

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA AVALIAÇÃO EMPÍRICA SOBRE A
APRENDIZAGEM COLABORATIVA EM
CODING DOJO RANDORI NO CONTEXTO DE
DESENVOLVIMENTO DE SOFTWARE**

BERNARDO JOSÉ DA SILVA ESTÁCIO

Tese apresentada como requisito parcial à
obtenção do grau de Doutor em Ciência da
Computação na Pontifícia Universidade
Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Rafael Prikladnicki

Porto Alegre
2017

Ficha Catalográfica

E82 Estácio, Bernardo José da Silva

Uma avaliação empírica sobre a aprendizagem colaborativa em coding dojo randori no contexto de desenvolvimento de software / Bernardo José da Silva Estácio . – 2017.

173 f.

Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rafael Prikladnicki.

I. Engenharia de Software. 2. Práticas de Codificação Colaborativa. 3. Coding Dojo Randori. 4. Programação em Par. 5. Aprendizagem Colaborativa. I. Prikladnicki, Rafael. II. Título.

BERNARDO JOSÉ DA SILVA ESTÁCIO

**UMA AVALIAÇÃO EMPÍRICA SOBRE A APRENDIZAGEM
COLABORATIVA EM *CODING DOJO RANDORI* NO CONTEXTO DE
DESENVOLVIMENTO DE SOFTWARE**

Tese apresentada como requisito parcial para obtenção do grau de Doutor em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 28 de Março de 2017.

BANCA EXAMINADORA:

Prof. Dr. Duncan Dubugras Ruiz (PPGCC/PUCRS)

Prof. Dr. Cleidson Ronald Botelho de Souza (PPGCC/UFPA)

Prof. Dr. Igor Fábio Steinmacher (UFTPR)

Prof. Dr. Rafael Prikladnicki (PPGCC/PUCRS - Orientador)

DEDICATÓRIA

Dedico este trabalho à minha avó Beatriz Estácio

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus, por toda sua bondade e graça comigo.

Aos meus pais, que mesmo longe me deram toda força necessária, assim como toda a minha família. À minha avó Beatriz que tanto me ama.

À Kamila Baltazar por sua amizade e companheirismo durante os anos de doutorado.

Ao Professor Rafael Prikładnicki por ter acreditado em mim desde o início. Muitas vezes me ajudou a voltar ao foco da pesquisa, motivando e instruindo. Aprendi muito contigo! Muito obrigado!

Ao Professor Lutz Prechelt por ter me recebido muito bem em Berlim. Ao grupo AGSE (em especial Gesine, Franz e Holger), onde recebi excelente feedback e pude vivenciar um pouco da cultura alemã. Aos amigos que fiz enquanto estive na Alemanha: Isabela Binotti, Rodrigo Pastl, Tayana, Analice, Caroline Visotto, Tiago Tasca, Marcelo, Gabriela, Alberto e Vanessa.

À Professora Sabrina Marczak pelos conselhos e discussões sobre pesquisas científicas. Muito obrigado!

À Professora Tayana Uchoa Conte por toda sua ajuda no período que visitei o grupo de pesquisa USES.

Ao Professor Alessandro Garcia pela colaboração científica que desenvolvemos. E também ao Roberto Oliveira e Marcos Kalinowski em tornar isto possível.

À Thoughtworks Inc. por ter financiado e por ter incentivado a minha pesquisa. Tive a oportunidade de participar de um projeto inovador e conhecer pessoas fantásticas. Em especial, gostaria de agradecer ao Alejandro Ochik, Michael Morá, Amanda Scopel, Ana Steffens, João Stocker, Pedro Guidoux, Thais Hamilton, Leonardo Iglesias e tantos outros que tive a oportunidade de interagir durante estes quatro anos.

Aos amigos que fiz durante este tempo na FACIN, em especial: Alessandra Dutra, Sílvia Nunes, Alan Santos, Samuel Souza, Aline Zanin, Joaquim Assunção e Thiago Paes. Em especial à Josiane Kroll por ter ajudado na revisão do texto, e por ter compartilhado a estrada acadêmica em tantos momentos.

Aos amigos do grupo de pesquisa: Letícia dos Santos Machado, Carolina Toscani, Julia Couto e Greice Roman. Obrigado pelo companheirismo e por compartilhar vários almoços juntos.

Aos grandes amigos que o mestrado me deu, e que seguiram firme nessa amizade:
Luciana Espindola e Claiton Corrêa.

À secretaria do PPGCC por ter colaborado de forma eficiente e eficaz em todas as vezes em quem precise de alguma ajuda. Em especial: Régis Escobal e Diego Cintrão.

Aos meus amigos de Belém por entenderem o período que fiquei tão distante da terrinha, sem poder visitá-los.

Às amizades que desenvolvi durante os congressos e simpósios científicos, e as colaborações que desenvolvemos. Em especial: Rodrigo Santos, Natasha Valentim, Awdren Fontão e Raoul Vallon.

Aos participantes dos estudos empíricos conduzidos por terem aceitado a contribuir com esta pesquisa.

UMA AVALIAÇÃO EMPÍRICA SOBRE A APRENDIZAGEM COLABORATIVA EM *CODING DOJO RANDORI* NO CONTEXTO DE DESENVOLVIMENTO DE SOFTWARE

RESUMO

O desenvolvimento de software tem se tornado cada vez mais uma atividade que necessita de um esforço social e colaborativo. Neste cenário, não há apenas a necessidade de desenvolver competências técnicas, mas também de saber trabalhar em equipe. As práticas de codificação colaborativa representam uma importante ferramenta de aprendizagem colaborativa e treinamento entre desenvolvedores. O *Coding Dojo Randori* (CDR) é uma prática de codificação colaborativa que tem sido adotada de forma crescente na indústria de software e seu principal propósito é a aprendizagem colaborativa. Esta tese tem por objetivo avaliar a aprendizagem colaborativa em CDR. A metodologia de pesquisa foi estruturada em três etapas: Exploratória, Avaliação e Consolidação. Na primeira etapa, realizou-se a caracterização do estado da arte de práticas de codificação colaborativa, assim como a execução de dois estudos de viabilidade de CDR. Na segunda etapa foram conduzidos estudos observacionais e estudos de caso, onde se pôde avaliar com mais ênfase a aprendizagem colaborativa em CDR. A última etapa consolidou os resultados por meio da concepção de um conjunto de diretrizes para apoiar a adoção de CDR com foco na aprendizagem colaborativa. A metodologia de pesquisa deste trabalho possibilitou uma avaliação extensiva de CDR, e a identificação de abordagens que podem ser utilizadas para avaliar a aprendizagem colaborativa em CDR. Os resultados apresentaram uma percepção positiva dos participantes em relação a prática, assim como um entendimento de níveis de colaboração que podem contribuir para a aprendizagem colaborativa.

Palavras chaves: Engenharia de Software, Práticas de Codificação Colaborativa, *Coding Dojo Randori*, Programação em Par, Aprendizagem Colaborativa.

EMPIRICAL EVALUATION OF COLLABORATIVE LEARNING IN CODING RANDORI DOJOS IN SOFTWARE DEVELOPMENT CONTEXT

ABSTRACT

Software development has become increasingly an activity that requires a social and collaborative effort. In this context, there is not only a need to develop technical skills but also to know how to develop teamwork skills. Collaborative programming practices are important tools for collaborative learning and training among software developers. The Coding Randori Dojo (CRD) is a collaborative coding practice that has been increasingly adopted in the software industry, and its main purpose is to provide collaborative learning. The goal of this dissertation is to evaluate collaborative learning in CRD. The methodology is structured in three stages: Exploratory, Evaluation and, Consolidation. In the first stage, a characterization of collaborative coding practices was performed, as well as an execution of two feasibility studies. In the second stage, we conducted observational studies and case studies, where collaborative learning has been evaluated with more emphasis. The last stage consolidated the results by the conception of a set of guidelines. The research methodology of this dissertation leads to an extensive evaluation of CRD, and the identification of approaches that can be used to evaluate collaborative learning during the practice. The findings present benefits perceived by the participants regarding CRD, as well an understanding of collaboration levels that can contribute to the collaborative learning.

Keywords: Software Engineering, Collaborative Programming Practices, *Coding Randori Dojo*, Pair Programming, Collaborative Learning.

LISTA DE FIGURAS

Figura 2.1 – Estágios da Zona de Desenvolvimento Proximal [GAR13][THA88].....	40
Figura 3.1 – Desenho de Pesquisa.....	43
Figura 3.2 – Visão do software Atlas.ti para análise das sessões de CDR.....	46
Figura 4.1 – Três passos seguidos na RSL e a quantidade respectiva de artigos selecionados.....	53
Figura 4.2 – Qualidade dos estudos primários	54
Figura 4.3 – Métodos de pesquisas utilizados	55
Figura 4.4 – Métodos de pesquisa de acordo com a classificação de Wieringa.....	55
Figura 4.5 – Ano de publicação dos artigos identificados	56
Figura 4.6 – Contexto das Publicações	57
Figura 4.7 – Veículo de publicação dos artigos primários.....	57
Figura 4.8 – Evidências identificadas nos artigos primários de PP.....	60
Figura 4.8 – Evidências identificadas nos artigos primários de CDR.....	61
Figura 4.10 – Categorias de tipo de avaliação nos estudos primários.....	62
Figura 4.11 – Classificação dos estudos primários quanto ao tamanho da amostra participantes nos cursos e treinamentos mapeados.....	65
Figura 5.1 – Dimensões de motivação no grupo de PP.....	74
Figura 5.2 – Dimensões de motivação no grupo de CDR.....	75
Figura 5.3 – Dimensões de experiência de usuário no grupo de PP	76
Figura 5.4 – Dimensões da experiência de usuário no grupo de CDR	76
Figura 5.5 –Item aprendizagem de longo prazo para os grupos de PP.....	77
Figura 5.6 – Item aprendizagem de longo prazo para o grupo de CDR.	78
Figura 5.7 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de PP ..	78
Figura 5.8 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de CDR.	78
Figura 6.1 – Configuração do <i>crossed design</i> no experimento.....	86

Figura 6.2 – Dimensões de motivação no grupo de PP.....	88
Figura 6.3 – Dimensões de motivação no grupo de CDR.....	89
Figura 6.4 – Item aprendizagem de longo prazo para o grupo de PP.	90
Figura 6.5 – Item aprendizagem de longo prazo para o grupo de CDR.	90
Figura 6.6 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de PP ..	91
Figura 6.7 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de CDR	91
Figura 6.9 – Dimensões de experiência de usuário no grupo de PP	92
Figura 6.10 – Dimensões de experiência de usuário no grupo de CDR	92
Figura 8.1 – Níveis de Colaboração identificados nas sessões de CDR.	102
Figura 8.2 – Caracterização do Nível de Conceito	103
Figura 8.3 – Caracterização do Nível de Tarefa	103
Figura 8.4 – Caracterização do Nível de Programação	104

LISTA DE TABELAS

Tabela 2.1 – Visão geral do Modelo de Kirkpatrick [KIR94] extraído de [SAV11].....	36
Tabela 2.2 – Estrutura da Taxonomia de Bloom, adaptado de [SAV11].....	38
Tabela 4.1 – Termos utilizados na abordagem PICO	49
Tabela 4.2 – Checklist de Avaliação de Qualidade do Artigos.....	51
Tabela 4.3 – Quantidade de artigos retornados por base de dados	53
Tabela 4.4 – Caracterização dos cursos que adotam práticas de codificação colaborativa	64
Tabela 5.1 – Experiência por participante em cada grupo.....	70
Tabela 5.2 Argumentos adaptados do questionário pós-estudo de Savi et al. [SAV11]	72
Tabela 6.1 – Experiência dos participantes em cada grupo	85
Tabela 7.1 – Característica das sessões de CDR de Estudos Observacionais.....	98
Tabela 7.2 – Características técnicas das sessões de CDR de Estudos Observacionais.	99
Tabela 7.3– Característica das sessões e dos participantes de CDR do Estudos de Caso	100
Tabela 7.4 – Características técnica das sessões de CDR dos Estudos de Caso	100

LISTA DE ABREVIATURAS

- CBO** – *Coupling between Objects*
- CD** – *Coding Dojo*
- CDR** – *Coding Dojo Randori*
- ES** – Engenharia de Software
- FUB** – Freie Universität Berlin
- GT** – *Grounded Theory*
- PASQI** – *Pan American Software Quality Institute*
- PP** – Programação em Par
- PUC-RIO** – Pontifícia Universidade Católica do Rio de Janeiro
- PUCRS** – Pontifícia Universidade Católica do Rio Grande do Sul
- RSL** – Revisão Sistemática da Literatura
- TDD** – *Test-Driven Development*
- TU Wien** – Technische Universität Wien
- UFAM** – Universidade Federal do Amazonas
- UNIRIO** – Universidade Federal do Estado do Rio de Janeiro
- XP** – *eXtreme Programming*
- WMC** – *Weighted Method per Class*
- ZDP** – Zona de Desenvolvimento Proximal

SUMÁRIO

1	INTRODUÇÃO	26
1.1	Objetivos	27
1.2	Justificativa e Relevância	27
1.3	Publicações e Apresentações Científicas	28
1.4	Organização do Volume	29
2	REFERENCIAL TEÓRICO	31
2.1	Desenvolvimento de Software	31
2.2	Coding Dojo	33
2.2.1	Estilos de Coding Dojo	33
2.2.2	Coding dojo randori	34
2.3	Teorias de Aprendizagem e Modelos de Avaliação	35
2.3.1	Rooksby et al. [ROO14]	35
2.3.2	Savi [SAV11]	35
2.4	Aprendizagem em Coding dojo randori (CDR)	38
2.4.1	Construtivismo Social	39
3	METODOLOGIA DE PESQUISA	41
3.1	Questões de Pesquisa	41
3.2	Etapas e Fases de Pesquisa	41
3.2.1	Metodologia de Schull et al. [SCH01] com extensão de Mafra et al. [MAF06]	41
3.2.2	Desenho de Pesquisa	42
3.2.3	Etapa 1: Exploratória	43
3.2.4	Etapa 2: Avaliação	44
3.2.5	Etapa 3: Consolidação	44
3.3	Aspectos Metodológicos	44
3.3.1	Fase 1 – Revisão Sistemática da Literatura	45
3.3.2	Fase 2 – Estudos de Viabilidade	45
3.3.3	Fase 3 e 4 – Estudos de Observação e Estudos de Caso	45
3.3.4	Consentimento dos Participantes	46
3.4	Visitas de Pesquisa e Parcerias	46
4	FASE 1: RSL SOBRE PRÁTICAS DE CODIFICAÇÃO COLABORATIVA	48
4.1	Protocolo da RSL	48
4.1.1	Objetivo e Questões de Pesquisa	48

4.1.2	Pesquisa na Literatura.....	49
4.1.3	Seleção dos Artigos.....	50
4.1.4	Extração dos dados.....	51
4.1.5	Avaliação da qualidade dos estudos.....	51
4.2	Execução.....	52
4.3	Resultados da Análise Quantitativa.....	54
4.3.1	Qualidade dos estudos primários.....	54
4.3.2	Método de Pesquisa.....	54
4.3.3	Análise por Ano.....	55
4.3.4	Tipo de Publicação.....	56
4.3.5	Análise por veículo de publicação.....	57
4.4	Resultados da Análise Qualitativa.....	58
4.4.1	RQ1: Principais evidências de práticas de codificação colaborativa.....	58
4.4.2	RQ3: Configuração dos cursos e treinamentos identificados nos estudos primários..	63
4.5	Discussão dos resultados.....	66
4.6	Limitações desta Revisão Sistemática.....	67
4.7	Considerações Finais do Capítulo.....	67
5	FASE 2: ESTUDO DE VIABILIDADE I.....	69
5.1	Contexto e Objetivo.....	69
5.2	Participantes.....	69
5.3	Procedimento e Materiais.....	71
5.4	Resultados.....	73
5.4.1	Motivação.....	73
5.4.2	Experiência do Usuário.....	75
5.4.3	Aprendizagem.....	77
5.4.4	Benefícios.....	79
5.4.5	Desvantagens de CDR.....	80
5.4.6	Mockups.....	80
5.5	Discussão.....	81
5.6	Ameaças à validade.....	81
5.6.1	Validade de Construto.....	81
5.6.2	Validade Interna.....	81
5.6.3	Validade Externa.....	82
5.6.4	Validade de Conclusão.....	82
5.7	Considerações Finais do Capítulo.....	82
6	FASE 2: ESTUDO DE VIABILIDADE II.....	84
6.1	Contexto e Objetivo.....	84

6.2	Participantes	84
6.3	Procedimento e Materiais	86
6.4	Resultados	87
6.4.1	Motivação	88
6.4.2	Aprendizagem	90
6.4.3	Experiência do Usuário.....	91
6.4.4	Benefícios de CDR	93
6.4.5	Desvantagens de CDR	93
6.4.6	Impacto no Curso de CDR.....	94
6.5	Discussão	94
6.6	Ameaças a validade	94
6.6.1	Validade do Construto	94
6.6.2	Validade Interna	95
6.6.3	Validade Externa	95
6.6.4	Validade de Conclusão.....	95
6.7	Considerações Finais	95
7	FASES 3 E 4: ESTUDOS DE OBSERVAÇÃO E ESTUDOS DE CASO (CICLO DE VIDA REAL)	97
7.1	Fase 3: Estudos de Observação.....	97
7.2	Fase 4: Estudos de Caso	99
8	FASES 3 E 4: RESULTADOS	101
8.1	Notação e Formato	101
8.2	Níveis de Colaboração	101
8.2.1	Nível de Conceito	102
8.2.2	Nível de Tarefa	103
8.2.3	Nível de Programação	104
8.3	Papéis	104
8.3.1	Mestre	104
8.3.2	Novato	105
8.4	Intervenção do Mestre	105
8.4.1	Intervenção Reativa do Mestre	105
8.4.2	Intervenção Proativa do Mestre	106
8.5	Participação dos Novatos	106
8.6	Participação da Audiência	107
8.6.1	Sincronização com os pares.....	107
8.6.2	Audiência como Segundo Copiloto.....	108
8.7	O Facilitador como Mestre	109

8.8	A Aprendizagem na ZDP	109
8.9	Melhorias para CDR	111
8.10	Retrospectiva.....	111
8.10.1	Estudos de Observação.....	111
8.10.2	Estudos de Caso	112
8.11	Diferenças entre estudos de observação e de estudos de caso.....	113
8.12	Ameaças a validade do estudo.....	114
8.13	Considerações Finais do Capítulo	114
9	FASE 5: UM CONJUNTO DE DIRETRIZES PARA APOIAR A ADOÇÃO DE CDR 115	
9.1	Diretrizes	115
9.2	Oportunidades de Pesquisa oriunda das diretrizes propostas.....	119
9.3	Avaliação das Diretrizes	120
9.4	Limitações das Diretrizes	121
9.5	Considerações finais do capítulo.....	121
10	CONCLUSÃO	122
10.1	Resumos dos resultados e objetivos.....	122
10.2	Principais Contribuições	123
10.3	Trabalhos Futuros	124
10.4	Reflexão Final	125
	REFERÊNCIAS BIBLIOGRÁFICAS	126
	APÊNDICE A	134
	APÊNDICE B	137
	APÊNDICE C	140
	APÊNDICE D	148
	APÊNDICE E	160
	ANEXO I	161

1 INTRODUÇÃO

O desenvolvimento de software tem se estabelecido como uma atividade que exige um esforço colaborativo [KII14]. A crescente adoção de abordagens como métodos ágeis evidenciam ainda mais este cenário. Neste contexto, as práticas de codificação colaborativa exercem um papel importante de desenvolver competência não apenas técnicas, mas de trabalho em equipe [ARO12] [ST012].

Os primeiros estudos com práticas de codificação colaborativa focaram principalmente na Programação em Par (PP), onde os autores buscaram apresentar uma comparação entre PP e programação individual em termos de custos e benefícios [NOS98] [COC11]. Nas últimas décadas, PP se tornou muito popular na indústria, devido a crescente adoção de métodos ágeis como *Extreme Programming* – XP, e na academia.

As práticas de codificação colaborativa, especialmente a PP, estão bem relacionadas com a aprendizagem, seja por estudantes [SAL11], seja por profissionais [BEC04]. Em um cenário mais recente, outras práticas de codificação colaborativa emergiram, como o *Coding Dojo* (CD).

O CD é uma prática onde um grupo de desenvolvedores se reúne para aprender e praticar algum conceito de tecnologia (linguagem de programação, *framework*) juntos. Existem diferentes estilos de dinâmica de CD, um deles se chama Randori (*Coding Dojo Randori* – CDR) e é amplamente adotado na indústria [ROO14].

Entretanto, apesar da sua adoção, poucos estudos empíricos investigaram CDR. Os estudos iniciais demonstram que a prática é promissora na aprendizagem de *Test-Driven Development* (TDD) [HEI13][LUZ13] e na prática de linguagens de programação, como Ruby [SAT08]. Tais estudos discutem, em um primeiro momento, a efetividade da prática, mas ainda se faz necessário compreender seu funcionamento e como avaliá-la, especificamente quanto à sua principal característica: a aprendizagem colaborativa [ROO14].

A aprendizagem colaborativa em CDR é bem peculiar, haja vista que em CDR a efetividade não é medida pela quantidade de tarefas terminadas, mas pelas interações e discussões entre os participantes e a prática constante de exercícios de programação [SAT08].

Desta forma, CDR pode ser considerada uma prática onde a aprendizagem vem por meio do treinamento, uma prática deliberada [ERI93][SAT08] que leva à proficiência. É também uma prática social estritamente relacionada ao construtivismo social, no qual o

aprendizado é condicionado ao contexto de grupo e à colaboração entre os participantes do processo [VYG78].

Entender e avaliar um cenário de aprendizagem colaborativa tão complexo como o de CDR ainda é uma lacuna que precisa ser preenchida para impulsionar e aprimorar a adoção desta prática na indústria.

A presente tese visa preencher este espaço, tanto do ponto de vista do estado da arte de CDR, quanto do ponto de vista prático. Portanto, a questão de pesquisa que norteia este trabalho é: ***como se pode avaliar a aprendizagem colaborativa durante a execução da prática de CDR no contexto de desenvolvimento de software?***

1.1 Objetivos

O **objetivo geral** dessa pesquisa é **avaliar empiricamente a aprendizagem colaborativa da prática de CDR**.

De forma a alcançar o objetivo geral proposto, os seguintes **objetivos específicos** foram definidos:

- OE1: Identificar as principais características da adoção de práticas de codificação colaborativa para fins de ensino e treinamento de programação;
- OE2: Identificar as principais vantagens e desvantagens de CDR.
- OE3: Analisar as estratégias de avaliação da aprendizagem colaborativa em CDR.
- OE4: Propor um conjunto de diretrizes que auxilie na adoção de CDR com foco na aprendizagem colaborativa.

1.2 Justificativa e Relevância

No atual cenário de desenvolvimento de software, há uma demanda crescente para que desenvolvedores estejam aptos a atuar em equipe [ARO12] [STO12]. Esta mudança de paradigma se reflete também na aprendizagem [KIL14] e há um esforço para que o ensino em Engenharia de Software seja mais colaborativo [SAN08].

Apesar do empenho em incluir o desenvolvimento de software colaborativo nos currículos de engenharia de software [ACM13] e também a crescente discussão de práticas de codificação colaborativa por praticantes da indústria [KIL14], há poucas evidências científicas sobre como a aprendizagem colaborativa oriunda destas práticas podem auxiliar a suportar as habilidades e competências necessárias para os desenvolvedores.

Neste contexto, CDR é uma prática de codificação colaborativa que vem sendo adotada e discutida por profissionais da indústria [ROO14], tendo como principal característica a aprendizagem colaborativa. Rooksby et al. ratificam a importância de pesquisar mais sobre CDR:

CD é sem dúvidas a única abordagem disponível para desenvolvedores profissionais aprenderem continuamente. Entretanto o formato de *Coding Dojo Randori* é especialmente praticado e merece atenção. [ROO14]

Embora alguns autores como Sato et al. [SAT08] e Rooksby et al. [ROO14] discutam sobre as características e a importância da aprendizagem colaborativa em CDR, pouco se sabe como avaliar e entender esta variável na prática. Este estudo se situa nesta lacuna e busca agregar dados empíricos para caracterizar os efeitos provenientes de CDR e o apoio à crescente demanda por práticas de ensino em ES que atendam o cada vez mais colaborativo cenário de desenvolvimento de software.

1.3 Publicações e Apresentações Científicas

Durante o período desta tese, foram publicados um conjunto de artigos científicos, em temas direta ou indiretamente relacionados com os resultados desta pesquisa. A lista com artigos é apresentada abaixo:

Artigos em *Journal*

- Estácio, B.; Prikładnicki, R. "Distributed Pair Programming: A Systematic Literature Review". *Information and Software Technology*, Vol. 63, 2015, pp. 1-10.

Artigos em Conferências/Workshops

- Estácio, B., Zieris, F.; Prechelt, L.; Prikładnicki, R. "On the randori training dynamics". In: *Cooperative and Human Aspects of Software Engineering Workshop*, 2016. p. 44 - 47.
- Oliveira, R.; Estácio, B.; Garcia, A.; Marczak, S.; Prikładnicki, R.; Kalinowski, M.; Lucena, C. "Identifying Code Smells with Collaborative Practices: A Controlled Experiment". In: *10th Brazilian Symposium on Software Components, Architectures and Reuse*, pp. 61-70, 2016.
- Estácio, B.; Valentim, N.; Rivero, L.; Conte, T. U. ; Prikładnicki, R. "Evaluating the Use of Pair Programming and Coding Dojo in the Teaching of Mockup Development: An Empirical Study". In: *48th Hawaii International Conference on System Sciences*, 2015, pp. 5084 - 5093.

- Valentim, Natasha M.; Conte, T.; Estácio, B.; Rafael Prikladnicki. "How do software engineers apply an early usability inspection technique? A qualitative study." In: 27th International Conference on Software Engineering and Knowledge Engineering, 2015, p.6.
- Estácio, B., Oliveira, R., Marczak, S., Kalinowski, M., Garcia, A., Prikladnicki, R.; Lucena, C. "Evaluating Collaborative Practices in Acquiring Programming Skills: Findings of a Controlled Experiment". In: 29th Brazilian Symposium on Software Engineering, 2015, pp. 150 - 159.
- Estácio, B.; Prikladnicki, R.; Mora, M.; Notari, G.; Caroli, P.; Olchik, A. "Software Kaizen: Using Agile to Form High-Performance Software Development Teams". In: Agile Conference, 2014, 10p.

Artigo Apresentado em Workshop de Teses e Dissertações

- Estácio, B. "On the evaluation of Distributed Collaborative Programming". Apresentação da Proposta de Tese de Doutorado. ICSE Warm Up. Cbsoft Maceió, 2015.

Envolvimento na Comunidade Científica

- Coordenação do Comitê de Programa. Workshop Brasileiro de Métodos Ágeis, Curitiba 2016.

1.4 Organização do Volume

O restante desta tese está organizado da seguinte forma: o **Capítulo 2** apresenta o referencial teórico desta pesquisa, envolvendo os principais conceitos e implicações das áreas de estudos: Desenvolvimento de Software, Desenvolvimento Ágil, Método XP, PP, e CD. Além disso, apresenta-se a teoria dos modelos de aprendizagem relacionadas a esta pesquisa.

O **Capítulo 3** apresenta a abordagem metodológica desta pesquisa, onde são descritos cada uma das etapas e fases. Além disso, apresenta-se a abordagem metodológica, e os principais métodos utilizados como revisão sistemática da literatura, experimento controlado e estudo de caso.

No **Capítulo 4** é apresentada uma revisão sistemática da literatura sobre práticas de codificação colaborativa, o protocolo utilizado, a execução e os principais resultados.

No **Capítulo 5** apresenta-se o primeiro estudo de viabilidade executado com o objetivo de avaliar a prática de CDR e PP.

O **Capítulo 6** apresenta um segundo estudo de viabilidade, onde utilizando um grupo controle foi possível avaliar a experiência dos participantes em três contextos diferentes (programação individual, PP e CDR).

O **Capítulo 7** descreve o contexto e as principais características dos estudos de observação e estudos de caso conduzidos.

No **Capítulo 8** são apresentados e discutidos os resultados dos estudos de observação e estudos de caso.

No **Capítulo 9** foi desenvolvido um conjunto de diretrizes para CDR, a partir da consolidação dos resultados desta tese.

Por fim, no **Capítulo 10**, são apresentadas as considerações finais desta tese, enfatizando as principais contribuições. A conclusão destaca as oportunidades futuras de pesquisas a partir deste trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico relacionado aos principais conceitos desta pesquisa. A Seção 2.1 apresenta os principais conceitos em torno de Desenvolvimento de Software, destacando o desenvolvimento ágil de software, o método *Extreme Programming* e PP. Na Seção 2.2, aborda-se os principais estilos, e principalmente CDR, a prática que é o foco desta pesquisa. A Seção 2.3 apresenta teorias de aprendizagem e os modelos utilizados no âmbito desta pesquisa para avaliá-la. Por fim, a Seção 2.4 apresenta a perspectiva da aprendizagem colaborativa em CDR.

2.1 Desenvolvimento de Software

O software se tornou imprescindível em diferentes contextos da sociedade [PRE11]. A Engenharia de Software (ES) é a disciplina que norteia o desenvolvimento de software. A partir dela, teorias, métodos e ferramentas são aplicados a fim de apoiar o processo de desenvolvimento de software. A IEEE [PRE11] traz a definição de ES como sendo a “aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software”.

A Engenharia de Software pode ser vista em camadas embasadas na qualidade [PRE11]. Enquanto a camada dos métodos fornece a técnica para desenvolver o software, a camada de ferramentas suporta de forma automatizada ou semi-automatizada para os métodos. A camada de processo é o alicerce, pois estabelece o contexto no qual os métodos e as ferramentas serão aplicados. A camada de processo é responsável pelo relacionamento com as outras camadas, permitindo o desenvolvimento racional do software [PRE11].

Em um processo de software são descritas uma série de atividades e resultados associados à produção deste [SOM11]. A representatividade e a descrição de um processo são feitas por meio de modelos que possuem características específicas, podendo estas serem semelhante na teoria, mas diferentes na prática [SOM11].

Modelo Prescritivo versus Adaptativo

Neste contexto, pode-se classificar modelos de processo em: modelo prescritivo e modelo adaptativo. O primeiro grupo, prescritivos, possui em sua estrutura uma ordem mais rígida e formal de elementos do processo, tais como atividades, tarefas e produtos de trabalho. Além disso, possuem um fluxo de trabalho que descreve como cada um destes elementos se relaciona uns com os outros. Estes modelos possuem como características a estrutura e a ordem dos elementos [PRE11].

Ao contrário dos modelos prescritivos, os modelos adaptativos, como o nome sugere, são abertos a mudanças [PRE11]. Estes modelos possuem um forte embasamento em dados empíricos, isto é, nas experiências vivenciadas pelos praticantes [BAS08] [PRE11]. Entre os métodos da abordagem adaptativa estão os métodos ágeis .

Desenvolvimento Ágil de Software

A base de conceitos dos métodos ágeis, como o desenvolvimento iterativo e incremental e a abordagem adaptativa, é discutida desde a década de 70 [FLO13]. Entretanto, foi em 2001 que os princípios da agilidade ficaram esclarecidos, quando um grupo de profissionais se reuniu para estabelecer o Manifesto Ágil [BEC16].

O Manifesto Ágil enfatiza Indivíduos e Interações acima de Processos e Ferramentas, Software Operacional acima de Documentação Completa, Colaboração dos Clientes acima de Negociação Contratual, Respostas a Mudanças acima de Seguir um Plano. Diferentes métodos emergiram sob os princípios do Manifesto Ágil e ganharam popularidade na indústria. Entre eles, pode-se citar: o *framework* Scrum, Crystal, FDD e o *Extreme Programming* (XP) [PRE14].

Extreme Programming

O XP surgiu no início de 2000, quando Kent Beck reuniu suas experiências em projetos na indústria de desenvolvimento de software e propôs um método que se tornou bem conhecido e adotado. Desta forma, o método XP surgiu baseado em valores como comunicação, simplicidade, *feedback*, respeito e coragem.

O XP é um método leve que foca em entregas iterativas e incrementais. Na segunda versão do método proposta em 2004 [BEC04][BAS08], treze práticas primárias o integram, entre elas se destacam *Test-Driven Development* (TDD) e Programação em par (PP).

Programação em Par

Com a popularidade do método XP, a PP foi impulsionada na indústria e na academia, sendo discutida e investigada ao longo dos anos. Em uma sessão clássica de PP, um dos desenvolvedores atua como o piloto que controlará o teclado e o mouse enquanto outro desenvolvedor atua como copiloto, auxiliando o piloto, principalmente revisando o código e discutindo a solução [BEC04].

Com o tempo, pesquisadores investigaram diferentes aspectos de PP no contexto acadêmico e na indústria. Em relação ao contexto da indústria estudo empíricos demonstraram aspectos importantes da prática no desenvolvimento de software como a

produtividade [HUL05], a qualidade do código [VAN07] e o entendimento de PP em tarefas ou sistemas mais complexos [CHO07].

No contexto acadêmico, PP foi investigada como uma ferramenta educacional para ensino em cursos de computação. Desde Nosek [NOS98], um dos primeiros estudos que investigaram PP [NOS98], um conjunto de evidências empíricas foi coletado e diferentes aspectos de PP no ensino foram analisados, tais como: o impacto no desempenho dos alunos [MCD03, WIL03, MEN05], a confiança [MCD03, SAL10], a motivação [MCD03]. No geral, PP se mostrou ser efetiva como uma prática de codificação colaborativa no ensino em cursos de computação [SAL11], despertando a atenção para outras práticas de codificação que envolvessem colaboração.

2.2 *Coding Dojo*

Ainda no contexto de práticas de codificação colaborativa, o CD é uma prática onde um grupo de desenvolvedores de software trabalha em conjunto. O principal objetivo do CD é promover um ambiente de aprendizagem com colaboração e sem competição entre os membros [SAT08]. A perspectiva por trás de CD é de uma prática que promova aprendizagem contínua para os desenvolvedores [ROO14].

O termo *Dojo* é uma palavra de origem japonesa [OXF17], que significa o lugar onde se é praticado judô e outros estilos de artes marciais. Bossavilt e Gaillot se inspiraram neste formato de treinamento colaborativo e propuseram utilizar em desenvolvimento de software [BOS05]. Desta forma, emergiu o conceito de *Coding Dojo* (CD), apresentado em um workshop da conferência internacional de métodos ágeis (XP *Conference*) [ROO14].

2.2.1 Estilos de *Coding Dojo*

Na literatura existem diferentes estilos de CD com peculiaridades em relação a dinâmica da prática. Abaixo, são detalhados quatro dos estilos reportados na literatura.

- **Kata:** Algumas tarefas são executadas previamente e então são realizadas novamente em um segundo momento, com apresentação para a audiência. A apresentação dura entre 10 e 30 minutos, seguida de feedback [ROO14]. As tarefas nesse estilo adotam o nome de “exercícios *Kata*”, sendo este termo também utilizado nos outros estilos, quando se refere a um exercício utilizado.
- **Wasa:** É um estilo muito parecido com o *Kata*, porém realizado em pares. Em *Wasa*, duas pessoas utilizam PP e discutem o exercício, enquanto uma audiência os acompanha. Este formato também é muito ligado à TDD. Enquanto um

desenvolvedor escreve o teste unitário, o outro no turno seguinte implementará esta funcionalidade [ROO14].

- **Kake:** Este estilo consiste em eventos onde uma mesma funcionalidade é desenvolvida simultaneamente em diferentes plataformas ou linguagens de programação [ROO14].
- **Randori:** Segundo Rooksby et al. [ROO14] é o estilo mais adotado. Aqui, um participante atua como piloto enquanto outro atua como copiloto. Os outros participantes compõem a audiência que, na maioria das vezes, participa de forma coordenada, prestando atenção aos pares. Após cada turno de 5-7 minutos, o piloto retorna à audiência, o copiloto assume como piloto e um membro da audiência se candidata a ser o novo copiloto. Desta forma, todos participantes atuam ao menos uma vez como piloto e copiloto [SAT08].

2.2.2 Coding Dojo Randori

Randori é uma palavra de origem japonesa e significa estilo livre [OXF17]. Em algumas artes marciais é permitido que, durante uma sessão de Randori, os participantes usem qualquer tipo de método, desde que seja de forma amistosa e não competitiva. A ideia por trás do *Coding Dojo Randori* (CDR) é bem semelhante, haja vista que várias abordagens de desenvolvimento de software podem ser utilizadas pelos participantes durante em uma sessão de CDR. O objetivo da prática é, sobretudo, a aprendizagem dos participantes [ROO14] [SAT08].

Poucos estudos na literatura exploram evidências empíricas sobre CDR. Sato et al. [SAT08] reportaram que o *Coding Dojo* impacta positivamente no processo de aprendizagem e apresentaram uma série de lições aprendidas em um ambiente acadêmico e da indústria.

Da Luz et al. [LUZ13] executaram um estudo envolvendo a adoção do CD com a aprendizagem de TDD e observaram, conforme os resultados, que a prática suporta a aprendizagem, em específico o tipo Randori, pois, ao fazer uso da programação em par, ajuda no diferente nivelamento entre os pares.

Heinonen et al. [HE113] conduziram algumas sessões de CDR dentro de tópicos ágeis de um curso para alunos da graduação em Engenharia de Software. A partir de uma *survey*, os participantes reportaram bons resultados quanto à aprendizagem, em particular de TDD. Um ponto negativo apresentado pelo estudo foi o tempo curto da sessão de 5 minutos de duração.

Rooksby et al. [ROO14] atentam para a falta de um método avaliativo da efetividade da aprendizagem em CDR, ao mesmo tempo em que defendem que a literatura sobre educação em computação é vasta e existe uma oportunidade para se explorar mais os efeitos desta prática. Neste contexto, eles citam o trabalho de Schön [SCH84] que caracteriza práticas reflexivas como uma possível teoria para nortear o entendimento da prática.

A aprendizagem é uma das principais características de CDR [ROO14] [SAT08]. E a maior parte dos estudos identificados investiga como a prática pode afetar a aprendizagem, seja com alunos [HEI13] ou profissionais [SAT08]. Na literatura, o Coding Dojo está relacionado com a aprendizagem de práticas ágeis (principalmente do método XP), como TDD e refatoração [LUZ13].

2.3 Teorias de Aprendizagem e Modelos de Avaliação

O objetivo desta tese é analisar a aprendizagem colaborativa de CDR e, a partir disso, a literatura foi analisada em busca de teorias de aprendizagem que suportassem o propósito que CD se caracteriza. As subseções a seguir apresentam o trabalho de Rooksby et al [ROO14] que se relaciona a esta tese e os modelos de avaliação do aprendizagem que são utilizados nesta pesquisa.

2.3.1 Rooksby et al. [ROO14]

Os autores investigaram a prática de CDR sob uma perspectiva teórica ao analisar como a aprendizagem pode ser compreendida em CDR. A teoria de aprendizagem discutida pelos autores é a prática reflexiva [SCH84], onde os autores discutem que mesmo as atitudes alternativas (talvez consideradas estratégias ruins durante o desenvolvimento de software) podem ser consideradas parte do processo de aprendizagem.

A principal contribuição deste estudo é apresentar uma teoria de aprendizagem que pode ser usada para avaliar a aprendizagem em CDR. Este estudo por ser preliminar, não contempla uma avaliação de efetividade de CDR, quanto ao seu valor [ROO14].

2.3.2 Savi [SAV11]

Savi [SAV11] propôs um modelo de avaliação de jogos educacionais para desenvolvimento de software. Tal modelo é baseado no modelo de avaliação de treinamento de Kirkpatrick [KIR94], nas estratégias motivacionais do modelo ARCS

(Atenção, Relevância, Confiança e Satisfação), na experiência do usuário e na avaliação de competências educacionais de acordo com a taxonomia de Bloom [BLO56].

Neste estudo, o autor aplicou o modelo em três jogos educacionais, obtendo resultados satisfatórios quanto à aprendizagem dos participantes. Por avaliar este quesito, no âmbito desta tese, adotou-se parcialmente o modelo de Savi [SAV11] para avaliar a aprendizagem, a motivação e a experiência do usuário em CDR. As adaptações do uso deste modelo são descritas nos próximos capítulos (especificamente no Estudo de Viabilidade 1 e 2). Abaixo, são apresentados dois modelos utilizados por Savi [SAV11] que avaliaram a aprendizagem.

Modelo de Kirkpatrick

Donald Kirkpatrick [KIR94] descreveu um modelo de quatro níveis onde é possível avaliar programas ou práticas de treinamentos. Os quatro níveis de avaliação são: reação, aprendizagem, comportamento e resultados. A Tabela 2.1 apresenta uma visão geral do modelo, acompanhada de exemplo de como mensurar cada nível.

Tabela 2.1 – Visão geral do Modelo de Kirkpatrick [KIR94] extraído de [SAV11]

Nível	Avaliação	Descrição e Características	Exemplo de ferramentas e métodos
1	Reação	Avaliar como os participantes se sentiram após o treinamento ou após a experiência de aprendizagem	<i>Happy-sheets</i> ; formulários de feedback; reações verbais; questionários pós-treinamento.
2	Aprendizagem	Avalia o aumento de conhecimento ou capacidade.	Avaliações/testes antes e depois do treinamento; entrevistas ou observações.
3	Comportamento	Avalia os efeitos da nova aprendizagem no ambiente de trabalho	Observações e entrevistas ao longo do tempo para avaliar mudanças, relevância das mudanças, e sustentabilidade das mudanças.
4	Resultados	Avalia os efeitos do treinamento do aluno no negócio da empresa	Questionários pós-treinamento; observação como parte

			de um treinamento sequencial e de <i>coaching</i> durante um período de tempo; medições de retrabalho, erros, etc., entrevistas com os participantes, seus gerentes e grupos de clientes.
--	--	--	---

Todos os níveis do modelo de Kirkpatrick representam uma evolução na escala de avaliação de um treinamento [SAV11]. Esta tese tem como base focar nos níveis 1 e 2, com foco na reação dos participantes ao utilizarem CDR e avaliação de aprendizagem. Os outros níveis apresentam um custo mais elevado e podem ser de difícil obtenção.

Na primeira etapa do modelo é importante avaliar a reação de diferentes perfis, como alunos e profissionais. Isto é realizado por meio da avaliação da experiência de aprendizagem e da percepção dos participantes. Para mensurar este aspecto, utiliza-se como ferramenta de avaliação formulários de feedback, pesquisas após o treinamento ou questionários [SAV11].

De acordo com Kirkpatrick [KIR94], avaliar a reação é como medir a satisfação de um cliente. Se um treinamento deve ser efetivo, é importante que os participantes tenham uma reação favorável a ele, caso contrário não se sentirão motivados para aprender [SAV11]. A segunda etapa se refere à avaliação da aprendizagem em si a partir de um treinamento, isto é realizado com a utilização de observações ou entrevistas sobre o que fora produzido durante o treinamento [KIR94].

Taxonomia de Bloom

A taxonomia de Bloom [BLOM56] foi desenvolvida com o objetivo de apoiar os processos de projeto e avaliação educacional. Ela é categorizada em três grandes domínios: cognitivo, afetivo e psicomotor

De acordo com Britto e Usman [BRI15], os estudos em ES se concentram no domínio cognitivo. Para o domínio cognitivo, as categorias foram estruturadas em seis níveis, apresentados e descritos na Tabela 2.2.

Tabela 2.2 – Estrutura da Taxonomia de Bloom, adaptado de [SAV11].

Nível	Descrição
Conhecimento	Habilidade de lembrar informações e conteúdos previamente abordados como fatos, datas, palavras, teorias, métodos, classificações, lugares, regras, critérios, procedimentos etc.
Compreensão	Habilidade de compreender e dar significado ao conteúdo. Essa habilidade pode ser demonstrada por meio da tradução do conteúdo compreendido para uma nova forma (oral, escrita, diagramas etc.) ou contexto. Nessa categoria, encontra-se a capacidade de entender a informação ou fato, de captar seu significado e de utilizá-la em contextos diferentes.
Aplicação	Habilidade de usar informações, métodos e conteúdos aprendidos em novas situações concretas. Isso pode incluir aplicações de regras, métodos, modelos, conceitos, princípios, leis e teorias.
Análise	Habilidade de subdividir o conteúdo em partes menores com a finalidade de entender a estrutura final. Essa habilidade pode incluir a identificação das partes, análise de relacionamento entre as partes e reconhecimento dos princípios organizacionais envolvidos. Identificar partes e suas interrelações. Nesse ponto é necessário não apenas ter compreendido o conteúdo, mas também a estrutura do objeto de estudo.
Síntese	Habilidade de agregar e juntar partes com a finalidade de criar um novo todo. Essa habilidade envolve a produção de uma comunicação única (tema ou discurso), um plano de operações (propostas de pesquisas) ou um conjunto de relações abstratas (esquema para classificar informações). Combinar partes não organizadas para formar um “todo”.
Avaliação	Habilidade de julgar o valor do material (proposta, pesquisa, projeto) para um propósito específico. O julgamento é baseado em critérios bem definidos que podem ser externos (relevância) ou internos (organização) e podem ser fornecidos ou conjuntamente identificados.

Nesta tese, de semelhante forma ao estudo de Savi [SAV11], adotou-se os três primeiros níveis: lembrar, compreender e aplicar. Eles foram utilizados como forma de medir a aprendizagem por competências técnicas específicas (objetivos de aprendizagem) no nível 1 do modelo Kirkpatrick, na percepção dos participantes em relação à CDR.

2.4 Aprendizagem em *Coding Dojo Randori*

Quando se analisa a aprendizagem por trás de CDR, poucos estudos avaliam a teoria da aprendizagem colaborativa de CDR. Sato et al. [SAT08] relacionam a

aprendizagem em CDR como uma prática deliberada. Segundo Ericsson et al. [ERI93], após um período extenso de tempo (geralmente mais que dez anos) de utilização de determinada prática deliberada, um indivíduo se torna especialista na respectiva área de domínio. CDR, como uma prática deliberada, exerceria um importante papel no processo de aprendizagem, levando um desenvolvedor iniciante a um nível mais proficiente [SAT08].

Rooksby et al. [ROO14] investigaram a aprendizagem de CDR sob a perspectiva de prática reflexiva, seguindo a teoria de Schön [SCH84]. Nesta teoria, a aprendizagem ocorre por meio da prática, do pensar enquanto há a existência da ação, ao considerar os envolvidos na prática como profissionais reflexivos.

A teoria de aprendizagem é importante para entender como é interpretado o processo de aprendizagem na prática de CDR. Não há uma teoria mais ou menos correta, apenas perspectivas diferentes de aprendizagem, como as exploradas por Rooksby et al. [ROO14] e Sato et al. [SAT08] em relação à CDR.

2.4.1 Construtivismo Social

Nesta tese, entende-se que a perspectiva de aprendizagem em CDR também é uma prática que abrange muitos aspectos da sociabilidade. Desta forma, compreende-se aqui que o construtivismo social de Vygotsky [VYG78], isto é, a aprendizagem por meio de uma atividade social, se enquadra também na natureza da aprendizagem colaborativa CDR. De acordo com esta teoria, a interpretação da aprendizagem depende da configuração do social e de onde ela foi criada, ou seja, depende do contexto.

Um dos conceitos chave discutidos por Vygotsky é a Zona de Desenvolvimento Proximal (ZDP). A ZDP é definida como a distância entre o nível de desenvolvimento real, determinado pela resolução de problemas de forma independente, e o nível de desenvolvimento em potencial, determinado pela assistência de um adulto ou de outra pessoa mais capaz.

Tharp e Gallimore classificaram a ZDP em quatro estágios: Assistência Externa, Auto-assistência, Automatização e Desautomatização [THA88][GAR13]. A Figura 2.1 apresenta os quatro estágios propostos.

No primeiro estágio (Assistência Externa), a compreensão sobre a situação, tarefa e meta ainda é limitada, necessitando assim da assistência externa de indivíduos mais capacitados. Neste estágio, a aprendizagem ocorre de forma gradual e de maneira irregular [THA88]. No segundo estágio (Auto-Assistência), o indivíduo já é capaz de

realizar uma tarefa sem a ajuda de outros. No entanto, o conhecimento ainda não está desenvolvido e automatizado, ocorrendo uma auto-assistência para que ele possa alcançar a própria aprendizagem.

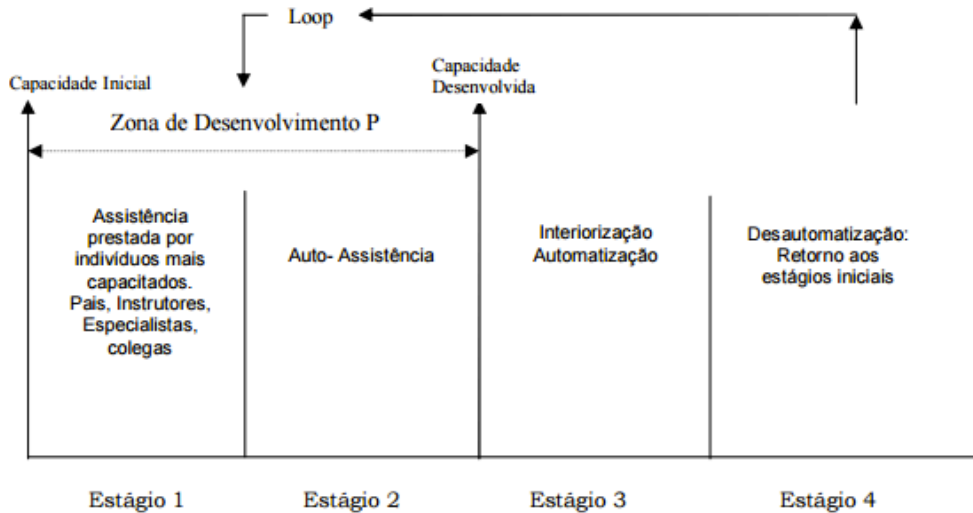


Figura 2.1 – Estágios da Zona de Desenvolvimento Proximal [GAR13][THA88].

O terceiro estágio (Automatização) consiste na automatização do conhecimento, desta forma a execução da atividade acontece de forma tranquila e integrada. O desempenho já está desenvolvido. O indivíduo, portanto, sai da ZDP, e a aprendizagem é alcançada. No último estágio (Desautomatização), há a desautomatização do conhecimento, levando a recorrência de estágios da ZDP. Durante a vida, um indivíduo passa por estas repetições desde a assistência por pessoas até a auto-assistência [THA88].

3 METODOLOGIA DE PESQUISA

Neste capítulo é apresentada a metodologia da pesquisa adotada neste trabalho. Na Seção 3.1, apresenta-se as questões de pesquisa, por meio da descrição das etapas e fases. Na Seção 3.2 são apresentados os aspectos metodológicos, especificamente ao apresentar os principais métodos de pesquisa utilizados.

Tendo por base o objetivo deste estudo e pelo fato de CDR ser uma prática ainda pouco explorada na literatura, a pesquisa desenvolvida nesta tese é do tipo exploratória, com natureza aplicada. Nesta tese foram utilizados métodos quantitativos e qualitativos de pesquisa, sendo a última abordagem a principal forma de análise dos resultados.

3.1 Questões de Pesquisa

A questão de pesquisa norteadora (QPN) deste estudo (Como se pode avaliar a aprendizagem colaborativa durante a execução da prática de CDR no contexto de desenvolvimento de software?) é desdobrada nas seguintes questões secundárias:

1. QP1: Quais as principais características das práticas de codificação colaborativa no contexto dos cursos de computação e treinamentos?
2. QP2: Quais os principais benefícios e desvantagens de CDR do ponto de vista dos participantes?
3. QP3: Quais estratégias podem ser utilizadas para avaliar a aprendizagem colaborativa em CDR?
4. QP4: Como apoiar a aprendizagem colaborativa em CDR?

3.2 Etapas e Fases de Pesquisa

A metodologia de pesquisa é constituída por um conjunto de estratégias, evidenciado por uma série de etapas e fases. A metodologia deste trabalho é inspirada na metodologia definida em Schull et al. [SCH01] e estendido por Mafra et al. [MAF06]. Esta metodologia é explicada na subseção abaixo. A Subseção 3.2.1 apresenta detalhes sobre o desenho de pesquisa desta tese.

3.2.1 Metodologia de Schull et al. [SCH01] com extensão de Mafra et al. [MAF06]

Schull et al. [SCH01] propuseram uma metodologia para introdução de tecnologias (processos, práticas) de software na indústria, desde sua concepção até a transferência para a indústria [MAF06]. A metodologia inicialmente possuía quatro etapas: estudo de viabilidade, estudo de observação, estudo de caso (ciclo de vida) e estudo de caso na

indústria. Mafra et al. [MAF06] propuseram como primeira etapa o estudo secundário da literatura. Desta forma, as cinco etapas são explicitadas abaixo.

1. Estudo Secundário da Literatura: consiste na caracterização do estado de arte, onde é possível compreender o grau de evidência empírica da área [MAF06]. Esta etapa é importante para constituir a definição da tecnologia a ser introduzida na indústria.
2. Estudo de Viabilidade: o objetivo de um estudo de viabilidade é desenvolver um entendimento inicial sobre o tema tratado. Este tipo de estudo deve responder ao pesquisador se a prática a ser avaliada atende razoavelmente aos objetivos definidos no primeiro momento.
3. Estudo de Observação: é possível fazer uma observação de como a prática é adotada em um ambiente onde a execução desta possa ser acompanhada por pesquisadores.
4. Estudo de Caso (Ciclo de Vida): nesta etapa é possível avaliar a prática de forma menos controlada e não isolada. O objetivo é caracterizar a prática no contexto de um ciclo de vida de desenvolvimento de software, e entender a sua implicação em outras tecnologias e processos.
5. Estudo de Caso (Indústria): esta etapa tem por objetivo caracterizar o uso da prática na indústria, em longo prazo. Quando se chega nesta etapa, entende-se que a prática atingiu um certo nível de maturidade que possibilite seu uso contínuo na indústria.

No contexto deste trabalho, foram utilizadas as quatro primeiras etapas de avaliação da metodologia de Schull et al. [SCH01]. A última etapa não foi alcançada no escopo desta tese.

3.2.2 Desenho de Pesquisa

Para alcançar os objetivos deste estudo, definiu-se um desenho de pesquisa (Figura 3.1) constituído por 3 etapas: exploratória, avaliação e consolidação. Ao todo, cinco fases contemplam esta tese.

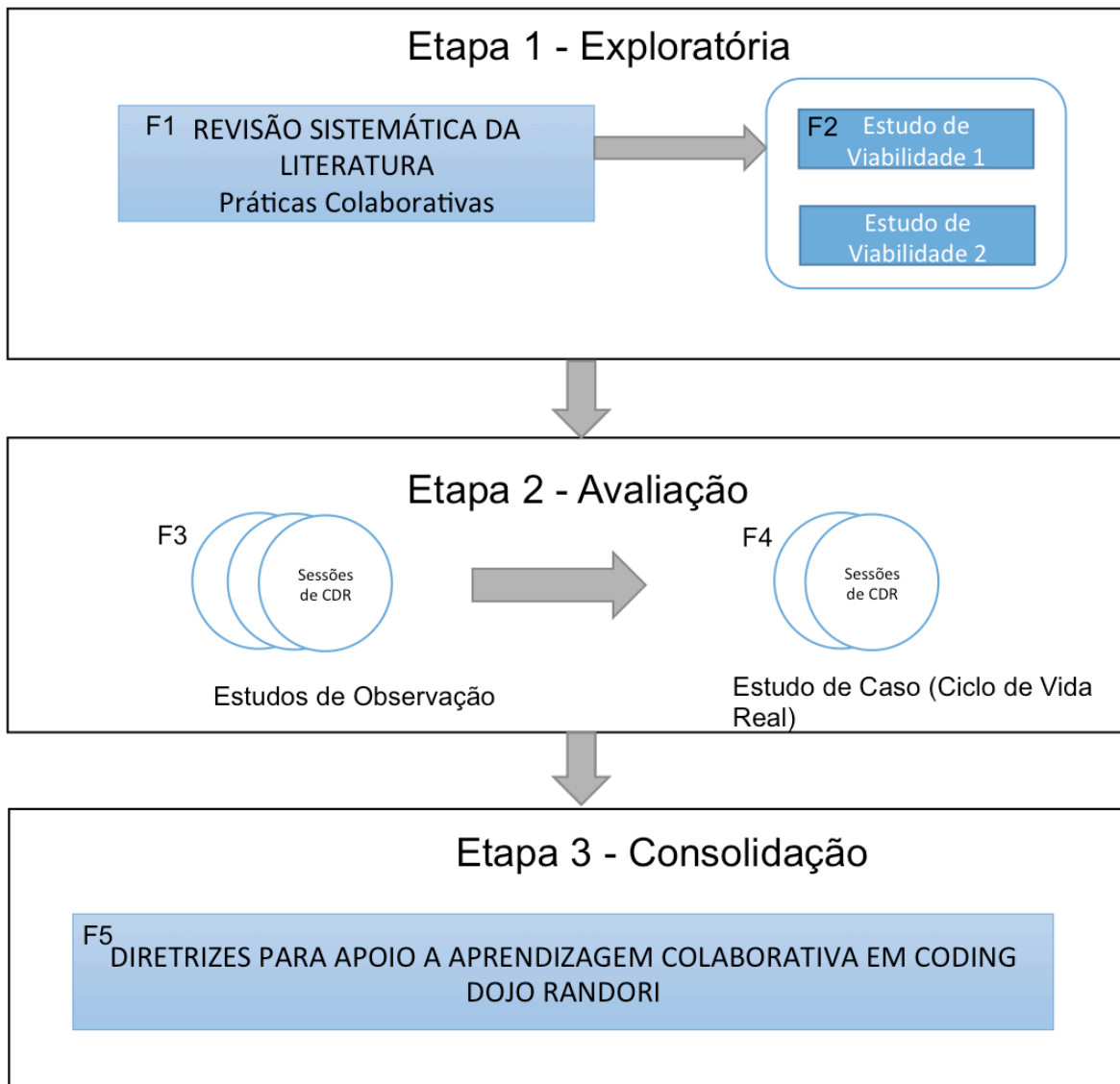


Figura 3.1 – Desenho de Pesquisa

3.2.3 Etapa 1: Exploratória

A primeira etapa constitui uma etapa exploratória na condução da pesquisa, consistindo em um estudo secundário (fase 1) e dois estudos de viabilidade (fase 2).

Fase 1: a primeira fase desta pesquisa foi constituída da compreensão do estado da arte do tema proposto. Desta forma, uma revisão sistemática da literatura sobre práticas de codificação colaborativa foi executada. Esta fase tinha por objetivo caracterizar as principais evidências sobre práticas de codificação colaborativa assim como suas características, a saber: o tipo de avaliação que é utilizado, e os cursos e treinamentos onde essas práticas são adotadas.

Fase 2: foram realizados dois estudos experimentais de viabilidade. Estes estudos tinham por objetivos 1) avaliar os benefícios e desafios das práticas de codificação

colaborativa, 2) avaliar a percepção da aprendizagem colaborativa, e 3) compreender a percepção dos participantes em relação à motivação e experiência de uso destas práticas.

3.2.4 Etapa 2: Avaliação

Ao perceber os efeitos iniciais das práticas de codificação colaborativa, optou-se por uma investigação qualitativa sob a prática de CDR. As fases 3 e 4 da Etapa 2 representam dois ciclos de análise sob esta prática.

Fase 3: foram realizadas três sessões de estudos observacionais de CDR com programadores novatos. Estas sessões foram organizadas pelo autor deste trabalho. Tais sessões tinham como objetivo analisar comportamentos e padrões durante as sessões de CDR.

Fase 4: visando avaliar CDR em um ambiente menos controlado e com pessoas mais experientes, foram conduzidas duas sessões em um estudo de caso dentro de um ciclo de vida real. Sessões de CDR foram empreendidas com desenvolvedores de um projeto da indústria em um ambiente menos controlado ao primeiro ciclo.

3.2.5 Etapa 3: Consolidação

A última etapa foi constituída da consolidação dos resultados da Etapa 1 e 2, e então na identificação de diretrizes para apoiar CDR.

Fase 5: esta fase tinha por objetivo integrar os resultados empíricos das fases anteriores. Com base nos resultados empíricos de CDR e nos instrumentos de avaliação utilizados, um conjunto de diretrizes para apoio a aprendizagem colaborativa em CDR foi desenvolvido.

3.3 Aspectos Metodológicos

Durante as etapas e as respectivas fases desta pesquisa, procurou-se seguir um processo científico, isto é, uma série de recomendações da literatura foram seguidas e o desenvolvimento de protocolos que pudessem garantir a confiabilidade dos resultados. Desta forma, cada fase possui uma característica específica em relação à abordagem metodológica.

3.3.1 Fase 1 – Revisão Sistemática da Literatura

Nesta fase, foi executada uma RSL seguindo as recomendações de Kitchenham et al. [KIT07]. Esta atividade contou com a revisão de um terceiro membro do grupo de pesquisa, além do orientador. O autor da presente pesquisa realizou as principais atividades de execução da RSL e os outros pesquisadores auxiliaram na revisão das etapas.

3.3.2 Fase 2 – Estudos de Viabilidade

A fase 2 é caracterizada por dois estudos de viabilidade. Ambos seguiram as recomendações de Wholin [WHO00] para estudos experimentais. A partir de protocolos de planejamento, estabeleceu-se o objetivo, as hipóteses e a instrumentalização a ser utilizada. A execução foi coordenada com o suporte de membros do grupo de pesquisa da PUCRS e grupos de pesquisas parceiros. Nesta fase também foram utilizados questionários para a coleta de dados de pesquisa. Congruente ao objetivo desta pesquisa, foram utilizados questionários baseado nos instrumentos elaborados por Savi [SAV11].

3.3.3 Fase 3 e 4 – Estudos de Observação e Estudos de Caso

Nas fases 3 e 4, optou-se pela análise mais aprofundada do ponto de vista da abordagem qualitativa. Para tal, adotou-se técnicas do método *Grounded Theory* (GT) [STR90]. As técnicas utilizadas de GT foram: codificação aberta, codificação axial e sensibilidade teórica.

Para aplicar GT nestas fases, foram conduzidas gravações por vídeo das sessões de CDR, além da captura de tela dos participantes. Um segundo canal de áudio por meio de um gravador foi utilizado a fim de resguardar a qualidade do áudio a ser analisado. O software Camtasia Studio¹ foi utilizado para captura de tela e para a integração dos vídeos(o da captura da tela e dos participantes). A análise usando os elementos de GT foi realizada com o software Atlas.ti². A Figura 3.2 apresenta a visão de uso deste software, onde “1” apresenta o conjunto de citações selecionadas no documento analisado, “2” o documento em si (por meio do vídeo da captura de tela e dos participantes), “3” o trecho

¹ www.techsmith.com/camtasia.html

² www.atlasti.com/

específico da citação selecionado com a codificação (aberta e axial), e “4” o trecho transcrito da citação selecionada.

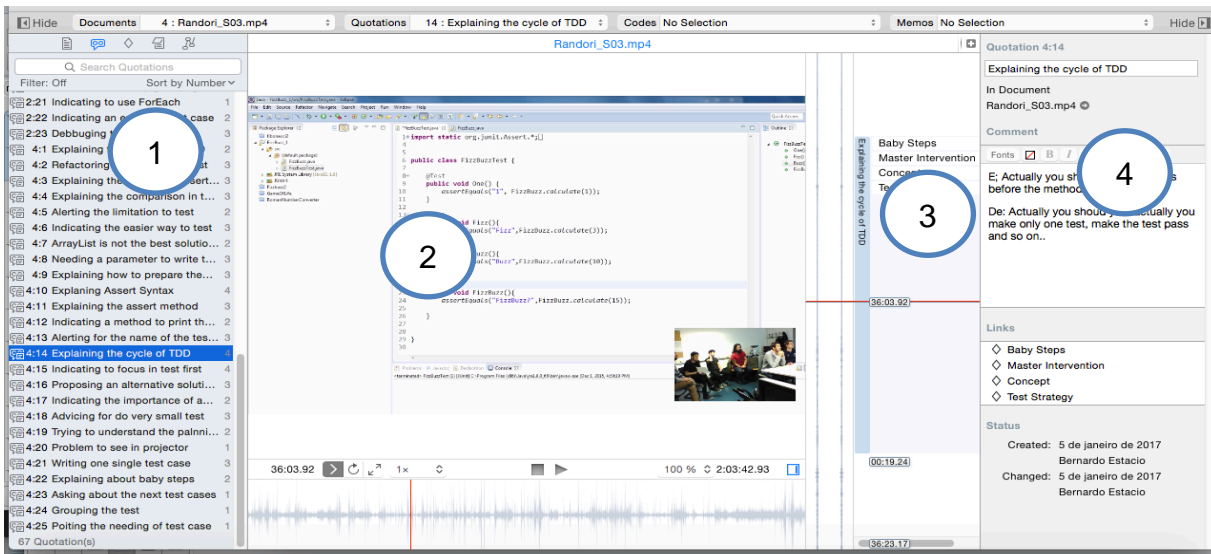


Figura 3.2 – Visão do software Atlas.ti para análise das sessões de CDR.

3.3.4 Consentimento dos Participantes

Quando os estudos envolviam participantes voluntários (fases 2, 3 e 4), garantiu-se o aceite de participação e o consentimento de cada indivíduo. Além disso, os pesquisadores deixaram claro o objetivo do estudo, a confidencialidade do anonimato e a autorização para gravação via vídeo.

3.4 Visitas de Pesquisa e Parcerias

Durante o percurso acadêmico desta tese, foram realizadas visitas de pesquisa e parcerias com grupos de pesquisa do Brasil e do exterior. No início do curso, o aluno foi selecionado para participar do PASQI (*Pan American Software Quality Institute*), uma escola de engenharia de software na Costa Rica. Além disso, a partir da participação em eventos científicos foi possível desenvolver uma colaboração com o grupo de pesquisa de engenharia de software da TU Wien (AMMA), onde se desenvolveu um estudo sobre práticas ágeis no desenvolvimento de software distribuído, em parceria com o então aluno de doutorado Raoul Vallon e seu orientador, Prof. Dr. Thomas Grechenig.

O primeiro estudo de viabilidade da fase 2 foi executado com a cooperação do grupo de pesquisa USES da Universidade Federal do Amazonas (UFAM), liderado pela Professora Dra. Tayana Conte. Neste contexto, aprimorou-se a troca de conhecimento sobre estudos experimentais controlados.

O segundo estudo de viabilidade da fase 2 foi realizado em parceria com o grupo de pesquisa OPUS da Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO), liderado pelo Prof. Dr. Alessandro Garcia. Neste estudo foi possível fazer a replicação dos questionários utilizados do estudo de viabilidade 1. Além disso, este estudo permitiu a análise a partir de um novo *design* de experimento.

As sessões na fase 3 foram realizadas durante o período de doutorado sanduíche do autor. Elas ocorreram na Freie Universität Berlin (FUB) com o suporte do grupo AGSE, liderado pelo Prof. Dr. Lutz Prechelt. Nesta etapa, os resultados preliminares foram discutidos com o grupo de pesquisa do exterior.

4 FASE 1: RSL SOBRE PRÁTICAS DE CODIFICAÇÃO COLABORATIVA

Neste capítulo é apresentada a RSL conduzida neste trabalho. A Seção 4.1 apresenta o protocolo adotado para esta RSL. A seção 4.2 apresenta os resultados, enquanto na Seção 4.3 são apresentadas as considerações finais deste capítulo, e na Seção 4.4 as limitações deste estudo.

Uma revisão inicial da literatura da área realizada no início dessa pesquisa, em 2013, identificou somente um conjunto de estudos iniciais da prática. Desta forma, optou-se pela utilização de um método secundário de pesquisa, a RSL com o objetivo de caracterizar as práticas de codificação colaborativa.

Desta forma, conduziu-se uma RSL em de março até outubro de 2014, ainda na fase exploratória da pesquisa. Com o objetivo de atualizar os estudos primários desta RSL, uma nova rodada utilizando o mesmo protocolo foi conduzida de setembro até novembro de 2016, coletando artigos dos anos não cobertos na primeira execução da RSL.

4.1 Protocolo da RSL

A revisão sistemática da literatura foi executada seguindo as recomendações documentadas por Kitchenham et al. [KIT07] e outras experiências de RSL na literatura com tema relacionado [DYB08, SAL11]. Desta forma, a condução da RSL seguiu etapas definidas estrategicamente. Cada etapa será explicada, em detalhes, nas próximas seções.

4.1.1 Objetivo e Questões de Pesquisa

O objetivo desta RSL é de caracterizar as evidências (aqui referido como resultados do estado da arte) da literatura de práticas de codificação colaborativa. Um protocolo de revisão sistemática (Apêndice A) que possuía as diretrizes para a condução da RSL foi desenvolvido. O objetivo da revisão foi responder as seguintes questões de pesquisa:

- QP1: Quais os principais tópicos investigados e reportados em práticas de codificação colaborativa em curso de computação/engenharia de software?
- QP2: Quais os tipos de avaliação que são utilizados para avaliar a efetividade no ensino em práticas de codificação colaborativa?
- QP3: Quais as configurações dos cursos que adotam práticas de codificação colaborativa?

4.1.2 Pesquisa na Literatura

Para facilitar a condução de pesquisa na literatura e a definição da *string* de busca, optou-se por seguir a recomendação da abordagem PICO [BIO05]: população (*population*), intervenção (*intervention*), controle (*control*) e efeitos (*outcomes*). Dada a característica exploratória desta pesquisa, a variável de controle não foi utilizada. Os termos utilizados na abordagem PICO são apresentados na Tabela 4.1.

Tabela 4.1 – Termos utilizados na abordagem PICO

Population	Intervention	Outcomes
Computer Software Education and Training	Collaborative programming	Method
Software Engineering Education	Pair programming	Course
Computer Software Higher Education	Group Programming	Practice
Computer Science	Coding Dojo	Programming
	Cooperative Programming	Coding

Inicialmente foram realizadas algumas tentativas para avaliar a *string* de busca não considerando termos como “*Computer Science*”, “*programming*” e “*software*”, de forma explícita. Em um segundo momento, observou-se que esses termos trouxeram mais possíveis candidatos a estudos primário para RSL. Portanto, a *string* de busca utilizada nesta RSL foi:

```
( "Computer Science" OR "Training" OR "Computer Higher Education"
  OR "Software Engineering" OR "Software" )
AND
( "Collaborative Programming" OR "Pair Programming" OR "Group
  Programming" OR "Coding Dojo" OR "Cooperative programming" )
AND
( " Course*" OR "Programming " )
```

De forma semelhante ao estudo de Salleh [SAL11a] a base de dados referência da pesquisa selecionada foi a Scopus, devido ao fato dela retornar uma grande quantidade de artigos de outras bases. Foram consultadas as bases IEEEExplorer, Science Direct, Wiley e ACM Digital Library. Cada artigo retornado das outras bases foi comparado com a lista já existente da Scopus com o intuito de evitar duplicações. Adicionalmente também foram feitas buscas manuais nos anais da conferência XP (*International Conference on Agile Software Development*) com objetivo de procurar possíveis artigos não indexados pela Scopus.

A escolha por estas bases de dados *online* se deu pelas bases utilizadas em outras revisões sistemáticas da literatura [DYB08] [SAL11a]. Além disso, a escolha das bases de dados também foi feita a partir do conhecimento dos pesquisadores sobre bases de dados que indexavam artigos de práticas de codificação colaborativa e por meio das bases disponibilizadas pela PUCRS.

4.1.3 Seleção dos Artigos

O critério de inclusão tinha por objetivo apenas estudos que usassem alguma prática de codificação colaborativa no contexto do desenvolvimento de software. A pesquisa apenas cobriu estudos que foram publicados entre 2007 e novembro de 2016. A data de 2007 foi escolhida em função de outras RSL (como a [SAL11]). A data fim de novembro de 2016 está relacionada ao período de término de execução da revisão.

Para exclusão dos artigos, foram adotados os seguintes critérios:

- Artigos escritos em outra língua que não a língua inglesa;
- Artigos publicados em eventos não acadêmicos da área de computação;
- Artigos que não abordam nenhuma prática colaborativa;
- Artigos sem resultados empíricos (*Opinion papers e Philosophical Papers*)
- Tutoriais, *artigos curtos* e palestras.

O processo de seleção de estudos consistiu em três passos. No primeiro passo, foi executado a *string* de busca para identificação dos artigos. No segundo passo, para cada artigo retornado, foi analisado o título, o resumo e as palavras chaves. Os artigos relevantes foram armazenados para uma avaliação posterior. No terceiro passo, cada estudo relevante foi lido na íntegra para determinar se era um estudo primário da revisão sistemática.

4.1.4 Extração dos dados

Para a extração dos dados foi utilizada uma ficha de leitura desenvolvida no MS Excel³. Os itens da ficha foram selecionados conforme alinhamento com as questões de pesquisa. A ficha possuía os seguintes itens: Nome do Artigo, Ano, Autor, Veículo (onde foi publicado), Tipo (*Journal*, Conferência e Workshop), Objetivo, Contexto (Educativo, Indústria), Contribuição, Evidências, Ferramentas/Infraestrutura, Tipo de Curso, Trabalhos Futuros, Metodologia, Com Equipes Distribuídas (Sim, Não), Status (Incluso ou Não Incluso), Justificativa (referente ao status).

4.1.5 Avaliação da qualidade dos estudos

A extração de dados foi auxiliada pelo uso de um *checklist*. O desenvolvimento desse instrumento foi inspirado no estudo de Salleh [SAL11a]. O *checklist* é composto por sete perguntas com a seguinte escala: Sim = 1 ponto, Parcialmente = 0,5 ponto e Não = 0 pontos. O resultado total de cada estudo tem uma faixa de 0 (Muito ruim) até 7 (Muito bom). A Tabela 4.2 apresenta o *checklist* de qualidade de avaliação dos artigos acompanhado das diretrizes usadas.

Tabela 4.2 – Checklist de Avaliação de Qualidade do Artigos

Item	Resposta
1 - O trabalho é bem/adequadamente referenciado (apresenta trabalhos relacionados/semelhantes e baseia-se em modelos e teorias da literatura)?	() Sim () Parcialmente () Não
2 - O objetivo da pesquisa é claro (*)?	() Sim () Parcialmente () Não
3 - O método de pesquisa foi apropriado para alcançar os objetivos da pesquisa?	() Sim () Parcialmente () Não
4 - Existe uma clara descrição do contexto no qual a pesquisa foi realizada?	() Sim () Parcialmente () Não
5 - A coleta de dados foi realizada adequadamente(**)?	() Sim () Parcialmente () Não
6 - A análise de dados foi realizada adequadamente (***)?	() Sim () Parcialmente () Não
7 - Os resultados possuem credibilidade (****)?	() Sim () Parcialmente () Não

³ <http://office.microsoft.com/pt-br/excel/>

Indicadores/Diretrizes de Qualidade
(*) O artigo é baseado em uma pesquisa ou é apenas um conjunto um relatório de lições aprendidas baseado na opinião de um <i>expert</i> ?
(**) Há uma Discussão de: Quem conduziu o de coleta de dados? Procedimentos / documentos utilizados para a coleta / Áudio ou gravação de vídeo de entrevistas / debates / conversas (Se não foram registrados há uma justificativa dada?) Como os métodos de estudo de campo ou estudo de caso aplicado podem ter influenciado nos dados coletados.
(***) Existe uma descrição profunda na análise de dados? Tem dados suficientes para apoiar os resultados? Os dados contraditórios foram levados em consideração? Métodos de controle de qualidade foram usados para verificar os resultados?
(****) Resultados são suportados por dados / estudos empíricos (ou seja, o leitor pode ver como o pesquisador chegou a seus / suas conclusões, a metodologia da análise e interpretação são evidentes) Os resultados e considerações possuem uma lógica coerente? Uso de evidências para apoiar ou refinar resultados?

Em relação à categoria de método de pesquisa, foi utilizada a classificação usada por Wieringa [WIE05]. Esta classificação é definida da seguinte forma:

1. *Evaluation Research*: Técnicas ou soluções são implementadas e avaliadas na prática, e as consequências investigadas.
2. *Validation Research*: Técnicas que foram propostas, mas ainda não foram executadas na prática.
3. *Solution Proposal*: A solução para um problema é proposta e os seus benefícios são discutidos. A diferença entre um artigo *Solution Proposal* para *Validation Research* é o tipo de abstração das soluções sugeridas, a qual possui um nível maior nos artigos *Solution Proposal*.
4. *Philosophical Paper*: Estrutura a área em forma de taxonomia ou *framework conceitual*.
5. *Experience Paper*: Inclui a experiência pessoal do autor na percepção de como ocorreu na prática.
6. *Opinion Paper*: A opinião pessoal do autor sobre um problema sem trabalhos relacionados e métodos de pesquisa.

4.2 Execução

A *string* de busca final levou à identificação de 366 artigos. Após o título e leitura do resumo, foram obtidos 76 artigos e, finalmente, após a leitura integral do texto, foram selecionados 31 trabalhos para extração de dados. A lista com os artigos pode ser

encontrada no Apêndice B desta tese. A Figura 4.1 mostra os três passos seguidos nesta revisão e o número dos artigos identificados em cada passo.

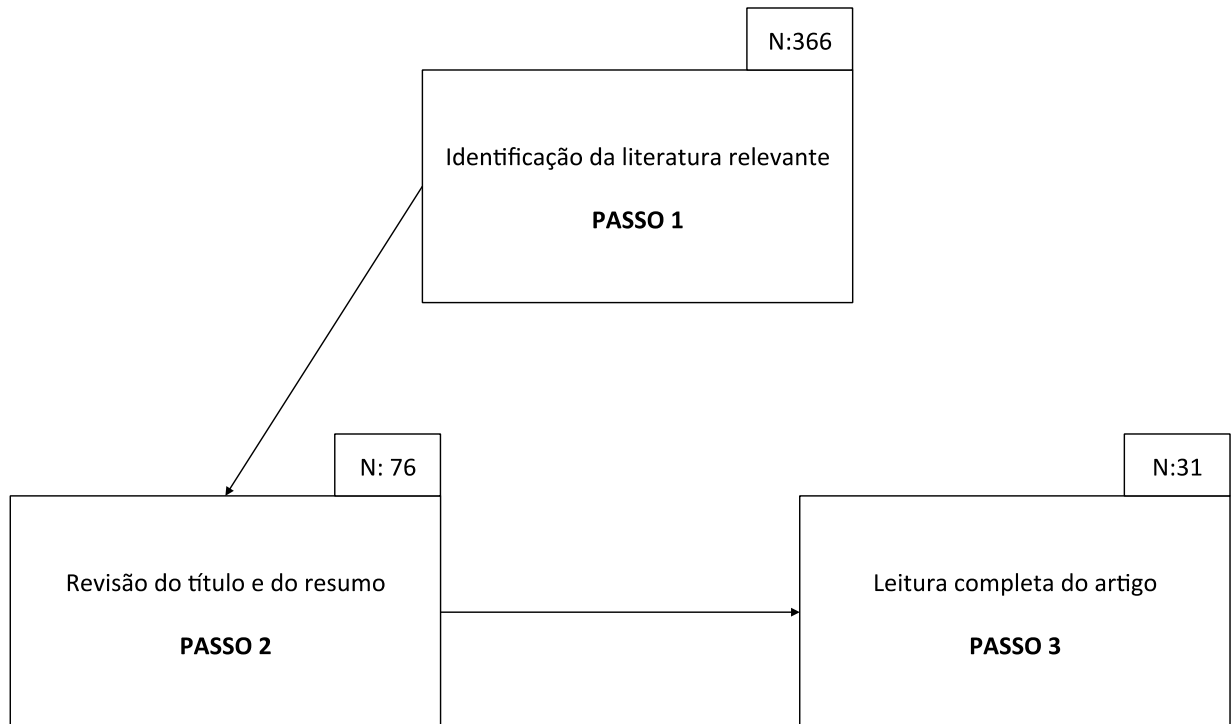


Figura 4.1 – Três passos seguidos na RSL e a quantidade respectiva de artigos selecionados.

A tabela 4.3 apresenta de onde vieram os artigos extraídos, sem duplicação. Scopus foi a base de dados que retornou mais primeiros estudos.

Tabela 4.3 – Quantidade de artigos retornados por base de dados

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	Total
ACM	0	0	1	1	1	0	1	1	1	1	7
IEEE	0	0	0	0	0	1	0	0	0	1	2
Manual (XP Conference)	0	1	0	0	0	0	0	0	0	0	1
Science Direct	0	2	0	0	0	0	0	0	0	0	2
Scopus	3	0	1	0	0	1	3	3	3	1	15
Springer Link	0	0	1	0	0	0	1	2	0	0	4
Willey	0	0	0	0	0	0	0	0	0	0	0
Total	3	3	3	1	1	2	6	6	4	3	31

4.3 Resultados da Análise Quantitativa

Nesta seção é apresentada a visão geral dos 31 artigos primários selecionados sob a perspectiva da análise quantitativa. Estes resultados foram divididos em: qualidade dos artigos primários, ano de publicação, método de pesquisa, classificação por Wieringa e os principais veículos de publicação.

4.3.1 Qualidade dos estudos primários

A Figura 4.2 apresenta o resultado da avaliação da qualidade dos artigos. A maioria dos estudos primários apresenta a classificação ‘Muito Bom’ (Vinte e dois) em termos de qualidade.

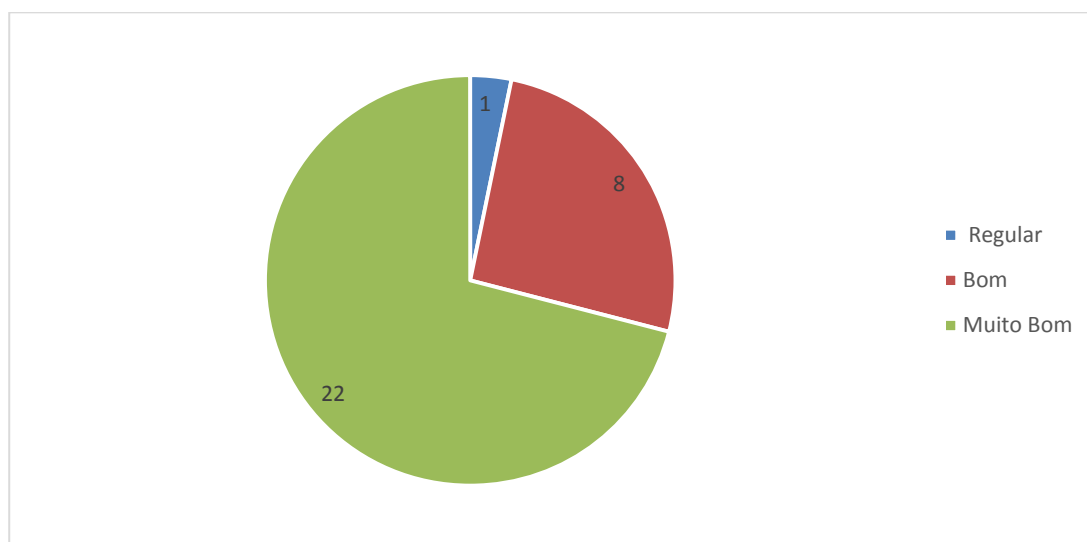


Figura 4.2 – Qualidade dos estudos primários

Oito artigos foram classificados como “Bom”, e apenas um artigo foi classificado como regular. Desta forma, nenhum artigo foi excluído com base na avaliação de qualidade.

4.3.2 Método de Pesquisa

Os artigos também foram avaliados quanto ao método de pesquisa (Figura 4.3). A maior parte dos estudos primários desta RSL são do Estudo de Caso (19 artigos), enquanto 10 artigos são experimentos controlados. Um artigo foi classificado como relato de experiência.

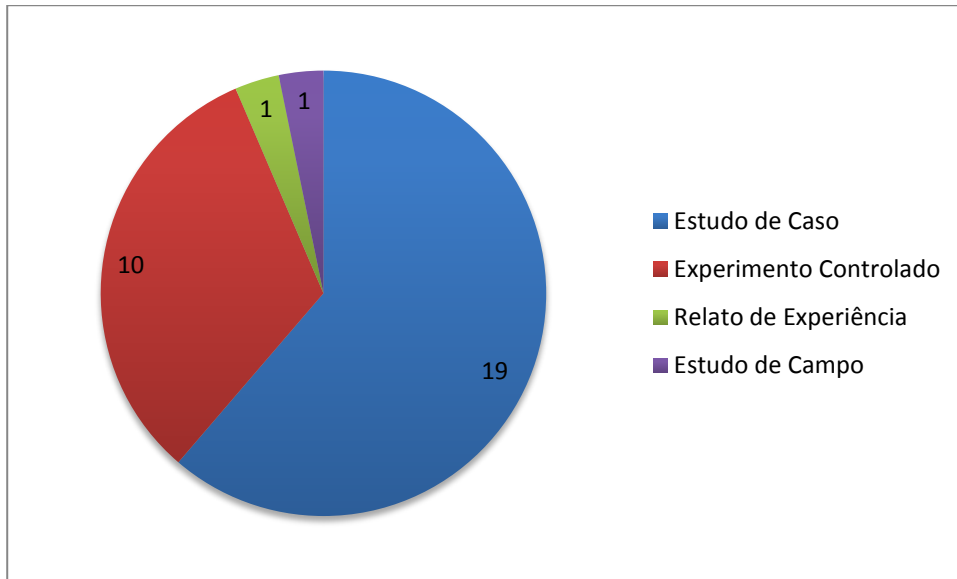


Figura 4.3 – Métodos de pesquisas utilizados

Segundo a classificação de Wieringa (Figura 4.4), a maior parte dos artigos é do tipo *Evaluation Research* com quatorze artigos, enquanto treze artigos foram classificados como *Validation Research*. Na categoria *Experience Paper*, quatro artigos foram identificados.

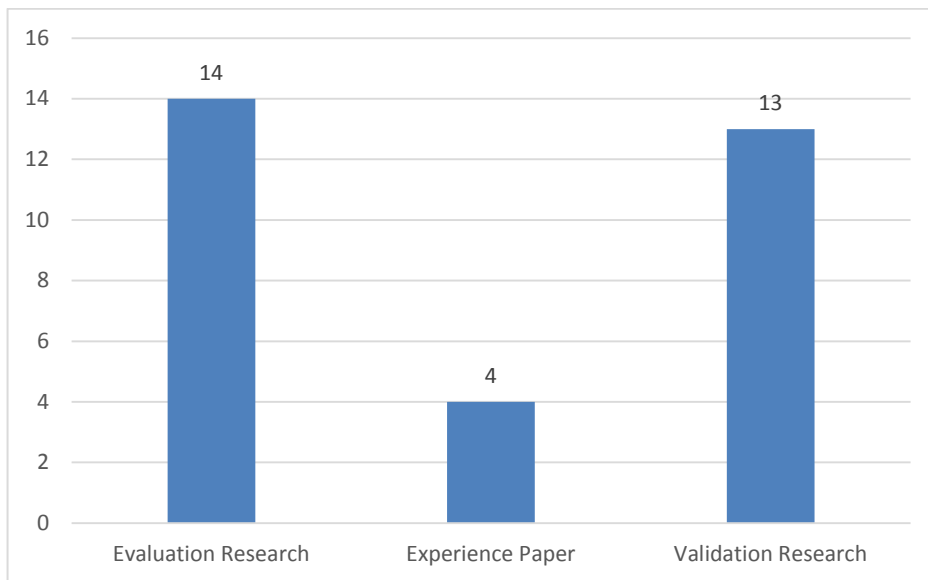


Figura 4.4 – Métodos de pesquisa de acordo com a classificação de Wieringa

4.3.3 Análise por Ano

Quanto ao ano de publicação dos estudos primários, percebe-se que o estado da arte de práticas de codificação colaborativa ainda continua em evolução. Os anos de 2014

e 2015 apresentaram maior quantidade de artigos publicados (cinco em cada). A Figura 4.5 apresenta a classificação por ano.

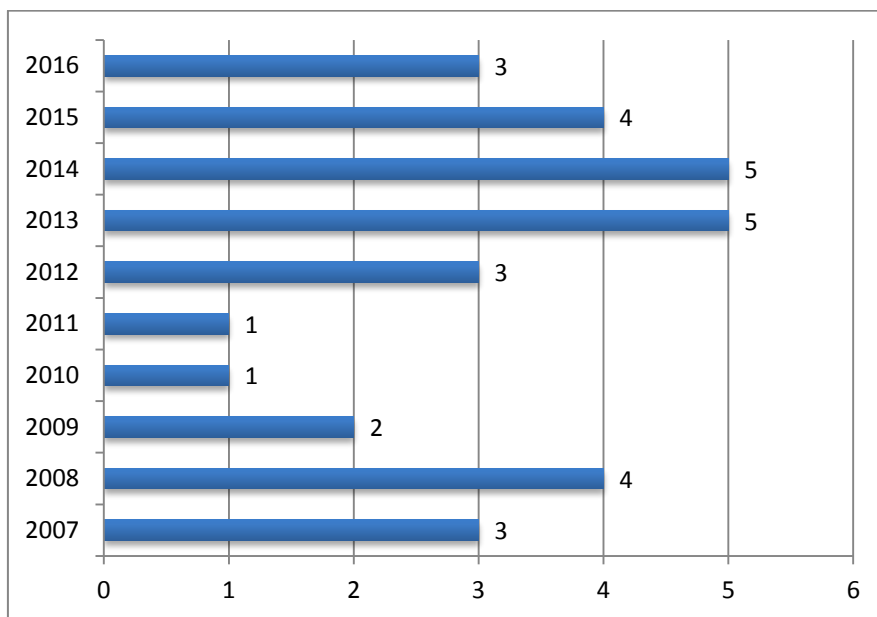


Figura 4.5 – Ano de publicação dos artigos identificados

Os anos que apresentaram menor quantidade de artigos foram 2010 e 2011, ambos com um artigo. Para o ano de 2016, considerou-se até novembro daquele ano, correspondente a finalização da data de execução desta RSL.

4.3.4 Tipo de Publicação

Quanto ao tipo de publicação, a maior parte dos estudos foi publicada em conferências (21 artigos). Enquanto 10 dos artigos selecionados foram publicados em *journal*. Nenhum artigo de workshop foi selecionado na lista final dos artigos primários desta RSL. A Figura 4.6 apresenta os dados em relação ao tipo de publicação dos artigos primários.

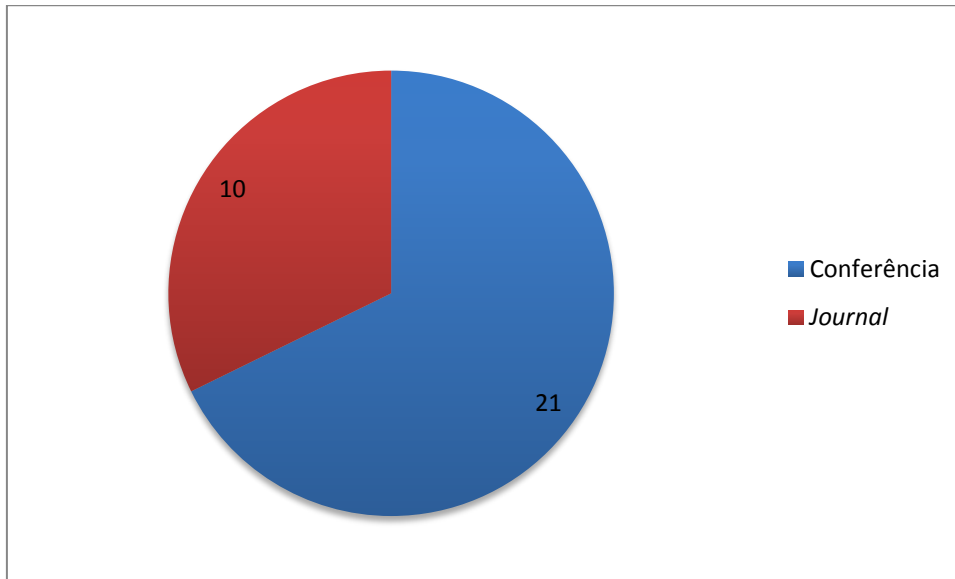


Figura 4.6 – Contexto das Publicações

4.3.5 Análise por veículo de publicação

Em relação à análise dos veículos de publicação, a conferência SIGCSE (*Special Interest Group on Computer Science Education*) é a que possui mais artigos primários nesta RSL. A Figura 4.7 (para fins de visualização, considerou-se os cinco primeiros resultados) apresenta os veículos que possuem mais artigos primários publicados nesta RSL.

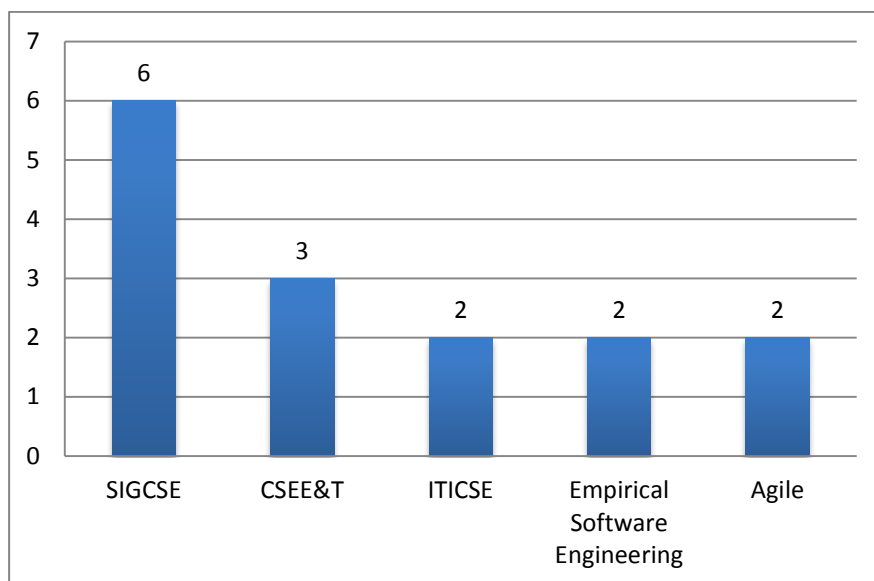


Figura 4.7 – Veículo de publicação dos artigos primários

4.4 Resultados da Análise Qualitativa

Nesta seção são agrupados os principais resultados em resposta às três questões de pesquisa desta RSL e discutidos sobre uma perspectiva de análise qualitativa.

4.4.1 RQ1: Principais evidências de práticas de codificação colaborativa

Para facilitar o entendimento, os resultados foram divididos em PP e Programação em Grupo.

#Programação em Par

As evidências de PP foram agrupadas em cinco categorias. Abaixo, cada categoria é descrita.

Compatibilidade entre os pares: Braught et al. [BRA10] analisam os benefícios de PP quando a formação é por habilidade. Os autores relatam que pares formados por habilidade ajudou a melhorar os estudantes com mais dificuldades. Salleh et al. [SAL12] investigam o impacto da personalidade em diferentes estudos, principalmente em relação ao modelo de personalidade *Big Five Factor* [SAL12]. Eles analisaram que o traço de personalidade *Openness* teve um impacto no desempenho dos alunos. Além disso, a variação no traço de personalidade entre os participantes apoiou a motivação e a diversão entre os alunos.

Por meio de outro modelo de avaliação de personalidade, chamado MBTI, Choi et al. também investigam personalidade em PP. Os autores relatam que um grupo com traços de personalidade diversas entre si foi mais produtivo [CHO08]. De semelhante forma, ao utilizar o mesmo modelo, Sfetsos et al. avaliam a influência dos traços de personalidade sobre a eficiência de PP [SFE08], e replicam a experimento [SFE12]. Os estudos deles relatam que os pares com traços de personalidades e temperamentos mistos tiveram melhor desempenho do que pares com personalidades semelhantes.

Lewis e Shah [LEW15] avaliam a equidade e a desigualdade entre os membros que usam PP, investigando principalmente a discussão entre os alunos. Os autores observaram que pares menos equivalentes entre si eram mais rápidos em completar as tarefas [LEW15].

Comunicação e Transferência de Conhecimento: A comunicação é considerada um fator importante em uma sessão de PP, Zarb et al. [ZAR14] propuseram *guidelines* inspirados na indústria para melhorar esta variável entre os alunos. Em primeiro lugar, eles observaram que essas *guidelines* podem ser eficazes para os alunos e educadores, e em uma segunda avaliação [ZAR15], eles confirmam esta hipótese, identificando a

eficiência na criação de um novo código fonte e uma diferença significativa entre os pares que foram expostos e o grupo de controle.

Na literatura, PP mostrou ser uma ferramenta importante para a efetividade da transferência de conhecimento [KAV13], especialmente quando esse benefício é percebido pelos alunos. Bipp et al. [BIP08] também relatam a transferência de conhecimento como um benefício de PP, e também destaca a qualidade do código fonte produzido pelos pares.

Evidências sobre Treinamento na Indústria: No contexto da indústria, poucos estudos têm investigado como o PP pode influenciar na aprendizagem ou sua aplicação em programas de treinamento. Vanhanem e Lasenius [VAN07] relatam efeitos positivos de PP na aprendizagem, qualidade e espírito de equipe. Schmidt et al [SCH14] relatam sobre a adoção de métodos ágeis na SAP AG. Os autores observaram que PP poderia ajudar não só na qualidade do código, mas também na transferência de conhecimento e motivação da equipe. Coman et al. [COM14] investigou em projetos *Capstones* da indústria diferentes padrões de colaboração em PP.

Evidências sobre Contexto Educacional: Nessa categoria, observou-se evidências que atingem a finalidade educacional como: aprendizagem, desempenho acadêmico e retenção. PP apresentou boas taxas de retenção, especialmente para desenvolvedores novatos e mulheres [LI13]. Além disso, Wood et al. [WOO13] relatam bons resultados da adoção do PP nas duas primeiras semanas de um curso introdutório de programação. Rong et al. [RON12] observaram resultados positivos em PP na relação ao desempenho dos alunos e também na qualidade do código produzido. Urness [URN09] relata que PP auxiliou na melhoria do desempenho acadêmico entre semestres, ao analisar os testes aplicados.

Simon et al. [SIM07] investigaram junto aos alunos sobre as experiências deles em um curso introdutório de programação. Os resultados reforçaram a perspectiva de aprendizagem de PP, e que a prática os ajudou a ter sucesso no curso. Os alunos também mencionaram como fatores de compatibilidade de desafios com os parceiros. McChesney [MCC16] também relatou que PP apoia a confiança e o desempenho dos alunos. Nawahdah e Taji [NAW16] observaram que o PP contribuiu com a diversão e a interatividade dos alunos, e com a qualidade do código produzido pelos pares, que apresentaram menos defeitos.

Estudos exploratórios em PP: Apesar da PP não ser uma prática nova, alguns estudos exploratórios ainda são executados. Seyam e McCrickard investigaram o uso do

PP no ensino de desenvolvimento mobile [SEY15], observaram que o ensino com PP para aplicativos mobile se difere dos cursos de computação regulares, especialmente para problemas envolvendo experiência do usuário ou interface gráfica [SEY16]. Zacharis [ZAC09] investiga a PP com pares distribuídos, em relação à satisfação dos alunos. Madeyski [MAD07] investiga o impacto de PP na detecção de falhas no teste unitário, porém os resultados do experimento não apoiaram o benefício do PP nesse contexto.

A Figura 4.8 sumariza as principais categorias e evidências encontradas para PP. Assim como, o tipo de amostra dos estudos: seja com estudo ou profissionais da indústria.

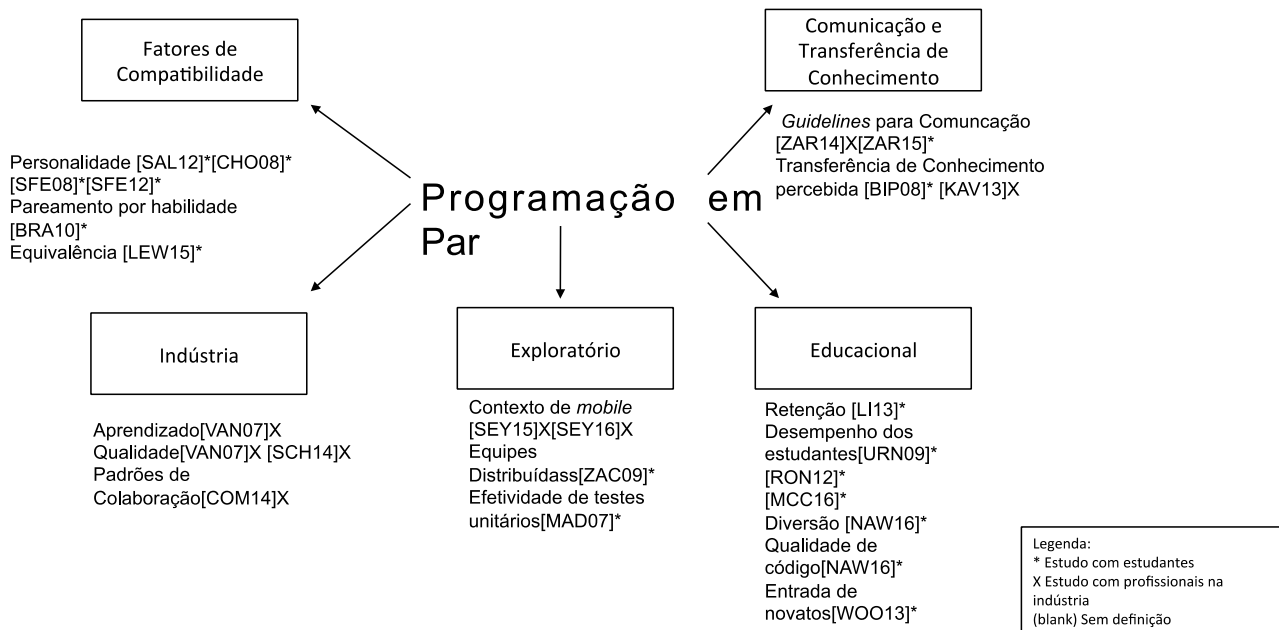


Figura 4.8 – Evidências identificadas nos artigos primários de PP

#Programação em Grupo

Em relação à Programação em Grupo, a maioria dos estudos desta RSL adotou a prática de CDR. Os tópicos foram classificados em quatro categorias, descritas abaixo. A Figura 4.9 apresenta as categorias e as evidências sobre os principais tópicos investigados na programação em grupo. Também são apresentados os tipos de amostra de cada estudo.

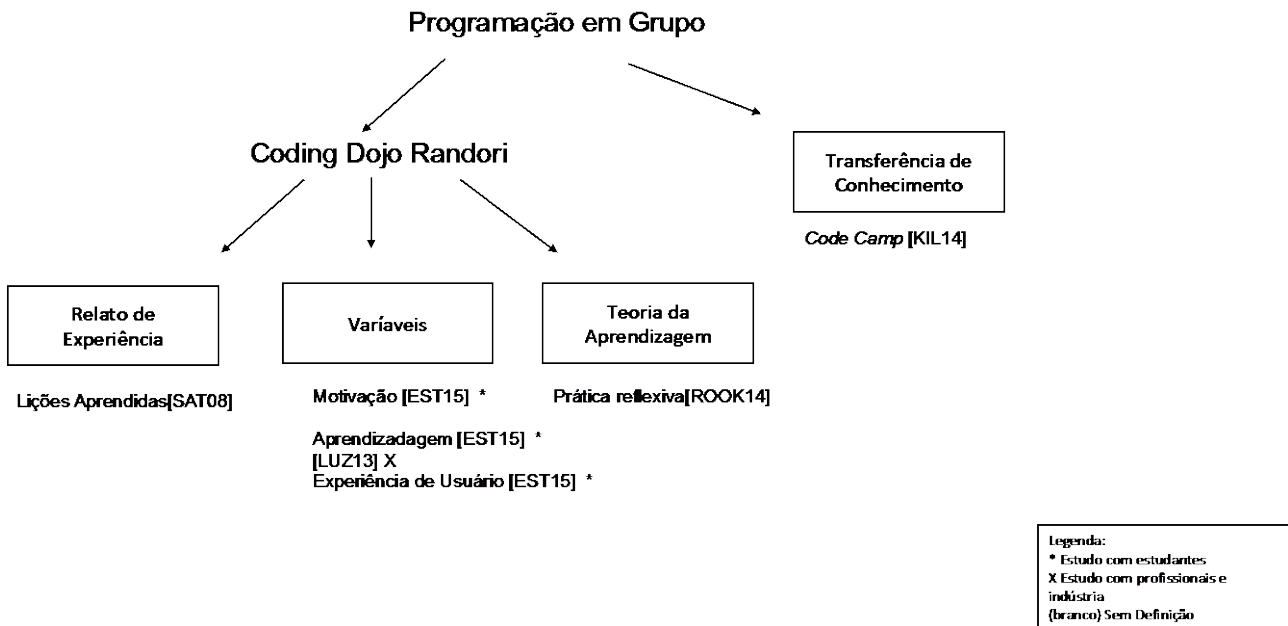


Figura 4.8 – Evidências identificadas nos artigos primários de CDR

Relatório de Experiência - Lições Aprendidas: Sato et al. [SAT08] apresentam um conjunto de lições aprendidas para adotar CDR com base em sua experiência na indústria e também na academia. Entre as lições aprendidas, são citadas a condução de retrospectivas e o tempo de turno de 5-7 minutos na rotação dos pares.

Variáveis (Aprendizagem, Motivação e Experiência do Usuário): O autor desta tese, em uma primeira tentativa de avaliar CDR [EST15], comparou CDR e PP em relação à aprendizagem, motivação e experiência do usuário. Observou-se que ambas as práticas contribuíram para a aquisição de novas habilidades de codificação. No entanto, CDR apresentou alguns desafios, especialmente em relação à motivação, quando os desenvolvedores buscavam obter consenso nas decisões.

TDD é um conceito-chave em relação à aprendizagem de CDR, Heinonen et al. [HE113] reportaram que CDR ajuda a uma melhor compreensão de TDD. Nesta linha, Luz et al. [LUZ13] também constaram que CDR apoia na aprendizagem de TDD.

Teoria da Aprendizagem: De acordo com Rooksby et al. [ROO14], há uma falta de teoria de aprendizagem para avaliar o Dojo, especialmente para o estilo Randori. Depois de analisar duas sessões, os autores afirmam que Randori poderia ser analisado sob a perspectiva da prática reflexiva.

Transferência de conhecimento: Kilamo et al. [KIL14] investigaram a transferência de conhecimento em um *code camp* de duas semanas. Por meio de uma

ferramenta colaborativa, eles observaram que as edições no código fonte ocorriam entre a maioria dos membros mais experientes do grupo. A partir de um questionário e de uma entrevista, os participantes relataram que a atividade foi importante na transferência de conhecimento.

RQ2: Tipos de avaliação de práticas de codificação colaborativa

Para responder a RQ2, identificou-se um conjunto de diferentes tipos de avaliação conduzidas em cursos e treinamentos de práticas de codificação colaborativa. Foram consideradas métricas ambas as formas de medida, seja de perspectiva qualitativa, seja de perspectiva quantitativa.

Os tipos de avaliações foram classificados em quatro categorias: percepções dos sujeitos, aspectos técnicos, desempenho acadêmico e comportamento. A Figura 4.10 apresenta o quadrante com as categorias propostas.

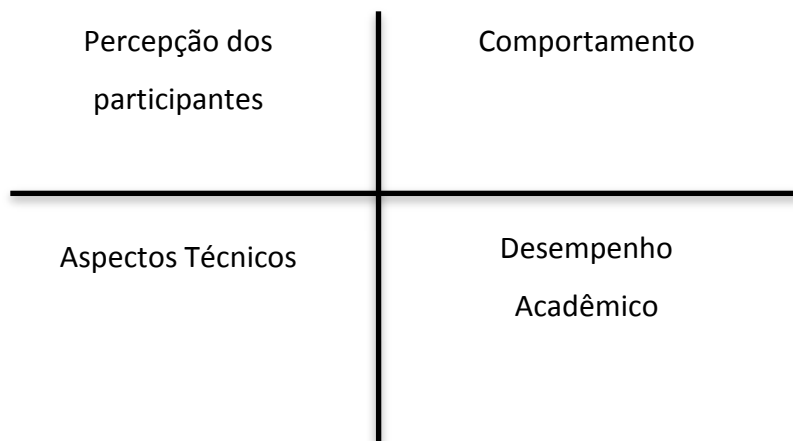


Figura 4.10 – Categorias de tipo de avaliação nos estudos primários

Em relação à categoria Percepções dos Participantes, foram identificados vários estudos que objetivaram analisar os efeitos percebidos das práticas de codificação colaborativa [MCC16] [URN09] [VAN07] [SIM07] [ZAR14] [HEI13] [LUZ13] [EST15]. Alguns estudos de PP tinham por objetivo analisar um contexto específico, como em Zarb [ZAR14], quando os participantes deveriam analisar uma *guideline*. Outros estudos [HEI13] [LU13] [EST15] tinham por objetivo compreender os benefícios e desafios percebidos na adoção de CDR.

Quanto à categoria de Aspectos Técnicos, identificou-se um conjunto de estudos visando analisar os benefícios do PP na qualidade do código. Bipp et al. investigaram os benefícios do PP através de métricas técnicas diferentes [BIP08]. Em um experimento,

eles relatam que algumas métricas, como CBO (*Coupling between Objects*, Acoplamento entre Objetos), não apresenta uma diferença significativa entre a equipe com pares e a equipe individual. No entanto, em relação à métrica WMC (*Weighted Method per Class*, Métodos ponderados por classe), as equipes individuais apresentam um valor maior que os pares. De acordo com os autores, WMC indica dificuldades para testar devido classes muito grandes [BIP08].

Semelhante à Bipp et al. [BIP08], Estácio et al. investigaram a quantidade de anomalias de código inseridas por pares e o grupo [EST15]. Madeyski [MAD07] investigou o impacto do PP em testes unitários, analisando através da cobertura de código e das pontuações de mutação. No entanto, os resultados não foram significativos em relação à eficácia do PP em conjuntos de unidades de teste. Kilamo et al. [KI14] avaliaram os dados de colaboração através do número de edições no código.

Para avaliar a categoria de desempenho acadêmico, a maioria dos estudos utilizou notas obtidas pelos alunos [BRA11] [BRA10] [KAV13] [MCC16] [URN09] [ZAC09]. Braught et al. [BRA11] analisaram notas através de diferentes tipos de desempenho, como a conclusão do curso, por exemplo. Os alunos que receberam uma nota de C (70% da atribuição), eles classificaram como bem-sucedido. Em um estudo de três anos, McChesney relata uma evolução no desempenho acadêmico, especialmente para os alunos "com menor desempenho", mas nenhuma mudança significativa para "alunos com maior desempenho" [MCC16]. Além disso, o desempenho acadêmico também não foi exclusivamente investigado por graus, mas também por tutores ou sujeitos externos. Em Choi et al. [CHO08], dois juízes independentes avaliaram o código dos sujeitos, com o objetivo de analisar a produtividade deste.

Na categoria de Comportamento, que analisa a colaboração entre os participantes, Rooksby et al. avaliaram se a teoria da prática reflexiva se assemelha a dinâmica de CDR. Kilamo et al. [KI14] trabalharam os dados coletados de uma ferramenta colaborativa por meio do número de edições no código. Em [SEY15], os autores executaram um passo a passo. Coman et al. [COM14] analisaram por meio do registro de tipo diferente de colaboração entre os pares em PP.

4.4.2 RQ3: Configuração dos cursos e treinamentos identificados nos estudos primários

Para responder a RQ3, identificou-se que em PP e na PG foram adotados diferentes tipos de cursos e treinamentos. A Tabela 4.4 apresenta os cursos e treinamentos, onde as práticas de codificação colaborativa foram adotadas.

Tabela 4.4 – Caracterização dos cursos que adotam práticas de codificação colaborativa

Cursos	Estudo
Cursos Introdutórios (por exemplo <i>Computer Science</i> “CS1”, “CS2”, <i>Introductory Programming Course</i>)	[BRA11][BRA10][SIM07][WOO13] [URN09][ZAC09][NAW16]
Curso de Imersão	[EST15]
Treinamento em Empresas	[SCH14][VAN07]
<i>Code Camp</i>	[KIL13]
Cursos de Desenvolvimento Mobile	[SEY15] [SEY16]
Projeto <i>Capstone</i>	[COM14]
Curso de uma Linguagem de Programação Específica (Java, C++, C#)	[LI13] [MAD07] [SAL12][MCC16][KAV13]
Cursos de curta duração (Escolas de Verão)	[LEW15][RON12]
Curso de Engenharia de Software	[SFE12] [HEI13]

A maioria dos estudos primários adotou práticas de codificação colaborativa em cursos introdutórios de programação [BRA11] [BRA10] [SIM07] [BRA11][BRA10][SIM07][WOO13][URN09][ZAC09][NAW16] como uma estratégia para introduzir novatos na área. Wood et al. [WOO13] exploraram PP nas primeiras semanas da disciplina de introdução a programação (CS1). Enquanto Zacharis [ZAC09] utilizou PP de forma remota em um curso de introdução a programação. Além disso, observou-se que existem diferentes formatos de cursos que adotam práticas de codificação colaborativa como *Code Camps* [KIL13] e programas de Treinamento da Empresas [SCH14]. Ainda na indústria, Coman et al. [COM14] analisaram atividades do tipo *back up behavior* – quando os desenvolvedores tentam resolver um problema juntos – em dois projetos *Capstones*.

Alguns cursos de programação em módulos mais avançados, envolveram linguagens de programação específicas, como Java e C++. Em Li et al. [LI15], PP foi adotada em curso intermediário de C++ e Unix. O de McChesney [MCC16] apesar de não em um estágio avançado, foca em um curso de Programação Windows. Em Madeyski [MAD07], o contexto do experimento foi um curso de programação em Java, de 7 aulas de 90 minutos e cinquenta sessões de laboratório.

Em Sfetsos et al. [SFE12], o curso de programação tem um enfoque mais técnico, envolvendo o ensino de TDD, PP, Java e Orientação a Objetos. Em Heinonen [HEI13], o foco do curso consistia em práticas do desenvolvimento ágil de software. Práticas de codificação colaborativa também foram utilizadas em cursos de curta duração, como o apresentado em Lewis e Shaw [LEW15], onde foi ofertado um curso de 36h em Logo em Scratch. Rong et al. [RON12] ofertaram um curso de verão após o primeiro ano acadêmico com foco *Personal Software Process* (PSP) utilizando PP.

Seyam e Mccrickard facilitaram 75 minutos de sessões de PP para 53 alunos, explorando aspectos relacionados ao desenvolvimento móvel, tais como o uso da câmera e de imagens processadas, a interface e a gerência de dados, e a conexão de dois dispositivos via *bluetooth* [SEY16].

Além disso, identificou-se o número de participantes envolvidos nestes cursos, especialmente para entender como é o cenário dos estudos de caso e experimentos controlados executados usando práticas de codificação colaborativa. Observou-se um amplo espectro quanta a quantidade de participantes entre os estudos primários.

A Figura 4.11 apresenta as características dos estudos primários em relação ao número da amostra nos estudos. Classificou-se como “Pequena”, estudos com amostras menos de 20 participantes, “Médio” com 21 participantes e mais, e “Grande” estudos com mais de 50 participantes.

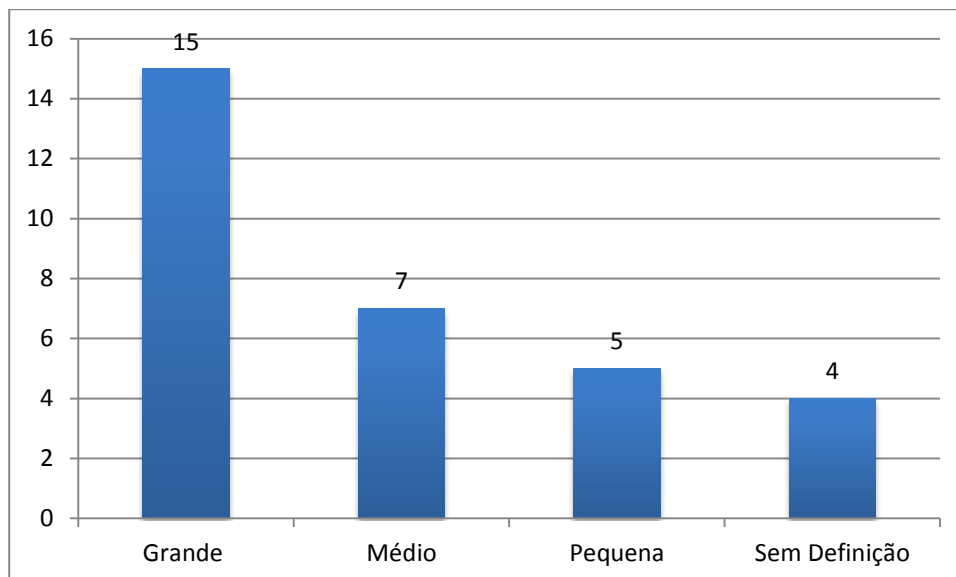


Figura 4.11 – Classificação dos estudos primários quanto ao tamanho da amostra participantes nos cursos e treinamentos mapeados

4.5 Discussão dos resultados

A partir da extração dos resultados, identificou-se uma série oportunidades de pesquisas sobre o estado da arte de práticas de codificação colaborativa no contexto de ensino e treinamento.

#1. Práticas de codificação colaborativa na indústria: a maioria dos estudos que avaliam as práticas de codificação colaborativa está no contexto acadêmico. Apesar disso, identificamos alguns estudos realizados na indústria que investigam o impacto do PP ou CDR no treinamento com profissionais. Faz-se necessário investigar mais aspectos de treinamento e componentes que podem interferir na adoção no contexto da indústria.

#2. Avaliação da aprendizagem: para Beck [BEC04] a aprendizagem é um dos principais benefícios de PP. Apesar da adoção de PP em um ambiente de sala de aula, há uma necessidade de avaliar a perspectiva de aprendizagem colaborativa em relação ao PP e CD. A maioria dos estudos avaliou a aprendizagem por meio ou da percepção dos participantes, ou do desempenho dos alunos, como notas. No entanto, a eficiência da aprendizagem em práticas de codificação colaborativa na literatura sugere que pode ser avaliada sob uma teoria de aprendizagem, como a prática reflexiva [ROO14] ou como prática deliberada [SAT08].

#3. Programação em grupo: ao longo da faixa de dez anos dos estudos pesquisados, poucos estudos empíricos investigaram práticas em grupo. CDR foi adotado na indústria e pela comunidade de software [ROO14], entretanto poucas evidências empíricas foram identificadas na literatura. Ainda existe uma necessidade de investigar sobre outras abordagens, como *Mob Programming* [WIL15]. Além disso, a maioria dos estudos no CD estão investigando o estilo CDR. Outra oportunidade seria investigar outros formatos de Dojo como Kake e Kata.

#4. Comportamento dos participantes: poucos estudos investigaram o comportamento dos participantes durante sessões de práticas de codificação colaborativa. Apesar de termos identificado algumas iniciativas na literatura, como em Zarb et al. [ZAR15], há uma necessidade de entender como podemos avaliar aspectos mais específicos, como a comunicação e a colaboração.

#5. Fatores de compatibilidade: A personalidade tem sido uma das variáveis mais investigadas em práticas de codificação colaborativa, especialmente em PP, ao longo deste intervalo de dez anos analisados nesta RSL. Os diferentes traços de personalidade, e especificamente alguns traços tal como o *Openness* no modelo *Big Five* [SAL12] mostrou-se importante para a eficiência da prática. Por outro lado, há uma falta

de estudos que investigam os fatores de compatibilidade em grupos maiores, como os de uma sessão de CDR.

#6. Cursos não tradicionais: existe uma oportunidade de pesquisa para investigar diferentes formatos de curso para adoção de práticas de codificação colaborativa. A maioria dos estudos adota cursos no formato tradicional, como parte de um semestre regular de uma universidade. Especialmente na indústria, outros formatos podem ser mais explorados.

4.6 Limitações desta Revisão Sistemática

As principais limitações desta RSL consistem no número de base dados selecionadas e a confiabilidade do método de classificação de cada artigo. As seis bases foram selecionadas com base em experiências passada do autor e de outros RSLs (como por exemplo, a [SAL11]).

Optou-se por não executar revisões em livros, nem em outras fontes de artigos que pudessem se concentrar em estudos de uma perspectiva não científica. Deste modo, infere-se que os resultados apresentados fornecem uma boa indicação do "estado da arte" e do "estado da prática" em codificação colaborativa.

O processo de seleção e classificação possui subjetividade, porém, para minimizar esta limitação, esse processo sempre foi conduzido em pares (e, às vezes, com o auxílio de uma terceira pessoa, membro do grupo de pesquisa). Todos os artigos foram revisados pelo menos três vezes pelo mesmo pesquisador.

4.7 Considerações Finais do Capítulo

Este capítulo apresentou uma RSL sobre práticas de codificação colaborativa. Os resultados desta RSL possibilitaram a caracterização destas práticas quanto as principais evidências investigadas no estado da arte. Considerando dez anos de tempo de pesquisa, ainda foi possível observar os principais tipos de avaliação utilizados por estes estudos para avaliar práticas de codificação colaborativa, além de uma caracterização sobre o contexto: os diferentes cursos e treinamentos, onde estas práticas foram utilizadas.

Esta RSL consistiu também em um esforço para analisar as possíveis oportunidades de pesquisas e lacunas existentes na literatura. Desta forma, percebeu-se especificamente que a aprendizagem ainda é pouco avaliada e discutida entre estes estudos, principalmente naqueles que envolvem CDR.

Os estudos de CDR apresentam que esta prática é promissora, porém mais estudos se fazem necessários para saber da sua viabilidade e efeitos. A RSL conduzida está alinhada com a primeira questão de pesquisa desta tese (QP1).

5 FASE 2: ESTUDO DE VIABILIDADE I

Este capítulo apresenta o planejamento e a condução do estudo de viabilidade 1. Após a execução da RSL (Fase 1), percebeu-se a necessidade de se investigar empiricamente CDR. Ao mesmo tempo, PP se mostrou uma prática de codificação colaborativa com um conjunto mais estabelecido de evidências no estado da arte. Desta forma, optou-se por avaliar PP e CDR no contexto de desenvolvimento de *mockups*.

A Seção 5.1 detalha o contexto e objetivo deste estudo de viabilidade, a Seção 5.2 apresenta as principais características quanto os participantes. Na Seção 5.3, apresenta-se os procedimentos e materiais utilizados. A Seção 5.4 detalha os resultados do estudo de viabilidade, enquanto a Seção 5.5 traz a discussão dos resultados. As ameaças à validade deste estudo são listadas na Seção 5.6. Por fim, a Seção 5.7 traz as considerações finais deste capítulo.

5.1 Contexto e Objetivo

Este estudo foi conduzido com alunos da graduação, onde a principal atividade envolveu o desenvolvimento de *mockups*. Para este estudo, a definição de *mockup* consistia em um protótipo semi-funcional de interface gráfica de usuário [PRE05]. O planejamento do estudo foi feito a partir das recomendações de Wohlin et al. [WOH00]. O objetivo detalhado do estudo, de acordo com a estrutura GQM [BAS94], é apresentado abaixo:

Analisar: CDR e PP no ensino de desenvolvimento de *mockups*

Com o propósito de: Avaliar;

Com respeito de: motivação, experiência de usuário e aprendizagem;

Do ponto de vista: dos estudantes;

No contexto do: desenvolvimento de *mockups*;

5.2 Participantes

O estudo foi conduzido no primeiro semestre de 2014 na turma de Análise e Projeto do curso de Ciência da Computação da UFAM. Esta disciplina é do terceiro ano deste curso e possui como pré-requisito as disciplinas de Engenharia de Software e Programação 1. Dezesete alunos participaram do estudo empírico.

Conforme mencionado no Capítulo 3, os alunos assinaram um termo de consentimento para participarem da pesquisa. Além disso, preencheram um formulário de caracterização com questões objetivas em ordem para informar a experiência e

conhecimento deles sobre: (a) programação, (b) Qt⁴ (*framework* de desenvolvimento utilizado para o desenvolvimento de *mockups* durante a disciplina), (c) a experiência deles com programação em pares e (d) a experiência deles com CD.

Para cada participante, os dados foram coletados e classificados em: nenhum (N), baixo (B), médio (M) e alto (A) em termos do nível de experiência informada. Por exemplo, em relação à programação e conhecimento em Qt, o participante foi avaliado: (a) sem experiência, se o participante nunca teve contato ou prática com o *framework* Qt; (b) baixa experiência, se teve contato com programação apenas na disciplina ou em matérias de apoio; (c) média experiência, se teve contato com um projeto acadêmico; ou (d) alta, se teve experiência na indústria. Da mesma forma, a experiência em PP e CD foram avaliadas de acordo com o número de sessões: (a) sem experiência; (b) baixa com 1 sessão; média com mais de 1 sessão e menos de 4 sessões e alta com mais de 4 sessões. A tabela 5.1 apresenta a experiência de cada participante de acordo com as respostas fornecidas.

Tabela 5.1 – Experiência por participante em cada grupo

Grupo Programação em Par									
	Par 1		Par 2		Par 3		Par 4		
Participante ID	P1	P3	P2	P4	P5	P8	P6	P7	
Programação	A	N	B	N	B	N	M	N	
Qt	B	N	N	N	B	N	N	N	
Programação em Par	N	N	M	N	M	M	N	N	
Coding Dojo	N	N	N	N	N	N	N	N	
Grupo - Coding dojo randori									
Participante ID	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9
Programação	N	B	N	B	N	B	N	B	M
Qt	N	N	N	N	N	N	N	N	B
Programação em Par	B	M	N	M	N	B	N	N	N
Coding Dojo	N	N	N	N	N	N	N	N	N

Além dos estudantes, dois pesquisadores atuaram como observadores e instrutores, dando apoio para cada grupo em suas respectivas salas. Outros dois pesquisadores ajudaram na preparação de materiais.

⁴ <https://www.qt.io/>

5.3 Procedimento e Materiais

Todos os participantes tiveram um treinamento sobre o *framework* Qt, especificamente sobre como desenvolver *mockups* e as transições entre eles. Este treinamento tinha por objetivo capacitá-los a desenvolver *mockups* funcionais, isto é, que pudessem ser usados para mostrar a interação gráfica.

No dia do estudo, os estudantes foram igualmente distribuídos nos dois grupos com base nos resultados da caracterização realizados anteriormente. Cada grupo foi encaminhado para uma sala diferente, a fim de evitar o viés de comunicação entre os alunos de cada grupo. Cada instrutor deu uma palestra de 10 minutos em cada sala sobre a respectiva prática (CDR ou PP). Dos 17 participantes, 9 alunos foram distribuídos ao grupo de CDR e 8 ao grupo de Programação em Par. O número reduzido de participantes no grupo de PP foi necessário para evitar a formação de trios.

Os objetos de estudo foram *mockups* de um aplicativo mobile chamado *Dona Know*. O *Dona Know* fornece uma lista de eventos (por exemplo: concertos, shows, eventos sociais, entre outros) para consulta pública. O aplicativo ainda está em fase de desenvolvimento e um conjunto de *mockups* foi disponibilizado para pesquisa.

Todos os participantes receberam imagens de cinco *mockups* do aplicativo *Dona Know* e um modelo do fluxo de interação que eles deveriam desenvolver. Este conjunto de cinco *mockups* foi escolhido pela equipe de desenvolvimento do *Dona Know*, pois apresentavam funcionalidades essenciais do aplicativo.

Cada grupo tinha um intervalo de tempo de 10 minutos para trocar os papéis (piloto e copiloto) entre os pares. Os pares do grupo de Programação em Par foram definidos com base na experiência de cada membro do grupo (ver Tabela 5.1). Dessa forma, um estudante com mais experiência juntamente com um aluno menos experiência formaram um par, de modo que se pode avaliar o processo de aprendizagem. No grupo de CDR, a sequência dos pares foi feita por conveniência (à escolha dos alunos). No entanto, a audiência também poderia participar de uma forma coordenada, caso os pares que tinham o controle no momento da sessão concordassem com a intervenção.

O estudo durou cerca de duas horas para cada grupo simultaneamente em salas diferentes. O grupo de PP realizou 10 sessões, enquanto o grupo de CDR realizou 11 sessões. No final do estudo, cada aluno respondeu a um questionário e enviou o código de execução dos *mockups* aos pesquisadores. Todos os alunos colaboraram neste processo com o estudo, e nenhum dado foi descartado.

O questionário pós-estudo e a avaliação foram adaptados de Savi [SAV11] (Apêndice C). O estudo de Savi et al. [SAV11] foi executado no contexto de jogos. Itens específicos foram selecionados e adaptados do referido estudo para avaliar as subcomponentes, tais como: aprendizagem, experiência do usuário e motivação, utilizando o modelo de treinamento de Kirkpatrick [KIR96] e a taxonomia de Bloom [BLO56] como referência.

O questionário pós-estudo é composto por 12 itens fixos divididos em 3 três subcomponentes (A motivação, experiência do usuário e aprendizagem) e 8 dimensões em uma escala de *Likert* com alternativas de resposta variando de discordo (-2) a concordo muito (2).

Como o estudo tem o foco em práticas ágeis e não em jogos ágeis, adaptamos os argumentos e removemos algumas dimensões do questionário originário de [WAN13] que não eram o foco deste estudo. As dimensões e respectivos argumentos adaptados e mantidos estão listadas na Tabela 5.2.

Tabela 5.2 Argumentos adaptados do questionário pós-estudo de Savi et al. [SAV11]

Motivação		
Atenção	Houve algo interessante na prática que capturou minha atenção	Adaptado
Relevância	O funcionamento desta prática está adequado ao meu jeito de aprender	Adaptado
Confiança	Ao utilizar a prática senti confiança de que estava aprendendo	Adaptado
	Foi fácil entender a prática e começar a usa-la	Adaptado
Experiência Usuário		
Competência	Tive sentimentos positivos de eficiência no desenrolar da prática	Adaptado
Diversão	Eu me diverti com a prática	Adaptado
	Eu recomendaria esta prática aos meus colegas	Adaptado
	Gostaria de usar esta prática novamente	Adaptado
Desafio	A prática é adequadamente desafiadora para mim, as tarefas não são muito fáceis nem muito difíceis	Adaptado
Interação Social	Eu me diverti com o grupo (par).	Mantido
	A prática promove momentos de cooperação entre os participantes.	Adaptado
Aprendizagem		
Aprendizagem longo prazo	A experiência com a prática vai contribuir para meu desempenho profissional	Adaptado

Em relação à aprendizagem, questões foram adicionadas para identificar o nível de conhecimento adquirido antes e após a prática em relação aos conceitos ensinados de: *Mockups*, Interação de *Mockups* e Qt, baseado na taxonomia de Bloom [BLO56]. Para esta parte do questionário, os alunos deveriam atribuir uma nota em uma escala de 1 a 5 para cada questão, baseada em suas percepções acerca de evolução da aprendizagem antes e depois da prática.

O questionário pós-estudo também foi customizado com duas questões abertas, onde os estudantes poderiam explicar benefícios e desvantagens da PP e CD. Após a coleta dos dados dos questionários, executamos uma análise temática em relação a cada categoria com a informação coletada.

5.4 Resultados

Foram coletados dados das questões *likert* (representando a escala com a seguinte notação: -2 Discordo Fortemente; -1 Discordo; 0 Neutro; 1 Concordo; 2 Concordo Fortemente) no questionário pós-estudo e dados qualitativos das questões abertas, além de comentários adicionais. Estes dados foram ao encontro do modelo Kirkpatrick nível 1 [KIR94] sob a perspectiva dos participantes.

A Seção 5.4.1 detalha os resultados de Motivação de ambas as práticas, enquanto a Seção 5.4.2 apresenta os resultados sobre Experiência de Usuário. A Seção 5.4.3 apresenta os resultados referentes a aprendizagem. Por fim, as Seções 5.4.4 e 5.4.5 apresentam respectivamente os benefícios e desvantagens percebidas em CDR.

5.4.1 Motivação

Em relação à motivação, os participantes reportaram a percepção deles para PP (Figura 5.1) e CDR (Figura 5.2) durante o desenvolvimento de *mockups*. Esta percepção foi concebida em três dimensões: Atenção, Relevância e Confiança [SAV11].

Quanto à dimensão Atenção, ambas as práticas apresentaram resultados positivos. O principal aspecto que chamou a atenção dos alunos nos dois contextos analisados foi a interação social.

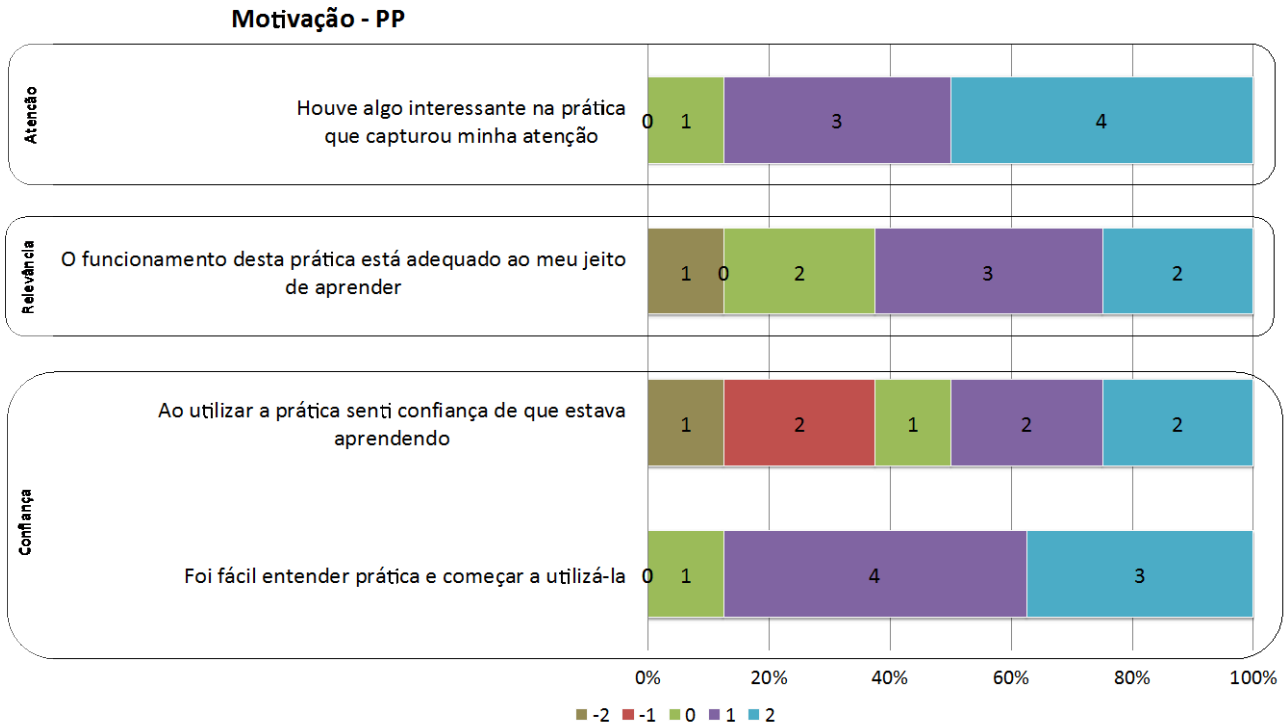


Figura 5.1 – Dimensões de motivação no grupo de PP

Quanto à dimensão Relevância (referente a pertinência em uma prática ser adotada), PP apresentou resultados positivos com a maioria dos estudantes em relação à aceitação da prática, mas em CDR não foi amplamente aceita, cinco estudantes relataram que a prática não combinava com a sua forma de aprender. Em CDR, o participante 1 relatou: "*Coding Dojo [Randori] não é adequado para o meu modo de aprendizagem devido a muitas ideias diferentes do grupo*".

No que diz respeito à dimensão Confiança, os itens relacionados à compreensão e facilidade de uso da prática, tanto PP e CDR apresentaram resultados positivos. PP apresentou resultados mais positivos sobre a impressão do estudante em relação a esta dimensão. Por outro lado, no grupo de CDR a maioria dos estudantes relataram uma diminuição ou efeito nulo na dimensão de Confiança em relação à aprendizagem.

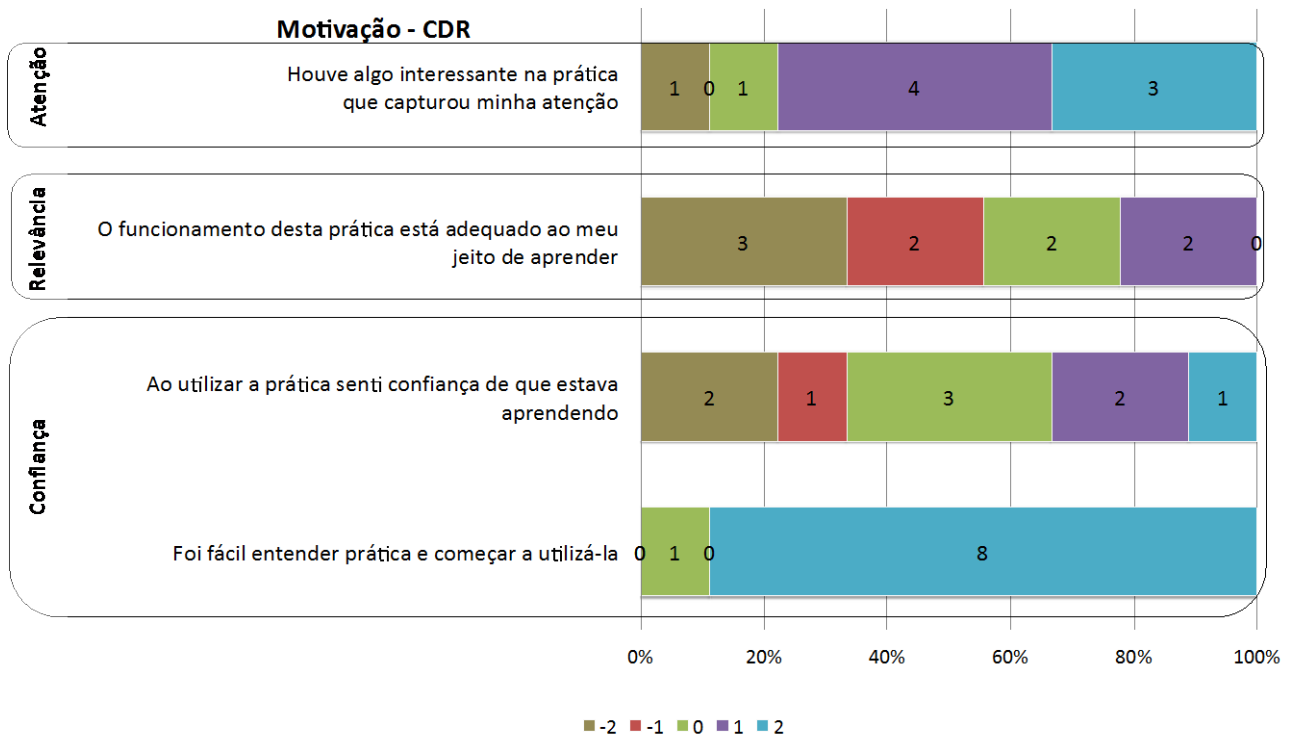


Figura 5.2 – Dimensões de motivação no grupo de CDR

Tanto CDR quanto PP apresentaram resultados positivos para motivação. Na comparação direta, os resultados de motivação para CDR não foram tão positivos quanto os de PP, principalmente em relação a dimensão Relevância. A confiança no aprendizado de ambas as práticas também teve respostas bem diversas (algumas mais positivas e outras mais negativas), entretanto todos os participantes ratificaram que ambas as práticas são de fácil entendimento e adoção.

5.4.2 Experiência do Usuário

A experiência do usuário, a percepção ou reação do aluno em relação à adoção da prática. A Figura 5.3 apresenta os resultados para PP, enquanto a Figura 5.4 para CDR.

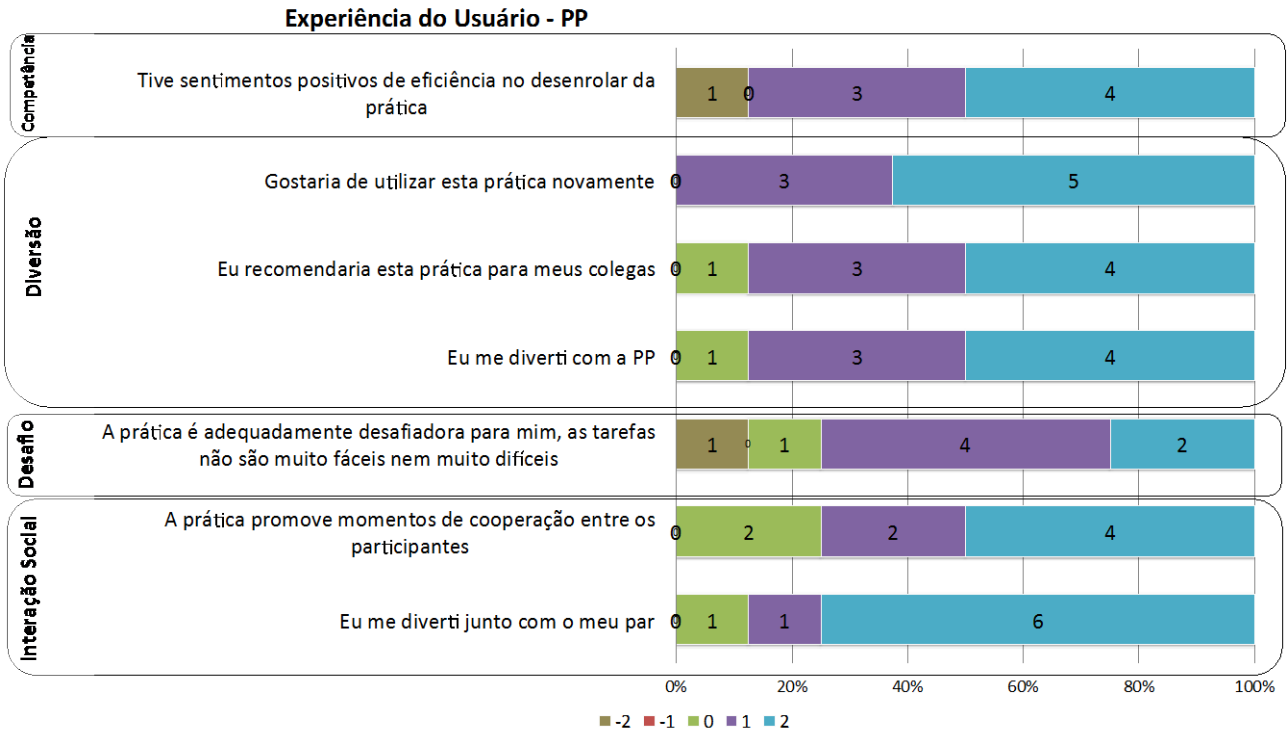


Figura 5.3 – Dimensões de experiência de usuário no grupo de PP

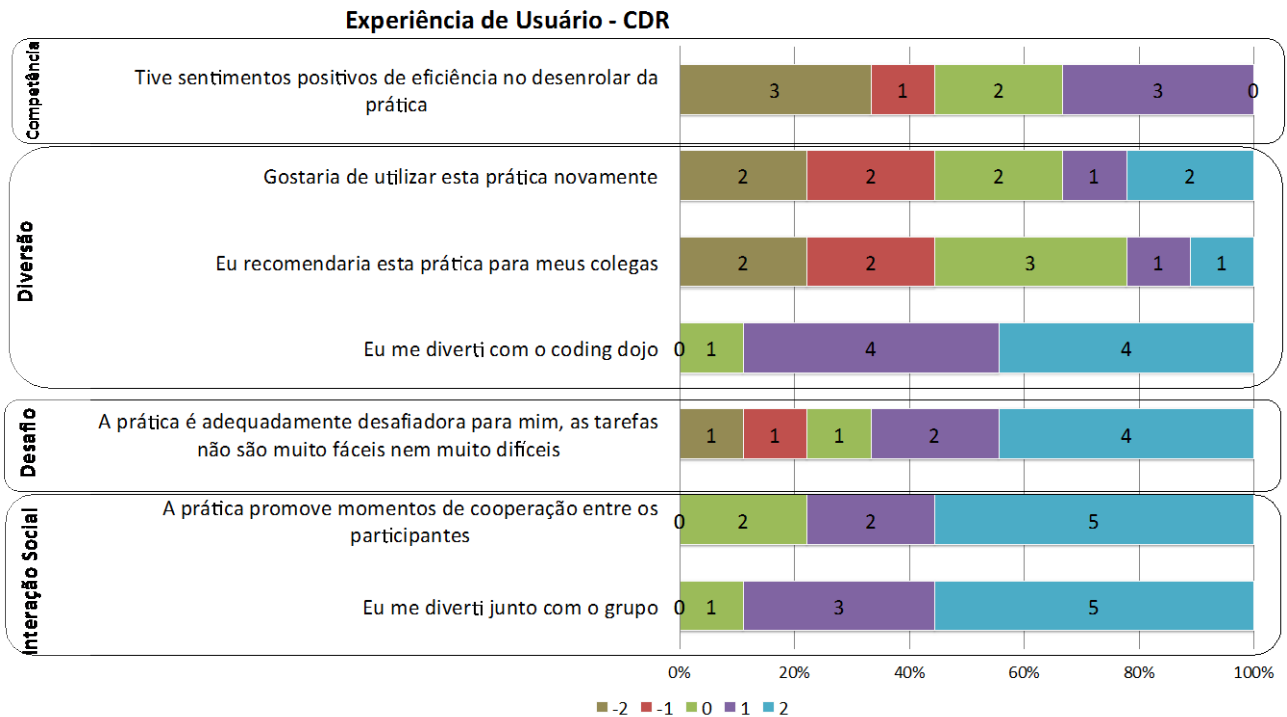


Figura 5.4 – Dimensões da experiência de usuário no grupo de CDR

Em relação à Competência (a capacidade de ser eficiente com a prática), a maioria dos estudantes expressaram positivamente a sua crença de que a PP tinha sido uma forma eficiente de aprender. No grupo de CDR, o feedback foi mais diversificado.

Quanto à dimensão de "diversão", no grupo PP todos os alunos relataram que eles querem usar a prática de novo e a maioria deles disse que eles se divertiram e gostaria de recomendar a prática de um colega. Por outro lado, os resultados do grupo de CDR não foram todos positivos, especificamente na recomendação da prática e no sentido de utilizá-la em outro momento.

No que diz respeito ao desafio da utilização de PP e CD, ambas as práticas mostraram não serem de alta complexidade para cada grupo. A interação social foi a dimensão melhor classificada para ambas as práticas pelos alunos. Todos os participantes se divertiram com o par ou grupo e relataram que a prática promoveu a cooperação entre os participantes.

Ao analisar a experiência de usuário em todas as dimensões, CDR apresentou um resultado mais diversos, especificamente em relação a Competência e Diversão. Percebeu-se que a experiência de CDR não marcou tanto os participantes, uma vez que a maioria dos participantes deu uma nota negativa para um segundo uso da prática. PP apresentou resultados mais positivos em todas dimensões.

5.4.3 Aprendizagem

A aprendizagem foi avaliada por duas formas: a percepção da aprendizagem em longo prazo e a aprendizagem segundo a taxonomia de Bloom [BLO56]. Em relação à primeira forma, a Figura 5.5 apresenta os resultados para PP e a Figura 5.6 para CDR, onde é apresentado que a experiência com a prática poderá ser útil no desenvolvimento de *mockups* na vida profissional.

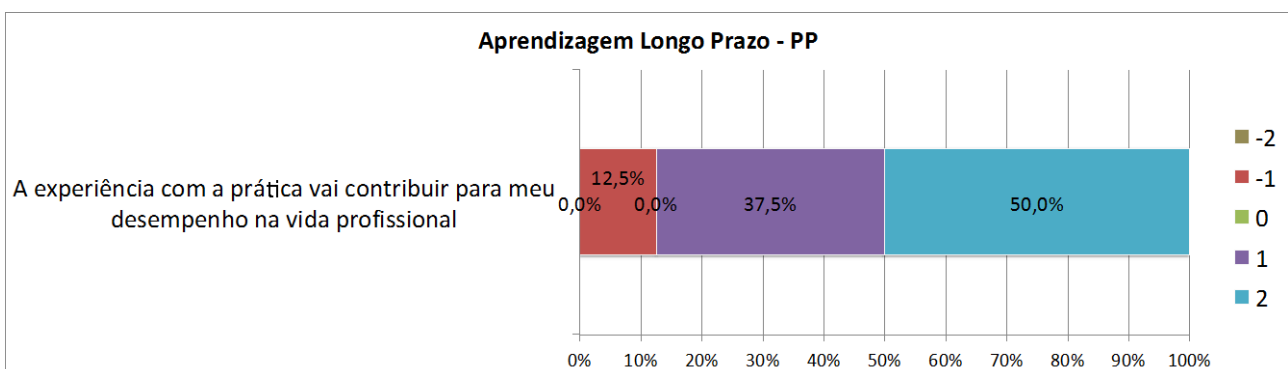


Figura 5.5 –Item aprendizagem de longo prazo para os grupos de PP.

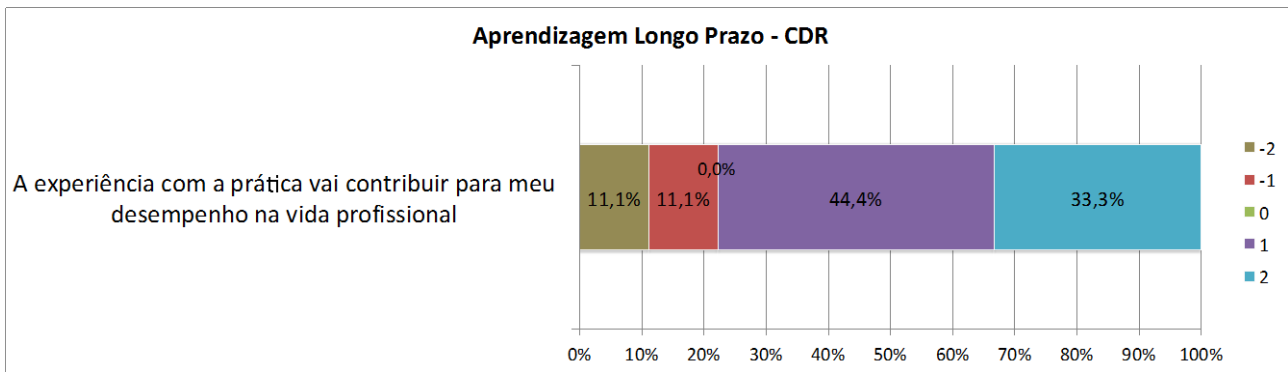


Figura 5.6 – Item aprendizagem de longo prazo para o grupo de CDR.

O principal aspecto citado pelos alunos foi o de saber como trabalhar em equipe, o participante 1 PP disse: "A PP ajuda a aprender a trabalhar em equipe para obter um melhor desempenho". O participante 3 do grupo de CDR relatou que a prática vai ajudar no uso futuro da programação em par: "Com Coding Dojo [Randori] vou saber melhor como se comportar em caso de uma programação em par."

A aprendizagem também foi ratificada pelas respostas do aluno em relação ao impacto percebido sobre os níveis de conhecimento, de acordo com a taxonomia de Bloom [BLO56]. Nesta taxonomia, os alunos informam em uma faixa que varia de 1-5 seus conhecimentos identificados nos conceitos antes e depois da utilização da prática.

Os alunos de ambos os grupos (Figura 5.7 para PP e Figura 5.8 para CDR) relataram um aumento significativo de conhecimento no que diz respeito a todos os três conceitos ensinados: *mockups*, a interação entre *mockups* e a sintaxe de Qt em todos os três níveis de conhecimento.



Figura 5.7 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de PP



Figura 5.8 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de CDR.

Em relação ao objetivo de aprendizagem do conceito de *Mockups*, a variação de PP (antes e depois) da prática foi superior à de CDR. Especificamente, em relação ao nível de “Aplicar”, PP teve uma variação positiva de 2,25 pontos em relação aos 2,0 de CDR.

Quanto ao conceito de Interação entre *Mockups*, PP também teve resultados superiores ao de CDR, especificamente em relação ao nível de “Lembrar o que é”, onde apresentou uma variação positiva de 2,15 pontos em relação a 1,56 pontos de CDR. Quanto ao nível “Entender como funciona”, CDR apresentou uma variação baixa (de 0,67 pontos), mas ainda sim positiva.

Em relação ao conceito de Qt, PP teve resultados superiores nos níveis “Lembrar o que é” e “Aplicar”. Especificamente quanto ao nível de “Lembrar o que é” PP teve uma variação maior, 2,25 pontos positivos em relação a 1,77 de CDR. CDR apresenta uma pequena margem superior no nível “Entender como funciona”, onde obteve 2,06 pontos em relação a 2,0 pontos de PP.

5.4.4 Benefícios

Os participantes que utilizaram a PP reafirmaram característica de interação e cooperação da prática. O Participante 1 relatou: "*A interação com o par é crucial para a prática ser desenvolvida com sucesso, o que melhora a aprendizagem.*" O Participante 4 reforça isso, dizendo: "*A interação com o par facilita a aprendizagem.*"

Outro benefício da Programação em Par relatado pelos alunos é sobre a criação de um padrão de programação. De acordo com os estudantes, PP ajuda a equipe a criar uma solução que possa ser comum em entendimento para os ambos os alunos. O Participante 6 disse: "*A prática nos leva a criar padrões que podem ser entendidos por ambos os alunos.*"

Em relação aos benefícios de CDR, os participantes relataram a detecção de defeitos no código durante o desenvolvimento. O participante 3 disse: "*Outras pessoas colaborando podem encontrar defeitos no código de forma mais rápida comparado quando estamos desenvolvendo de forma individual.*"

A aprendizagem também foi relatada como benefício, como participante 3 disse: "*No Coding Dojo, há mais chance de aprender técnicas de programação.*" Os estudantes reforçam a diversão em Coding Dojo. Participante 2 relatou: "*Coding Dojo promove um ambiente não-competitivo entre os participantes, tornando a prática programação um pouco mais divertida.*"

5.4.5 Desvantagens de CDR

A falta de consenso foi a maior desvantagem citada no grupo de PP, segundo a percepção dos alunos. O participante 1 relatou: "*Se o par não concordar em como fazer, pode se ter um tempo de desenvolvimento maior e também afetar a qualidade do produto*". A infraestrutura foi relatada como essencial para a prática. O participante 4 disse: "*ambiente muito tranquilo, a prática requer uma maior comunicação*". Os estudantes relataram outros pontos negativos, tais como a diferença de conhecimento entre os pares. O participante 8 disse: "*Se um dos dois não tem conhecimento suficiente do conteúdo da sessão, podem contribuir menos*".

O tempo de estudo e o momento da troca entre papéis foram citados como desvantagens no grupo de PP. O participante 1 disse: "*Quando houver uma interrupção para alternar os pares, a nossa concentração pode ser perdida, por vezes, o que prejudica a programação*".

Quanto à CDR, a principal desvantagem citada pelos alunos foi o conflito de objetivos entre os pares, e da falta de consenso. O participante 1 disse: "*CD não é difícil de compreender ou de praticar, mas é difícil aceitar as ideias do grupo em tudo*".

O participante 2 disse: "*Em vários momentos tentei convencer outras pessoas a fazer o que eu faço, porque eu julguei ser a melhor forma, mesmo sem ver o resultado do que estava sendo feito*". Os alunos também citaram a curta duração da troca dos pares, a dispersão dos alunos entre a audiência e a infraestrutura dos ambientes utilizados (devido a má visualização do código para toda a audiência).

5.4.6 Mockups

Foram distribuídos cinco *mockups* a serem desenvolvidos pelos alunos. No entanto, nenhum dos grupos terminou todos. No grupo de PP, apenas um dos quatro pares desenvolveu dois *mockups*, os outros três pares desenvolveram o primeiro e começaram a desenvolver o segundo *mockup*.

Além disso, apenas dois pares conseguiram executar os códigos em Qt, outros dois pares apresentaram *mockups* com erro de compilação. No grupo de CDR, os alunos desenvolveram apenas um *mockup*. Durante sete sessões os alunos tentaram corrigir um erro, e só nas últimas 3 sessões eles finalmente alcançaram a solução.

5.5 Discussão

Os feedbacks obtidos de ambas práticas corroboram que as duas podem ser efetivas para o ensino de desenvolvimento de *mockups*. Entretanto, tanto PP quanto CDR apresentaram desafios, o principal neste estudo foi em relação ao consenso durante as decisões de codificação. Como analisado na RSL (fase 1), fatores de compatibilidade entre os participantes é uma das evidências do estado da arte das práticas de codificação colaborativa.

Ao analisar os *mockups* desenvolvidos por ambos os grupos, percebe-se que ambos os grupos entregaram código com erros, sendo que um dos exercícios dos pares não executava. Durante a condução do estudo empírico, observou-se que no grupo de PP, os desenvolvedores eram mais focados na solução, porém tinham pouco cuidado com o código desenvolvido. Enquanto no grupo de CDR, o tempo utilizado para resolver uma tarefa era maior, porém com um cuidado maior na estratégia de codificação.

No geral, os estudantes tiveram um interesse maior em PP comparado à CDR, os números de PP foram sempre mais favoráveis do que CDR em aprendizagem, motivação e experiência de usuário. Além de PP ser uma prática consolidada na literatura, uma possível explicação para essa diferença, consistiu no nível de dispersão entre CDR e PP durante a sessão.

5.6 Ameaças à validade

Estudos empíricos envolvem a ameaça à validade que pode afetar a validade dos resultados. Nesta seção, apresenta-se as potenciais ameaças e como elas foram mitigadas.

5.6.1 Validade de Construto

A validade de construto se refere a relação entre teoria e observação, ou seja, se o resultado reflete o efeito do construto [WOH00]. Neste estudo, para avaliar aprendizagem, motivação e experiência do usuário, seguimos a construção proposta por Savi et al. [SAV11].

5.6.2 Validade Interna

Ameaças à validade interna podem ter influência nas conclusões sobre uma possível relação causal entre o tratamento (PP e CDR) e os resultados de um estudo. Neste estudo, foram consideradas quatro principais ameaças à validade interna: (a)

efeitos de formação, (b) a experiência em programação e em Qt dos participantes, (c) a experiência em PP e CD, e (d) o tipo de tarefas.

5.6.3 Validade Externa

A validade externa se refere à generalização dos resultados para uma população maior ou para outros contextos que se diferem deste estudo. Em relação ao efeito da formação, há um risco que envolve diferenças na qualidade da formação entre um grupo e outro. No entanto, nós controlamos essa ameaça, balanceando os grupos em termos de experiência. Além disso, a fim de minimizar a ameaça de conhecimento de programação dos participantes, realizou-se a divisão em grupos equilibrados de acordo com a experiência dos envolvidos.

Esta medida evitou que a experiência dos participantes afetasse os resultados globais das práticas. Quanto ao tipo de tarefas, esta ameaça foi controlada em ambas as práticas, utilizando o mesmo conjunto de *mockups* para o desenvolvimento. Em ambos os grupos os participantes também receberam as mesmas configurações de Qt e do material.

5.6.4 Validade de Conclusão

A validação de conclusão (ou validação estatística de conclusão) considera a relação entre o tratamento e os resultados em termos da significância estatística [WOH10]. Uma das limitações deste estudo foi o pequeno número de participantes, onde apenas foi possível criar um grupo de CDR.

Outra ameaça identificada é que todos os alunos no estudo vieram da mesma universidade. Por este motivo, os dados extraídos a partir deste estudo apresentam resultados importantes relacionados com a motivação, experiência do usuário e de aprendizagem, mas não podem ser amplamente generalizados no momento. Mais estudos e replicações são necessárias como trabalhos futuros.

5.7 Considerações Finais do Capítulo

O Estudo de Viabilidade 1 foi um primeiro esforço de avaliar os efeitos de CDR sob o ponto de vista dos participantes (QP2). Neste primeiro momento, optou-se por comparar CDR com PP, por esta última ser uma prática de codificação colaborativa mais consolidada na literatura (conforme observado na fase 1).

Além disso, nesta fase se avaliou aprendizagem em uma primeira fase (QP3) com uma estratégia de analisar a percepção dos participantes, onde ambos os tratamentos (PP e CDR) apresentaram resultados favoráveis, sendo possível inferir que estas práticas podem ser efetivas no processo de aprendizagem. Os resultados do estudo de viabilidade 1 foram publicados em Estácio et al. [EST15b], em parceria com o grupo USES-UFAM.

Neste primeiro estudo, alguns aspectos importantes foram levantados como: a infraestrutura do ambiente e a maior dispersão entre os alunos em CDR desenvolvido por ambos os grupos. Ainda foi possível perceber que CDR e PP apresentaram resultados positivos para a interação social, transferência de conhecimento e cooperação entre os participantes.

Uma limitação quanto à viabilidade de CDR ainda neste primeiro estudo, foi de avaliar a prática com tarefas que envolvem pouca lógica de programação, focando mais nos componentes de visualização, particularidades de *mockups*.

6 FASE 2: ESTUDO DE VIABILIDADE II

Este capítulo apresenta o planejamento e a condução do estudo de viabilidade 2. Após o estudo de viabilidade 1, percebeu-se a necessidade investigar mais os resultados de CDR com tarefas de codificação que envolvessem mais lógica de programação, diferentemente do estudo de viabilidade 1 que focou em *mockups*. Portanto, o estudo de viabilidade 2 é um complemento ao primeiro estudo da fase 2, como forma de analisar os efeitos iniciais de CDR.

A próxima seção detalha o contexto e objetivo deste estudo. A Seção 6.2 apresenta características do participante deste estudo. A Seção 6.3 apresenta os procedimentos e materiais utilizados no estudo. A Seção 6.4 apresenta os resultados deste estudo. A Seção 6.5 apresenta as ameaças a validade, e por fim na Seção 6.6 são apresentadas as considerações finais do capítulo.

6.1 Contexto e Objetivo

Como parte da execução da etapa de exploratória, foi conduzida a realização de um segundo estudo de viabilidade. O foco neste estudo consiste em observar e avaliar como a aprendizagem de conceitos de orientação a objetos, quando os participantes experimentassem as duas práticas PP e CDR, além de um grupo controle (a programação individual). O objetivo do Estudo de Viabilidade 2, de acordo com a estrutura GQM [BAS94], é apresentado abaixo:

Analisar: CDR e PP;

Com o propósito de: Avaliar;

Com respeito de: motivação, experiência de usuário e aprendizagem;

Do ponto de vista: desenvolvedores;

No contexto do: Desenvolvedores iniciantes.

Como mencionado no Capítulo 3, o Estudo de Viabilidade 2 foi conduzido em parceria com o grupo OPUS da PUC-RIO. Neste contexto, contemplou-se também a análise da qualidade do código, especificamente por meio da identificação de anomalias. Estes resultados não fazem parte do escopo desta pesquisa, mas podem ser consultados em Estácio et al.[EST15] e Oliveira et al. [OLI16].

6.2 Participantes

O estudo foi conduzido com desenvolvedores novatos no contexto do projeto da Aceleradora [EST14]. O projeto utiliza métodos ágeis como abordagem de processo de

desenvolvimento de software e tem por objetivo o ensino de práticas de desenvolvimento de software em 4 meses. Ao todo 14 desenvolvedores participaram do estudo.

O CDR é utilizado neste projeto como prática de treinamento durante o desenvolvimento de software, tendo a frequência de utilização por conveniência da equipe quando necessita compartilhar determinado tópico durante o programa. Assim como no estudo de viabilidade 1, foram coletadas informações de cada participante e classificados em: nenhum (N), baixo (B), médio (M) e alta (A) em termos do nível de experiência informada.

Desta forma, os participantes do estudo preencheram um formulário (Apêndice C) de caracterização com questões objetivas com objetivo de informar a experiência e conhecimento deles sobre: (a) programação, (b) a experiência deles com programação em pares e (c) a experiência dele com CDR. Para realizar o balanceamento em cada grupo resguardou-se que nenhum dos grupos tivesse mais participantes experientes que os outros, com o objetivo de assegurar o viés dos resultados, onde um grupo poderia ter um desempenho melhor que o outro. A Tabela 6.1 apresenta a experiência de cada participante nos respectivos grupos, foram marcados os participantes com mais experiência em cada grupo.

Tabela 6.1 – Experiência dos participantes em cada grupo

Grupo	Identificação	Programação	Java	Programação em Par	<i>Coding Dojo</i>
1	Participante 1	A	B	M	M
	Participante 2	M	M	M	B
	Participante 3	B	B	B	B
	Participante 4	B	B	B	B
	Participante5	M	B	B	B
	Participante 6	B	B	M	M
2	Participante 7	M	M	B	B
	Participante 8	B	B	M	M
	Participante 9	M	M	B	B
	Participante10	A	M	M	B
3	Participante 11	A	M	M	B
	Participante12	B	B	B	B
	Participante 13	M	B	B	B
	Participante 14	A	M	M	M

6.3 Procedimento e Materiais

Neste estudo, abordou-se um fator (identificação do impacto das práticas de codificação colaborativa em contraste com as práticas individuais) e três tratamentos (programação individual, PP e CDR). Além disso, foram selecionados três objetos de estudo, exercícios específicos de orientação a objetos com base nos conceitos vistos durante o contexto do curso. Os materiais utilizados neste estudo se encontram no Apêndice D.

O design adotado pelo estudo foi do tipo *crossed design* [PFL94]. Esta abordagem possibilitou que cada grupo experimentasse as três práticas e todos os objetos de estudos. Dois pesquisadores, um da PUCRS e outro da PUC-RIO acompanharam a execução das atividades. O *design* definido possibilita isolar a variável de aprendizagem, haja vista que cada grupo utiliza as práticas em diferentes sequências. A Figura 6.1 apresenta o *design* do experimento.

	Exercício A (Primeiro Passo)	Exercício B (Segundo Passo)	Exercício C (Terceiro Passo)
Programação Individual	Participantes 1-6	Participantes 7-10	Participantes 11-14
Programação em Par	Participantes 11-14	Participantes 1-6	Participantes 7-10
Coding Dojo Randori	Participantes 7-10	Participantes 11-14	Participantes 1-6

Figura 6.1 – Configuração do *crossed design* no experimento

Como pode ser observado na Figura 6.1, a primeira etapa consistia na resolução do exercício “A” aplicado para os participantes de 1 a 6 com programação individual, para os participantes de 11 a 14 para o tratamento de PP, e para os participantes de 7 a 10 com o tratamento de CDR.

Na segunda etapa, o exercício “B” foi aplicado para os participantes de 7 a 10 para programação individual, para os participantes de 1 a 6 para o tratamento de PP, e para os participantes de 11 a 14 para o tratamento de CDR.

O terceiro passo, o exercício “C”, foi aplicado para os participantes de 11 a 14 para programação individual, para os participantes de 7 a 10 para PP, e para os participantes de 1 a 6 para o tratamento de CDR. A formação dos pares em PP dentro de cada grupo foi realizada de forma similar a Braught et al [BRA10].

Cada exercício tinha uma complexidade particular. Por exemplo, o primeiro exercício consistia no desenvolvimento de um jogo de adivinhação animal, enquanto segundo consistia no desenvolvimento de funcionalidades para um caixa eletrônico. No exercício “C”, os alunos tinham que desenvolver um estoque de uma livraria. Para realizar a tarefa, o estudo foi conduzido *online* (simultaneamente e observado pelos pesquisadores), tendo a duração de duas horas ao total. Ao final do estudo, cada participante respondeu a um questionário e enviou para os pesquisadores. Foi utilizado o mesmo questionário do estudo de viabilidade 1.

6.4 Resultados

Assim como no estudo de Viabilidade 1, os dados foram coletados utilizando os instrumentos propostos por Savi [SAV11]. Para este estudo, optou-se pela retirada de dois itens do estudo de viabilidade 1, a motivação para esta retirada foram as dúvidas dos participantes sobre esses itens em relação à dinâmica das práticas. A Tabela 6.2 apresenta os itens removidos em relação ao estudo de viabilidade 1.

Tabela 6.2 – Itens removidos para o Estudo de Viabilidade 2 do questionário pós-experimento

Eu me diverti junto com o meu par
A prática é adequadamente desafiadora para mim, as tarefas não são muito fáceis nem muito difíceis

Com o objetivo de entender como ambas práticas poderiam afetar parte do treinamento que os participantes já tinham recebido, no estudo de viabilidade 2 foi adicionada uma nova questão aberta ao questionário pós- experimento: “Quais conceitos esta atividade te ajudou a lembrar (que foram vistos durante a aceleradora [EST14])? ”.

As próximas subseções apresentam resultados específicos em relação a motivação, experiência de usuário, aprendizagem, benefícios, desvantagens da prática e

o impacto de CDR no curso. A Seção 6.4.1 apresenta os resultados referentes à motivação, enquanto a Seção 6.4.2 apresenta os resultados de aprendizagem. A experiência do usuário é detalhada na Seção 6.4.3.

6.4.1 Motivação

Quando as dimensões foram analisadas individualmente, a dimensão de Atenção obteve dados positivos de ambas as práticas. Um dos pontos que chamou a atenção dos participantes em ambas as práticas foi a interação entre os participantes. O Participante 3 disse em relação ao PP: *Eu gosto de discutir a solução e o design da tarefa com o meu par, isso me ajuda a manter a situação durante o tempo de sessão*”.

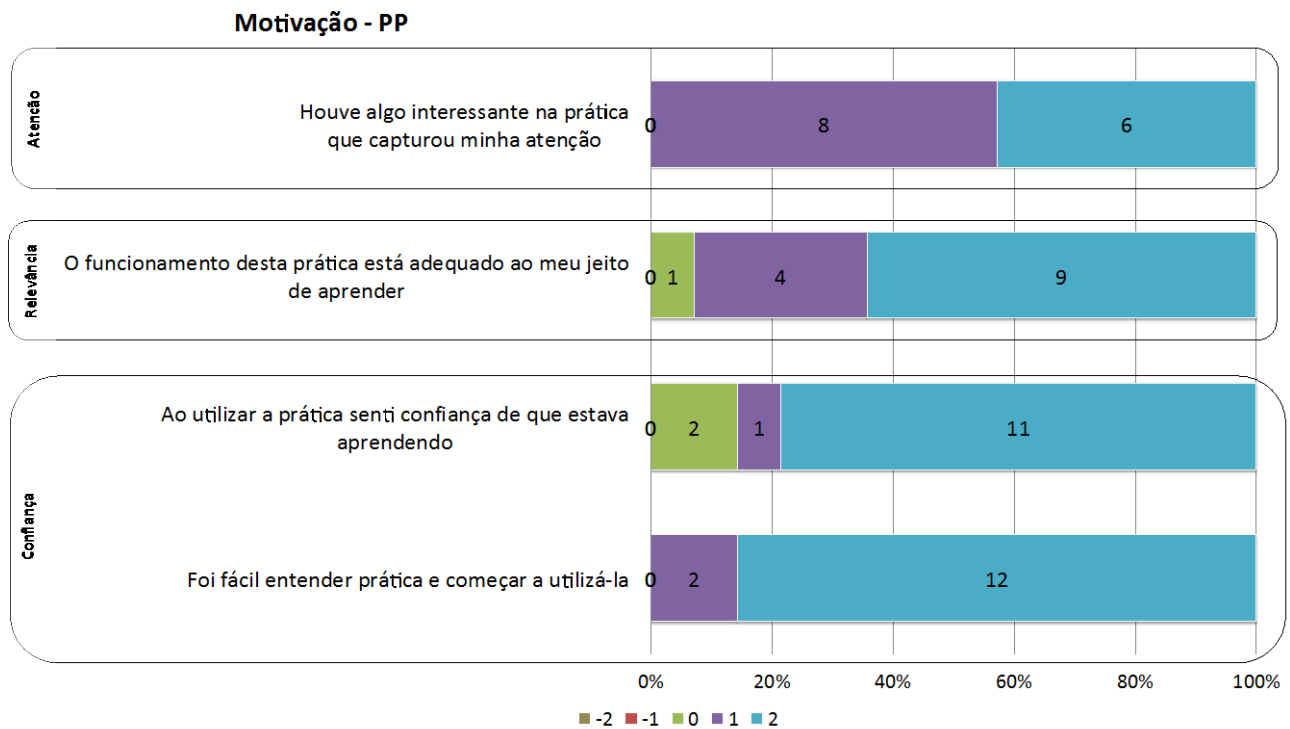


Figura 6.2 – Dimensões de motivação no grupo de PP

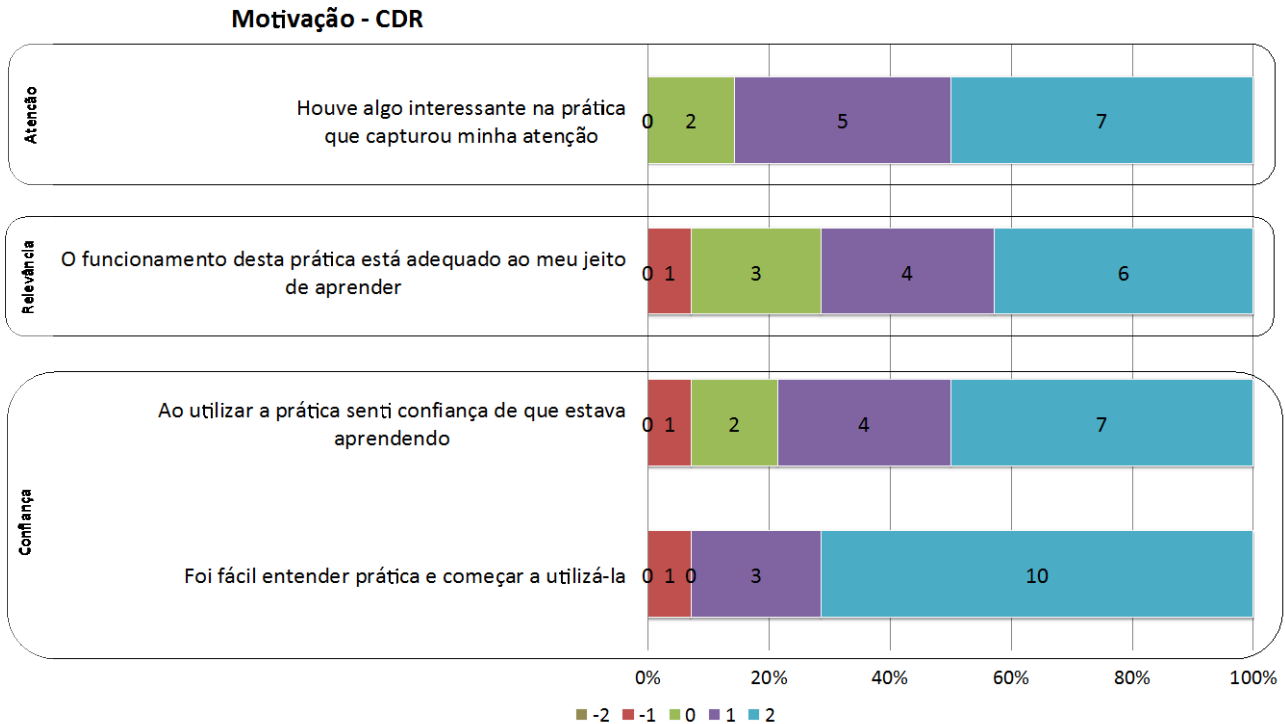


Figura 6.3 – Dimensões de motivação no grupo de CDR

Em relação à dimensão de Relevância, a maioria dos participantes reportou a o nível de aceitação de PP, mas CDR não foi amplamente aceita. Quatro participantes reportaram que a prática não se adequa ao jeito de aprender. Em PP, o Participante 4 relatou: “*PP é melhor para obter consenso que CDR, quando estávamos em uma sessão de CDR, era muito difícil concorda com todos*”.

Na dimensão Confiança, os itens relacionados ao entendimento do uso da prática, tanto PP quanto CDR apresentaram resultados positivos. PP apresentou mais resultados positivos em relação à impressão dos estudantes quanto a confiança de aprendizagem. Contudo, CDR também obteve respostas positivas. O Participante 4 disse: “*Em uma sessão de CDR, programadores experientes podem ajudar durante toda sessão, então nós temos a confiança de estarmos fazendo a coisa certa.*” A Figura 6.2 apresenta os resultados de Motivação em PP, enquanto a Figura 6.3 em CDR.

Ambas as práticas apresentaram um resultado positivo para a Motivação, entretanto os resultados de PP foram mais favoráveis aos resultados de CDR. Especialmente na Relevância e na Confiança da prática, CDR apresentou resultados mais nulos e negativos.

6.4.2 Aprendizagem

A maioria dos participantes expressou que ambas as práticas contribuíram positivamente para a aprendizagem em longo prazo (Figura 6.4 para PP e Figura 6.5 para CDR). Adicionalmente, os estudantes indicaram que estas práticas podem ser úteis em um ambiente de trabalho real. Além disso, foram relatados que CDR auxilia em um conhecimento inicial em como eles podem fazer decisões no projeto, por meio de discussões de pontos específicos no código fonte.

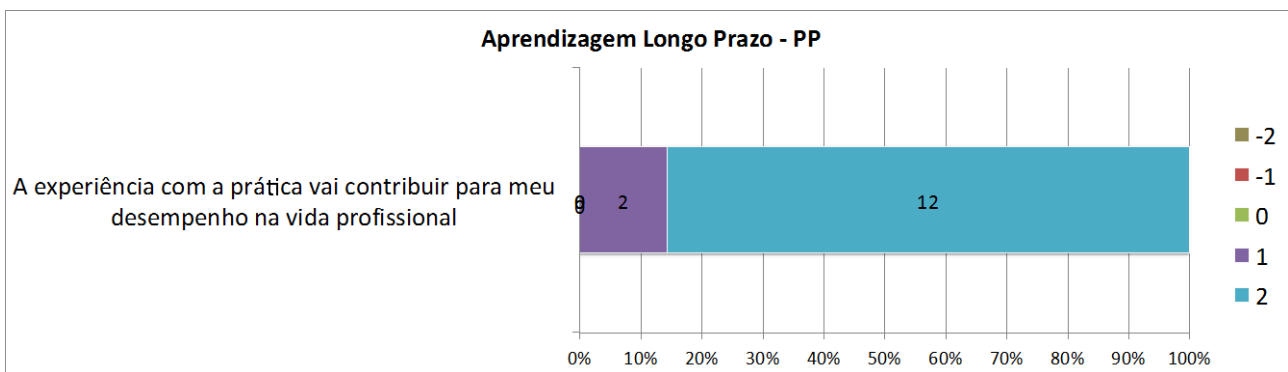


Figura 6.4 – Item aprendizagem de longo prazo para o grupo de PP.

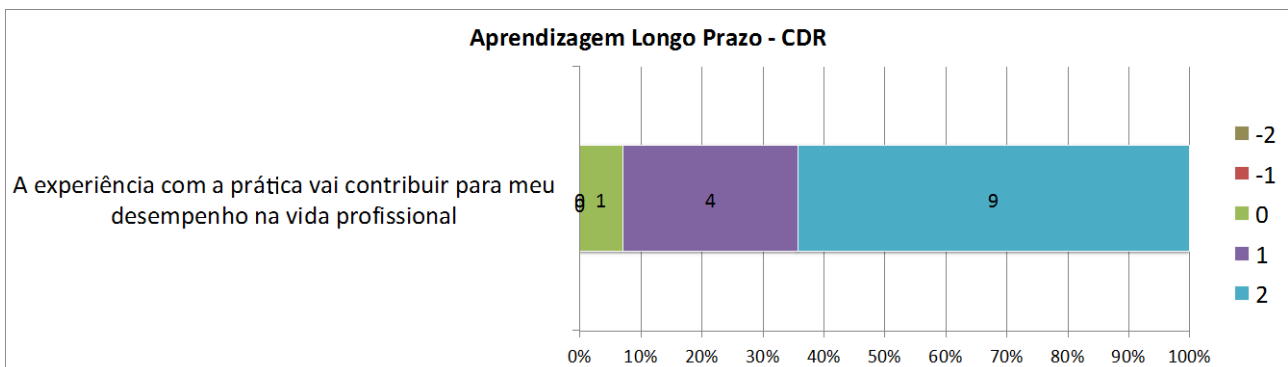


Figura 6.5 – Item aprendizagem de longo prazo para o grupo de CDR.

Em relação à aprendizagem baseada na Taxonomia de Bloom (Figura 6.6 para PP e 6.7 para CDR), foi avaliado a aprendizagem nas três categorias de habilidades presentes no questionário: Sintaxe do Java, Programação Orientada a Objetos e boas práticas de *design* do código desenvolvido.

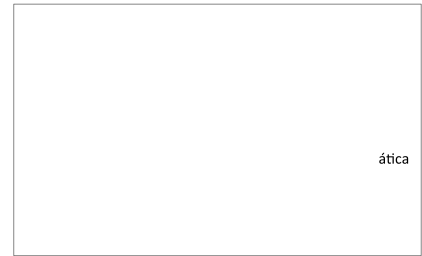
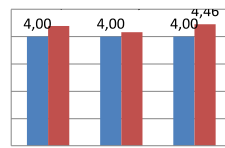


Figura 6.6 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de PP

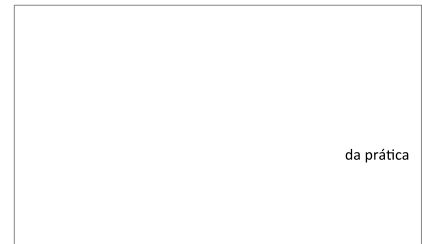
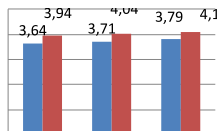


Figura 6.7 – Notas de aprendizagem na taxonomia de Bloom [BLO56] no grupo de CDR

Quanto ao objetivo de aprendizagem do conceito de Sintaxe da Linguagem Java, a variação de PP (antes e depois) da prática foi superior à de CDR. Especificamente, em relação ao nível de “Aplicar”, PP teve uma variação positiva de 0,74 pontos em relação aos 0,26 de CDR.

Quanto ao conceito de Programação Orientada a Objetos, PP também teve resultados mais favoráveis ao de CDR, especificamente em relação ao nível de “Aplicar”, onde apresentou uma variação positiva de 0,46 pontos em relação a 0,05 pontos de CDR. Quanto ao nível “Entender como funciona”, CDR apresentou uma variação superior à de PP com 0,25 pontos positivos em relação a 0,18 pontos.

Em relação ao conceito de Boas Práticas de *Design*, CDR teve resultados superiores em todos os níveis comparado com PP. Especificamente quanto ao nível de “Aplicar”, CDR teve uma variação maior, 0,55 pontos positivos em relação a 0,4 pontos de CDR. No nível “Lembrar o que é”, PP apresentou uma variação nula.

6.4.3 Experiência do Usuário

Por já terem uma experiência prévia em PP, os participantes reportaram de forma positiva a utilização da prática (Figura 6.9). Especificamente em relação à Diversão, o Participante 12 relatou: “Eu *me diverti com a prática quando estava tentando entender o conceito do problema.* ” PP recebeu notas positivas em todas as dimensões, sendo apenas o de Competência com uma nota neutra.

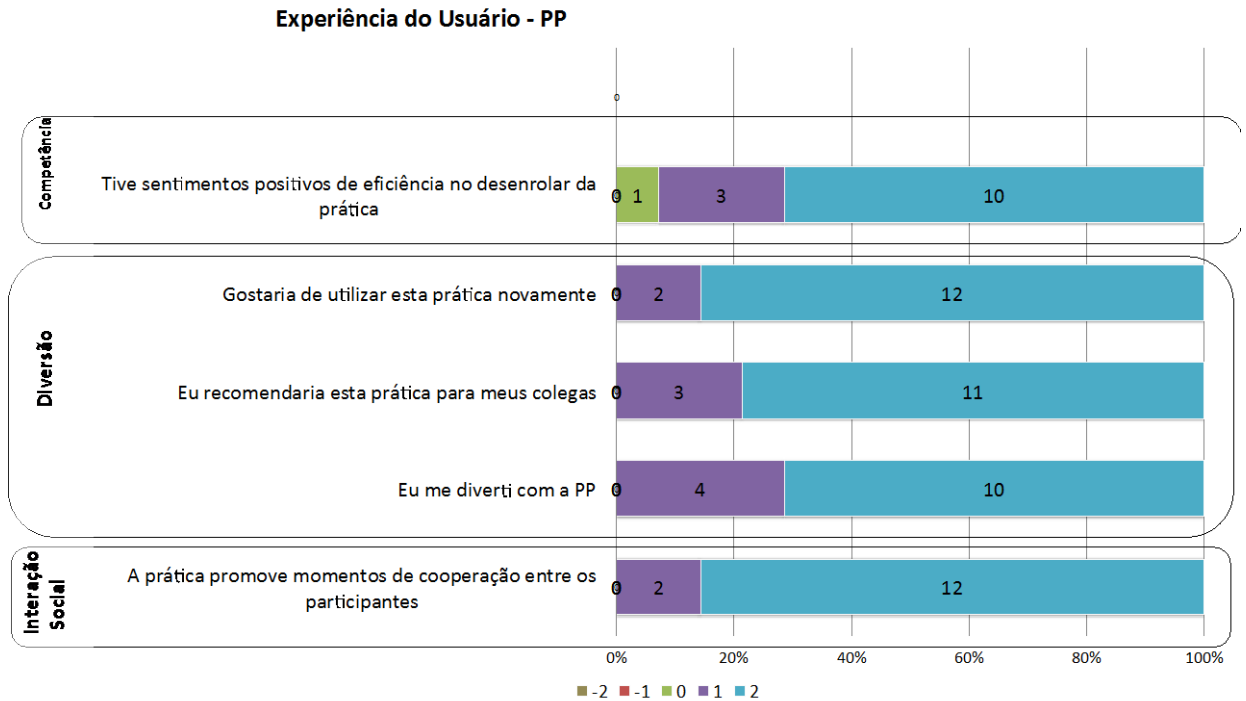


Figura 6.9 – Dimensões de experiência de usuário no grupo de PP

As respostas para CDR (Figura 6.10) tiveram um número favorável, porém com uma pontuação mais baixa em relação à PP. Em relação à Diversão, o participante 10 relatou: *“foi super tranquila [a sessão de CDR], mas porque já conhecia as pessoas que participaram”*.

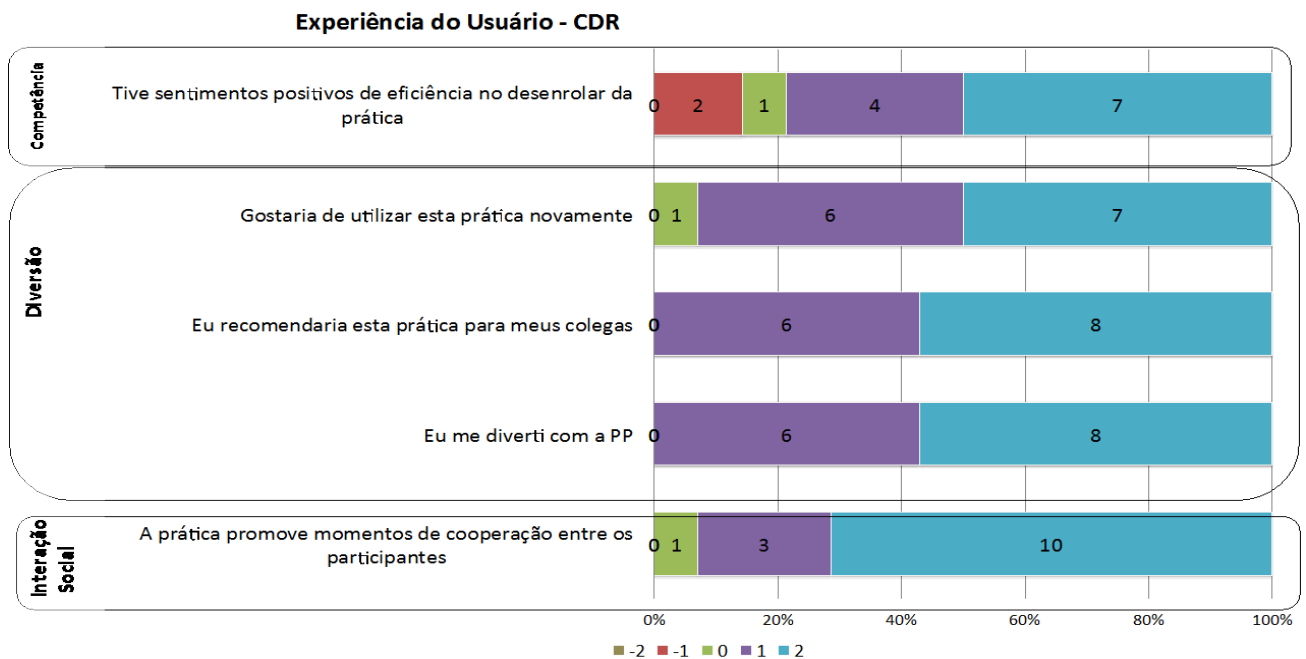


Figura 6.10 – Dimensões de experiência de usuário no grupo de CDR

Em relação à Interação Social, o participante 12 disse: “*Para mim essa é a principal característica e vantagem de CDR*”. Essa dimensão foi reportada como favorável pelos participantes tanto em PP quanto em CDR. Em relação à dimensão Competência, CDR apresentou duas repostas negativas, enquanto PP teve mais resultados positivos.

6.4.4 Benefícios de CDR

Assim como no Estudo de Viabilidade 1, os participantes apontaram como principais benefícios: a característica de colaboração, a cooperação, o trabalho equipe de PP, destacando a importância disso para a equipe. O Participante 12 relatou: “*A PP ajudou em conhecer melhor o jeito que os outros programam*”. O Participante 13 reportou a natureza colaborativa de resolução das tarefas: “*Com a PP vi novas formas de interpretar as tarefas*”. Outros pontos citados foram o feedback rápido e a facilidade de imersão ao problema.

Em relação aos benefícios de CDR, a cooperação e a colaboração foram os mais citados. O Participante 12 disse: “*CDR tinha um ambiente descontraído, sem pressão, pessoas com boa vontade [de colaborar]*.” Os papéis em CDR também foram citados como um ponto positivo da prática pelo Participante 2: “*Poder ser o copiloto e também avaliar de fora [como audiência]*.” Outros benefícios citados foram a qualidade de código, a aprendizagem e a resolução colaborativa de problemas.

6.4.5 Desvantagens de CDR

Entre as desvantagens de citadas pelos participantes em PP, destaca-se a falta de alinhamento com o par, o conflito de ideias. O Participante 13 relatou: “*Uma das desvantagens é seguir o raciocínio de outra pessoa*”. Alguns participantes relataram que algumas pessoas preferem codificar sozinhas e, para elas, a PP não se revela adequada.

Em relação às desvantagens de CDR, os participantes relataram desafios como conversas paralelas e a demora em resolver o problema. Especificamente no que se refere a troca de pares em CDR, o participante 2 relatou: “*Demora [pra eu ser piloto] pra chegar a ver*”. O tamanho do time em CDR também foi um tema discutido, o Participante 10 argumenta: “*Muitas pessoas no Dojo com pensamentos diferentes, podem gerar muita discussão*”.

6.4.6 Impacto no Curso de CDR

Quanto à terceira pergunta aberta (incluída no estudo de viabilidade 2), as respostas para PP e CDR foram bem similares. Os participantes reportaram que os principais conceitos lembrados durante a adoção da prática foram: Java e POO.

Algumas respostas foram mais específicas, o Participante 3 indicou que CDR ajudou no conceito de Polimorfismo, enquanto o Participante 8 e 11 relataram que em CDR questões relativas boas práticas de código foram mais discutidas.

6.5 Discussão

Os resultados para ambas práticas apresentaram de forma positiva as perspectivas analisadas, contudo os resultados não consistentemente o mesmo para todas as categorias, revelando peculiaridades na adoção de cada prática de codificação colaborativa.

Em relação à motivação, a percepção dos participantes por meio dos questionários apresentou que CDR apresenta desafios em relação ao consenso na tomada de decisões. Por outro lado, alguns participantes informaram que a audiência os ajudou a sentir mais confiança naquilo que estão aprendendo.

Quanto à experiência de usuário, a maioria dos participantes já tinha uma experiência prévia em PP e CDR, o que facilitou certa margem positiva dos resultados. Um ponto importante destacado por alguns dos participantes é o fato do grupo já trabalhar junto e se conhecer, o que pode melhorar a experiência de ambas as práticas.

No que tange a aprendizagem, analisou-se que ambas as práticas podem contribuir de forma favorável a este processo. CDR apresentou resultados mais positivos em relação as boas práticas de *design*, enquanto PP foi superior em relação a sintaxe do código em programação orientada a objetos. Na aprendizagem longo prazo, PP também se mostrou mais consolidada que CDR neste aspecto.

6.6 Ameaças a validade

Assim como no estudo de viabilidade 1, neste estudo há limitações quanto a validade dos resultados. Esta seção endereça estas possíveis ameaças, assim como as medidas para evitá-las.

6.6.1 Validade do Construto

Em relação à validade do construto, foi utilizado o instrumento elaborado por Savi [SAV11]. Para este experimento, considerou-se que os tratamentos (práticas de

codificação colaborativa) representam a causa e as métricas obtidas os efeitos. Algumas ameaças são de difícil controle, como o comportamento diferente do participantes devido a presença dos pesquisadores durante o experimento.

6.6.2 Validade Interna

Para a validade interna, foram consideradas as seguintes ameaças: (a) diferença entre os participantes em relação a experiência e habilidades (por exemplo, programação e conhecimento em Java, experiência em PP e CDR), e (b) diferença entre as tarefas experimentais (os exercícios utilizados podem ter influenciado nos resultados).

Para mitigar essas ameaças, foram aplicados princípios de *design* de experimento como balanceamento, atribuição aleatória, como sugerido em Wohlin et al. [WOH00]. O *design* do experimento auxiliou na mitigação das tarefas utilizadas, haja vista que o *crossed design* adotado permitia que todos os três tratamentos fossem aplicados aos três grupos.

6.6.3 Validade Externa

A ameaça a validade externa está principalmente relacionada à representatividade dos participantes, dos exercícios utilizados e do ambiente experimental específico. Os participantes deste estudo são do mesmo curso, desta forma, os dados extraídos nesta fase apresentam importantes indicadores em relação à motivação, aprendizagem e qualidade de código, porém deve se ter cuidado na generalização dos dados.

6.6.4 Validade de Conclusão

Quanto a validade de conclusão, a principal limitação deste estudo é a amostra pequena. Desta forma, mais resultados e replicações devem ser realizados para melhorar a validade na conclusão dos resultados.

6.7 Considerações Finais

Por meio de um *design* de experimento diferente do primeiro Estudo de Viabilidade, foi possível fazer um isolamento maior sobre o tratamento de aprendizagem, onde cada participante experimentou um tratamento diferente.

Além disso, as tarefas utilizadas neste estudo foram tarefas de codificação, diferentemente do Estudo de Viabilidade 1 que se concentrou em *mockups*. Isto

possibilitou uma análise sobre os efeitos de CDR em tarefas que envolvem lógica de programação.

Este estudo ratificou a importância de se investigar CDR sob a perspectiva dos participantes (QP2), utilizando o nível 1 do modelo de Kirkpatrick [KIR94]. Os resultados apresentados neste capítulo foram parcialmente publicados em Estácio et al. [EST15], em parceria com o grupo OPUS da PUC-RIO.

7 FASES 3 E 4: ESTUDOS DE OBSERVAÇÃO E ESTUDOS DE CASO (CICLO DE VIDA REAL)

Na etapa 1 deste trabalho foram executados estudos exploratórios para compreensão de uma avaliação inicial sobre a prática de CDR. Neste primeiro momento, procurou-se comparar a prática com a PP, por ambas apresentarem como característica a colaboração. No início da etapa 2, especificamente na fase 3, percebeu-se a necessidade de ter o foco em CDR, com o objetivo de alinhado a um dos objetivos deste estudo: o de entender como ocorre a aprendizagem colaborativa ocorre nesta prática. Desta forma, as fases 3 e 4 são caracterizadas pela execução de sessões de CDR, onde se buscou analisar sob uma perspectiva mais qualitativa. Este capítulo descreve os procedimentos e características de cada sessão de estudo observacionais (Seção 7.1) e do estudo de de caso (Seção 7.2).

7.1 Fase 3: Estudos de Observação

Nas três sessões de estudos observacionais, a organização foi realizada pelo autor deste estudo. Os participantes foram convidados a experimentar a prática, e os exercícios foram selecionados de acordo com a conveniência do autor desta tese. A infraestrutura utilizada nestas sessões consistia em um notebook e um projetor. O notebook foi configurado com as ferramentas necessárias (IDE Eclipse e JUnit). O principal objetivo destas sessões consistia em praticar TDD. Todos os participantes possuíam baixa experiência quanto a TDD. A linguagem de programação adotada foi Java e todas as sessões foram em inglês. Os participantes assinaram um termo de consentimento (Apêndice E).

Ao término de cada sessão, uma retrospectiva foi facilitada pelo autor desta pesquisa. Nos estudos de caso, a retrospectiva foi sugerida pelo autor desta pesquisa e acatada pelos participantes. Esta retrospectiva tinha por objetivo levantar pontos positivos, pontos negativos e melhorias para a prática de CDR. A dinâmica da retrospectiva consistia em colocar *post-its* em um espaço visual e discutir sobre as implicações dos pontos colocados.

Sessão I

A primeira sessão foi realizada com três alunos da FUB (de nacionalidade alemã), sendo dois de mestrado e um de graduação. Os membros já se conheciam antes da atividade. Os participantes possuíam uma experiência inicial em programação e um

conhecimento prévio em TDD. Os exercícios para esta sessão foram *Fizz Buzz*, *Fibonacci* e *Game of Life* (Anexo I).

Sessão II

Na segunda sessão, também três alunos participaram do estudo. Um aluno de mestrado e dois alunos do nível de graduação da FUB participaram da atividade (um homem e uma mulher). Os membros já se conheciam antes das atividades. Os participantes tinham experiência inicial em programação. Dois participantes já possuíam um conhecimento prévio em TDD, porém um participante não tinha tido nenhum contato. As tarefas utilizadas foram *Roman to Numerals* e *Mars Rovers* (Anexo 1).

Sessão III

A terceira sessão contou com a participação de sete participantes, seis alunos de graduação e um aluno de mestrado (seis homens e uma mulher). Nesta sessão, três alunos brasileiros que estavam visitando a FUB, participaram da sessão. Todos os participantes possuíam uma experiência inicial em programação. Três participantes não possuíam uma experiência prévia com TDD. As tarefas utilizadas foram *Fizz Buzz*, *Numbers in Words* e *City of Flatland* (Anexo1).

A Tabela 7.1 sumariza as principais características da sessão dos estudos observacionais. Apresenta-se informações sobre a nacionalidade dos participantes, onde foi executada, se os participantes já se conheciam anteriormente e o idioma utilizado.

Tabela 7.1 – Característica das sessões de CDR de Estudos Observacionais

Id Sessão	Participantes	Nacionalidade dos Participantes	Os participantes se conheciam?	Local	Idioma utilizado na Sessão
S1	P1 (Mestrado), P2 (Mestrado), P3 (Graduação)	Alemães	Sim	Alemanha	Inglês
S2	P1 (Mestrado), P2 (Graduação), P3 (Graduação)	Alemães	Sim	Alemanha	Inglês
S3	P1...P6(Graduação), P7 (Mestrado)	4 Brasileiros e 3 Alemães	Alguns não se conheciam	Alemanha	Inglês

A Tabela 7.2 apresenta características das ferramentas utilizadas e o tempo de gravação dos estudos observacionais. Destaca-se a linguagem de programação, a ferramenta de desenvolvimento e o *framework* de testes utilizado.

Tabela 7.2 – Características técnicas das sessões de CDR de Estudos Observacionais

Id Sessão	Tempo de Sessão	Linguagem de Programação	Ferramenta	<i>Framework</i> de Teste
S1	1h:16m	Java	Eclipse	JUnit
S2	1h:28m	Java	Eclipse	JUnit
S3	2h:03m	Java	Eclipse	JUnit

7.2 Fase 4: Estudos de Caso

Nos Estudos de Caso, a análise de CDR foi em um ambiente menos controlado, desta forma, o autor desta pesquisa estava presente na sessão. Porém, o planejamento da sessão como o tempo de troca de participantes, os exercícios e o objetivo a ser exercido ficaram a cargo dos participantes que organizaram. As sessões ocorreram no Brasil e foram realizadas em português.

Sessão IV

A sessão IV contou com a participação de quatro desenvolvedores de uma empresa multinacional de consultoria de software. As sessões ocorreram em um horário diferente do período de trabalho dos participantes. O objetivo da sessão era de praticar TDD na linguagem de Programação Go⁵.

Um dos participantes foi o organizador da sessão, sendo ele o único a ter uma experiência inicial nesta linguagem. Apesar de não conhecerem a linguagem, os outros participantes já possuíam experiência na indústria em TDD. O exercício selecionado pelo organizador da sessão foi o *Fizz Buzz*.

Sessão V

A sessão V teve a participação de 11 desenvolvedores do projeto Aceleradora [EST14]. Um desenvolvedor, mentor dos desenvolvedores iniciantes, facilitou esta sessão: organizando o ambiente, elaborando os exercícios. O objetivo desta sessão consistia em praticar Jasmine⁶ (*framework* de teste em Javascript).

⁵ <https://golang.org/>

⁶ <https://jasmine.github.io/>

Os participantes (com exceção do facilitador) tinham um conhecimento baixo na linguagem e nenhum contato com a *framework* e a prática de CDR foi utilizada neste contexto como forma de treinamento. Desta forma, esta sessão tornou-se parte do processo de aprendizagem do time em tecnologias que estavam sendo adotadas no projeto. A Tabela 7.3 apresenta as principais características dos estudos de caso.

Tabela 7.3– Característica das sessões e dos participantes de CDR do Estudos de Caso

Id Sessão	Participantes	Nacionalidade dos Participantes	Os participantes se conheciam?	Local	Idioma utilizado na Sessão
S1	P1..P4 (Desenvolvedores Experientes)	Brasileiros	Sim	Brasil	Português
S2	P1...P11 (Desenvolvedores Junior), P12 (Desenvolvedor Pleno)	Brasileiros	Sim	Brasil	Português

A Tabela 7.4 apresenta as características técnicas dos estudos de caso. Destaca-se a ferramenta utilizada, bem como o *framework* de testes utilizado, e o tempo de cada sessão.

Tabela 7.4 – Características técnica das sessões de CDR dos Estudos de Caso

Id Sessão	Participantes	Nacionalidade dos Participantes	Os participantes se conheciam?	Local	Idioma utilizado na Sessão
S1	P1..P4 (Desenvolvedores Experientes)	Brasileiros	Sim	Brasil	Português
S2	P1...P11 (Desenvolvedores Junior), P12 (Desenvolvedor Pleno)	Brasileiros	Sim	Brasil	Português

8 FASES 3 E 4: RESULTADOS

Este capítulo apresenta os resultados a partir da análise dos estudos de observação e estudos de caso. A Seção 8.1 apresenta a notação e formato de apresentação destes resultados. Os resultados são apresentados a partir da Seção 8.2, onde se apresenta os níveis de colaboração identificados e na 8.3 os papéis emergidos durante a prática. Nas Seções 8.4, 8.5, 8.6, 8.7 e 8.8 são descritos fenômenos de CDR. Na Seção 8.9 é apresentada a percepção dos participantes nas retrospectivas e na Seção 8.10 principais diferenças entre os estudos de observação e de estudos de caso. Por fim, na Seção 8.11 são apresentadas as ameaças à validade destes estudos, e na Seção 8.12 a conclusão deste capítulo.

8.1 Notação e Formato

Nos resultados deste trabalho, quando se refere a um conceito analisado por meio de GT, a notação utilizada é no formato negrito: **Algum Conceito**. Além disso, serão especificados 1) a sessão e 2) os participantes envolvidos nos exemplos utilizados. Os exemplos coletados em inglês foram traduzidos para português, e podem usar uma conotação um pouco diferente do original. Os exemplos apresentados neste estudo seguiram um formato de narrativa inspirado em Kumar [KUM16], conforme a estrutura abaixo;

Situação: a descrição do fato ocorrido, a caracterização do problema e da atividade envolvida.

Acontecimento: a transcrição do que ocorreu no momento daquela situação.

Discussão: uma discussão sobre o exemplo apresentado.

8.2 Níveis de Colaboração

O treinamento em CDR envolve diferentes níveis de colaboração. O desejável **Nível de Conceito** ocorre quando a colaboração (discussão e questões) entre os participantes se concentra no objetivo da sessão. No nosso contexto, isso ocorreu com a prática de TDD nas sessões.

O **Nível de Tarefa** se refere à programação da tarefa propriamente dita, quanto os participantes interagem buscando compreender e solucionar as tarefas de uma sessão de CDR. No **Nível de Programação**, a menos importante pro contexto de treinamento, é onde os participantes discutem sobre o aspecto específico de uma linguagem de programação. A figura 8.1 apresenta o grafo do nível de colaboração, o detalhamento dos três níveis é feito nas subseções abaixo.

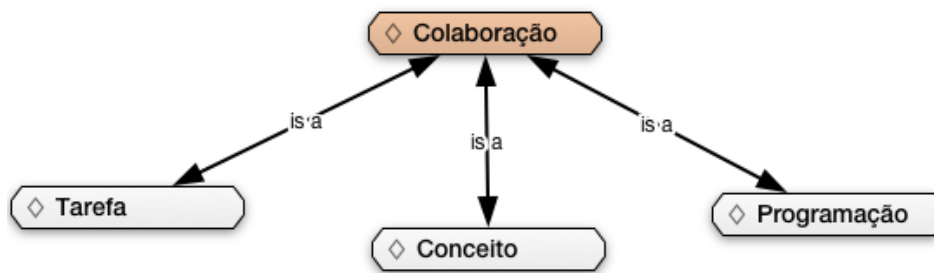


Figura 8.1 – Níveis de Colaboração identificados nas sessões de CDR.

8.2.1 Nível de Conceito

Quando se analisa os episódios em torno do **Nível de Conceito**, percebe-se que a maior parte da análise provenientes da GT são relacionados ao ciclo de TDD e a teste de software, haja vista que os objetivos das sessões consistiam em aprender ou praticar a abordagem de TDD. Sobretudo a **Estratégia de Teste**, que se refere a abordagem de teste utilizada neste tipo de teste, foi uma das identificadas. Uma estratégia bem adotada é a de **Baby Steps**, isto é, quando se é necessário fazer passos curtos para chegar a solução do código.

Ainda no **Nível de Conceito**, foram identificados recorrentes episódios nos quais o assunto entre os participantes foi sobre a **Execução dos Testes**, esta se refere ao processo de execução de um caso de teste e a comparação dos resultados (do esperado ao resultado real). A **Sintaxe de Teste** se refere ao formato necessário para escrita de um caso de teste, especificamente em determinada linguagem de programação ou *framework* específico de teste. A **Convenção de Nomes** acontece quando se é discutido a nomenclatura dos testes desenvolvidos. A Figura 8.2 apresenta o grafo do **Nível de Conceito**.

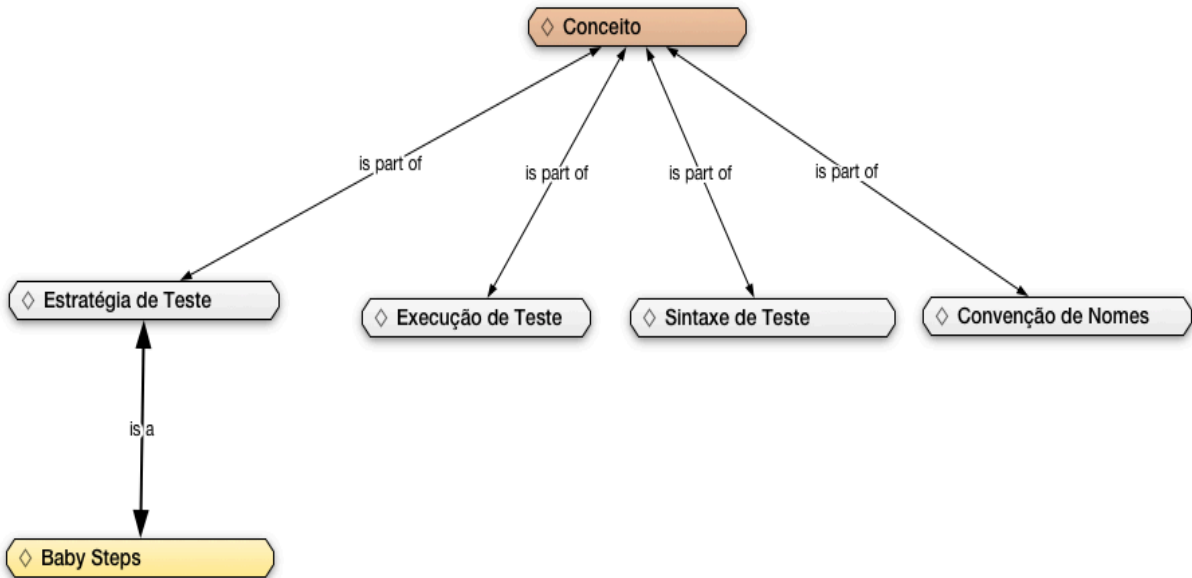


Figura 8.2 – Caracterização do Nível de Conceito

8.2.2 Nível de Tarefa

Quando a colaboração ocorreu no **Nível de Tarefa**, a principal causa foi uma **Pesquisa Externa** que consistia em uma investigação na *web* para compreender melhor a tarefa. Uma **Solução Alternativa** foi proposta como uma forma de resolver a tarefa em questão. Além disso, a **Depuração (*debugging*) de Código** foi um mecanismo utilizado para entender o que tarefa exigia. A Figura 8.3 apresenta o grafo do **Nível de Tarefa**.

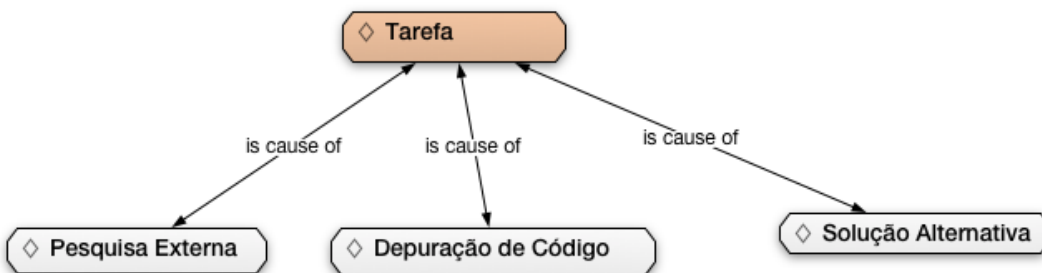


Figura 8.3 – Caracterização do Nível de Tarefa

8.2.3 Nível de Programação

Em relação ao **Nível de Programação**, a maior parte das discussões envolveu a **Sintaxe da Linguagem de Programação** utilizada. Especificamente na S5, por se tratar de uma linguagem nova para a maioria dos participantes, percebeu-se um número maior de episódios à **Nível de Programação**.

Durante um episódio, a discussão sobre o **Declaração de um Método Estático** envolveu uma **Pesquisa Externa**. Especialmente na S4 e S5, o **Nível de Programação** foi discutido em mais situações devido à pouca experiência dos participantes com as tecnologias utilizadas nestas sessões. A **Sintaxe** das linguagens utilizadas também foi recorrente no **Nível de Programação**. Por vezes se discute especificamente sobre uma linguagem de programação, como as **Características das Variáveis em Java**. A Figura 8.4 apresenta um grafo do **Nível de Programação**.

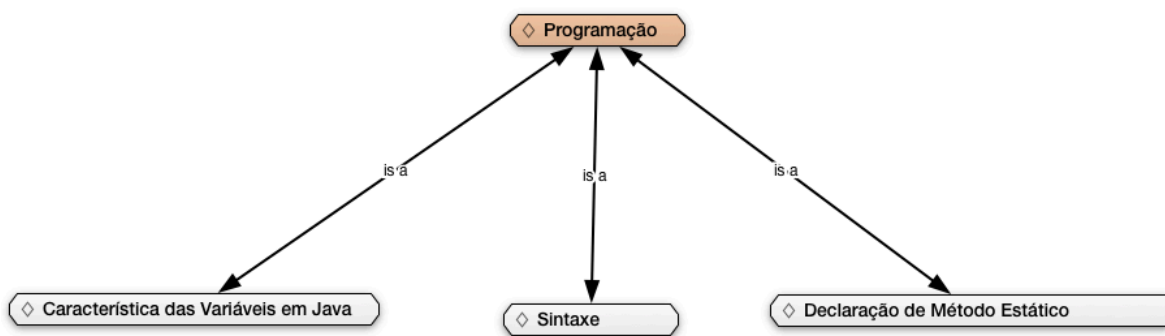


Figura 8.4 – Caracterização do Nível de Programação

8.3 Papéis

Durante a análise das sessões, observou-se que outros papéis que não os já estabelecidos da prática (Audiência, Piloto e Copiloto) foram emergidos dos dados analisados.

8.3.1 Mestre

O **Mestre** atua como um tutor para os outros participantes. Durante as sessões, percebeu-se que os participantes que assumiram este papel de forma implícita e natural tinham como principal característica a experiência ou já estarem mais familiarizados com a tecnologia (linguagem de programação) utilizada.

A participação mais ativa do **Mestre** acontece no **Nível de Conceito**, auxiliando os participantes nos conceitos de TDD. No **Nível de Tarefa**, ele ajuda principalmente nas questões mais difíceis, onde os participantes ficam bloqueados.

8.3.2 Novato

O **Novato** tem um papel de dualidade, enquanto o **Mestre** suporta com conhecimento, o **Novato** pergunta por isto, especificamente no **Nível de Conceito**. Nem todos os participantes inexperientes atuam no papel do **Novato**, na S5 alguns participantes tinham conhecimento prévio, porém a maioria dos participantes eram inexperientes.

O **Novato** parece aumentar a colaboração em torno da aprendizagem a **Nível de Conceito**. Durante as sessões, um **Novato** parece mais ativo quando atuando como piloto, enquanto permaneceram em silêncio quando desempenharam função de copiloto.

8.4 Intervenção do Mestre

O **Mestre** não ajuda no conhecimento sob demanda. Ele faz intervenções durante as sessões de CDR. Observou-se dois tipos distintos de intervenção: a reativa e a proativa.

8.4.1 Intervenção Reativa do Mestre

Ao perceber alguma ação equivocada, principalmente em relação ao **Nível de Conceito**, o **Mestre** faz uma intervenção, auxiliando os participantes. Este tipo de movimento também é percebido no **Nível de Tarefa**. Um exemplo que ocorreu na S1 é descrito abaixo.

Situação

Durante a primeira tarefa da S1, o desenvolvedor P3, atuando como piloto, inicia o desenvolvimento do primeiro teste unitário do exercício. O episódio se inicia em 05:50 da sessão, porém alguns minutos atrás, P3 já apresenta algum desentendimento e dificuldades em relação aos conceitos de TDD. Ao escrever o primeiro teste da tarefa, ele já inicia com três *asserts*. Ao analisar esta ação, P1 intervém com o objetivo de auxiliar no desenrolar da prática.

Acontecimento

Mestre (P2): *“Então, TDD seria para desenvolver [o teste] antes do desenvolvimento. Para todo teste ou não?”*

Copiloto (P1): *“Sim, você deveria fazer um único assert para um único caso que você deseja, e então escrever o código.”*

Piloto (P3): *“Ou nós iremos fazer múltiplos [asserts] para grandes casos [de teste].”*

Mestre (P2): *“Hum, era...mas a ideia de TDD é pegar um pequeno caso de tese, implementá-lo, e então fazer o próximo.”*

Discussão

O **Mestre** aqui percebe uma oportunidade de intervir no **Nível de Conceito**, especificamente alertando para a **Estratégia de Teste**, explicando o uso de **Baby Steps**. Apesar da **Intervenção Reativa do Mestre**, a primeira pergunta do **Mestre** é pouco intrusiva, no sentido de compreender o **Nível de Conceito** dos outros participantes. O segundo momento de participação do **Mestre** já acrescenta diretamente ao **Nível de Conceito**.

8.4.2 Intervenção Proativa do Mestre

Embora não seja tão necessário em algum momento uma intervenção, o **Mestre** antecipa um comentário como forma de precaver ou indicar uma situação de melhoria. Um exemplo de **Intervenção Proativa do Mestre** é apresentado abaixo.

Situação

Na segunda tarefa da S3, quando o P2 observou uma oportunidade de mudança no nome dos testes já escritos.

Acontecimento

Mestre (P3): “*Você não deve colocar mais [a palavra] Test Alguma Coisa.*”

Discussão

A mudança a **Nível de Conceito** possibilitou uma melhoria na **Convenção de Nomes**. Apesar de não ser uma mudança crucial no momento do episódio, a **Intervenção Proativa do Mestre** possibilitou parte da execução do Ciclo de TDD: o de Refatoração.

8.5 Participação dos Novatos

Um **Novato** participa principalmente no **Nível de Conceito**. Durante as sessões do estudo observacional, foi possível perceber uma interação maior na discussão em **Estratégia de Teste**. Um episódio da participação de um novato é apresentado abaixo.

Situação

Durante o primeiro caso de teste a ser feito na S1 (28:45:42 - 28:58:47), o desenvolvedor P4 tem dificuldade na escrita do método.

Acontecimento

Mestre (P3): “[*indicando o método a ser utilizado*] *Assert Equal.*”

Novato (P4): “*Em letra maiúscula?* ”

Mestre (P3): “*Equals com S.*”

Discussão

No **Nível de Conceito**, percebeu-se neste episódio que uma interação onde o **Novato** buscava compreender conceitos fundamentais em relação a TDD, especificamente em **Sintaxe de Teste**.

O papel de novato também foi percebido em outros níveis de colaboração. Abaixo, descreve-se um episódio no **Nível de Tarefa**.

Situação

Durante a resolução do exercício *Fizz Buzz* na S4 (36:04:66 – 36:17:02), um dos participantes era o único que não conhecia a tarefa proposta.

Acontecimento

Novato (P4): “*Desculpa me meter, mas o exercício é assim mesmo, tem que ter dois métodos, então?*”

Copiloto (P1): “*Não, é um método só, quando for 3 retorna Fizz, e quando for 5 retorna Buzz.*”

Piloto (P2): “*É bem facinho.*”

Discussão

Os outros participantes já estavam familiarizados com a tarefa, enquanto o **Novato** buscou entender mais a sua resolução, colaborando em **Nível de Tarefa**.

8.6 Participação da Audiência

Durante as sessões foi possível observar que a **Participação da Audiência** ocorreu com maior frequência nos grupos menores (S1 e S3). Em sessões maiores (S3 e S5) foi possível observar **Conversar Paralelas** e **Distrações** entre os participantes com tópicos não relacionados.

8.6.1 Sincronização com os pares

Em algumas situações, membros da audiência buscavam sincronizar o conhecimento com os então pares do turno, acontecendo a **Sincronização com os pares**. Na maioria dos episódios, essa situação foi motivada pela compreensão do *status* de codificação ou por uma tentativa de entender a abordagem seguida pelos pares. Um exemplo é descrito abaixo.

Situação

Na S2 (11:23:11 - 11:52:60), um membro da audiência buscou entender qual o próximo passo a ser testado.

Acontecimento

Audiência (P2): *“Mas... agora que temos um teste para um número fixo, o próximo passo seria testa um número romano?”*

Copiloto (P2): *“Não”*

Piloto (P1): *“Não!”*

Audiência (P2): *“Não?”*

Copiloto (P2): *“O próximo passo seria para teste se o número romano tem o mesmo formato e valor.”*

Discussão

A **Participação da Audiência** neste episódio ocorreu no âmbito de **Sincronização com os Pares**, onde o membro da audiência buscou entender se a **Estratégia de Teste** que ele estava cogitando era semelhante ao do piloto e copiloto. No episódio descrito acima, o copiloto clarificou que o passo de teste seria diferente daquele pensado pelo membro da audiência.

8.6.2 Audiência como Segundo Copiloto

A **Audiência como Segundo Piloto** aconteceu em alguns episódios, onde um membro da audiência auxiliou os pares no então turno. Este auxílio acontecia muito em **Nível de Tarefa**, como a sugestão de uma **Solução Alternativa**. Porém, também ocorreu em **Nível de Conceito** e **Nível de Programação**. O episódio abaixo exemplifica um momento.

Situação

Durante a segunda tarefa na S5 (26:51:60 - 27:34:88), P4 e P5 estão tentando realizar a segunda tarefa que consiste na extração de uma lista simplificada (autor e livro) a partir de uma lista completa. Os pares estão declarando esta lista simplificada e o P6 (um dos membros da audiência) faz uma sugestão.

Acontecimento

Audiência (P6): *“Faz [primeiro] ele receber uma lista e retornar o tamanho da lista. E testa se o retorno é o mesmo tamanho do que foi enviado”.*

Piloto (P4): *“Tá, então ele recebeu a lista e retorna o tamanha dela?”*

Audiência (P6): *“Testa se o tamanho da lista que retornou é que foi é o mesmo da que foi”.*

Piloto (P4): *“É o .length () para retornar”.*

Audiência (P6): *“É, agora coloca do outro lado lá [na classe de teste]”.*

Discussão

Neste episódio, a **Participação da Audiência** foi no sentido de sugerir uma **Solução Alternativa** para **Estratégia de Teste**. A **Audiência como Segundo Piloto** também aconteceu no **Nível de Programação**, como forma de ajudar na **Sintaxe** de JavaScript para a dupla.

8.7 O Facilitador como Mestre

Nas sessões S4 e S5, foram percebidos alguns episódios de **o Facilitador como Mestre**. Este participante, além de organizar as sessões, auxiliava na solução em algumas situações onde os participantes estavam bloqueados.

Situação

Na sessão S5 (19:13:39 – 19:39:49), o P1 que estava como piloto possuía uma dúvida em desenvolver a atividade, e resolveu consultar o facilitador da sessão de CDR.

Acontecimento

Piloto (P1): “*Dá pra acessar o Google?*”

Facilitador (P11): “*Claro, mas qual tua dúvida?*”

Piloto (P1): “*É o que a gente tem que comparar.*”

Facilitador (P11): “*A gente vai comparar item por item, é uma forma de fazer o resultset.*”

Discussão

Neste episódio, o P1 pediu uma permissão ao facilitador para fazer uma **Pesquisa Externa**. O facilitador respondeu tentando ajudar como **Mestre**. A interação ocorreu no **Nível de Tarefa**.

8.8 A Aprendizagem na ZDP

As interações apresentadas anteriormente como a **Intervenção do Mestre** e a **Participação do Novato** evidenciam uma **Assistência Externa** principalmente no **Nível de Conceito**. A **Assistência Externa** caracteriza o primeiro estágio da ZDP descrito no referencial teórico desta tese (ver seção 2.4.1).

Em alguns episódios, percebeu-se que a **Assistência Externa** resultou em um segundo momento em uma **Auto-Assistência**, segundo estágio de ZDP, onde o Novato adquire um conhecimento ou habilidade de fato. O exemplo abaixo, estruturado nos dois estágios da ZDP, apresenta esta transição.

Situação (S1) – Primeiro Estágio da ZDP – Assistência Externa (outra pessoa mais capacitada)

Ao escrever o primeiro caso de teste (06:57:04 – 07:32:39), um dos desenvolvedores não consegue avançar no desenvolvimento do código.

Acontecimento

Piloto (P3): “Ok, então...”

Mestre (P1): “Para o primeiro caso de teste, dê um número e ponha o que você espera.”

Situação (S2) – Segundo Estágio da ZDP – Auto-assistência

O participante explica como proceder para a escrita do primeiro caso de teste da tarefa (03:40:96 – 04:07:03).

Copiloto (P1): “O objetivo não pode ser modificado, então não temos como escrever métodos...”

Audiência (P3 - mesmo da S1): “Apenas precisamos de um método e continuar refatorando ele como TDD”

Discussão

Na primeira sessão, o **Mestre** percebeu uma oportunidade de intervir no **Nível de Conceito**. A **Intervenção Reativa do Mestre** foi motivada pela falta de conhecimento em TDD do participante, que não possuía habilidades suficientes para continuar sozinho.

Na segunda sessão, P3 (mesmo participante de ambas sessões) sugere que é necessário fazer um caso teste de único, onde se faz necessário adotar **Baby Steps**. Assim, percebe-se que P3 teve um avanço de desenvolvimento do primeiro estágio para o segundo estágio da ZDP, adquirindo **Auto-Assistência**.

O segundo estágio da ZDP, o da **Auto-assistência**, também foi particularmente evidenciado na S4 com a interação entre desenvolvedores mais experientes. Um exemplo é apresentado abaixo.

Situação

Na S4 (04:29:05 – 04:46:04), não há um desenvolvimento conciso de habilidades de como fazer TDD na linguagem Go.

Acontecimento

Piloto (P1): “Vamos ver como é o assert”

Copiloto (P2): “Coloca gol [indicando na busca] antes do assert”

Discussão

A **Pesquisa Externa** conduzida ilustra que os desenvolvedores já possuíam um conhecimento prévio sobre TDD. Entretanto, especialmente para a Linguagem Go, isto é, no **Nível de Programação** este conhecimento ainda não tinha sido desenvolvido. Este episódio se relaciona com o segundo estágio da ZDP.

8.9 Melhorias para CDR

Durante as sessões, foi possível identificar algumas melhorias apontadas pelos participantes. Essas melhorias emergiram durante a prática e, algumas vezes, não apareceram durante a retrospectiva. **Evitar uma tarefa conhecida pelos participantes** apareceu em um episódio, no qual um dos participantes já conhecia a tarefa da sessão. Um **BoilerPlate**, isto é, uma estrutura de código já existente ajudaria no fluxo de uma sessão de CDR. Outro conceito identificado foi o de **Infraestrutura Adequada**, como um projetor de boa qualidade, um teclado onde os membros se sentem à vontade para usar etc. Um exemplo é apresentado abaixo.

Situação

Na sessão S4 (15:27:03 – 15:34:11), o P1 no papel de piloto indica uma dificuldade ao usar o laptop de outro participante.

Acontecimento

Piloto (P1): “Usar o computador dos outros é complicado, não consigo acertar os atalhos.”

Copiloto (P2): “Também não gosto.”

Discussão

Neste episódio, percebeu-se que uma **Infraestrutura Adequada** se faz importante para a dinâmica da prática, especialmente um conjunto de atalhos que seja entendível para todos os participantes.

8.10 Retrospectiva

Durante as retrospectivas, foi possível discutir a percepção dos participantes durante as sessões. O resultado das retrospectivas dos estudos de observação e de caso são apresentados abaixo.

8.10.1 Estudos de Observação

Na S1, os participantes relataram que a existência de um **BoilerPlate** ajudaria a iniciar a sessão de CDR. Além disso, foi discutido a respeito da complexidade das tarefas. Segundo os participantes, a terceira tarefa, por ser a mais difícil, demandaria mais tempo para execução. Em relação à quantidade de participantes em uma sessão de CDR, os participantes relataram que 3 participantes é um número aceitável e citaram, todavia, que em algumas situações o membro da audiência participava como um copiloto, de uma forma pouco coordenada. Os participantes também apontaram que os turnos estavam

bem restritos (ao completar 7 minutos, o time deveria mudar imediatamente e fazer o giro) e que deveria haver mais bom senso na mudança.

Na retrospectiva da S2, os participantes relataram que seria melhor se os papéis (piloto, copiloto e audiência) fossem forçados a serem cumpridos. Segundo o P1 “*É muito bom ter todos [os três participantes] com foco, eu acredito que quatro seria melhor.*” Os participantes ainda discutiram que seria melhor reservar 15 minutos de discussão sobre a tarefa antes de iniciar a sessão de CDR. Os participantes relataram que a primeira atividade era bem complexa. Todos concordaram que a prática de CDR é uma prática divertida.

Na última retrospectiva da S3, o *BoilerPlate* foi novamente citado como uma importante melhoria pra CDR, especialmente para os participantes sem experiência. Sobre esta sugestão, o P1 disse: “*Nós gastamos um tempo para implementar...talvez um caso de teste [existente no arquivo]...*”. A participação da audiência também foi citada como algo desafiador, o P4 relatou: “*Eu perguntei a mim mesmo se o copiloto e a audiência eram a mesma pessoa participando*”. Devido ao time ser maior, os participantes também relataram uma certa dificuldade em tomar decisões que pudessem resolver o problema.

8.10.2 Estudos de Caso

Na S4, os participantes também discutiram sobre o número de participantes, relatando que o número de participantes da S4 foi bom. O P4 relatou: “*Se for um número maior, teria que ter alguém para controlar.*” O P3 ainda disse: “*No Chile que eu estava em um [uma sessão] de Dojo tinha muita gente, e grande parte da audiência estava no celular.*” Sobre a infraestrutura, o P1, que organizou a sessão, disse: “*A montagem do ambiente é importante, porque senão perde muito tempo*”.

No segundo estudo de caso, os participantes discutiram bastante sobre o papel do facilitador na sessão de CDR. Algumas opiniões, como de P2, eram a favor de um papel de facilitador, bem como a de P1: “*Eu coloquei [o post it sobre o papel do Facilitador] mas não é pra ficar destravando toda hora, é mais pra dar uma sugestão...*”. Outros participantes tiveram uma postura contrária, como a de P7: “*O dojo é para time, então temos que se virar mesmo*”. E outros ainda uma postura mais ponderada, semelhante a de P10:

Tem que ver bem qual é o objetivo de se fazer o *Dojo*, se é a aprendizagem, aí pode ter como copiloto alguém que já sabe... agora se for para resolver

problema, aí tem que ser o time. Aí se for a aprendizagem de algo que temos que aprender para o time, aí mais silêncio. (Participante 10)

Ainda sobre o objetivo da S5, o P3 relatou: “*Eu acho que a gente fez muito sobre teste e pouco sobre JavaScript*”. Alguns pontos positivos sobre a CDR e a S5 foram citados como a troca de conhecimento, a diversão em praticar em grupo e a disponibilização do material e de um *BoilerPlate*.

8.11 Diferenças entre estudos de observação e de estudos de caso

Durante as sessões de estudos observacionais, os participantes tiveram um maior suporte do autor desta tese na organização da sessão de CDR. Além disso, de uma sessão para outra, pequenas melhorias foram realizadas com base no *feedback* dado nas retrospectivas. As melhorias para cada sessão dos estudos observacionais são descritas abaixo.

Da S1 para S2 os turnos entre os pares foram menos restritos, isto é, mesmo se passassem alguns segundos ou minutos, era esperado um tempo para os pares terminarem a ação deles. Após as duas primeiras sessões possuem poucos participantes, então na S3 experimentamos com 6 participantes.

A experiência em relação a CDR é uma das principais diferenças entre os estudos de observação e os estudos de caso. Enquanto no primeiro os participantes não tinham experiência em CDR, no segundo os participantes tinham um nível considerado médio de experiência.

Em relação à finalização das tarefas, na S1 o grupo concluiu os dois exercícios (*Fizz Buzz* e *Fibonacci*) e começou o terceiro (*Game of Life*), na S2 os participantes pararam no primeiro exercício (*Roman to Numerals*) e na S3 o grupo concluiu o *Fizz Buzz* (primeira e segunda etapa) e iniciou o segundo exercício (*Number in Words*). Na S4, o *Fizz Buzz* não foi concluído e na S5 os participantes pararam no segundo exercício proposto. Os resultados foram ao encontro ao que é reportado por Sato et al. [SAT08], que esclarecem que o objetivo de CDR não é finalizar a tarefa, mas contribuir com a aprendizagem. Desta forma, a finalização de mais ou menos tarefas não foi contabilizada como um critério.

Além disso, em relação à experiência de programação, percebe-se que em S1, S2 e S3 os participantes tinham uma experiência bem diversa, desde um mais iniciante até um com mais experiência. Em contrapartida, na S4 e S5 os participantes tinham um nível de experiência mais próximo uns dos outros. Desta forma, foi possível perceber que nos estudos observacionais o **Nível de Conceito** era mais eminente, sendo possível analisar

de forma atrativa a aprendizagem durante as sessões de CDR. Enquanto que em S4 e S5, o **Nível de Programação** e o **Nível de Tarefa** foram mais identificados.

8.12 Ameaças a validade do estudo

O resultado a partir da análise de GT tem como objetivo caracterizar e explicar as observações das sessões de CDR, porém estes resultados não podem ser completamente generalizados. Se as práticas de GT foram usadas de forma correta, o resultado será sempre correto. A lista de fenômenos ainda está incompleta, sendo necessárias mais sessões no ciclo da indústria, conforme a metodologia de Schull et al. [SCH01].

8.13 Considerações Finais do Capítulo

Neste capítulo foi possível avaliar uma abordagem alternativa (QP3) para caracterizar o treinamento sob o nível 2 do modelo de Kirkpatrick: ao analisar os desenvolvedores interagindo em uma sessão de CDR, por meio do uso da GT. Parte dos resultados dos estudos de observação deste estudo foram publicados em Estácio et al. [EST16].

As sessões aconteceram sob duas perspectivas, a primeira com estudos de observação e a segunda com estudos de caso. Nos estudos de caso, percebeu-se a importância de um facilitador, enquanto nos estudos de observação o próprio pesquisador organizou os detalhes da sessão.

Ambas as fases também respondem a QP4 deste trabalho, evidenciando características dos papéis que emergiram em CDR, assim como os diferentes tipos de colaboração e fenômeno identificados. Estes permitiram um maior entendimento da dinâmica de treinamento em CDR, e como isto pode observado.

9 FASE 5: UM CONJUNTO DE DIRETRIZES PARA APOIAR A ADOÇÃO DE CDR

Neste capítulo, apresenta-se um conjunto de diretrizes para adoção de CDR. Estas diretrizes foram desenvolvidas por meio da consolidação dos resultados das etapas 1 (Exploratória) e 2 (Avaliação) desta pesquisa. Durante estas etapas, extraiu-se um conjunto de lições aprendidas em episódios recorrentes de adoção de CDR. Estes episódios foram convertidos em diretrizes. Desta forma, este conjunto de diretrizes destina-se a praticantes de CDR, profissionais da indústria e para professores que desejam adotar a prática em seus respectivos treinamentos e cursos.

A Seção 9.1 apresenta o conjunto de diretrizes identificados. A Seção 9.2 lista um conjunto de oportunidades de pesquisa a partir das diretrizes. A Seção 9.3 explica como se procedeu a avaliação de CDR. A Seção 9.4 apresenta as limitações das diretrizes propostas, e a Seção 9.5 apresenta as considerações finais deste capítulo.

9.1 Diretrizes

Para facilitar o entendimento e o uso das diretrizes, adotou-se uma notação para descrição dos padrões. A notação foi inspirada na estrutura utilizada por Schimdt et al. [SCH00] para padrões arquiteturais e é dividida em quatro categorias: o nome da diretriz, o problema, o contexto, e a solução que se propõe resolver. O nome é sempre em negrito, e os outros itens em itálico. Apresenta-se abaixo as sete diretrizes identificadas.

#1 - Tempo de Rotação Flexíveis

Problema

O tempo de rotação em CDR é um fator importante, porém pouco avaliado na literatura específica. Este tempo pode implicar diretamente na aprendizagem colaborativa.

Contexto

Nos estudos de observação, especificamente no S1, observou-se que o tempo de rotação pode interferir diretamente no **Nível de Conceito** (o principal nível onde a aprendizagem colaborativa acontece).

Na literatura (etapa exploratória), Sato et al. [SAT08] recomendam por meio de sua experiência a utilização de 5 a 7 minutos. Pouco se discute sobre qual intervalo de tempo deve ser realizada a rotação e como ela deve ser conduzida. Além de quem deve decidir o tempo de rotação entre os pares.

Solução

O tempo de rotação entre os participantes foi avaliado nas diversas fases desta pesquisa. Apesar disso, não se obteve um consenso quanto ao tempo ideal. Durante o estudo de viabilidade 1 foi utilizado 10 minutos, enquanto nas demais fases (estudos de observação e estudos de caso), 7 minutos. Ambos intervalos se mostraram viáveis num primeiro momento. A principal descoberta em relação a esta diretriz consiste ao fato dele não ser estritamente levado à risca.

Durante os estudos de observação, observou-se que a melhor estratégia foi que ao finalizar o tempo de rotação, deve-se esperar a última ação entre piloto e copiloto. Desta forma, foi possível evitar nas sessões interrupções nos diferentes níveis de colaboração, resguardando a aprendizagem colaborativa.

#2 – Balanceando a quantidade de participantes

Problema

Não há uma definição clara sobre a quantidade de participantes de uma sessão CDR e sobre como isto pode afetar a aprendizagem colaborativa.

Contexto

Nos estudos existentes da literatura não é mencionada a quantidade de participantes necessária para condução de uma sessão de CDR. Durante os estudos de viabilidade, este número foi bem diverso, de 10 para 6 participantes, e até grupos de 4 (estudo de viabilidade 2). Nos estudos de observação, realizou-se sessões com um número pequeno de participantes como os percebidos na S1 e S2 (três participantes cada) podem influenciar no cumprimento dos papéis estabelecidos pela prática: piloto, copiloto e audiência.

Entretanto, sessões com muitos participantes como a S5 ou do estudo de viabilidade 1 mostraram que a distração, as conversas paralelas e a divergência de muitas opiniões podem comprometer o trabalho. Como este estudo possui natureza exploratória, ainda se faz necessário avaliar como o número de participantes pode influenciar CDR.

Solução

Sessões com mais de três participantes apresentaram um resultado mais satisfatório em relação a aprendizagem colaborativa, haja vista que os papéis em CDR são mais respeitados e a colaboração flui de uma maneira mais efetiva. Entretanto, sessões com mais de 10 participantes podem gerar dispersão e conversas paralelas que afetam a aprendizagem colaborativa.

#3 – Tarefa Incremental

Problema

A escolha da tarefa pode afetar a aprendizagem colaborativa entre os desenvolvedores e o fluxo de uma sessão de CDR.

Contexto

Os estudos existentes na literatura em CDR não apresentam característica quanto ao tipo de tarefa e a sua complexidade. Durante as sessões conduzidas neste estudo, quando o pesquisador teve mais interferência na sessão (estudo de viabilidade e estudos de observação), optou-se por exercícios clássicos de programação. Em um dos estudos de caso, observou-se que o facilitador optou também por uma atividade clássica, o *Fizz Buzz*. Enquanto na outra sessão de estudo de caso, os exercícios foram propostos pelo próprio facilitador.

Solução

Em todos os estudos conduzidos nesta tese, uma importante inferência relativa ao tipo de tarefa consiste na progressão de dificuldade. Para uma adoção de CDR é apropriado propor uma dificuldade em nível incremental, isto é, uma tarefa mais simples até chegar ao nível mais complexo.

#4 – Uma estrutura de código de partida (*BoilerPlate*)

Problema

Um novato tem dificuldade de participar de uma sessão de CDR devido ao seu baixo conhecimento prévio na tecnologia.

Contexto

Durante a condução de sessões de CDR, percebeu-se que em alguns momentos os novatos tinham dificuldade em desenvolver a dinâmica da prática. O conhecimento prévio limitado era um dos principais motivos que impediam que a aprendizagem colaborativa se desenvolvesse.

Solução

Para facilitar isto, a presença de um *BoilerPlate* (uma estrutura de código) se mostrou útil em uma sessão de estudo de caso, sendo apontada como melhoria nos estudos de observação pelos participantes. A S5 do estudo de caso possuía também um exemplo já pronto, o que facilitou que os participantes da sessão (principalmente os novatos), pudessem dar um ponto de partida.

#5 – Treinamento Inicial para Novatos

Problema

Muitas vezes, apenas uma leitura prévia não é o suficiente para apoiar os novatos a participarem de uma sessão de CDR. Não se faz necessário um conhecimento extenso sobre o assunto para se engajar em uma sessão, mas um conhecimento mínimo é necessário.

Contexto

Nos estudos de viabilidade 1 e 2, os treinamentos foram essenciais para suportar o andamento dos novatos. Nos estudos de observação (especificamente na S3), percebeu-se que ausência de um treinamento prévio pode dificultar a participação de um novato, tornando CDR não tão eficiente na aprendizagem. No S5, os participantes tiveram sessões prévias do tipo *hands-on* na tecnologia com o facilitador da sessão, o que colaborou o andamento da sessão.

Solução

Estabelecer um treinamento inicial na tecnologia seja em forma tradicional, como aulas expositivas ou em outras formas, como grupo de estudos.

#6 – A presença de um facilitador

Problema

CDR exige uma operacionalidade para fazer a dinâmica da prática acontecer.

Contexto

Na literatura (etapa exploratória – RSL), Heinonen et al. [HEI13] citam a importância do papel de um facilitador, que atuaria como um tutor para os alunos, além de organizar a sessão. Nos estudos executados nesta tese, o facilitador durante os estudos de viabilidade e de observação foi o próprio autor, tal fato pode representar uma limitação.

Entretanto, as sessões de estudos de caso possuíam um facilitador emergido dos próprios participantes da sessão de CDR, que preparou e conduziu o andamento da prática e tomou decisões referentes ao tempo de rotação e as tarefas utilizadas.

O papel do facilitador apoiou também referente a algumas dúvidas sobre a prática, como por exemplo se a audiência poderia ser participativa ou não. No estudo de caso, ficou mais claro em certos momentos que a figura do facilitador como aquele que dita as regras do CDR, por exemplo se o piloto poderia fazer uma pesquisa externa ou não.

Solução

Um dos membros do grupo deve assumir a facilitação de uma sessão de CDR, em comum acordo com o grupo. Este facilitador será responsável por aspectos como: tempo de rotação, regras de participação da audiência, duração de uma sessão, entre outras coisas.

#7 – Grupo refletindo junto sobre CDR

Problema

É necessário promover um feedback entre os participantes sobre a prática, para a melhoria contínua da aprendizagem colaborativa.

Contexto

Na literatura (etapa exploratória), Sato et al [SAT08] compartilham sobre a importância de realizar retrospectivas ao fim de uma sessão de CDR, no intuito de levantar lições aprendidas. No âmbito dos estudos empíricos desta tese, retrospectivas foram conduzidas nos estudos de observação e estudos de caso. As retrospectivas duraram em torno de 15 a 30 minutos. Nos estudos de observação, elas foram conduzidas pelo autor desta tese e por um dos participantes nos estudos de caso. Esta diretriz vai ao encontro da teoria da aprendizagem colaborativa, proposta por Vygostky [VYG78].

Solução

Uma técnica que pode ser utilizada para promover a reflexão sob a prática consiste na condução retrospectivas de 15 a 30 minutos. Um dos participantes da sessão deve facilitar a dinâmica.

9.2 Oportunidades de Pesquisa oriunda das diretrizes propostas

Durante a concepção destas diretrizes, foram identificadas algumas oportunidades de pesquisa para melhoria e expansão das diretrizes propostas. As oportunidades de pesquisa são apresentadas abaixo.

#1 – Existe uma oportunidade de avaliar CDR com outras práticas colaborativa de grupo

Os estudos conduzidos no âmbito desta tese investigaram os efeitos em CDR, e a comparação com PP (etapa exploratória). Outros estilos de CD podem ser interessantes para avaliar a possibilidade de melhorias destas diretrizes de CDR. Desta forma, há a necessidade de se entender que diretrizes são comuns para outras práticas de grupos, e que diretrizes possuem uma especificidade maior.

#2 – Existe uma oportunidade de avaliar questões relativas à gênero em CDR

Na literatura de PP, alguns estudos [WER13][CHO13] buscaram analisar a relação da prática com o gênero, apresentando que a prática pode ajudar mulheres a se engajarem nos cursos de programação. Poucas mulheres participaram das sessões de CDR que foram conduzidas neste estudo, havendo a oportunidade de executar mais estudos com mulheres para compreender questões de gênero em CDR.

#3 – A cultura é uma variável que pode influenciar CDR

Durante a condução dessa pesquisa, foram executados três estudos em outro país, a Alemanha. E um destes estudos envolveu a participação de estudantes brasileiros e alemães. Esta pesquisa não tinha por objetivo analisar a influência da cultura dos participantes na prática de CDR, apesar deste aspecto ter se mostrado uma oportunidade de pesquisa.

#4 – A personalidade entre os participantes pode influenciar na aprendizagem e motivação

A personalidade foi amplamente investigada em PP, conforme apresentado no capítulo 4. Em CDR, pouco se sabe como os diferentes tipos de personalidade podem influenciar na característica de aprendizagem colaborativa da prática. Especificamente, os papéis emergidos pela GT, como **Mestre** e **Novato**, e a relação deles com traços de personalidade.

#5 – Técnicas alternativas de rotação devem ser investigadas

A técnica de rotação dos pares geralmente segue um giro entre todos os participantes. Nos estudos conduzidos no presente trabalho, em um segundo turno de rotação, os pares se repetem. Por exemplo: se P1 foi o piloto e P2 o copiloto, em um segundo momento esta formação irá se repetir. Desta forma, existe a oportunidade de investigar novas técnicas de rotação e como estas podem influenciar na dinâmica de CDR.

#6 – A voz da audiência em CDR

A participação da audiência em alguns episódios se mostrou um importante aliada durante resolução de alguns problemas. Entretanto, mais estudos devem ser conduzidos para analisar a efetividade da participação da audiência e como ela pode ser coordenada. Nos estudos conduzidos nesta pesquisa, a audiência teve uma coordenação mínima, deixada a cargo dos próprios participantes.

9.3 Avaliação das Diretrizes

A própria concepção destes diretrizes, por meio da metodologia de Shull et al. [SCH01], representou uma primeira etapa de avaliação do conjunto de diretrizes. Esta metodologia permitiu a coleta de evidências por meio de diferentes estudos empíricos, em diferentes cenários. Desde um desenvolvedor menos experiente até um caso na indústria, com desenvolvedores mais experientes.

Ainda se faz necessário a última etapa deste ciclo, ao avaliar este conjunto em um estudo a longo prazo na indústria. Esta etapa implicaria em possíveis melhorias para esta

versão inicial e no entendimento de como CDR pode ser transferido para a indústria, como uma prática que seja parte do ciclo de desenvolvimento de software.

9.4 Limitações das Diretrizes

A proposta dos diretrizes para CDR se encontra em estágio inicial. Apesar de passar por duas etapas (exploratória e avaliação) nesta pesquisa, ainda se faz necessário analisar a transferência destas diretrizes para o ambiente da indústria, de forma a observar os efeitos em longo prazo. Além disso, esta pesquisa possui foco qualitativo, havendo a necessidade ainda de investigar estas diretrizes sob uma perspectiva quantitativa.

O processo de concepção das diretrizes foi realizado sempre com um par como revisor, de forma a minimizar o viés do pesquisador desta tese. A consolidação dos resultados, sempre que possível, foi oriunda de uma análise dos resultados dos estudos empíricos e dos estudos da literatura (RSL), como forma de ratificar a importância de cada componente proposto.

9.5 Considerações finais do capítulo

Este capítulo apresentou um conjunto de diretrizes a partir da consolidação dos resultados desta pesquisa. O objetivo deste conjunto é facilitar a adoção da prática, evidenciando o foco na aprendizagem colaborativa.

Ainda, identificou-se seis oportunidades de pesquisa para expansão ou melhoria do conjunto inicial de diretrizes. Apesar de se tratar de uma proposta ainda em um estágio inicial, as diretrizes foram extraídas durante as etapas da metodologia desta pesquisa.

As diretrizes complementam as fases 3 e 4, ao responderem a QP4 desta tese. Cada diretriz caracteriza um contexto de adoção de CDR, levando em conta a aprendizagem colaborativa inerente a prática.

10 CONCLUSÃO

Este capítulo resume os principais resultados desta tese, além de apresentar os trabalhos futuros. Na Seção 10.1 é apresentado um resumo dos resultados alinhado aos objetivos desta pesquisa. A Seção 10.2 apresenta as principais contribuições desta tese. A Seção 10.3 apresenta os trabalhos futuros, e por fim a Seção 10.4 a reflexão final.

10.1 Resumos dos resultados e objetivos

A principal característica de CDR é a aprendizagem [ROO14]. Por ser uma prática que envolve mais de um desenvolvedor, é de natureza colaborativa. Entender e avaliar a perspectiva de aprendizagem colaborativa de CDR implica em entender como esta prática pode ser adotada de forma mais efetiva pela indústria.

Desta forma, o objetivo primário desta tese foi de avaliar a aprendizagem colaborativa de CDR. Este objetivo foi alcançado por meio da condução de uma extensiva metodologia de pesquisa, que incluía estudos secundários (condução de uma RSL) e estudos primários (experimentos controlados, estudos observacionais).

Ao se analisar os objetivos específicos, o primeiro objetivo (caracterização da adoção das principais características de práticas de codificação colaborativa) foi alcançado por meio da RSL apresentada no Capítulo 4. Os resultados permitiram uma caracterização destas práticas de codificação colaborativa em relação às principais evidências investigadas no ensino e treinamento de programação, as avaliações utilizadas nestes estudos e o contexto (tipos de cursos e treinamentos) onde foram aplicadas.

Os Capítulos 5 e 6 apresentam dois estudos de viabilidade sobre CDR, neles foi possível alcançar o segundo objetivo desta tese (identificar os benefícios e desvantagens de CDR de acordo com a percepção dos participantes), ao observar os principais benefícios e desvantagens de CDR. Ainda nesse capítulos, analisou-se o terceiro objetivo (identificar as estratégias de avaliar a aprendizagem em CDR) desta pesquisa, em uma primeira avaliação da aprendizagem de colaborativa em CDR. Esta primeira estratégia de avaliação consistia em analisar a percepção dos participantes em relação à CDR. Os estudos de viabilidade apresentaram que CDR tinha resultados favoráveis a aprendizagem, porém os valores se mostraram menos favoráveis que os de PP.

Desta forma, a partir do Capítulo 7 foram conduzidos estudos na etapa de avaliação, com o objetivo de avaliar CDR para além da viabilidade. Nesta segunda etapa, o terceiro objetivo foi alcançado e os resultados são apresentado no Capítulo 8. Nesta etapa, a estratégia para utilizar CDR foi analisar a interação durante as sessões de CDR,

tanto em estudos observacionais quanto em estudos de caso. Ainda nesta etapa, foi possível identificar diferentes fenômenos na prática, e a analisar a aprendizagem colaborativa em CDR por meio de diferentes conceitos.

O último objetivo desta tese (propor um conjunto de diretrizes que apoie a adoção de CDR com foco na aprendizagem colaborativa) foi apresentado no Capítulo 9, onde foi apresentado diretrizes desenvolvidas a partir da *consolidação* dos resultados desta tese.

Esta tese possui como questão norteadora (QN) de pesquisa “Como se pode avaliar a aprendizagem colaborativa durante a execução da prática de CDR no contexto de desenvolvimento de software? ”. Esta questão foi respondida pela execução da metodologia proposta nesta pesquisa, e pela consolidação dos resultados por meio do desenvolvimento de um conjunto de diretrizes para CDR.

10.2 Principais Contribuições

Esta tese possui quatro contribuições importantes, tanto para o estado da arte da Engenharia de Software (ES), quanto para o estado da prática. As contribuições são listadas e explicadas abaixo:

C1. Caracterização e Oportunidades de Pesquisa em Práticas de Codificação Colaborativa.

Por meio desta pesquisa, mapeou-se e analisou-se as principais evidências em práticas de codificação colaborativa na literatura, especificamente em cursos de computação e treinamentos. A C1 está relacionada com o primeiro objetivo específico desta pesquisa (OE1). Desta forma, lacunas no estado de arte e oportunidades de pesquisa foram levantadas. Esta caracterização pode servir como referência para novos estudos empíricos não somente em CDR, mas em outras práticas de codificação colaborativa.

C2. Identificação e aplicação de um conjunto de métodos para avaliar aprendizagem colaborativa em CDR.

Neste estudo, a metodologia de avaliação de jogos proposta por Savi [SAV11] foi instanciada como forma de avaliar a aprendizagem colaborativa. O modelo de Kirkpatrick, componente desta metodologia, orientou os níveis de avaliação (os dois utilizados neste trabalho, sendo o 1 de reação e o 2 de aprendizagem). A taxonomia de Bloom (também utilizada por Savi [SAV11] na sua metodologia) foi utilizada nas etapas exploratórias para avaliar a aprendizagem por competências técnicas específicas. Na etapa de Avaliação, utilizou-se GT para avaliar a dinâmica da aprendizagem colaborativa entre os desenvolvedores. A C2 contempla o segundo objetivo específico (OE2) desta tese.

C3. Identificação de um conjunto de benefícios e desvantagens de CDR.

Outra contribuição deste estudo foi de analisar a viabilidade de CDR quanto à sua adoção. Na fase exploratória desta pesquisa, analisou-se a percepção dos participantes. Em relação à CDR e também à outra prática de codificação colaborativa, a PP. Assim, um conjunto de benefícios e desafios foram coletados pelos participantes, não apenas na fase exploratória, mas na etapa de avaliação. Esta contribuição apresenta uma importante caracterização dos potenciais de CDR, e as suas respectivas limitações para adoção. A C3 vai ao encontro do terceiro objetivo específico (OE3) deste estudo.

C4. A consolidação dos resultados por meio da concepção de um conjunto de diretrizes para a adoção de CDR com foco na aprendizagem colaborativa.

A partir da execução do ciclo de Schull [SCH11], este estudo consolidou os resultados por meio de um conjunto de diretrizes para CDR. Com esta contribuição, espera-se facilitar a adoção de CDR por meio de um entendimento de como se pode avaliar a prática quanto a sua principal característica, a aprendizagem colaborativa. O quarto objetivo específico (OE4) desta tese é alcançado por meio da C4. Espera-se que estas diretrizes, ao contribuir para o estado da arte, igualmente apoiem a adoção de CDR por profissionais da indústria e aprimorem o estado da prática.

10.3 Trabalhos Futuros

Por meio deste estudo, percebeu-se a existência de várias oportunidades de pesquisa relativas à condução de estudos empíricos em CDR. Esta linha de pesquisa se faz importante para a indústria e academia. Desta forma, as seguintes pesquisas futuras são sugeridas:

- Continuação do Ciclo de Shull com a execução de Estudos na Indústria e aplicação do conjunto de diretrizes proposto. Pretende-se avaliar CDR como programa de treinamento em uma organização em um estudo de caso longitudinal.
- Explorar CDR no contexto de sala de aula e suas implicações em diferentes cursos de desenvolvimento de software. Além disso, avaliar se as diretrizes identificadas nesta tese possuem especificidades ou limitações em relação aos diferentes tópicos, bem como os diferentes níveis de experiência entre os estudantes.
- Avaliação de novos estilos de CD e de Mob Programming que já são adotados na indústria com o objetivo de entender a comparação de CDR com outros estilos de programação em grupo.

- A investigação da relação de Comunidade de Prática [WEN00] e CDR, com o objetivo de analisar possíveis peculiaridades de se adotar CDR não só por equipes da indústria (o âmbito deste trabalho), mas de uma comunidade de software.
- A investigação de CDR em outros contextos de desenvolvimento de software, como no de Ecossistemas de Software. Existe uma parceria inicial neste sentido entre PUCRS, UFAM e UNIRIO. O principal objetivo é avaliar o papel de CDR como uma prática de treinamento por evangelistas para desenvolvedores de plataformas mobile (como Google, Apple e Microsoft).

10.4 Reflexão Final

Práticas de codificação colaborativa exercem um papel importante em relação à característica social do desenvolvimento de softwares. A execução desta pesquisa envolveu um esforço na condução de estudos empíricos tanto na academia quanto na indústria, com o objetivo de entender a aprendizagem colaborativa desta prática.

A concepção do conjunto de diretrizes é uma primeira tentativa de expandir o estado da prática de CDR. Do ponto de vista da indústria, o resultado desta pesquisa é um importante guia para adoção de CDR por profissionais, e do ponto de vista acadêmico pode auxiliar no âmbito da avaliação do processo de aprendizagem colaborativa em CDR.

Espera-se que as diretrizes propostas possam ser incorporadas em um ciclo de desenvolvimento de software na indústria. E que possa ser evoluído constantemente, à medida que a avaliação de CDR se expanda, para além de novos contextos de desenvolvimento de software.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ACM13] ACM/IEEE. Computer Science Curricula. "Guidelines for Undergraduate Degree Programs in Computer Science", Final Report, ACM, 2013, 514p.
- [ARO12] Arora, R.; Goel, S. "Learning to Write Programs with Others: Collaborative Quadruple Programming". In: 25th Conference on Software Engineering Education and Training, 2012, pp.32-41.
- [BAS94] Basili, V.; Rombach, D.; Caldiera, G. Goal Question Metric Paradigm, Encyclopedia of Software Engineering. New York: John Wiley & Sons, 1994, 1929p.
- [BAS08] Bassi, D. Experiências com Desenvolvimento Ágil", Dissertação de Mestrado, Programação de Pós-Graduação em Ciência da Computação, IME-USP, 2008, 170p.
- [BEC01] Beck, K, et al. "Manifesto for Agile Software Development". Capturado em: www.manifestoagil.com.br, Agosto 2016.
- [BEC04] Beck, K.; Andres, C. "Extreme Programming Explained: Embrace Change". Boston: Addison-Wesley Professional, 2004, 224p.
- [BIO05] Biolchini, J.; Mian, P.G.; Natali, A.; Travassos, G. "Systematic review in software engineering", Relatório Técnico RT-ES 679/05, COPPE/UFRJ, 2005, 30p.
- [BIP08] Bipp, T.; Lepper, A.; Schmedding, D. "Pair programming in software development teams - An empirical study of its benefits". *Information and Software Technology*, vol. 50-3, 2008, pp. 231 - 240.
- [BLO56] Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; Krathwohl, D. R. "Taxonomy of educational objectives: The classification of educational goals". New York: David McKay Company, 1956, 207p.
- [BOS05] Bossavit, L.; Gaillot, E.: "The Coder ' s Dojo - A Different Way to Teach and Learn Programming". In: 6th International Conference on Agile Software Development, 2005, pp. 290 - 291.
- [BRA10] Braught, G.; MacCormick, J.; Wahls, T. "The benefits of pairing by ability". In: 41st ACM Technical Symposium on Computer Science Education, 2010, pp. 249 - 253.
- [BRA11] Braught, G.; Wahls, T.; Eby, L. "The case for pair programming in the computer science classroom". *ACM Transactions on Computing Education*,

vol.11-1, 201, 21p.

- [BRI15] Britto, R.; Usman, M. "Bloom's taxonomy in software engineering: A systematic mapping". In: 45th Frontiers in Education Conference, 2015, pp. 1 - 8.
- [CHO07] Chong, J.; Hurlbutt, T. "The Social Dynamics of Pair Programming". In: International Conference on Software Engineering, 2007, pp. 354-353.
- [CHO13] Choi, K. "Evaluating Gender Significance within a Pair Programming Context." In: 46th Hawaii International Conference on System Sciences, 2013, pp. 4817-4825.
- [CHO08] Choi, K.S.; Deek, F. P; Im, I. "Exploring the underlying aspects of pair programming: The impact of personality". *Information and Software Technology*, vol. 50-11, 2008, pp. 1114 -1126.
- [COM14] Coman, P; Robillard, N.; Sillitti, A.; Succi, G. "Cooperation, collaboration and pair programming: Field studies on backup behavior". *Journal of Systems and Software*, vol. 91-1, pp. 124 -134.
- [DYB08] Dyba, T.; Dingsøyr, T. "Empirical studies of agile software development: A systematic review". *Information Software Technology Journal*, vol. 50-10, 2008, pp. 833 - 859.
- [DYB09] Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F." The effectiveness of pair programming: A meta-analysis". *Information Software and Technology*, vol. 51-7, 2009, pp. 1110-1122.
- [EST14] Estácio, B.; Prikladnicki, R.; Mora, M.; Notari, G.; Caroli, P.; Olchik, A. "Software Kaizen: Using Agile to Form High-Performance Software Development Teams". In: Agile Conference, 2014, 10p.
- [EST15] Estácio, B., Oliveira, R., Marczak, S., Kalinowski, M., Garcia, A., Prikladnicki, R.; Lucena, C. "Evaluating Collaborative Practices in Acquiring Programming Skills: Findings of a Controlled Experiment". In: 29th Brazilian Symposium on Software Engineering, 2015, pp. 150 - 159.
- [EST15b] Estácio, B.; Valentim, N.; Rivero, L.; Conte, T. U. ; Prikladnicki, R. "Evaluating the Use of Pair Programming and Coding Dojo in the Teaching of Mockup Development: An Empirical Study". In: 48th Hawaii International Conference on System Sciences, 2015, pp. 5084 - 5093.
- [EST16] Estácio, B., Zieris, F.; Prechelt, L.; Prikladnicki, R. On the randori training dynamics". In: Cooperative and Human Aspects of Software Engineering

Workshop, 2016. p. 44 - 47.

- [ERI93] Ericsson, K A.; Krampe, R.T; Tesch-Römer, C. "The role of deliberate practice in the acquisition of expert performance". *Psychological Review*, vol.100-3, 1993, pp. 363 - 406.
- [FLO13] Flora, H.; Chande, S. "A Review and Analysis on Mobile Application Development Processes using Agile Methodologies". *International Journal of Research in Computer Science*, vol. 3-4, 2013, pp. 9–18.
- [GAR13] Garcia, L. "Metodologia Para Implementação De Estratégia Colaborativas Mediadas por Ferramentas de Interação Síncronas. Dissertação de Mestrado, Programa de Pós Graduação em Informática, UFPR, 2013, 96p.
- [HAN06] Hanks, B. "Student attitudes toward pair programming". In: 37th ACM Technical Symposium on Computer Science Education, 2006, pp.113 -117.
- [HEI13] Heinonen, K.; Hirvikoski, K; Luukkainen, M.; Vihavainen, A. "Learning agile software engineering practices using coding dojo". In: 14th Conference on Information Technology Education, 2013, pp. 97-102.
- [HUL05] Hulkko, H.; Abrahamsson, P. "A multiple case study on the impact of pair programming on product quality". In: 27th International Conference on Software Engineering, 2005, pp. 495 - 504.
- [KAV13] Kavitha, R. K.; Ahmed, M. S. I. "Knowledge sharing through pair programming in learning environments: An empirical study". *Education and Information Technologies*, vol. 20- 2, 2013, pp. 319–333.
- [KIR94] Kirkpatrick, D. L. *Evaluating Training Programs - The Four Levels*. San Francisco: Berrett-Koehler Publishers, 1994, 229p.
- [KIL14] Kilamo, T.; Nieminen, A.; Lautamäki, J.; Aho, T.; Koskinen, J.; Palviainen, J.; Mikkonen, T. Knowledge transfer in collaborative teams: Experiences from a "two-week code camp". In: 36th International Conference on Software Engineering - ICSE Companion, 2014, pp. 264- 271.
- [KIT07] Kitchenham, B.; Charters, S. "Guidelines for performing Systematic Literature Reviews in Software Engineering", Relatório Técnico, Keele University and Durham University, 2007, 65p.
- [KUM16] Kumar, Shreya. "Communication Patterns and Strategies in Software Development Communities of Practice". Tese de Doutorado, Michigan Technological University, 2016, 215p.
- [LEW15] Lewis, C. M.; Shah, N. "How Equity and Inequity Can Emerge in Pair

Programming". In: 11th International Conference on International Computing Education Research, 2015, pp. 41-50.

- [LI13] Li, Z.; Plaue, C.; Kraemer, E. "A spirit of camaraderie: The impact of pair programming on retention". In: 26th Software Engineering Education and Training, 2013, pp. 209-218.
- [LUZ13] Luz, R.B.; Neto, A.G.; Noronha, R. "Teaching TDD, the Coding Dojo Style". In: 13th International Conference Advanced Learning Technologies, 2013, pp. 371 - 375.
- [MAD07] Madeyski, L. "On the Effects of Pair Programming on Thoroughness and Fault-Finding Effectiveness of Unit Tests". In: 8th Product-Focused Software Process Improvement, 2007, pp. 207 - 221.
- [MAF06] Mafra, S. N.; Barcelos, R. F.; Travassos, G. H. "Aplicando uma Metodologia Baseada em Evidencia na Definição de Novas Tecnologias de Software". In: 20th Simpósio Brasileiro de Engenharia de Software, 2006, pp. 239-254.
- [MCC16] Mcchesney, I. "Three Years of Student Pair Programming". In: 47th ACM Technical Symposium on Computing Science Education, 2016, pp. 84-89.
- [MCD03] Mcdowell, C.; Werner, L.; Bullock, H.; Fernald, J. C. Mcdowell, L. "The impact of pair programming on student performance, perception and persistence". In: 25th International Conference on Software Engineering, 2003, pp.143-152.
- [MEN05] Mendes, E.; Basil, L.; Fakhri, A.; Reilly, A. "Investigating pair-programming in a 2-year software development and design computer science course". In: 10th Innovation and Technology in Computer Science Education, 2005, pp. 296 - 300.
- [NAW16] Nawahdah, M.; Taji, D. "Investigating students' behavior and code quality when applying pair-programming as a teaching technique in a Middle Eastern society". In: 7th Global Engineering Education Conference, 2016, pp. 32-39.
- [NOS98] Nosek, J. "The case for collaborative programming". *ACM Communication*, vol. 41- 3, 1998, pp. 105-108.
- [OLI16] Oliveira, R.; Estácio, B.; Garcia, A.; Marczak, S.; Prikladnicki, R.; Kalinowski, M.; Lucena, C. "Identifying Code Smells with Collaborative Practices: A Controlled Experiment". In: 10th Brazilian Symposium on Software Components, Architectures and Reuse, pp. 61-70, 2016.
- [OXF17] Oxford Dictionary Online. Capturado em: <https://en.oxforddictionaries.com/>, Março de 2017.

- [PRE05] Preece, J.; Rogers, Y.; Sharp, H. Design de interação: além da interação homem-computador. Porto Alegre, RS: Bookman, 2005, p. 548.
- [PRE11] Pressman, R. "Engenharia de software: uma abordagem profissional". Porto Alegre: Grupo A, 2011, 880p.
- [RON12] Rong, G.; Zhang, H.; Xie, M; Shao, D. "Improving PSP education by pairing: An empirical study".In: 34th International Conference on Software Engineering, 2012, pp.1245-1254.
- [ROO14] Rooksby, J.; Hunt, J.; Wang, X. "The Theory and Practice of Randori Coding Dojos". In: 15th International Conference on Agile Software Development, 2014, pp 251- 259.
- [SAL10] Salleh, N.; Mendes, E.; Grundy, J.; Burch, G. "An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model". In: 32nd International Conference on Software Engineering, 2010, pp. 577-586.
- [SAL11a] Salleh, N.; Mendes, E.; Grundy, J. "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review". *IEEE Transactions on Software Engineering*, vol.37-4, pp. 509 - 525.
- [SAL11b] Salleh, N.; Mendes, E.; Grundy, J. "The effects of openness to experience on pair programming in a higher education context". In: 24th Software Engineering Education and Training, 2011, pp.149 - 158.
- [SAL12] Salleh, N.; Mendes, E.; Grundy, J. "Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments". *Empirical Software Engineering*, vol. 19-3, 2012, pp. 714 -752.
- [SAN08] Santos, R.; M.; Werner, C.; Travassos, G. "Utilizando experimentação para apoiar a pesquisa em educação em engenharia de software no Brasil". In: Fórum de Educação em Engenharia de Software, 2008, p.10.
- [SAV11] Savi, R. "Avaliação de Jogos Voltados para a Disseminação do Conhecimento". Tese de Doutorado, Programa de Pós-graduação em Engenharia e Gestão do Conhecimento, UFSC, 2011, 236p.
- [SAT08] Sato, D.T; Corbucci, H.; Bravo, M. "Coding Dojo: An Environment for Learning and Sharing Agile Practices". In: Agile Conference, 2008, pp.459 - 464.
- [SCH84] Schön, D.A. The Reflective Practitioner: How Professionals Think in Action. New York: Basic Books, 1984, 384p.

- [SCH00] Schmidt, D. C., Stal, M., Rohnert, H. and Buschmann, F. "Pattern-Oriented Software Architecture, v2: Patterns for Concurrent and Networked Objects". New York: Wiley, 20, 666pp.
- [SCH14] Schmidt, C.T.; Venkatesha, S. G.; Heymann, J. "Empirical insights into the perceived benefits of agile software engineering practices: a case study from SAP". In: 36th International Conference on Software Engineering, 2014, pp. 84 - 92.
- [SEY15] Seyam, M.; Mccrickard, S. "Collaborating on mobile app design through pair programming: A practice-oriented approach overview and expert review". In: International Conference on Collaboration Technologies and Systems, 2015, pp.124-131.
- [SEY16] Seyam, M.; Mccrickard, S. "Teaching Mobile Development with Pair Programming". In: 47th ACM Technical Symposium on Computing Science Education, 2016, pp 96-101.
- [SFE08] Sfetsos, P.; Stamelos, I.; Angelis, L.; Deligiannis, I. "An experimental investigation of personality types impact on pair effectiveness in pair programming". *Empirical Software Engineering*, vol. 14- 2, 2008, pp. 187-226.
- [SFE12] Sfetsos, P.; Adamidis, P.; Angelis, L.; Stamelos, I.; Deligiannis, I., "Investigating the Impact of Personality and Temperament Traits on Pair Programming: A Controlled Experiment Replication". In: 8th Quality of Information and Communications Technology, 2012, pp.57,65.
- [SHU01] Shull F., Carver, J.; Travassos, G. In: "An empirical methodology for introducing software processes ". In: 8th European Software Engineering Conference, 2001, pp. 288-296.
- [SOM11] Sommerville, I. "Software Engineering". São Paulo: Pearson Addison-Wesley, 2011, 729p.
- [STO12] Storey, M. "The evolution of the social programmer". In: 9th Mining Software Repositories, 2012, pp. 140 – 140.
- [SIM08] Simon, B.; Hanks, B. "First-year students' impressions of pair programming in CS1". *Journal of Education Resources in Computing*, vol. 7- 4, 2008, 28p.
- [SIS08] Sison, R. "Investigating Pair Programming in a Software Engineering Course in an Asian Setting". In: 15th Asian Pacific Software Engineering Conference, 2008, pp.325 - 331.

- [THA88] Tharp, R. G.; Gallimore, R. Rousing mind to life: teaching, learning, and schooling in social context. Cambridge: Cambridge University Press, 1988, p. 332.
- [VAN07] Vanhanen, J.; Abrahamsson, P. "Perceived Effects of Pair Programming in an Industrial Context". In: 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, p.8.
- [VYG78] Vygotsky, L. Mind in Society: The Development of Higher Psychological Processes. Cambridge: Harvard University Press, 1978, 159p.
- [URN09] Urness, T. "Assessment using peer evaluations, random pair assignment, and collaborative programming in CS1". *Journal of Computer Science College*, vol. 25 - 1, 2009, pp.87-93.
- [WEN00] Wenger, E. "Communities of practice and social learning systems. *Organization*, vol. 7- 2, pp. 225–246, 2000.
- [WER13] Werner, L.; Denner, J.; Campe, S.; Ortiz, E.; DeLay, D.; Hartl, A.; Laursen., B. "Pair programming for middle school students: does friendship influence academic outcomes?". In: 44th ACM Technical symposium on Computer science education, pp. 421-426.
- [WIL03] Williams, L.; Mcdowell, C.; Nagappan, N.; Fernald, J.; Werner, L. "Building pair programming knowledge through a family of experiments". In: 2nd International Symposium on Empirical Software Engineering, 2003, pp. 1413-152.
- [WOH00] Wohlin, C et. al. "Experimentation in Software Engineering: An Introduction". Norwell: Kluwer Academic Publishers, 2000, 236p.
- [WOO13] Wood, K.; Parsons, D.; Gasson, J.; Haden, P. "It's never too early: Pair programming in cs1". In: 15th Australasian Computing Education Conference, 2013, pp. 13–21.
- [ZAC09] Zacharis, N. "Evaluating the Effects of Virtual Pair Programming on Students' Achievement and Satisfaction". *International Journal of Emerging Technologies in Learning (iJET)*, vol. 4- 3, 2009, pp. 34 - 39.
- [ZAR14] Zarb, M.; Hughes, J.; Richards, J. "Evaluating industry-inspired pair programming communication guidelines with undergraduate students". In: 45th ACM Technical Symposium on Computer Science Education, 2014, pp. 361-366.
- [ZAR15] Zarb, M.; Hughes, J.; Richards, J. "Further Evaluations of Industry-Inspired

Pair Programming Communication Guidelines with Undergraduate Students”.
In: 46th ACM Technical Symposium on Computer Science Education, 2015,
pp. 314-319.

APÊNDICE A

Protocolo Inicial da Revisão Sistemática da Literatura (Em inglês)

Empirical evaluation of collaborative programming practices in CS/SE courses and software training

1. Research question

Our systematic review intends to answer the following research questions:

- What main evidences are there about collaborative practices in CS/SE courses?
- What are the assessment strategies that have been used to assess the effectiveness of these collaborative methods?
- Which configurations are adopted in courses and training to adopt collaborative methods?

2. Question structure

In order to structure the research question, we defined the scope of this SLR following:

Population: Computer Software Education/ Software Engineering Education

Intervention: Collaborative programming methods

Outcome: Collaborative methods of teaching programming and then, their metrics, challenges and benefits identified.

Study design: Empirical studies

Comparison: -

3. Identifying Relevant Literature

We are going to execute the following search string:

Population	Intervention	Outcomes
Computer Software Education and Training	Collaborative programming	Method
Software Engineering Education	Pair programming	Course

Computer Software Higher Education	Group programming	Practice
Computer Science	Coding Dojo	Programming
	Cooperative Programming	Coding

P and I and O

("Training" OR "Computer Higher Education" OR "Software Engineering" OR "Software") AND ("Collaborative Programming" OR "Pair Programming" OR "Group Programming" OR "Coding Dojo" OR "Cooperative programming") AND (" Course" OR "Courses" OR "Programming ")

4. Data source and search strategy

The search strategy will include electronic databases. We are going to search in following electronic databases:

- IEEEXplore
- ACM Digital Library
- Scopus
- Science direct
- Willey

In addition, we are going to hand-search in the following proceedings: International Conference on Agile Software Development (XP)

5. Search Strategy

1. To identify relevant studies-search in electronic databases;
2. To exclude studies on the basis of titles, abstracts and keywords;
3. To exclude studies on the basis of introduction and conclusion;
4. To exclude studies on the basis of full reading;
5. To obtain primary papers and critically appraise studies.

6. Inclusion criteria

We are going to consider in the SLR papers following the criterion:

- Empirical studies of collaborative programming that involved students or professionals in programming courses or training and passed in quality assessment.

7. Exclusion criteria

We are not going to consider in the SLR papers following the criteria:

- Opinion papers, presenting only the point of view of the researcher, without evidence to support the claim.
- Papers without the perspective of collaborative programming practices.
- Paper not written in English.

APÊNDICE B

Estudos Primários Seleccionados da Revisão Sistemática da Literatura

1. Bipp, T.; Lepper, A.; Schmedding, D. "Pair programming in software development teams - An empirical study of its benefits". *Information and Software Technology*, vol. 50-3, pp. 231 - 240.
2. Braught, G.; MacCormick, J.; Wahls, T. "The benefits of pairing by ability". In: 41st ACM Technical Symposium on Computer Science Education, 2010, pp. 249 - 253.
3. Braught, G.; Wahls, T.; Eby, L. "The case for pair programming in the computer science classroom". *ACM Transactions on Computing Education*, vol.11-1, 2011, 21p.
4. Choi, K.S.; Deek, F. P; Im, I. "Exploring the underlying aspects of pair programming: The impact of personality". *Information and Software Technology*, vol. 50-11, pp. 1114–1126.
5. Coman, P; Robillard, N.; Sillitti, A.; Succi, G. "Cooperation, collaboration and pair programming: Field studies on backup behavior". *Journal of Systems and Software*, vol. 91-1, pp. 124 -134.
6. Estácio, B., Oliveira, R., Marczak, S., Kalinowski, M., Garcia, A., Prikladnicki, R.; Lucena, C. "Evaluating Collaborative Practices in Acquiring Programming Skills: Findings of a Controlled Experiment". In: 29th Brazilian Symposium on Software Engineering, 2015, pp. 150 - 159.
7. Heinonen, K.; Hirvikoski, K; Luukkainen, M.; Vihavainen, A. "Learning agile software engineering practices using coding dojo". In: 14th Conference on Information Technology Education, 2013, pp. 97-102.
8. Kavitha, R. K.; Ahmed, M. S. I. "Knowledge sharing through pair programming in learning environments: An empirical study". *Education and Information Technologies*, vol. 20- 2, pp. 319–333.
9. Kilamo, T.; Nieminen, A.; Lautamäki, J.; Aho, T.; Koskinen, J.; Palviainen, J.; Mikkonen, T. Knowledge transfer in collaborative teams: Experiences from a "two-week code camp". In: 36th International Conference on Software Engineering - ICSE Companion, 2014, pp. 264- 271.
10. Lewis, C. M.; Shah, N. "How Equity and Inequity Can Emerge in Pair Programming". In: 11th International Conference on International Computing Education Research, 2015, pp. 41-50.

11. Li, Z.; Plaue, C.; Kraemer, E. "A spirit of camaraderie: The impact of pair programming on retention". In: 26th Software Engineering Education and Training, 2013, pp. 209-218.
12. Luz, R.B.; Neto, A.G.; Noronha, R. "Teaching TDD, the Coding Dojo Style". In: 13th International Conference Advanced Learning Technologies, 2013, pp. 371 – 375.
13. Madeyski, L. "On the Effects of Pair Programming on Thoroughness and Fault-Finding Effectiveness of Unit Tests". In: 8th Product-Focused Software Process Improvement, 2007, pp. 207 - 221.
14. Mcchesney, I. "Three Years of Student Pair Programming". In: 47th ACM Technical Symposium on Computing Science Education, 2016, pp. 84-89.
15. Nawahdah, M.; Taji, D. "Investigating students' behavior and code quality when applying pair-programming as a teaching technique in a Middle Eastern society". In: 7th Global Engineering Education Conference, 2016, pp. 32-39.
16. Rong, G.; Zhang, H.; Xie, M; Shao, D. "Improving PSP education by pairing: An empirical study". In: 34th International Conference on Software Engineering, 2012, pp.1245-1254.
17. Rooksby, J.; Hunt, J.; Wang, X. "The Theory and Practice of Randori Coding Dojos". In: 15th International Conference on Agile Software Development, 2014, pp 251- 259.
18. Salleh, N.; Mendes, E.; Grundy, J. "Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments.". *Empirical Software Engineering*, vol. 19-3, 2012, pp. 714–752
19. Sato, D.T; Corbucci, H.; Bravo, M. "Coding Dojo: An Environment for Learning and Sharing Agile Practices". In: Agile Conference, 2008, pp.459 - 464.
20. Schmidt, C.T.; Venkatesha, S. G.; Heymann, J. "Empirical insights into the perceived benefits of agile software engineering practices: a case study from SAP". In: 36th International Conference on Software Engineering, 2014, pp. 84-92.J.
21. Seyam, M.; Mccrickard, S. "Collaborating on mobile app design through pair programming: A practice-oriented approach overview and expert review". In: International Conference on Collaboration Technologies and Systems, 2015, pp.124-131.
22. Seyam, M.; Mccrickard, S. "Teaching Mobile Development with Pair

- Programming". In: 47th ACM Technical Symposium on Computing Science Education, 2016, pp 96-101.
23. Sfetsos, P.; Adamidis, P.; Angelis, L.; Stamelos, I.; Deligiannis, I., "Investigating the Impact of Personality and Temperament Traits on Pair Programming: A Controlled Experiment Replication". In: 8th Quality of Information and Communications Technology, 2012, pp.57,65.
 24. Sfetsos, P.; Stamelos, I.; Angelis, L.; Deligiannis, I. "An experimental investigation of personality types impact on pair effectiveness in pair programming". *Empirical Software Engineering*, vol. 14- 2, 2008, pp. 187-226.
 25. Simon, B; Hanks, B. "First-year students' impressions of pair programming in CS1". *Journal of Education Resources in Computing*, vol. 7- 4, 2008, 28p.
 26. Urness, T. "Assessment using peer evaluations, random pair assignment, and collaborative programing in CS1". *J. Comput. Sci. College*, Vol. 25 - 1, 2009, pp.87-93.
 27. Vanhanen, J.; Abrahamsson, P. "Perceived Effects of Pair Programming in an Industrial Context". In: 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, p.8.
 28. Wood, K.; Parsons, D.; Gasson, J.; Haden, P. "It's never too early: Pair programming in cs1". In: 15th Australasian Computing Education Conference, 2013, pp. 13–21
 29. Zacharis, N. "Evaluating the Effects of Virtual Pair Programming on Students' Achievement and Satisfaction". *International Journal of Emerging Technologies in Learning (IJET)*, vol. 4- 3, 2009, pp. 34 - 39..
 30. Zarb, M.; Hughes, J.; Richards, J. "Evaluating industry-inspired pair programming communication guidelines with undergraduate students". In: 45th ACM Technical Symposium on Computer Science Education, 2014, pp. 361-366.
 31. Zarb, M.; Hughes, J.; Richards, J. "Further Evaluations of Industry-Inspired Pair Programming Communication Guidelines with Undergraduate Students". In: 46th ACM Technical Symposium on Computer Science Education, 2015, pp. 314-319.

APÊNDICE C

Questionário Pré-experimento do estudo de viabilidade

CARACTERIZAÇÃO DO PARTICIPANTE

Participante: _____

Prezado (a),

Este questionário tem por objetivo avaliar seu grau de familiaridade em relação a diferentes aspectos relacionados a este estudo. **Toda informação coletada neste questionário será tratada de forma confidencial.**

1. Experiência com programação

Em relação à sua experiência com programação, assinale os itens que mais se aplicam ao seu grau de experiência:

- Meu contato com programação se restringe as disciplinas da graduação
- Tenho experiência de ___anos (___meses) em programação em projetos pessoais
- Tenho experiência de ___anos (___meses) em programação em projetos no ambiente acadêmico (exemplo: iniciação científica, participação em um projeto de pesquisa)
- Tenho experiência de ___ anos (___meses) com programação em projetos na indústria

2. Conhecimento de Qt

Em relação ao conhecimento prévio sobre o framework QT, marque as alternativas mais adequadas. **Por favor, não considere os conhecimentos adquiridos durante as aulas ministradas nesta disciplina.**

- Não possuo nenhum conhecimento prévio sobre QT.

- () Tenho conhecimento de QT a partir de leituras e materiais de apoio.
- () Utilizei QT no ambiente acadêmico em ___projeto(s).
- () Utilizei QT na indústria em ___projeto(s).

3. Programação em par

A programação em par é caracterizada quando **dois** programadores desenvolvem juntos um algoritmo, ou seja, um programa de computador. Você já teve alguma experiência com programação em par? Assinale as opções mais adequadas.

- () Sim, na Faculdade em ___sessão(ões)
- () Sim, no Trabalho em ___sessão(ões)
- () Sim, na Comunidade de Software Local (grupo de usuários de determinada linguagem de programação) em ___sessão(ões)
- () Sim, em Eventos em ___sessão(ões)
- () Outros: _____ em ___ sessão(ões)
- () Nunca tive uma experiência com programação em par

*sessão: intervalo de tempo específico para execução da prática

4. Coding dojo

O *coding dojo* é caracterizado quando um **grupo de 3 ou mais** programadores desenvolvem juntos um algoritmo, ou seja, um programa de computador. Você já teve alguma experiência com coding dojo? Assinale as opções mais adequadas.

- () Sim, na Faculdade e participei de ___sessão(ões)
- () Sim, no Trabalho e participei ___sessão(ões)
- () Sim, na Comunidade de Software Local (exemplo: grupo de usuários de determinada linguagem de programação) e participei ___sessão(ões)
- () Sim, em Eventos e participei de ___sessão(ões)
- () Outros: _____ e participei ___ sessão(ões)
- () Nunca tive uma experiência com *coding dojo*

Questionário de avaliação - Programação em Par

Gostaríamos que você respondesse as questões abaixo para nos ajudar a melhorar a adoção desta prática. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa. Algumas fotografias poderão ser feitas como registro desta atividade, mas não serão publicadas em nenhum local sem autorização. Por favor, circule **um número** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

Afirmações	Sua avaliação			Comentários sobre a questão
	-2 Discordo Fortemente	-1 Discordo 0 Neutro 1 Concordo	2 Concordo Fortemente	
Houve algo interessante na prática que capturou minha atenção.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
O funcionamento desta prática está adequado ao meu jeito de aprender.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Foi fácil entender prática e começar a utilizá-la.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Ao utilizar a prática senti confiança de que estava aprendendo.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Pude interagir com outra pessoa durante a prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Eu me diverti junto com o meu par.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A prática é adequadamente desafiadora para mim, as tarefas não são muito fáceis nem muito difíceis.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Eu me diverti com a programação em par.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

Eu recomendaria esta prática para meus colegas.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Gostaria de utilizar esta prática novamente.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Tive sentimentos positivos de eficiência no desenrolar da prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A prática contribuiu para a minha aprendizagem na disciplina.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A experiência com a prática vai contribuir para meu desempenho na vida profissional.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

– Atribua uma nota de 1,0 a 5,0 para seu nível de conhecimento antes e depois da prática aos conceitos listados na tabela abaixo (1,0 – pouco;5,0 – muito).

Conceitos	Lembrar o que é		Compreender como funciona		Aplicar na prática	
	Antes	Depois	Antes	Depois	Antes	Depois
<i>Mockups</i>						
Interação entre <i>mockups</i>						
Sintaxe (linguagem) de QT						

– Cite 3 pontos fortes da prática:

– Cite 3 pontos negativos da prática:

Questionário de avaliação - Coding Dojo

Gostaríamos que você respondesse as questões abaixo para nos ajudar a melhorar a adoção desta prática. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa. Algumas fotografias poderão ser feitas como registro desta atividade, mas não serão publicadas em nenhum local sem autorização. Por favor, circule **um número** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

Afirmações	Sua avaliação			Comentários sobre a questão
	-2 Discordo Fortemente	-1 Discordo 0 Neutro 1 Concordo	2 Concordo Fortemente	
Houve algo interessante na prática que capturou minha atenção.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O funcionamento desta prática está adequado ao meu jeito de aprender.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Foi fácil entender prática e começar a utilizá-la.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Ao utilizar a prática senti confiança de que estava aprendendo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
A prática promove momentos de cooperação entre as pessoas que participam.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me diverti junto com o grupo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
A prática é adequadamente desafiadora para mim, as tarefas não são muito fáceis nem muito difíceis.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me diverti com o coding dojo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Eu recomendaria esta prática para meus colegas.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	

Gostaria de utilizar esta prática novamente.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Tive sentimentos positivos de eficiência no desenrolar da prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A experiência com a prática vai contribuir para meu desempenho na vida profissional.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

– Atribua uma nota de 1,0 a 5,0 para seu nível de conhecimento antes e depois da prática aos conceitos listados na tabela abaixo (1,0 – pouco;5,0 – muito).

Conceitos	Lembrar o que é		Compreender como funciona		Aplicar na prática	
	Antes	Depois	Antes	Depois	Antes	Depois
<i>Mockups</i>						
Interação entre <i>mockups</i>						
Sintaxe (linguagem) de QT						

– Cite 3 pontos fortes da prática:

– Cite 3 pontos negativos da prática:

O uso da Programação em Par e *Coding Dojo* no desenvolvimento de *mockups*

Faculdade Informática/PUCRS
Avenida Ipiranga, 6681 – Prédio 32 - 90619-900 –
Porto Alegre – RS Tel: (51) 3320-3558

Termo de Consentimento Livre e Esclarecido

A PUCRS, através do **grupo de pesquisa de Metodologias Ágeis e Desenvolvimento Distribuído de Software (Munddos)** da Faculdade de Informática em parceria com o grupo USES da UFAM, agradece a sua atenção e a inestimável contribuição que prestarão para o auxílio do avanço da pesquisa na área de Engenharia de Software.

O objetivo desta pesquisa é investigar questões relacionadas a adoção da Programação em Par e *Coding Dojo* no desenvolvimento de *mockups*. Para isto, os participantes são convidados a experimentar uma das práticas para desenvolver *mockups* e a responderem dois questionários (antes e após as atividades). Durante a atividade os participantes serão orientados por um ou mais pesquisadores.

Lembramos que o objetivo deste estudo **não é avaliar o participante, mas sim** avaliar a prática que o participante usará durante o experimento. O uso que se faz dos registros efetuados durante o experimento é **estritamente** limitado a atividades de pesquisa e desenvolvimento, garantindo-se para tanto que:

1. O anonimato dos participantes será preservado em todo e qualquer documento divulgado em foros científicos (tais como conferências, periódicos, livros e assemelhados) ou pedagógicos (tais como apostilas de cursos, *slides* de apresentações, e assemelhados).
2. Todo participante poderá ter acesso a cópias destes documentos após a publicação dos mesmos.
3. Todo participante que se sentir constrangido ou incomodado durante a realização da atividade pode interromper a sua participação e estará fazendo um favor à equipe se registrar por escrito as razões ou sensações que o levaram a esta atitude. A equipe fica obrigada a descartar os dados do participante para fins da avaliação a que se destinaria.
4. Todo participante tem direito de expressar por escrito, na data de realização da atividade, qualquer restrição ou condição adicional que lhe pareça aplicar-se aos itens acima enumerados (1, 2 e 3). A equipe se compromete a observá-las com rigor e entende que, na ausência de tal manifestação, o participante concorda que rejam o comportamento ético da equipe somente as condições impressas no presente documento.
5. A equipe tem direito de utilizar os dados das atividades, mantidas as condições acima mencionadas, para quaisquer fins acadêmicos, pedagógicos e/ou de desenvolvimento contemplados por seus membros.

Declaro que estou de pleno acordo com os termos acima.

Assinatura do participante

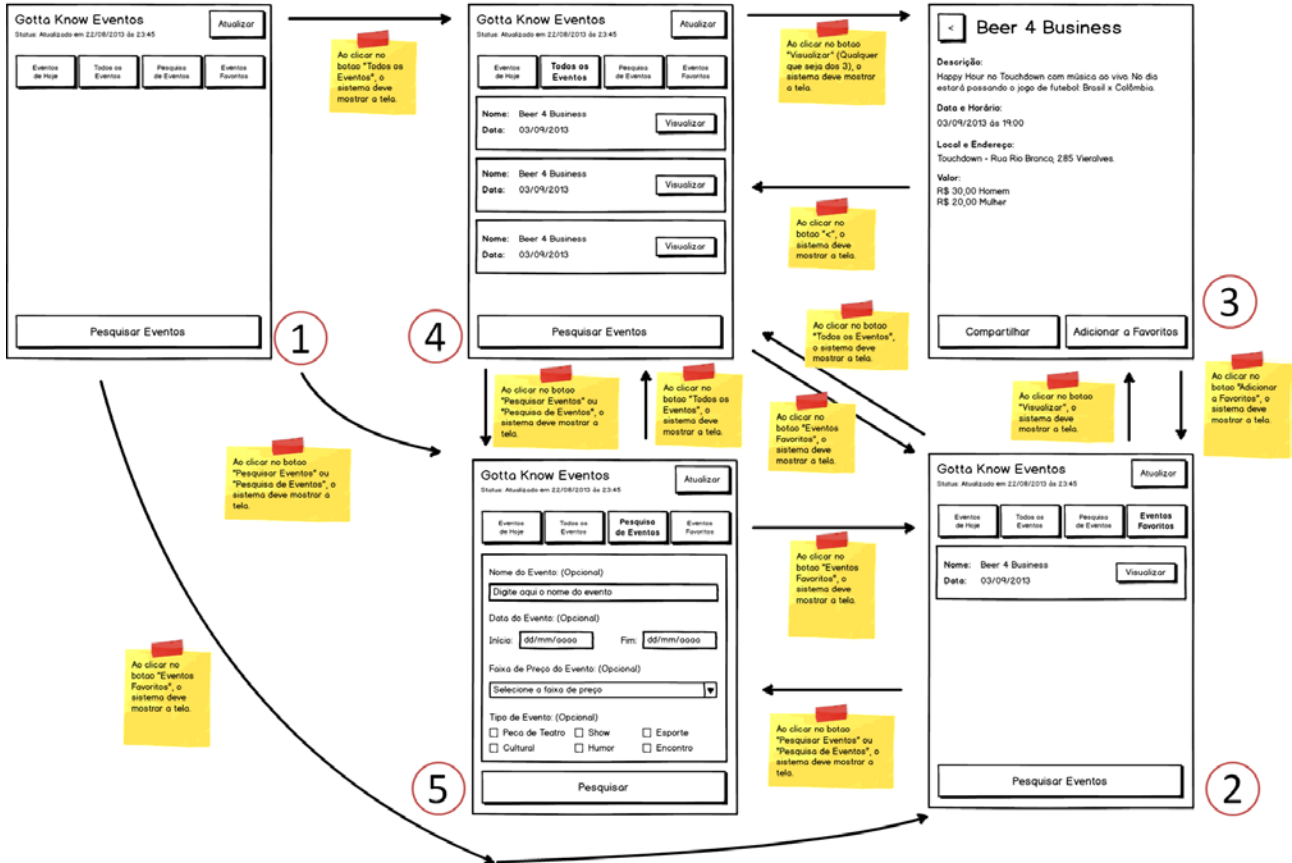
Bernardo Estácio
Pesquisador Responsável

Nome do Participante: _____

Pesquisadores Responsáveis: Bernardo Estácio e Dr. Rafael Prikładnicki

Contato: bernardo.estacio@acad.pucrs.br e rafael.prikladnicki@pucrs.br

O Status do Sistema é representado ao colocar a fonte do botão em negrito.



Tarefa Do Estudo de Viabilidade 1

APÊNDICE D

CARACTERIZAÇÃO DO PARTICIPANTE

Participante: _____

Prezado (a),

Este questionário tem por objetivo avaliar seu grau de familiaridade em relação a diferentes aspectos relacionados a este estudo. **Toda informação coletada neste questionário será tratada de forma confidencial.**

1. Dados demográficos

Gênero: () Masculino () Feminino Idade: _____ anos

Curso (nível mais alto): _____

Tempo de experiência profissional na área de TI: _____ anos

Tempo de experiência profissional com Métodos Ágeis: _____ anos Qntde. de Projetos: _____

Departamento/área: _____

Vínculo empregatício: _____ Tempo de empresa: _____ anos

Função atual: _____

2. Experiência com programação

Em relação a sua experiência com programação, assinale os itens (múltipla escolha) que mais se aplicam ao seu grau de experiência:

- () Meu contato com programação se restringe as disciplinas da graduação
- () Tenho experiência de _____ anos (____ meses) em programação em projetos pessoais

() Tenho experiência de ___anos (___meses) em programação em projetos no ambiente acadêmico (exemplo: iniciação científica, participação em um projeto de pesquisa)

() Tenho experiência de ___ anos (___meses) com programação em projetos na indústria

3. Conhecimento de Java

Em relação ao conhecimento prévio sobre Java, marque as alternativas mais adequadas.

() Não possuo nenhum conhecimento prévio sobre Java.

() Tenho conhecimento de Java a partir de leituras e materiais de apoio.

() Utilizei Java no ambiente acadêmico/pessoal em ___projeto(s).

() Utilizei Java na indústria em ___projeto(s).

4. Programação em par

A programação em par é caracterizada quando **dois** programadores desenvolvem juntos um programa em uma linguagem de programação. Você já teve alguma experiência com programação em par? Assinale as opções mais adequadas.

() Sim, na Faculdade em ___sessão(ões)

() Sim, no Trabalho em ___sessão(ões)

() Sim, na Comunidade de Software Local (grupo de usuários de determinada linguagem de programação) em ___sessão(ões)

() Sim, em Eventos em ___sessão(ões)

() Outros: _____ em ___ sessão(ões)

() Nunca tive uma experiência com programação em par

*sessão: intervalo de tempo específico para execução da prática

5. Coding dojo

O *coding dojo* é caracterizado quando um **grupo de 3 ou mais** programadores desenvolvem juntos desenvolvem juntos um programa em uma

linguagem de programação. Você já teve alguma experiência com coding dojo?
Assinale as opções mais adequadas.

- Sim, na Faculdade e participei de ___sessão(ões)
- Sim, no Trabalho e participei ___sessão(ões)
- Sim, na Comunidade de Software Local (exemplo: grupo de usuários de determinada linguagem de programação) e participei ___sessão(ões)
- Sim, em Eventos e participei de ___sessão(ões)
- Outros: _____ e participei ___ sessão(ões)
- Nunca tive uma experiência com *coding dojo*

*sessão: intervalo de tempo específico para execução da prática

EXERCÍCIO DE CODIFICAÇÃO A

Enunciado: A atividade consiste na implementação de parte de um sistema em Java capaz de simular o comportamento do jogo de adivinhação de animais (<http://www.animalgame.com/>). O objetivo central do trabalho é implementar um entrevistador que deverá adivinhar qual animal, dentre um grupo de animais pré-definidos, o entrevistado está “pensando”.

Exemplo:

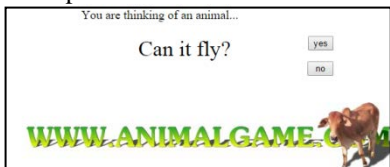


Fig. 01 - Exemplo Questão 01

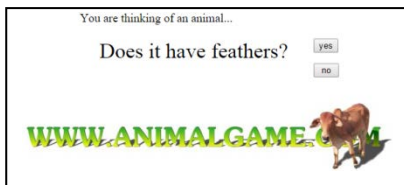


Fig. 02 - Exemplo Questão 02

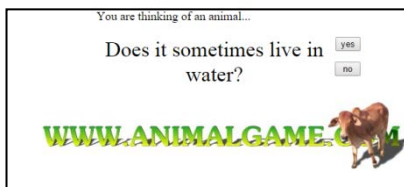


Fig. 03 - Exemplo Questão 03

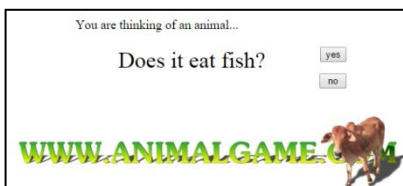


Fig. 04 - Exemplo Questão 04

Exemplo Escrito:

1) Ele pode voar? S ou N

S: Será que ela tem penas? S ou N

S: Será que, que voa? S ou N

S: Será que começar com um 'P'?

S: O animal que você esta pensando é a "Papagaio"

N: O animal que você esta pensando é a "Galinha"

N: Será que vive na água? S ou N

S: Será que começar com um 'B'?

S: O animal que você esta pensando é a "Baleia"

N: O animal que você esta pensando é a "Peixe"

N: Será que começa com "T"?

S: O animal que você esta pensando é o "Tatu"

N: O animal que você esta pensando é a "Cachorro"

EXERCÍCIO DE CODIFICAÇÃO B

Enunciado: Desenvolva um programa que simule a entrega de notas quando um cliente efetuar um saque em um caixa eletrônico. Os requisitos básicos são os seguintes:

- Entregar o menor número de notas;
- É possível sacar o valor solicitado com as notas disponíveis;
- Saldo do cliente infinito;
- Quantidade de notas infinito (pode-se colocar um valor finito de cédulas para aumentar a dificuldade do problema);
- Notas disponíveis de R\$ 100,00; R\$ 50,00; R\$ 20,00 e R\$ 10,00

Exemplos:

Valor do Saque: R\$ 30,00 – Resultado Esperado: Entregar 1 nota de R\$20,00 e 1 nota de R\$ 10,00.

Valor do Saque: R\$ 80,00 – Resultado Esperado: Entregar 1 nota de R\$50,00 1 nota de R\$ 20,00 e 1 nota de R\$ 10,00.

EXERCÍCIO DE CODIFICAÇÃO C



Enunciado: Desenvolva um sistema de apoio interno à livraria obedecendo as seguintes características:

- 1) Ao ser executado o sistema deverá apresentar uma mensagem inicial "**Bem-vindo à Livraria Sofia**", no entanto esta mensagem pode ser modificada pelo dono da livraria;
- 2) A segunda mensagem informa que o primeiro item deve ser cadastrado. Este item deve ter: código, título, um autor, uma descrição, quantidade e preço;
- 3) Os itens da livraria são: código 1 (livro), código 2 (mídia) e código 3 (revista);
- 4) Utilize os conceitos de heranças para diferenciar os atributos específicos de cada item.
- 5) A livraria está dando desconto, de 20% para livros, 15% para mídia e 10% para revistas. Exiba o preço desses itens **com e sem desconto**.

Termo de Consentimento Livre e Esclarecido

Faculdade Informática/PUCRS
Avenida Piranga, 6681 – Prédio 32-90619-900–
Porto Alegre – RS Tel: (51) 3320-3558

A PUCRS e PUC-RIO, através dos **grupos de pesquisa de Engenharia de Software**, agradecem a sua atenção e a inestimável contribuição que prestarão para o auxílio do avanço da pesquisa na área de Engenharia de Software.

O objetivo desta pesquisa é investigar questões relacionadas a adoção de práticas colaborativas no contexto de entendimento dos conceitos de programação. Para isto, os participantes são convidados a experimentar as práticas e a responderem dois questionários (antes e após as atividades). Durante a atividade os participantes serão orientados por um ou mais pesquisadores.

Lembramos que o objetivo deste estudo **não é avaliar o participante, mas sim** avaliar a prática que o participante usará durante o experimento. Ou do que se faz dos registros efetuados durante o experimento é **estritamente** limitado a atividades de pesquisa e desenvolvimento, garantindo-se para tanto que:

1. O anonimato dos participantes será preservado em todo e qualquer documento divulgado em foros científicos (tais como conferências, periódicos, livros e assemelhados) ou pedagógicos (tais como apostilas de cursos, *slides* de apresentações, e assemelhados).
2. Todo participante poderá ter acesso a cópias destes documentos após a publicação dos mesmos.
3. Todo participante que se sentir constrangido ou incomodado durante a realização da atividade pode interromper a sua participação e estará fazendo um favor à equipes e registrar por escrito as razões ou sensações que o levaram a esta atitude. A equipe fica obrigada a descartar os dados do participante para fins da avaliação a que se destinaria.
4. Todo participante tem direito de expressar por escrito, na data de realização da atividade, qualquer restrição ou condição adicional que lhe pareça aplicar-se aos itens acima enumerados (1, 2 e 3). A equipe se compromete a observá-las com rigor e entende que, na ausência de tal manifestação, o participante concorda que rejeita o comportamento ético da equipe somente as condições impressas no presente documento.
5. A equipe tem direito de utilizar os dados das atividades, mantidas as condições acima mencionadas, para quaisquer fins acadêmicos, pedagógicos e/ou de desenvolvimento contemplados por seus membros.

Declaro que estou de pleno acordo com os termos acima.

Assinatura do participante

Bernardo Estácio
Pesquisador Responsável

Nome do Participante: _____

Pesquisadores Responsáveis: Bernardo Estácio, Roberto Oliveira,

Dr. Rafael Prikladnicki, Dr. Alessandro Garcia, Dr.^a Sabrina Marczak e DR. Marcos Kalinowski

Contato: bernardo.estacio@acad.pucrs.br e rafael.prikladnicki@pucrs.br

Contato: rfelicio@inf.puc-rio.br e afgarcia@inf.puc-rio.br

Questionário de avaliação - Programação em Par

Gostaríamos que você respondesse as questões abaixo para nos ajudar a melhorar a adoção desta prática. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa.

Por favor, circule **um número** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

Afirmações	Sua avaliação			Comentários sobre a questão
	-2 Discordo Fortemente	-1 Discordo 0 Neutro 1 Concordo	2 Concordo Fortemente	
Houve algo interessante durante prática que capturou minha atenção.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
O funcionamento da prática está adequado ao meu jeito de aprender.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Foi fácil entender a dinâmica do prática e começar a utilizá-la.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Ao utilizar a prática senti confiança de que estava aprendendo.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A prática promove momentos de cooperação entre as pessoas que participam.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Eu me diverti junto com a prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Eu recomendaria esta prática para meus colegas.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

Gostaria de utilizar esta prática novamente.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Tive sentimentos positivos de eficácia no desenrolar da prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A experiência com a prática vai contribuir para meu desempenho na vida profissional.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

– Atribua uma nota de 1,0 a 5,0 para seu nível de conhecimento antes e depois da prática aos conceitos listados na tabela abaixo (1,0 – pouco;5,0 – muito).

Conceitos	Lembrar o que é		Compreender como funciona		Aplicar na prática	
	Antes	Depois	Antes	Depois	Antes	Depois
Sintaxe (linguagem) Java						
POO						
Boas Práticas de Design						

– Cite 3 pontos fortes da prática de Coding Dojo/Programação em Par:

– Cite 3 pontos negativos (ou de melhoria) da prática Coding Dojo:

- Quais conceitos esta atividade te ajudou a lembrar (os que foram vistos durante a aceleradora)?

Questionário de avaliação – *Coding dojo randori*

Gostaríamos que você respondesse as questões abaixo para nos ajudar a melhorar a adoção desta prática. Todos os dados são coletados anonimamente e somente serão utilizados no contexto desta pesquisa.

Por favor, circule **um número** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo.

Afirmações	Sua avaliação			Comentários sobre a questão
	-2 Discordo Fortemente	-1 Discordo 0 Neutro 1 Concordo 2 Concordo Fortemente	Concordo Fortemente	
Houve algo interessante durante prática que capturou minha atenção.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
O funcionamento da prática está adequado ao meu jeito de aprender.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Foi fácil entender a dinâmica do prática e começar a utilizá-la.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Ao utilizar a prática senti confiança de que estava aprendendo.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A prática promove momentos de cooperação entre as pessoas que participam.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Eu me diverti junto com a prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Eu recomendaria esta prática para meus colegas.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

Gostaria de utilizar esta prática novamente.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
Tive sentimentos positivos de eficácia no desenrolar da prática.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	
A experiência com a prática vai contribuir para meu desempenho na vida profissional.	Discordo Fortemente	-2 -1 0 +1 +2	Concordo Fortemente	

– Atribua uma nota de 1,0 a 5,0 para seu nível de conhecimento antes e depois da prática aos conceitos listados na tabela abaixo (1,0 – pouco;5,0 – muito).

Conceitos	Lembrar o que é		Compreender como funciona		Aplicar na prática	
	Antes	Depois	Antes	Depois	Antes	Depois
Sintaxe (linguagem) Java						
POO						
Boas Práticas de Design						

– Cite 3 pontos fortes da prática de Coding Dojo/Programação em Par:

– Cite 3 pontos negativos (ou de melhoria) da prática Coding Dojo:

- Quais conceitos esta atividade te ajudou a lembrar (os que foram vistos durante a aceleradora)?

APÊNDICE E

Informed Consent Form for Participants

Please read the following information carefully. You can also request a copy for future reference.

DESCRIPTION: You are invited to participate in a research study of the Software Engineering Research Group of Freie Universität Berlin and MunDdos Research Group from PUCRS-Brazil, hereinafter the “Research Groups”. In this study, you will participate in a session of a collaborative practice called *Coding Randori Dojo*, working on two software tasks. After the session, you will be interviewed about your experience with the practice.

PARTICIPATION: The participation in this study is voluntary and anonymous. You might stop the activity anytime without giving reasons.

RECORDING: Throughout the session, you will be both video and audio recorded.

RESULTS: The data will be analyzed and used by the Research Groups. The results might be published in conferences, journal, book chapters, magazine, and other types of publication. The Research Group will keep your identity confidential.

TIME INVOLVEMENT: The session will take approximately 120 minutes.

By signing the form, you express that you understood and agree with the above terms.

Name:

Date:

Signature:

ANEXO I

Fizz Buzz (Utilizando em S1, S3, S5)

Extraído de <http://codingdojo.org/cgi-bin/wiki.pl?action=browse&diff=2&id=KataFizzBuzz>

Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz?".

Sample output:

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

16

162

17

Fizz

19

Buzz

... etc up to 100

Stage 2 (Utilizado apenas na S3)

* A number is fizz if it is divisible by 3 or if it has a 3 in it

* A number is buzz if it is divisible by 5 or if it has a 5 in it

Game of Life (S1)

Extraído de : [/http://codingdojo.org/cgi-bin/wiki.pl?KataGameOfLife](http://codingdojo.org/cgi-bin/wiki.pl?KataGameOfLife)

This Kata is about calculating the next generation of Conway's game of life, given any starting position. See http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life for background.

You start with a two dimensional grid of cells, where each cell is either alive or dead. In this version of the problem, the grid is finite, and no life can exist off the edges. When calculating the next generation of the grid, follow these rules:

1. Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.
2. Any live cell with more than three live neighbours dies, as if by overcrowding.
3. Any live cell with two or three live neighbours lives on to the next generation.
4. Any dead cell with exactly three live neighbours becomes a live cell.

You should write a program that can accept an arbitrary grid of cells, and will output a similar grid showing the next generation.

Clues

The input starting position could be a text file that looks like this:

Generation 1:

4 8

.....

.... * ...

164

... ** ...

.....

And the output could look like this:

Generation 2:

4 8

.....

... ** ...

... ** ...

.....

Fibonacci (S1)

Extraído de: <http://chrismepham.co.uk/blog/programming/computing-a-list-of-the-first-100-fibonacci-numbers/>

Write a function that computes the list of the first 100 Fibonacci numbers. By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two. As an example, here are the first 10 Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, and 34.

Mars Rovers (S2)

Extraído de: <https://marsroverexercise.codeplex.com/>

A squad of robotic rovers are to be landed by NASA on a plateau on Mars.

This plateau, which is curiously rectangular, must be navigated by the rovers so that their on board cameras can get a complete view of the surrounding terrain to send back to Earth.

A rover's position is represented by a combination of an x and y co-ordinates and a letter representing one of the four cardinal compass points. The plateau is divided up into a grid to simplify navigation. An example position might be 0, 0, N, which means the rover is in the bottom left corner and facing North.

In order to control a rover, NASA sends a simple string of letters. The possible letters are 'L', 'R' and 'M'. 'L' and 'R' makes the rover spin 90 degrees left or right respectively, without moving from its current spot.

'M' means move forward one grid point, and maintain the same heading.

Assume that the square directly North from (x, y) is (x, y+1).

Input (whether hard coded or input from keyboard):

The first line of input is the upper-right coordinates of the plateau, the lower-left coordinates are assumed to be 0,0.

The rest of the input is information pertaining to the rovers that have been deployed. Each rover has two lines of input. The first line gives the rover's position, and the second line is a series of instructions telling the rover how to explore the plateau.

The position is made up of two integers and a letter separated by spaces, corresponding to the x and y co-ordinates and the rover's orientation.

Each rover will be finished sequentially, which means that the second rover won't start to move until the first one has finished moving.

Output:

The output for each rover should be its final co-ordinates and heading.

Plateau max X and Y, Starting coordinates, direction and path for two rovers:

5 5

1 2 N

LMLMLLMM

3 3 E

MMRMMRMRRM

Output and new coordinates:

1 3

5 1 E

Numbers in Words (S3)

Extraído de: <https://web.archive.org/web/20161225084757/http://codingdojo.org/>

It occurs now and then in real life that people want to write about money, especially about a certain amount of money. If it comes to cheques or contracts for example some nations have laws that state that you should write out the amount in words additionally to the amount in numbers to avoid fraud and mistakes. So if you want to transfer 745 \$ to someone via cheque you have to fill out to fields:

745.00 \$ (amount in numbers)

seven hundred and fourty five dollars (amount in words)

Step 1

The Kata is now to write a little converter class or function (depends on your favourite language and flavour) to convert numbers into words.

Step 2

Convert it back.

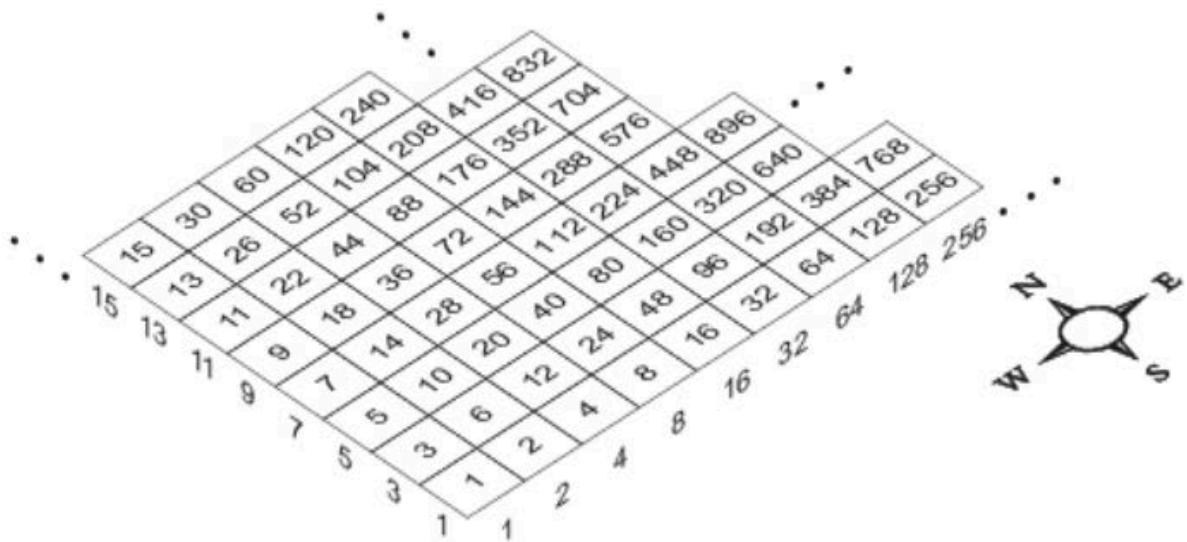
Step 3

Do all of it test driven.

City of Flatland (S3)

Extraído de: <https://icpcarchive.ecs.baylor.edu/external/25/2550.pdf>

In recognition to the number of famous mathematicians of its residents, the City of Flatland has decided to rename all its streets as numbers (positive integers to be more precise.) The streets of Flatland are organized as a grid. The city decided to number all its North-South streets using powers of two (1, 2, 4, 8,...) and all its East-West streets using odd numbers (1, 3, 5,...) . The city also decided to re-number all its buildings so that the number of each building is the result of multiplying the numbers of the two streets the building is on. For example, building #40 is at the intersection of streets 5 and 8.



The problem with this numbering scheme is that it is not easy for the residents to determine the distance between buildings. The distance between any two buildings is the number of buildings one needs to cross to go from one building to another. One can only move parallel to the streets (no diagonals or any other shortcuts.) For example, to go from building #6 to building #40, one has to travel one building north and two buildings east, so the distance is 3. Similarly, the distance from building #80 to building #88 is 4. Help the residents of Flatland by writing a program that calculates the distance between any two given building

Help the residents of Flatland by writing a program that calculates the distance between any two given buildings. Input The input is made of one or more pairs of building numbers. Each pair $\langle S, T \rangle$ appears on a single line with a single space between the two numbers.

Note that $S, T < 1,000,000,000$. The end of the input is identified by the pair $< 0, 0 >$ (which is not part of the test cases.) Output For each input pair $< S, T >$, the output file should include a line of the form:

Input

The input is made of one or more pairs of building numbers. Each pair $< S, T >$ appears on a single line with a single space between the two numbers. Note that $S, T < 1,000,000,000$. The end of the input is identified by the pair $< 0, 0 >$ (which is not part of the test cases.)

Output

For each input pair $< S, T >$, the output file should include a line of the form: 2550 - City of Flatland 1/2 The distance between S and T is D. The output file should be in the same order as the input file. Sample Input 12 14 20 30 40 50 0 0 Sample Output The distance between 12 and 14 is 3. The distance between 20 and 30 is 6. The distance between 40 and 50 is 12.

Dojo de Javascript (Exercícios da Sessão 5)

Extraído de: <https://github.com/aceleradora-TW/JasmineRailsPOC>

Primeira etapa: Jasmine

Nesta etapa, vamos utilizar o projeto base para configurar o o Jasmine no HemoHeroes. O esquema é o seguinte:

Pull da versão mais recente do projeto na máquina do dojo

Configurar Gemfile do jasmine-rails conforme Gemfile do projeto base

`bundle install`

`rails generate jasmine_rails:install` para gerar os specs e os arquivos de configuração

Criar primeiro arquivo de spec

Criar primeiro arquivo testado por uma spec

`rails s` e então acessamos `/specs` para ver resultado dos testes

E rodando pelo terminal: `rake spec:javascript`

Segunda etapa: Exercícios

Regras

Para cada arquivo de exercício, há um arquivo de teste unitário com o mesmo nome, porém estendido por `.spec.js`

Os exercícios precisam ser implementados e testados, sempre tentando escrever o teste antes da implementação.

Os exercícios e somente estes, devem, preferencialmente, ser escritos usando ES6.

Sempre que um exercício for finalizado, commitar a solução. Verificar o resultado da execução dos testes no Snap CI## Lista de Exercícios

Exercício 01: Soma

Crie um novo teste para a função soma que consiga testar resultados com valores diferentes de 2.

Exercício 02: Transformar a lista complexa

Função 1: Há uma lista de livros e autores no arquivo de exercício. Desta lista, retorne uma lista simplificada, contendo apenas o nome do livro e do autor

Função 2: Transformar a lista complexa utilizando `Array.prototype.map`

Exercício 03: Filtrar a lista de livros

Função 1: Da lista de livros, a solução deve retornar somente os livros que tenham sido escritos depois do ano 2000.

Função 2: A solução deve receber um ano por parametro e retornar somente os livros que tenham sido escritos depois do ano recebido. A solução deve utilizar `Array.prototype.filter`

Exercício 04: Some todos os elementos ímpares de uma lista

Função 1: Utilize um loop convencional para resolver o exercício

Função 2: Utilize `forEach`

Função 3: Utilize `reduce`

Exercício 05: Filtrar caracteres x

A função receberá uma string por parâmetro e devolverá uma nova string sanitizada. Se a string contiver x como primeiro ou último caractere, ele deve ser removido. Caso haja um x na palavra que não seja o primeiro ou o último caractere, ele deve ser mantido.

Exemplos:

xolá -> olá

oláx -> olá

oxlá -> oxlá

Exercício 06: Geolocalização

A função receberá uma string por parâmetro. Desta string, deverão ser extraídos dois números separados por vírgulas, latitude e longitude, respectivamente. O retorno deve ser um objeto com a seguinte estrutura:

```
{  
  Latitude: Double,  
  Longitude: Double  
}
```

Observações:

Números de latitude somente são válidos se estão entre -90 e 90

Números de longitude somente são válidos se estão entre -180 e 180

Exercício 07: Escada de asteriscos

Escreva uma função que monte uma escada de asteriscos com um número de degraus recebido por parâmetro.

Exemplo:

```
montaEscada(6)
```

Saída:

```
#
```

```
##
```

```
###
```

```
####
```

```
#####
```

```
#####
```

Exercício 07: Cifra de César

A solução deve receber uma palavra em formato string S e um número inteiro N por parâmetro. Ela deve substituir todas as letras de S por uma letra N posições à frente no alfabeto.

Por exemplo: Se a palavra for “Oca” e o número for 1, o retorno deve ser: Pdb

Dicas:

Uma string pode ser representada por um número da tabela ASCII, facilitando possíveis cálculos. Para isso:

`String.prototype.charCodeAt(n)`: Retorna o caractere no índice n da String como inteiro.

`String.fromCharCode`: Retorna um caractere à partir de um inteiro n recebido

Bônus: Implementar a solução usando `Array.prototype.map`.

Bônus 2: Implementar uma solução usando `Array.prototype.map` em apenas uma linha.