao questionário no qual haviam perguntas explicitando a existência de uma comunicação entre *designer* e usuário, os usuários explicaram trechos de seus códigos. Durante a explicação podemos notar nas falas deles, evidenciando P1, P3 e P14, que estes veem os usuários finais como receptores das mensagens que estavam passando. Os participantes relatam em trechos que o sistema vai “solicitar para o usuário inserir um nome de jogo” (P3), ou que “este trecho estava na classe App, pois ele tem interação com o usuário” (P14). Participante P1 cita que o sistema está “perguntando para o usuário, mostrando na tela”. Destacamos aqui que o participante P14 cita “interação com o usuário” em vários momentos durante sua explicação, mostrando que sabia com quem “seu sistema” estava interagindo. No entanto, ambos 3 participantes apesar de por diversas vezes terem expressado que a mensagem ia ao usuário final, em todas as mesmas circunstâncias eles afirmavam que era o sistema que estava “falando” a mensagem, e não eles (os próprios participantes).

O participante P15 faz uma vaga referência a existência de um processo comunicativo entre sistema e usuário. Suas falas “aqui retornava a quantidade de participantes” e “o sistema dá um retorno, se for cadastrado ou não”, não expressam para quem estes retornos eram fornecidos. O mais próximo que ele chega de um usuário final é na fala “se o cara respondesse”, fala esta que evidencia o conhecimento por parte entrevistado de que existia um alguém (o cara) que usaria seu sistema.

A respeito das respostas dadas pelos participantes para o *template* de metacomunicação, através do *qual*, os participantes podem elaborar estas mensagens. Tais mensagens podemos ter um último indicador a respeito da apropriação que eles têm sobre a mensagem que estão passando (tendo usado ou não os exemplos).

Os participantes P1, P3 e P14 relatam que acreditavam que através do sistema podem se comunicar com o usuário final, e que percebem que determinadas características do *designer* podem percebidas pelos usuários. Estes participantes citam, por exemplo, que “pessoas mais comunicativas tendem a utilizar mais mensagens para o usuário e pessoas mais detalhistas vão usar mais métodos, controles precisos” (P1) ou mensagens de interface mais pessoais, lembrando uma “comunicação entre amigos” (P3). Além disso, P3 destaca que também acredita que pode estar se comunicando com outros programadores, estes “situados em outras partes do planeta”.

Após responderem esta pergunta, e antes de preencherem o *template* de metacomunicação, os participantes receberam uma breve introdução sobre o que era a Engenharia Semiótica, e qual a finalidade do *template*<sup>5</sup>. Munidos destas

informações os participantes tentaram elaborar as mensagens que eles acreditavam estar passando aos seus usuários. Os participantes P3, P14 e P15 conseguem identificar com clareza o usuário com quem estão se comunicando, sendo este descrito como “uma pessoa que está buscando novas ferramentas p/ aplicação didática” (P3), “uma pessoa comum, aluno ou professor” (P14) ou “um professor avaliando uma nova ferramenta de ensino que lhe foi apresentada” (P15). Apesar de o participante P15 não ter explicitado isso durante suas explicações, a questão talvez tenha o levado a uma reflexão, no qual este foi capaz de identificar o usuário final como receptor da mensagem.

Em relação ao item “O que aprendi que você quer ou precisa fazer, de que maneiras prefere fazer, e por quê...” as mensagens dos participantes P3 e P14 são precisas em relação ao que o usuário precisa. Eles relatam que o usuário “precisa selecionar um aplicativo que melhor se adequa aos seus alunos e possua um bom nível de conhecimento a ser transmitido de forma clara e objetiva, com uma interface agradável que atraia a atenção do aluno” (P3) e que este usuário “quer armazenar e manipular informações sobre jogos” (P14).

E o sistema que os participantes P3 e P15 dizem ter projetado é, respectivamente, “um sistema que permite que você identifique de onde são as outras pessoas que optaram por determinado aplicativo, sua idade, suas qualificações e seu parecer com relação ao mesmo” (P3) e um sistema com o qual “o administrador poderá ter um pequeno banco de dados a respeito dos participantes, capaz de organizar e transformar estes em informação útil. Para o usuário será uma forma simples, porém eficaz de expor suas percepções sobre as ferramentas educacionais avaliadas” (P15) a forma como seus usuários deveriam utilizá-lo para alcançar os objetivos é “Oferecendo em instituições que possuam laboratórios de informática ou estejam desenvolvendo aplicações para este fim” (participante 3) ou, de forma mais específica em relação a interação com o sistema “você pode seguir os menus, de forma intuitiva” (P14 e P15).

As mensagens do participante P1 foram feitas de forma genérica ou com poucas informações em relação ao sistema em si. As mensagens do participante P15 por vezes também exibiam trechos genéricos. No entanto as mensagens dos participantes P3 e P14 foram elaboradas de forma precisa, clara e objetiva, mostrando a apropriação que os mesmos possuíam das mensagens que estavam passando a seus usuários.

A seguir, apresentaremos uma discussão embasada com os dados previamente apresentados.

<sup>5</sup> A introdução a estes conceitos era feita somente após responderem à pergunta sobre com quem acreditavam estar se comunicando, afim de não influenciar nas respostas desta pergunta.

## DISCUSSÃO

Os achados da pesquisa nos mostram três cenários distintos. Podemos ver claramente a ocorrência de “os meus”, sendo esse o código de autoria do participante, sem influência do exemplo, “os seus”, sendo este o código que reusou do exemplo, porém com baixa apropriação do mesmo, e por fim “os nossos”, sendo este o código com reuso do exemplo, mas que demonstra clara apropriação. Acreditamos que apropriar-se vai além de simplesmente reusar o código do outro programador. Para apropriar-se realmente deste código, o programador deveria se apropriar do discurso e da mensagem que está sendo passada, compreendendo-a, e só assim incorporando-a ao seu próprio discurso. Os cenários serão discutidos a seguir.

### Os meus...

Durante a etapa 1, já foi possível verificar que os códigos produzidos por P14 e P15 eram discrepantes do exemplo dado. Seus códigos possuíam diferenças em relação ao código do exemplo, e diversos itens únicos, criados pelos próprios participantes, que não se assemelhavam sequer às abordagens vistas em sala de aula.

A etapa 2 mostrou-nos que os participantes alcançaram apropriação sobre seu código, conseguindo identificar com clareza os traços de sua autoexpressão ao explicar os trechos selecionados, não somente em relação à sua sintaxe e semântica, mas também em relação a aspectos pragmáticos dos mesmos.

Em relação a como estes participantes usam exemplos, suas respostas mostraram que para eles, exemplos só são usados em casos onde não conseguiram solucionar o problema sozinhos. As mensagens de metacomunicação elaboradas por ambos mostravam que eles sabiam quem eram seus usuários e qual era a finalidade de seus sistemas.

### Os seus...

No caso de P1, observamos o reuso do exemplo, no entanto o participante não conseguiu relatar com precisão o funcionamento de seu próprio código. Este participante levou um tempo longo e teve alto grau de incerteza para encontrar seu código, mostrando que sua autoexpressão não era marcada em seu trabalho.

Este participante demonstrou dúvidas em relação aos códigos 1 (seu próprio), 2 e 3, todos semelhantes ao exemplo, e, segundo ele, “todos estavam muito parecidos”. Observamos aqui que o participante conseguiu identificar a voz do *designer* do exemplo, presente nestes casos.

Em sua mensagem de metacomunicação, o participante mostrou pouco domínio em relação há quem era seu usuário e o que era seu programa, levando-nos a acreditar o reuso do código do exemplo, embora tenha levado o participante a atingir o objetivo final, não foi apropriado, gerando prejuízos em relação à autoexpressão do participante e a comunicação através do software.

### Os nossos...

O participante P3 nos leva a um terceiro cenário, onde o reuso de exemplo ocorreu, porém, o participante se apropriou da mensagem do *designer* do exemplo, e sobre esta mensagem, construiu a sua própria, através da adição de itens autorais.

O participante foi capaz de identificar seu código mais rapidamente que qualquer outro participante (inclusive aqueles cujo sistema era totalmente autoral), e apresentar com clareza o funcionamento dos códigos. Embora em alguns trechos da explicação ocorresse alguma confusão, isso ocorreu poucas vezes.

Outro ponto que reforça o achado é a mensagem de metacomunicação elaborada por este participante. O participante sabia quem era seu usuário, o que era seu sistema, como este deveria ser usado, e desenvolveu uma mensagem com clareza e objetividade. Mesmo tendo-se apropriado de uma mensagem de metacomunicação diferente, para um contexto diferente (projeto Livraria), soube readaptar o código utilizando-o (como descrito pelo próprio participante) apenas como um “esqueleto” para a modelagem.

## CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Exemplos são importantes no ensino de programação. É através deles que conceitos são introduzidos e muitas vezes trabalhados, sendo comum, na área de programação, mesmo dentre programadores experientes, o reuso de códigos de exemplos. Neste trabalho abordamos o reuso de código sob outra perspectiva: a apropriação do código do exemplo, adaptando não somente o código, mas incorporando o discurso presente na mensagem de metacomunicação do programador exemplo ao discurso daquele que está fazendo o reuso.

No entanto, consideramos a apropriação um processo que vai além da simples incorporação: ao nosso ver, a apropriação abrange a compreensão desta mensagem, e somente a partir desta compreensão dá-se a incorporação da mesma à expressão do programador.

Neste trabalho realizamos uma pesquisa qualitativa na qual códigos de alunos de programação foram analisados à procura de traços de reuso de exemplo e de autoexpressão. Parte destes alunos foi também entrevistada, para uma análise mais aprofundada sobre a apropriação do discurso do exemplo.

A pesquisa nos mostrou que, mesmo em casos onde o exemplo é totalmente reusado (o exemplo é usado como estrutura na qual o aluno apenas renomeia o que é necessário para a adaptação ao novo contexto), o aluno foi capaz de apropriar-se plenamente do metadiscurso, fazendo, deste, sua autoexpressão. Além disso, é possível observar que mesmo sem terem sido apresentados previamente a conceitos de Engenharia Semiótica e de computador como mídia, que

esses programadores têm certa consciência a respeito de quem é seu usuário e do que querem dizer a eles.

Unimos as respostas dadas ao *template* de metacomunicação dos participantes (P3 e P14) formando uma mensagem que, em nossa opinião, retrata os sistemas, e por sua vez, retrata que os participantes têm consciência da mensagem que estão passando: “*Você é uma pessoa, aluno ou professor (P14) que está buscando novas ferramentas para aplicação didática, porém com receio de optar por algo inapropriado, recebendo auxílio de outros usuários (P3). Você precisa selecionar um aplicativo que melhor se adeque aos seus alunos e possua um bom nível de conhecimento a ser transmitido de forma clara e objetiva, com uma interface agradável que atraia a atenção do aluno (P3). Para isso precisará cadastrar as pessoas, que respondem questionários, e jogos, no caso o objeto do questionário. Por fim, você poderá consultar, de diferentes formas as estatísticas sobre os questionários (P14). Desta forma, projetei para você um sistema que permite que você identifique de onde são as outras pessoas que optaram por um determinado aplicativo, sua idade, suas qualificações e seu parecer com relação ao mesmo (P3). Você pode seguir os menus, de forma intuitiva. Inicialmente, terá que cadastrar os respondentes e os jogos. A partir disso, os questionários poderão ser respondidos (pelos respondentes, sobre os jogos). Por fim, você poderá consultar as informações geradas, a partir de estatísticas e relatórios. (P14)*”.

O destaque deste *template* certamente recai sobre as falas de P3, que mesmo tendo feito reuso do exemplo Livraria, mostra apropriação sobre a mensagem que desejava passar a seu usuário.

Novos estudos sobre tal tema são necessários, e tão importante quanto, é necessário compreender como esses casos, onde há o reuso dos exemplos (com ou sem a real apropriação), emergem ao usuário final, àquele que interpreta a mensagem do programador.

Ainda, é necessário compreender melhor como exemplos impactam não só na comunicação, mas no processo cognitivo dos alunos de programação, e como incentivar e motivar o uso/reuso com apropriação da mensagem.

### Agradecimentos

Clarisse de Souza agradece ao CNPq e FAPERJ pelo apoio parcial de sua pesquisa neste tópico.

### REFERÊNCIAS

1. Ali R. Abasi, Nahal Akbari and Barbara Graves. Discourse appropriation, construction of identities, and the complex issue of plagiarism: ESL students writing in graduate school. *Journal of Second Language Writing*, 15, 2 (2006), 102-117.
2. Peter Bøgh Andersen, Berit Holmqvist and Jens F. Jensen. *The Computer as Medium*. Cambridge University Press, Cambridge, 1993.
3. Humberto Eco. *A theory of semiotics*. Indiana University Press, Bloomington, 1976.
4. Alessio Gaspar and Sarah Langevin. Restoring "coding with intention" in introductory programming courses. In *8th ACM SIGITE conference on Information technology education* (New York 2007), ACM, 91-98.
5. Mitchell S. Green. *Self-Expression*. Clarendon Press - Oxford, Oxford, 2007.
6. Walid Maalej, Rebecca Tiarks, Tobias Roehm and Rainer Koschke. On the Comprehension of Program Comprehension. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 23, 4 (2014).
7. Ingrid Teixeira Monteiro and Clarisse Sieckenius de Souza. The representation of self in mediated interaction with computers. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems* (Porto Alegre 2012), SBC, 219-218.
8. Ingrid Teixeira Monteiro, Clarisse Sieckenius de Souza and Eduardo Tolmasquim. My program, my world: Insights from 1st-person reflective programming in EUD education. In *Fifth International Symposium on End-User Development - IS-EUD 2015* (2015).
9. Ingrid Teixeira Monteiro. *Autoexpressão e engenharia semiótica do usuário-designer*. PUC-Rio, Rio de Janeiro, 2015.
10. Lisa Rubin Neal. A System for Example-Based Programming. In *CHI '89 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York 1989), ACM, 63-68.
11. Charles Sanders Peirce. *Collected Papers of Charles Sanders Peirce*. Harvard University Press, Cambridge, 1931-1958.
12. Clarisse Sieckenius de Souza. *The semiotic engineering of human-computer interaction*. The MIT Press, Cambridge, 2005.
13. Lúcia Santaella. *O que é semiótica?* Brasiliense, São Paulo, 1983.
14. Saul Schleimer, Daniel S. Wilkerson and Alex Aiken. Winnowing: Local Algorithms for Document Fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (New York 2003), ACM.
15. James V. Wertsch. *Mind as Action*. Oxford University Press, Oxford, 1998.