

A utilização da Programação em Par Distribuída no ensino de programação

Bernardo José da Silva Estácio, Rafael Prikladnicki

Programa de Pós-Graduação em Ciência da Computação – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre - RS – Brasil

{bernardo.estacio@acad.pucrs.br, rafael.prikladnicki@pucrs.br}

Abstract. *Courses taken at a distance are an increasingly present reality in different areas of knowledge. In Computing is not different, programming teaching at a distance requires new challenges and practices for learning to be effective. In the literature, the practice of Pair Programming (PP) has proved to be an effective practice in the learning process. However few studies have examined the PP in the distributed context as a teaching tool and its importance to compose a resume of a computer course. This paper aims to present a conceptual basis of Distributed Pair Programming (DPP) in teaching, presenting the main lessons learned from studies conducted in the area.*

Resumo. *Cursos realizados a distância são uma realidade cada vez mais presente em diversas áreas do conhecimento. Na Computação não é diferente, o ensino de programação a distância requer novos desafios e práticas para que o aprendizado seja efetivo. Na literatura a prática de Programação em Par (PP) mostrou ser uma prática efetiva no processo de aprendizado. Contudo, poucos estudos analisaram a PP no contexto distribuído como ferramenta de ensino e a sua importância de compor um currículo de um curso de Computação. Este artigo tem como objetivo apresentar uma base conceitual de Programação em Par Distribuída (PPD) no ensino, apresentando as principais lições aprendidas dos estudos conduzidos na área.*

1. Introdução

O advento de plataformas virtuais de ensino consolidadas aliada à difusão de cursos semipresenciais e a distância contribui para que atividades pedagógicas sejam cada vez mais realizadas de forma remota, distribuída [Edwards et al. 2010]. A crescente adoção do Desenvolvimento Distribuído de Software (DDS) nas organizações também corrobora na importância do tema e da experiência no currículo dos alunos de um curso de Computação em nível de graduação [L'Erario et al. 2008].

A Programação em Par (PP) é uma prática do método XP que consiste em dois desenvolvedores trabalhando em um computador sobre o mesmo algoritmo, código ou projeto [Mcdowell et al. 2002]. Esta prática já mostrou muitos benefícios no ensino, tais como: a facilidade de aprendizado [Carver et al. 2007], a motivação dos alunos [Mcdowell et al. 2002] e a melhora na qualidade do código [Ramli et al. 2008]. No contexto distribuído, a Programação em Par Distribuída (PPD) é uma alternativa nos cursos que envolvem o desenvolvimento de software remotamente. Poucos estudos na literatura exploram empiricamente a PPD como ferramenta de ensino. Desta forma, o principal objetivo deste artigo é apresentar os resultados de uma revisão da literatura

sobre os estudos já existentes, apresentando as principais variáveis em torno da prática, bem como um conjunto de lições aprendidas.

Este artigo está estruturado da seguinte maneira: a próxima seção apresenta os principais estudos envolvendo a PP no ensino de programação. Na Seção 3, são apresentados os principais conceitos em torno da PPD. A Seção 4 apresenta uma caracterização de PPD no ensino de programação a partir dos estudos presentes na literatura. A Seção 5 apresenta um conjunto de lições aprendidas obtidas a partir destes estudos. Por fim, na Seção 6 as considerações finais deste trabalho apresentado.

2. O uso da PP no ensino de programação

Alguns estudos da literatura buscaram avaliar PP no contexto educacional. Han et al. (2011) e Ramli et al. (2008) avaliaram a qualidade do código produzido pelos alunos a partir de algumas heurísticas que apresentaram bons resultados, tais como: pouca quantidade de erros gramaticais e lógicos, mais tarefas finalizadas dentro de um espaço de tempo e uso de comentários e variáveis significantes ao longo do código desenvolvido pelos alunos.

Quanto ao desempenho dos alunos, três estudos mostraram que PP ajudou os programadores a desenvolverem melhor as suas atividades. Ramli et al. (2008) relata que os alunos obtiveram um melhor desempenho nas atividades em pares do que individualmente [Ramli et al. 2008]. Mcdowell et al. (2002) apontou como desempenho duas métricas: a qualidade dos programas desenvolvidos e a capacidade dos estudantes aplicarem os conceitos ensinados durante o curso, as notas finais obtidas pelos alunos detalhada foi usada como parâmetro para esses critérios [Mcdowell et al. 2002].

Alguns estudos apresentaram que a PP é uma prática propícia para o aprendizado. Nagappan e Carver afirmam que PP auxilia na não evasão de alunos nos cursos de introdução a programação [Carver et al. 2007], [Nagappan et al. 2003]. Alguns estudos do tipo *survey* mostraram que os alunos ficaram mais motivados e tiveram mais satisfação ao usar a PP [Chigona et al. 2008], [Mcdowell et al. 2002]. Quanto à diferença de personalidade no uso de PP no ensino alguns estudos mostraram que não há um efeito significativo na PP e enfatizam que mais estudos precisam ser realizados [Salleh et al. 2009], [Salleh et al. 2010].

Outra variável importante que tem um efeito positivo no ensino em PP é a produtividade. Nagappan et al. (2003) relatou que no curso de introdução de programação os alunos foram mais produtivos e menos frustrantes [Nagappan et al. 2003]. A medida usada foi a nota dos alunos. Chigona et al. (2008) também ratifica esses resultados por meio de um experimento, onde os alunos possuíram menos dúvidas e informaram por meio de uma *survey* que foram mais produtivos usando PP [Chigona et al. 2008].

A PP também mostrou ser eficiente por meio do aumento das notas dos alunos. Seis estudos apontaram essa métrica positiva [Braught et al. 2008], [Braught et al., 2011], [Brereton et al. 2009], [Mendes et al. 2005]. Outros seis estudos apresentaram que PP aumentou a confiança do programador e todos eles foram aplicados em um contexto educacional em experimentos envolvendo alunos. Para medir a confiança os estudos aplicaram *surveys* após cursos que utilizavam PP [Hanks 2008], [Han et al. 2011], [Mcdowell et al. 2002], [Ramli et al. 2008], [Salleh et al. 2011]. Os resultados

nestes estudos apontam que PP tem um efeito positivo na confiança ao desenvolver as atividades de programação.

3. A Programação em Par Distribuída

A adoção crescente do DDS pelas organizações implica na investigação de como as práticas de desenvolvimento de software são utilizadas com equipes distribuídas. Com a estratégia de adoção de Métodos Ágeis no DDS, a PP se torna alvo desse contexto, sendo uma prática ágil que exige comunicação face a face e uma colaboração com a interação constante entre os programadores. A distribuição geográfica levanta alguns desafios nesses aspectos. Desta forma, a Programação em Par Distribuída (PPD), também conhecida como Programação em Par Virtual ou ainda Programação em Par Remota, é uma prática que possui a necessidade de mais investigações e estudos empíricos.

Baheti (2002) atestou que os benefícios de PP são os mesmos em PPD, tais como a produtividade e qualidade do código [Baheti 2002]. Outro benefício de PPD é que dada as suas características, ela ajuda à promoção do trabalho e da comunicação dentro de equipes distribuídas [Baheti 2002]. Contudo, o foco aos aspectos de infraestrutura deve ser redobrado, e isto se reflete na adoção de um ferramental específico para a prática. Canfora et al.(2006) em seu estudo relatou que a qualidade do código sofre uma pequena diminuição devido as questões de infraestrutura.

Ho et al. (2004) aponta que o uso de ferramentas específicas para PPD ajuda a tornar a prática mais rápida [Ho et al. 2004]. Canfora corrobora dizendo que uma ferramenta específica evita que o programador fique alternando entre diferentes ferramentas, gastando tempo [Canfora et al. 2006]. Natsu et al. (2003) e Stotts et al. (2004) propuseram diferentes ferramentas de apoio à PPD. Este ferramental tem que prover um bom canal de comunicação por meio de *chat* e divisão da área de trabalho entre os programadores. Natsu et al. (2003) e Stotts et al. (2004) propuseram diferentes ferramentas de apoio à PPD.

Outro ponto a ser abordado é a ideia de time, com a distância geográfica são necessárias estratégias para que todos conheçam o projeto de maneira igual para que não haja empecilhos no desenvolvimento do sistema e não afetar a colaboração entre os pares. A colaboração também pode enfrentar outros obstáculos como o fuso horário, a cultura e o idioma [Canfora et al. 2006]. Quanto ao esforço não há evidências que relatam um aumento [Canfora et al. 2006].

A PPD também mostrou ser uma prática que gera benefícios no ensino de programação. Alguns efeitos positivos foram encontrados em relação ao aprendizado, qualidade do Código e ao desempenho do programador [Hanks 2005]. Em um experimento realizado por Hanks (2005), os alunos que realizaram a PPD tiveram um desempenho tão bom quanto os alunos que estavam com pares co-locados, passando no curso com notas similares. A próxima seção terá um detalhamento maior sobre as variáveis e efeitos em torno dos estudos da prática e do ensino em PPD.

4. O uso de PPD no ensino de programação

Na literatura são encontrados vários estudos sobre ferramentas de apoio e a exploração de PPD como prática, havendo poucos trabalhos que lidam com PPD no ensino de programação. Neste sentido, a tabela 1 apresenta quatro estudos, identificados em uma

revisão sistemática de PPD executada por Estácio (2013), e as suas principais características que analisam PDD sob a perspectiva pedagógica.

Tabela 1. Estudos sobre PPD no ensino de programação

Estudos	Variáveis Independentes	Principais variáveis dependentes	Tipo de coleta das variáveis
[Hanks 2005]	PP X PPD	Desempenho Confiança Satisfação	Nota dos alunos <i>Survey</i> aplicada aos alunos
[Zin et al. 2006]	PPD	Confiança Aprendizado	<i>Survey</i> aplicada aos alunos
[Edward et al. 2010]	PP X PPD	Satisfação Compromisso Aprendizado	<i>Survey</i> aplicada aos alunos
[Zacharis et al. 2011]	PI X PPD	Desempenho Satisfação Qualidade Produtividade de código	Nota dos alunos <i>Survey</i> aplicada aos alunos LOC/H

Hanks (2005) realizou um experimento com alunos a respeito de verificar o desempenho do uso de uma ferramenta de apoio a programação em par distribuída em um curso introdutório de programação [Hanks 2005]. O experimento envolveu 112 alunos, sendo que 57 utilizaram PPD e 55 fizeram pareamento co-locado, executando em três sessões. Os resultados mostraram que os alunos que realizaram PPD tiveram um desempenho tão bom quanto os alunos que estavam com pares co-locados, passando no curso com notas similares. *Surveys* foram aplicadas no início e ao final do curso e os resultados indicaram que o nível de confiança se mostrou estatisticamente semelhante.

O experimento realizado por Zin et al. (2006) buscou avaliar a efetividade de PPD. A limitação principal desse estudo é que a PPD foi utilizada de forma assíncrona devido ao sistema utilizado na universidade. O experimento foi realizado em um curso de orientação a objetos, onde 147 participaram da pesquisa respondendo uma *survey* aplicada. Os resultados apontaram que a PPD ajudou na confiança dos alunos e no aprendizado. Os alunos também indicaram que seria mais interessante utilizar a PPD de forma síncrona, por meio de ferramentas de compartilhamentos, isto facilitaria o dialogo entre os pares.

O estudo realizado por Edwards et al.(2010) tinha por objetivo avaliar a efetividade de PPD no currículo de um curso de informática *online*. Um experimento foi realizado com 100 alunos da universidade de Indiana Bloomington de um curso de introdução de Computação. As atividades foram divididas em duas sessões: a primeira utilização PP com pares co-locados e a segunda usando PPD. Ao término, os alunos responderam uma *survey* sobre o compromisso, motivação e o aprendizado deles. Quanto aos resultados, a PP teve um faixa de 80-90% de aprovação pelos alunos, já a PPD teve um índice um pouco menor em uma faixa de 72%- 80%. Edwards aponta que existe uma diferença entre PP e PPD, principalmente em relação à metodologia, mas o uso de uma ferramenta específica é uma solução viável.

Zacharis *et al.* (2011) realizou um estudo com alunos para investigar a efetividade de PPD no desempenho dos alunos e a sua motivação em um curso de introdução de Java [Zacharis et al. 2011]. O experimento foi conduzido com 129 alunos. A comparação foi feita com a programação individual (PI) e os resultados apontaram que os alunos que usaram PPD tiveram 50% menos defeitos, evidenciando a qualidade do código, e foram mais produtivos baseado na métrica de LOC/h [Zacharis et al. 2011]. O desempenho também aumentou com as notas dos alunos. Os alunos também responderam uma *survey* ao final do curso, relatando que PPD deu mais motivação e satisfação.

5. Lições Aprendidas

Os estudos conduzidos por Hanks (2005), Zacharis et al. (2011), Edwards et al. (2010) e Zacharris et al. (2011) apresentam características do uso e aplicação de PPD como ferramenta de ensino. Baseado nestes estudos, algumas lições aprendidas foram identificadas para apoiar a adoção e o uso de PPD no ensino de programação.

Lição 1: *A adoção de PPD necessita de uma ferramenta que possibilite o compartilhamento entre os pares:* o experimento de Hanks (2005) e Zacharis *et al.* (2010) utilizaram uma ferramenta própria para PPD, onde é possível simular outro cursor em forma de “gestos” apontando partes do código, semelhante ao que é feito com os dedos em pareamentos co-localizados. Uma ferramenta específica em PPD é recomendada também Canfora et al. (2006) e Ho et al. (2004) que afirmam que tais ferramenta reduzem o tempo de alternar entre diversos programas. Edwards *et al.* (2010) utilizou uma ferramenta proprietária de conferência que exigia apenas acesso a internet, possuindo facilidade na customização. Apesar de Zin et al. (2006) ter implementado a PPD de forma assíncrona, os alunos atestaram a importância de uma ferramenta específica.

Lição 2: *A PPD retorna bons resultados no desempenho, confiança, motivação, satisfação dos alunos:* Hanks (2005) e Zhacharis et al.(2011) obtiveram bons resultados na análise de desempenho dos alunos em função das notas dos alunos, porém Hanks (2005) afirma que cursos introdutórios, em geral, os alunos possuem notas elevadas. Uma limitação neste caso é que todos os quatro estudos utilizaram a PPD em cursos introdutórios. Hanks (2005) e Zin *et al.* obtiveram de retorno dos alunos que com a PPD eles se sentiram mais confiantes nas tarefas de programação. Já em relação em a satisfação e motivação dos alunos em realizar a prática, os resultados obtidos por Hanks (2005), Edward et al. (2010) e Zhacharis et al. (2011) apresentam resultados positivos.

Lição 3: *Existe uma necessidade de investigar mais aspectos de PPD, como a diferença de personalidade e a qualidade e produtividade do código dos alunos:* Nenhum dos quatro trabalhos analisou a colaboração em PPD com diferentes tipos de personalidades. Segundo a pesquisa de Walle (2009), baseado nas evidências de um estudo com empresas, a diferença de traços na personalidade pode aumentar a quantidade de comunicação entre um par em PP [Walle et al. 2009]. Outro ponto importante a ser levantado são variáveis de aspectos técnicos como a qualidade e produtividade do código, apenas o trabalho de Zacharis et al. (2010) analisou este tipo de métrica que em PP possui vários estudos empíricos como Mcdowell et al. (2002) e Chigona et al. (2008).

Lição 4: *É importante que a PPD seja implementada em seções:* Três trabalhos analisados utilizaram a prática de PPD em seções. Edward et al. (2010) utilizou a primeira seção para explicar sobre o conceito de PP e PPD e suas implicações no DDS aos alunos. Hanks (2005) atribui cinco tarefas de programação em cada uma das seções. Zacharis et al. (2010) no seu curso utilizou 4 seções com PPD, mesclando as outras com programação individual e PP. Apesar de Canfora et al. (2006) afirmar que não há aumento de esforço em PPD, seria interessante conduzir mais estudos para analisar o esforço da prática no contexto pedagógico.

Lição 5: *Existe a necessidade de avaliar o papel de um instrutor em PPD:* Segundo Hannay et al. (2010) a presença de um instrutor ajuda a evitar impasses entre os pares e possibilita que as dúvidas referentes a prática de PP possam ser rapidamente respondidas. Um estudo anterior de Edwards (1997) afirma que cursos a distância que tratem de programação é importante que se tenham encontros presenciais. No estudo realizado por Zin et al. (2006), os alunos afirmaram que a presença de encontros presenciais com o tutor é importante. Os autores indicam que na cultura asiática é comum os alunos terem encontros face a face com os instrutores.

Lição 6: *Existe a necessidade de se avaliar a importância de PPD no ensino como instrumento de capacitação do currículo do aluno para o DDS:* Nenhum dos quatro estudos buscou avaliar a importância da PPD não apenas como estratégia de ensino de programação de forma remota, mas também como parte de formação e preparo dos alunos a indústria. Muitas empresas tem adotado o DDS e as práticas ágeis tem sido uma estratégia para diminuir os desafios [Paasivara et al. 2009]. Neste caso, a PPD no ensino de programação também auxiliaria na capacitação dos alunos frente a demanda por profissionais por parte das empresas que utilizam DDS.

6. Considerações Finais

A partir dos resultados dos estudos foi possível identificar os benefícios da PPD no processo de ensino. Assim como a PP, a PPD também pode ser utilizada de forma efetiva como ferramenta pedagógica, colaborando em variáveis como o desempenho, confiança e motivação dos alunos. Do tema foram identificadas seis lições aprendidas, as quais caracterizam o estado da arte de PPD no ensino e listam possíveis oportunidades de pesquisa na área.

Em relação a trabalhos futuros, pretende-se avaliar como o ensino de programação por meio da prática de PPD pode colaborar para a formação de profissionais mais bem preparados para o mercado de DDS. Além disso, muitas oportunidades de pesquisa ainda existem como identificadas em algumas lições aprendidas levantadas neste trabalho. Essa pesquisa visa apoiar tanto a educação, dando suporte para PPD como prática de ensino e parte do conteúdo a ser aplicado no DDS, como a indústria, capacitando para que o currículo dos estudantes seja de acordo com a realidade e necessidades das empresas.

Agradecimentos

O desenvolvimento deste trabalho tem apoio financeiro do convênio PUCRS/Thoughtworks.

Referências

- Baheti, P. (2002) "Assessing distributed pair programming". In: OOPSLA, 2002, p. 50–51.
- Brereton, P.; Turner, M.; Kaur, R. (2009) "Pair programming as a teaching tool: a student review of empirical studies". In: Software Engineering Education and Training, 2009, p. 240-247.
- Brought, G.; Eby, L.; Wahls, T. (2008) "The effects of pair-programming on individual programming skill". In: SIGCSE, 2008, p. 200-204.
- Brought, G.; Wahls, T.; Eby, L. (2011) "The case for pair programming in the computer science classroom". *ACM Transactions on Computing Education*, Fev-2011, vol.11-1.
- Canfora, G.; Cimitle, A.; Visaggio, C.; Di Lucca, G. (2006) "How distribution affects the success of pair programming". In: International Journal of Software Engineering and Knowledge Engineering, p. 293-313.
- Carver, C.; Henderson, L.; Lulu, H.; Hodges, J.; Reese, D. (2007) "Increased Retention of Early Computer Science and Software Engineering Students Using Pair Programming". In: Software Engineering Education & Training, p.115-122.
- Chigona, W.; Pollock, M. (2008) "Pair programming for information systems students new to programming: Students' experiences and teachers' challenges," In: Portland International Conference on Management of Engineering & Technology, p.1587-1594.
- Edwards, W.K., Elizabeth D. Mynatt, K. P., Mike J. S., Douglas B. T. & Marvin M. T. (1997) "Designing and implementing asynchronous collaborative applications with Bayou". In: Proceedings of the 10th annual ACM symposium on User interface software and technology.
- Edwards, R. L., Stewart, J. K., & Ferati, M. (2010) "Assessing the Effectiveness of Distributed Pair Programming for an Online Informatics Curriculum". In: ACM Inroads, 1(1), 48-54
- Estácio, B. (2013) "Desenvolvimento de um conjunto de boas práticas para a programação em par distribuída". Dissertação de Mestrado, PPGCC-PUCRS.
- Han, L; Wenjuan X. (2011) "An experimental research of the pair programming in java programming course," In: International Conference on e-Education, Entertainment and e-Management (ICEEE), p.257-260.
- Hanks, B. (2005) "Student performance in CS1 with distributed pair programming". In: SIGCSE Bull, p. 316-320.
- Hanks, B. (2008) "Problems encountered by novice pair programmers". *ACM Journal on Educational Resources in Computing*, vol. 7-4.
- Hannay, J.; Arisholm, E.; Engvik, H.; Sjoberg, D. (2010) "Effects of personality on pair programming". In: IEEE Transactions on Software Engineering, vol. 36-1, pp.61–80.
- Han, L; Wenjuan X. (2011) "An experimental research of the pair programming in java programming course," In: International Conference on e-Education, Entertainment and e-Management (ICEEE), p.257-260.

- Ho, C.; Raha, C.; Gehringer, E.; Williams, L. (2004) "Sangam: a distributed pair programming plug-in for eclipse". In: OOPSLA, p. 73–77.
- L'Erario, A. Pessôa, M. (2008). "Um método de ensino e práticas de desenvolvimento distribuído de software para cursos de graduação. In: Anais do XXXVI COBENGE - Congresso Brasileiro de Educação em Engenharia, São Paulo.
- Mcdowell, C.; Werner, L. ; Bullock, H. ; Fernald, J. (2002) "The effects of pair-programming on performance in an introductory programming course". In: SIGCSE technical symposium on Computer science education, p. 38–42.
- Mendes, E.; Basil, L.; Fakhri, A.; Reilly, A." (2005) "Investigating pair-programming in a 2-year software development and design computer science course". In: SIGCSE Bull., p. 296–300.
- Natsu, H.; Favela, J.; Moran, A. L.; Decouchant, D.; Martinez-Enriquez, A. M. (2003) "Distributed pair programming on the Web". In: Proceedings of the Fourth Mexican International Conference, p. 81-88.
- Nagappan, N.; Williams, L., Ferzli, M.; Wiebe, E.; Miller, K.; Balik, S. (2003) "Improving the CS1 experience with pair programming". In: SIGCSE Bull., p. 359-362.
- Paasivaara, M.; Durasiewicz, S.; Lassenius, C. (2009) In: "Using Scrum in Distributed Agile Development: A Multiple Case Study". In: ICGSE, p. 195-204
- Ramli, N.; Fauzi S. (2008) "The effects of pair programming in programming language subject". In: International Symposium on Information Technology, p.1-4.
- Salleh, N.; Mendes, E.; Grundy, J.; Burch, G. (2009). "An empirical study of the effects of personality in pair programming using the five-factor model". In: International Symposium on Empirical Software Engineering and Measurement, 12p.
- Salleh, N.; Mendes, E.; Grundy, J. (2011) "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review". *IEEE Transactions on Software Engineering*, vol.37-4, Jul-Ago.. 509–525.
- Stotts, D.; Mc. Smith, J.; Gyllstrom, K. (2004) "Support for Distributed Pair Programming in the Transparent Video Facetop". Relatório Técnico, Universidade da Carolina do Norte.
- Walle, T.; Hannay, J.E. (2009) "Personality and the nature of collaboration in pair programming". In: Empirical Software Engineering and Measurement, p.203-213.
- Zin, A.M., Idris, S., & Subramaniam, N.K. (2006) "Improving Learning of Programming Through E-Learning by Using Asynchronous Virtual Pair Programming". In: Turkish Online Journal of Distance Education-TOJDE, vol.7, number 3, article 13.
- Zacharis, N. (2011) "Measuring the effects of virtual pair programming in an introductory programming java course". *IEEE Transactions on Education*, vol. 54-, Fev-2011, pp. 168–170.