

FÁBIO VINÍCIUS BINDER

ABORDAGEM DE APOIO PARA AMBIENTES
COMPLEXOS QUE ENVOLVEM ENSINO, INOVAÇÃO
E PRODUÇÃO DE SOFTWARE

Tese apresentada ao Programa de Pós-Graduação
em Informática da Pontifícia Universidade Católica
do Paraná para obtenção do título de Doutor em
Ciências.

Curitiba
2016

FÁBIO VINÍCIUS BINDER

ABORDAGEM DE APOIO PARA AMBIENTES
COMPLEXOS QUE ENVOLVEM ENSINO, INOVAÇÃO
E PRODUÇÃO DE SOFTWARE

Tese apresentada ao Programa de Pós-Graduação
em Informática da Pontifícia Universidade Católica
do Paraná para obtenção do título de Doutor em
Ciências.

Área de concentração: Ciência da Computação

Orientador: Profa. Dra Andreia Malucelli
Co-Orientadora: Profa. Dra. Sheila Reinehr

Curitiba
2016

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

B612a Binder, Fábio Vinícius
2016 Abordagem de apoio para ambientes complexos que envolvem ensino,
inovação e produção de software / Fábio Vinícius Binder ; orientadora: Andreia
Malucelli ; co-orientadora: Sheila Reinehr. – 2016.
131 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba,
2016

Bibliografia: f. 92-95

1. Software – Desenvolvimento – Estudo e ensino. 2. Aprendizagem.
3. Ensino auxiliado por computador. I. Malucelli, Andreia. II. Reinehr, Sheila.
III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação
em Informática. IV. Título.

CDD 22. ed. – 005.1

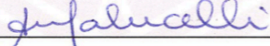
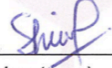

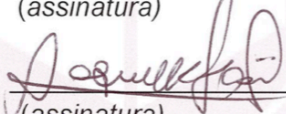
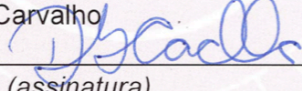
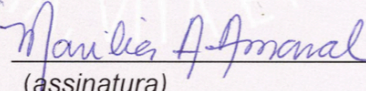


Pontifícia Universidade Católica do Paraná

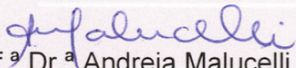
ATA DE DEFESA DE TESE DE DOUTORADO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO

DEFESA DE TESE DE DOUTORADO Nº 037/2015

Aos 07 dias de Março de 2016 realizou-se a sessão pública de Defesa da Tese de Doutorado intitulada “**Abordagem de Apoio para Ambientes Complexos que envolvem Ensino, Inovação e Produção de Software**” apresentada pelo aluno **Fábio Vinícius Binder** como requisito parcial para a obtenção do título de Doutor em Informática, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. ^a Dr. ^a Andreia Malucelli PUCPR (Orientador)	<u></u> (assinatura)	<u>APROVADO</u> (aprov/reprov.)
Prof. ^a Dr. ^a Sheila Reinehr PUCPR (Co-orientador)	<u></u> (assinatura)	<u>APROVADO</u> (aprov/reprov.)
Prof. ^a Dr. ^a Dilmeire Vosgerau PUCPR	<u></u> (assinatura)	<u>APROVADO</u> (aprov/reprov.)
Prof. ^a Dr. ^a Raquel Stasiu PUCPR	<u></u> (assinatura)	<u>APROVADO</u> (aprov/reprov.)
Prof. ^a Dr. ^a Deborah Ribeiro Carvalho PUCPR	<u></u> (assinatura)	<u>Aprovado</u> (aprov/reprov.)
Prof. ^a Dr. ^a Marília Amaral UTFPR	<u></u> (assinatura)	<u>APROVADO</u> (aprov/reprov.)

Conforme as normas regimentais do PPGIa e da PUCPR, o trabalho apresentado foi considerado APROVADO (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora registradas no Livro de Defesas do programa.


Prof.^a Dr.^a Andreia Malucelli
Coordenadora do Programa de Pós-Graduação em Informática.



DEDICATÓRIAS

Dedico este trabalho para a luz da minha jornada, Sueli.

AGRADECIMENTOS

Agradeço a todos os que me auxiliaram nessa caminhada, principalmente a PUCPR pelas oportunidades de estudo, ensino, empreendedorismo interno e por ter me liberado por seis meses para a realização do doutorado sanduíche. Agradecimentos também para a Capes pela oportunidade de estudar em outro país por meio do PDSE. Aos times de professores das instituições: IFCE, Mackenzie, FUCAPI, SENAC-SP, PUC-Rio, UFPE, UCB e PUCRS pela grande ajuda na execução dos questionários para os estudantes e, principalmente, no esforço em viabilizar as entrevistas com os respectivos professores. Aos criadores e evangelizadores do método CBL e também para a empresa parceira que possibilitou a realização dessa experiência de ensino rica, marcante e inovadora.

Agradecimentos especiais são devidos a várias pessoas que contribuíram de alguma forma para que este objetivo fosse alcançado e que merecem ser nomeadas:

Para quem me educou e sempre me empurrou para a frente, minha Super Mãe, Aldete (p.).

Para quem sempre confiou em meu potencial e ainda serve de exemplo para tudo o que eu faço, meu Pai, Eliseu (p.).

Para a melhor família de todos os universos paralelos e realidades alternativas, minha esposa Sueli, meus filhos Vinícius e Gustavo e minha irmã, Lisandra.

Para quem me inspirou a gostar de ciência, Carl Sagan (p.).

Para o sábio amigo "o que ele faria se estivesse nessa situação?", Parau.

Para quem ficou ao meu lado quando todos os dedos apontavam para mim, Luiz Hamilton Berton (p.).

Para quem me ajudou quando eu estava caído, meus amigos e quase-irmãos Jopz e Rubi.

Para o desconhecido que me deu a mão e me levantou, Caetano Mauro Tavares.

Para quem me mostrou que doenças não devem nos envergonhar mas nos fortalecer, Sílvio Reis e Juliana Vermelho Martins.

Para o companheiro de literatura e das conversas mais inteligentes e empolgantes, Rodrigo Webler.

Para quem me fez olhar para fora da água, Armênio Cardoso.

Para o mestre e amigo que, com sua incrível capacidade de criar realidades ao seu redor, me trouxe de volta ao lar, Vidal Martins.

Para quem reacendeu a centelha da esperança em um futuro próspero, Alessandro Piovezahn, Stewart Billmayer, Cláudio Carvilhe e Eric Onuki.

Para quem tornou esse trabalho melhor, por meio da sugestão de inclusão de uma técnica de entrevista mais elaborada no processo de seleção, professor Carlos Hairon e seu time da IFCE.

Para quem me ajudou no momento mais crítico, o início da escrita desta tese, Regina Albuquerque.

Para dois alunos brilhantes que propiciaram um dos momentos mais inesquecíveis da minha vida ao ganharem, por méritos próprios, um prêmio importantíssimo, Victor Lappas Giménez e Gabriel Mathias Rocha.

Para a equipe de professores e amigos que participaram ativa e decisivamente no desenvolvimento deste trabalho: Breno Azevedo, Maicris Fernandes, Mark Joselli e Vinícius Godoy de Mendonça.

Para as professoras da banca de qualificação que me ofereceram conselhos muito valiosos e uma nova oportunidade para continuar o doutorado, Marília Amaral, Dilmeire Vosgerau, Deborah Carvalho e Raquel Stasiu.

E finalmente, para aquelas que, contra todas as evidências, acreditaram que eu seria capaz de realizar meu maior sonho e fizeram tudo o que foi possível para que esta tese estivesse pronta a tempo, minhas orientadoras Andreia Malucelli e Sheila Reinehr.

*Sometimes when you innovate, you make mistakes. It is best
to admit them quickly, and get on with improving your other
innovations.*

Steve Jobs

RESUMO

O mercado de dispositivos móveis vem crescendo a cada ano, criando uma demanda por aplicativos pessoais e empresariais de missão crítica. Essa demanda tem como consequência direta, a necessidade de capacitação para desenvolvedores de aplicativos móveis. Com o intuito de atender a esta necessidade, foi desenvolvida uma abordagem para a formação de desenvolvedores, incluindo a seleção dos estudantes, qualificação por meio da metodologia ativa *Challenge Based Learning*, produção de software para dispositivos móveis e criação de startups. A abordagem IDEAS (Inovação, **D**esenvolvimento e **E**nsino em **A**mbientes complexos de **S**oftware) foi criada por meio de uma pesquisa-ação durante dois ciclos do referido curso. Para validar esta abordagem foram feitas avaliações com estudantes e professores por meio de reflexões periódicas além de questionários e entrevistas ao final de cada ciclo.

Palavras-chave: *Challenge Based Learning*, Ensino de Programação, Aprendizagem Ativa, Desenvolvimento de Software, Dispositivos Móveis, Inovação.

ABSTRACT

The market for mobile devices is growing each year, creating a demand for personal and mission critical business applications. Such demand has as a consequence, the need for training of mobile application developers. To fill this need a complete approach was developed including students' selection, training in the active learning method *Challenge Based Learning*, mobile devices software production and startup development. The approach IDEAS (Innovation, Development and Teaching in Complex Software Environments - Inovação, **D**esenvolvimento e **E**nsino em **A**mbientes complexos de **S**oftware) was created during two cycles of action research. To validate this approach, evaluations with students and teachers were executed: periodic reflections and also questionnaires and interviews at the end of each cycle.

Keywords: Challenge Based Learning, Programming Teaching, Active Learning, Software Development, Mobile Devices, Innovation.

SUMÁRIO

RESUMO.....	X
ABSTRACT	XI
LISTA DE FIGURAS.....	XV
LISTA DE QUADROS E TABELAS	XVI
LISTA DE ABREVIATURAS E SIGLAS	XVII
CAPÍTULO 1 - INTRODUÇÃO	1
1.1 MOTIVAÇÃO.....	3
1.2 OBJETIVOS	4
1.3 DELIMITAÇÃO DE ESCOPO	4
1.4 PROCESSO DE TRABALHO	5
1.5 ESTRUTURA DO DOCUMENTO DA TESE	5
1.6 CONSIDERAÇÕES SOBRE O CAPÍTULO	7
CAPÍTULO 2 - REVISÃO DA LITERATURA	8
2.1 METODOLOGIAS ATIVAS	8
2.2 O MÉTODO CBL.....	10
2.2.1 Etapas do CBL.....	11
2.3 METODOLOGIAS ATIVAS NO ENSINO DE PROGRAMAÇÃO.....	15
2.3.1 Uso de Metodologias não Tradicionais no Ensino de Programação	15
2.3.2 Desafios no Desenvolvimento de Software para Dispositivos Móveis.....	16
2.3.3 Comparativo Metodologias Ativas x CBL.....	17
2.4 TRABALHOS RELACIONADOS	20
2.5 CONSIDERAÇÕES SOBRE O CAPÍTULO	23
CAPÍTULO 3 - ESTRUTURAÇÃO DA PESQUISA.....	24
3.1 CARACTERIZAÇÃO DA PESQUISA	24
3.2 PESQUISA-AÇÃO.....	25
3.3 ESTRATÉGIA DE PESQUISA.....	28
3.3.1 Fase Preliminar – Contexto e Objetivo	28
3.3.2 Ciclo 1 – Turma 2013-2014	33
3.3.3 Ciclo 2 – Turma 2015.....	34
3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO	34

CAPÍTULO 4 - CICLO 1 – TURMA 2013-2014	35
4.1 PLANEJAMENTO DA AÇÃO DO CICLO 1	36
4.2 IMPLEMENTAÇÃO DO CICLO 1	37
4.3 AVALIAÇÃO DO CICLO 1.....	42
4.3.1 Avaliação dos Estudantes.....	43
4.3.2 Avaliação dos Professores.....	47
4.4 CONSIDERAÇÕES SOBRE O CAPÍTULO.....	53
CAPÍTULO 5 - CICLO 2 – TURMA 2015.....	55
5.1 PLANEJAMENTO DA AÇÃO DO CICLO 2	56
5.2 IMPLEMENTAÇÃO DO CICLO 2	57
5.2.1 Etapa 1: Selecionar Perfil.....	57
5.2.2 Etapa 2: Executar Curso	60
5.2.3 Etapa 3: Produzir Software.....	62
5.2.4 Etapa 4: Criar Startup	78
5.3 AVALIAÇÃO DO CICLO 2.....	80
5.4 CONSIDERAÇÕES SOBRE O CAPÍTULO.....	86
CAPÍTULO 6 - CONSIDERAÇÕES FINAIS	87
6.1 RELEVÂNCIA DO ESTUDO.....	87
6.2 CONTRIBUIÇÕES DA PESQUISA.....	88
6.3 LIMITAÇÕES DA PESQUISA	90
6.4 TRABALHOS FUTUROS	91
REFERÊNCIAS BIBLIOGRÁFICAS	92
GLOSSÁRIO	96
APÊNDICES	97
APÊNDICE A - QUESTIONÁRIO PARA ESTUDANTES	98
APÊNDICE B - QUESTIONÁRIO PARA PROFESSORES DA 1^A TURMA	101
APÊNDICE C - ENTREVISTA DOS CANDIDATOS DA 2^A. TURMA.....	103
APÊNDICE D - FORMULÁRIO DE INSCRIÇÃO PARA A HOTMILK.....	105
APÊNDICE E - FORMULÁRIO DE CAPTAÇÃO DE PROJETOS.....	106
APÊNDICE F - LISTA DE PROJETOS FINALIZADOS NA 1^A TURMA.....	109

APÊNDICE G	- LISTA DE PROJETOS FINALIZADOS NA 2^A TURMA	110
APÊNDICE H	- PROVA DA 1^A TURMA.....	112
APÊNDICE I	- ENTREVISTA DA 1^A TURMA.....	117
APÊNDICE J	- PROVA DA 2^A TURMA	118
APÊNDICE K	- PUBLICAÇÕES DO MINI DESAFIO DA 1^A TURMA.....	124
APÊNDICE L	- QUESTIONÁRIO PARA PROFESSORES DA 2^A TURMA.....	125
APÊNDICE M	- TABULAÇÃO DAS ENTREVISTAS DA 1^A TURMA.....	128
APÊNDICE N	- REFLEXÕES DOS PROFESSORES NA SEGUNDA TURMA...	129
APÊNDICE O	- CRONOGRAMA DA PESQUISA-AÇÃO.....	131

LISTA DE FIGURAS

Figura 1-1: Fases da execução da pesquisa.....	5
Figura 1-2: Organização dos Capítulos da Tese.....	6
Figura 2-1: O Método <i>Challenge Based Learning</i> [CBL; 2011].	11
Figura 2-2: Número de dispositivos móveis vendidos por ano. Fonte: Statista, 2016.....	16
Figura 3-1: Fases da Pesquisa-Ação. Adaptado de Coughlan e Coughlan (2002).....	26
Figura 3-2: Versão Inicial da Abordagem Proposta.....	29
Figura 4-1: Fases da Pesquisa-Ação – Ciclo 1.	35
Figura 4-2: Versão ajustada da abordagem proposta	36
Figura 4-3: Número de vezes que os estudantes usaram o CBL no Ciclo 1.....	44
Figura 4-4: Efetividade do CBL no Entendimento dos Desafios Propostos no Ciclo 1.	44
Figura 4-5: Efetividade das Questões Guias na Construção das Soluções do Ciclo 1.....	45
Figura 4-6: Efetividade do CBL na Melhoria das Habilidades de Programação no Ciclo 1. ..	46
Figura 4-7: Rede Temática de Aspectos Positivos, Negativos e Inconclusivos do Ciclo 1. ...	49
Figura 4-8: Rede Temática de Sugestões de Melhoria e de Outros Métodos no Ciclo 1.	52
Figura 5-1: Ciclo 2 da Pesquisa-Ação.	55
Figura 5-2: Abordagem IDEAS.....	57
Figura 5-3: Detalhamento da Etapa 3 da Abordagem IDEAS: Produzir Software	64
Figura 5-4: Trecho de WBS do projeto Chama Frete.	70
Figura 5-5: Trecho de Diagrama Gantt do Projeto ChamaFrete.	71
Figura 5-6: Quadro SCRUMM do Projeto ChamaFrete.....	73
Figura 5-7: Trecho de Relatório do Gitinspector	74
Figura 5-8: Exemplo de Diagrama Gantt - Classe da Noite da Turma 2015.....	76
Figura 5-9: Rede Temática de Sugestões de Melhorias dos Docentes da PUCPR - Ciclo 2.....	84
Figura 5-10: Rede Temática dos Docentes de Outras Universidades - Ciclo 2.	85
Figura O-1: Cronograma da Pesquisa-Ação	131

LISTA DE QUADROS E TABELAS

Quadro 2-1: Comparativo CBL x Outras Metodologias Ativas. O autor.	18
Tabela 3-1: Estudantes selecionados para a primeira turma.	30
Tabela 4-1: Egressos da primeira turma	43
Tabela 5-1: Estudantes x Cursos Ciclo 2 - Turma 2015.....	59
Quadro 5-2: Guia para a Etapa: Questões Guias, Atividades e Recursos.....	67
Quadro 5-3: Exemplos de Projetos Propostos para o Desafio Final da Turma 2015.....	69
Quadro 5-5: Dificuldades na Etapa de Implementação do Desafio Final da Turma 2015.	75
Quadro 5-6: Cronograma de Execução dos Projetos da Turma 2015.	78
Quadro 5-7: Quadro Comparativo dos Ciclos (Questionários dos Estudantes).....	82
Tabela M-1: Temas Básicos x Temas Organizadores.	128

LISTA DE ABREVIATURAS E SIGLAS

CBL	Challenge Based Learning
GDC	Game Developers Conference
IDEAS	Inovação, Desenvolvimento e Ensino em Ambientes Complexos de Software
iOS	Sistema Operacional de iPhones e iPads
MVP	Minimum Viable Product
NAJA	Núcleo Acadêmico de Jogos e Animação
PBL	Problem Based Learning
PjBL	Project Based Learning
PMBOK	Project Management Body of Knowledge
PUCPR	Pontifícia Universidade Católica do Paraná
SBGames	Simpósio Brasileiro de Games e Entretenimento Digital
SWOT	Planning Method: Strengths, Weaknesses, Opportunities and Threats
WBS	Work Breakdown Structure

CAPÍTULO 1 - INTRODUÇÃO

What you do makes a difference and you have to decide what kind of difference you want to make.

Jane Goodall.

O mercado de dispositivos móveis vem se desenvolvendo com taxa acelerada a cada ano, criando uma forte demanda por aplicativos pessoais e empresariais [LEWIS et al., 2013]. Essa demanda tem como consequência direta, a necessidade crescente por uma formação sólida de desenvolvedores de aplicativos para dispositivos móveis (conhecidos por Apps), que ofereça tanto o ensino de programação básica quanto o de desenvolvimento de aplicativos e produção de software para publicação.

Uma formação com essa envergadura apresenta, além das dificuldades comuns encontradas em cursos tradicionais de ensino de programação, desafios relativos às competências necessárias para se entrar no mercado de desenvolvimento profissional de aplicativos, como processos de desenvolvimento de software, gerenciamento de projetos e testes [NAISMITH et al., 2013].

A presente pesquisa se propõe a desenvolver uma abordagem que efetivamente ofereça uma formação completa, permitindo ao estudante a possibilidade de, a partir de suas próprias ideias de produtos, aprender programação, desenvolver e publicar Apps e eventualmente criar *startups* com base nas Apps desenvolvidas.

O ponto de partida da pesquisa foi um projeto de inovação tecnológica que visava suprir tais demandas de formação de desenvolvedores dentro de um ambiente universitário. Esse projeto de formação de desenvolvedores de software, envolveu diversas universidades (inicialmente apenas no Brasil, mas atualmente em outros dois países e com previsão para outros nos próximos anos) e contou com o apoio de uma empresa parceira da área de tecnologia.

A primeira premissa do projeto era de que as atividades deveriam ser orientadas por professores da área de computação, com habilidades e competências

para o desenvolvimento de aplicativos desta natureza e desenvolvidas por estudantes de cursos técnicos, de graduação, de especialização, de mestrado e doutorado.

A segunda premissa era que o método de ensino a ser utilizado deveria necessariamente ser uma metodologia de aprendizagem ativa denominada *Challenge Based Learning* (CBL)[JOHNSON; ADAMS, 2011] e que o ambiente físico deveria ser adaptado para comportar o uso deste método.

A terceira premissa era que os estudantes deveriam desenvolver aplicativos inovadores com apelo profissional para publicação ao término do projeto. Todos os aplicativos desenvolvidos dentro do curso deveriam ser de propriedade intelectual dos próprios estudantes desenvolvedores.

Dentro da PUCPR, a pesquisa foi realizada em dois ciclos de pesquisa ação, conforme explicação no capítulo 3 seguindo o cronograma apresentado no Apêndice O. Cada ciclo englobou uma etapa completa de formação de uma turma, incluindo preparação dos professores, seleção de alunos, execução do curso e desenvolvimento dos aplicativos.

O ambiente utilizado nos dois ciclos é dito complexo devido à grande quantidade de pessoas e fatores de risco envolvidos, oferecendo vários desafios para seu desenvolvimento e gerenciamento. Considerando-se os dois ciclos envolvidos, os números e fatores são os seguintes:

- Uso de um método de ensino desconhecido pelos professores e sem nenhuma literatura referente à sua aplicação no ensino de programação;
- Quase 300 estudantes (entre estudantes regulares e estudantes de equipes de apoio);
- A característica heterogênea dos estudantes, que vinham de cursos de níveis técnico, graduação, especialização, mestrado e doutorado;
- 10 professores (sendo 5 em período integral exclusivo ao curso);
- Pouco mais de 30 professores externos proponentes de projetos;
- Gerenciamento de 4 equipes de apoio nas áreas de usabilidade, interface, ilustração, modelagem 3D, animação, sonorização, trilhas sonoras, inovação e empreendedorismo. Conforme será explicado no

capítulo 4, estas equipes foram fornecedoras de recursos e serviços para todos os projetos;

- 1400 horas de curso, sendo 800 horas para o primeiro ciclo e 600 para o segundo ciclo;
- A obrigatoriedade do desenvolvimento de Apps inovadoras e com apelo profissional, ou seja, aplicativos prontos para serem publicados.
- 125 Apps com apelo profissional finalizadas e publicadas.

Não foi encontrado na literatura caso similar que envolvesse números desta magnitude, entretanto existem estudos com objetivos semelhantes envolvendo escopo menor e menos recursos. Em uma destas pesquisas [SCHARFF; VERMA, 2010], que teve resultados positivos, também foi oferecido curso para desenvolvimento de Apps. Entretanto, foram apenas seis semanas de curso (40 horas aula), havia o pré-requisito de conhecimento de programação na linguagem Java, método tradicional de ensino (aulas expositivas e práticas), não havia apoio de pessoal de outras áreas e o projeto foi único para todas as equipes.

1.1 Motivação

As motivações para a execução deste estudo foram: oportunidade de aplicar um método de ensino diferenciado e relativamente novo na formação de programadores; possibilidade de transformar efetivamente um curso de programação em uma fábrica de software para dispositivos móveis e de criar Apps inovadoras, que resolvessem problemas relevantes dentro desse mesmo curso. Tudo isso dentro de um ambiente heterogêneo composto de estudantes bolsistas, professores em tempo integral, equipes de apoio de áreas não associadas diretamente à programação, recursos financeiros fartos e alta motivação.

Cursos de programação apresentam grandes dificuldades e alta evasão. A aprendizagem ativa coloca o foco na autonomia do estudante, aumentando a motivação e consequente engajamento.

Em geral os trabalhos propostos em sala de aula são eminentemente acadêmicos e não voltados ao mercado. Existe portanto uma grande oportunidade para transformar todo o esforço dispendido pelos estudantes na elaboração dos trabalhos acadêmicos em produtos de mercado.

Os desafios normalmente encontrados em equipes de desenvolvimento de software permanecem escondidos em um ambiente acadêmico, devido à própria natureza dos objetivos do trabalho que nunca ou raramente exigem um produto acabado, preocupando-se mais com as questões de funcionalidade, entendimento dos tópicos estudados e qualidade do código fonte.

O desenvolvimento de software para aplicativos móveis mudou o panorama de publicação, permitindo a pequenos desenvolvedores uma grande visibilidade a um custo relativamente baixo. Entretanto, com mais de um milhão de Apps nas lojas de aplicativos, tornou-se difícil atualmente manter-se por mais de um dia na página inicial com algum destaque. Torna-se, portanto, necessário e imprescindível a criação de Apps inovadoras que possuam apelo suficiente para se tornarem populares.

1.2 Objetivos

Para atender às necessidades de formação de desenvolvedores para dispositivos móveis, utilizando aprendizagem ativa e o desenvolvimento de um software inovador e publicável, o objetivo deste trabalho é: **Propor uma abordagem para um ambiente complexo de formação de desenvolvedores de aplicativos para dispositivos móveis envolvendo aprendizagem ativa e inovação.**

Para atingir o objetivo geral, os seguintes objetivos específicos foram delineados:

- Desenvolver uma abordagem para ensino de programação e desenvolvimento de software inovador.
- Avaliar a abordagem proposta em um contexto real de ambiente complexo.

1.3 Delimitação de escopo

A abordagem proposta está necessariamente limitada pela empresa parceira ao uso do método CBL e ao ensino de programação para seus dispositivos móveis.

A abordagem necessita de um ambiente adequado para funcionar, sendo prejudicada caso este não exista.

Não existem elementos suficientes para extrapolar a presente abordagem para outras necessidades que não sejam as de desenvolvimento de software.

1.4 Processo de trabalho

Para a execução do estudo foi utilizada a metodologia da pesquisa-ação e as etapas definidas na Figura 1-1.

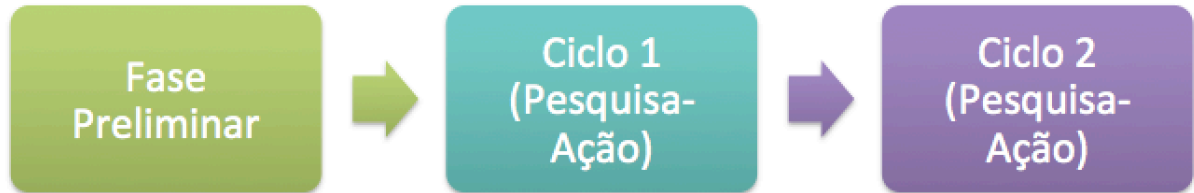


Figura 1-1: Fases da execução da pesquisa.

- **Fase Preliminar – Contexto e Objetivo:** fase que correspondeu ao entendimento inicial do projeto, seleção de estudantes e preparação dos professores.
- **Ciclo 1 – Turma 2013-2014:** fase na qual foi executada a primeira abordagem desenvolvida e a partir de avaliações periódicas foram feitas alterações imediatas. Também foi realizada uma avaliação final que originou outros resultados utilizados para novas alterações na etapa seguinte.
- **Ciclo 2 – Turma 2015:** fase na qual a abordagem atualizada e expandida com os resultados do Ciclo 1 foi executada e novamente avaliada.

1.5 Estrutura do documento da tese

O trabalho foi estruturado em seis capítulos, cuja organização está representada na Figura 1-2.

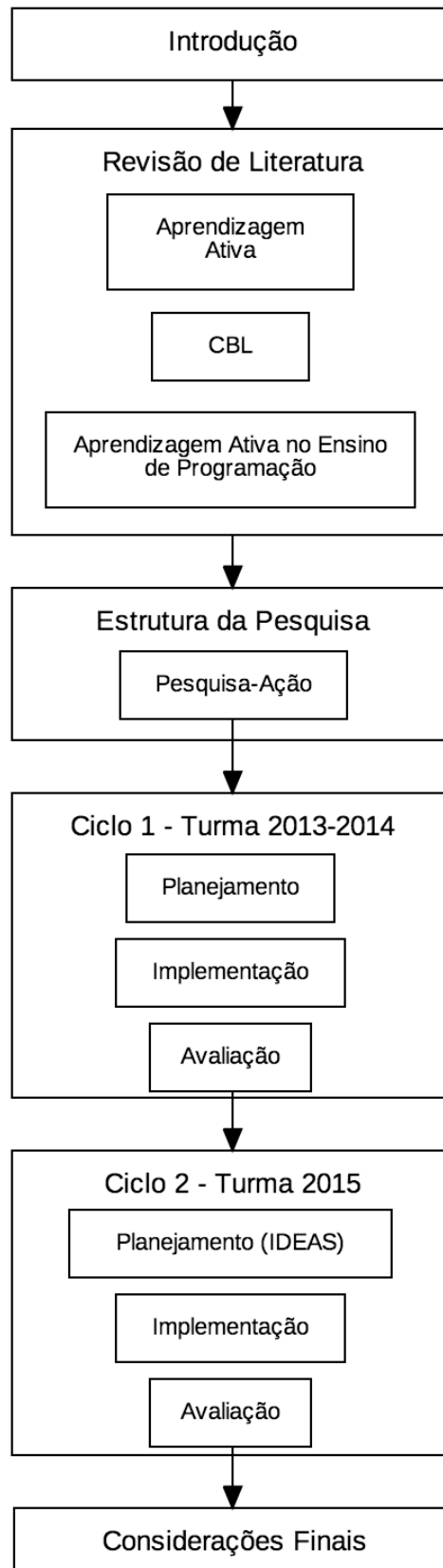


Figura 1-2: Organização dos Capítulos da Tese.

- O Capítulo 1: Introdução, aqui apresentada, contextualiza o problema, descreve as etapas da pesquisa e identifica os objetivos geral e específicos.
- O Capítulo 2: Revisão da Literatura apresenta a base teórica descrita inicialmente no Capítulo 1, focando nos temas: *Challenge Based Learning*, Aprendizagem Ativa, Desafios no Desenvolvimento de Software e Trabalhos Relacionados.
- O Capítulo 3: Estrutura da Pesquisa, que descreve a metodologia de pesquisa usada (pesquisa-ação) e identifica as etapas de pesquisa realizadas.
- O Capítulo 4: Ciclo 1 – Turma 2013-2014, apresenta a execução do primeiro ciclo de pesquisa-ação e os resultados obtidos.
- O Capítulo 5: Ciclo 2 – Turma 2015, apresenta a execução do segundo ciclo da pesquisa-ação no qual foi utilizada a abordagem: Inovação, Desenvolvimento e Ensino em Ambientes Complexos de Software (IDEAS), elaborada a partir dos resultados do ciclo 1. Também são discutidos os resultados alcançados por este ciclo da pesquisa.
- O Capítulo 6: Considerações Finais, que conclui este trabalho, apresentando a relevância do estudo, suas contribuições, limitações da pesquisa e trabalhos futuros.
- Os Apêndices A até N detalham formulários, roteiros de pesquisa, listas de produtos desenvolvidos e provas aplicadas durante o decorrer deste trabalho.

1.6 Considerações sobre o capítulo

Este capítulo apresentou a motivação para a realização deste estudo, os objetivos da pesquisa, seu escopo, estrutura da pesquisa e organização dos capítulos da tese.

CAPÍTULO 2 - REVISÃO DA LITERATURA

The excitement of learning separates youth from old age. As long as you are learning you're not old.

Rosalyn S. Yalow

Neste capítulo serão apresentados e contextualizados os conceitos utilizados ao longo deste trabalho: métodos de aprendizagem ativa, CBL e ensino de programação.

2.1 Metodologias Ativas

Metodologias ativas são uma forma de ensino com ênfase em qualidades construtivistas [NIEMI, 2002] que vem tendo destaque globalmente, sendo incluídas de forma recorrente em políticas educacionais de longo prazo em diversos países [DREW; MACKIE, 2011]. E apesar de não resolverem todos os problemas da educação, apresentam uma multiplicidade de fontes com resultados melhores do que o modelo de aulas expositivas [MICHAEL, 2006][CANALETA et al.,2014], principalmente entre educadores que buscam meios alternativos de ensino para aumentar o interesse dos estudantes [PRINCE, 2004]. Um recente meta-estudo [FREEMAN et al, 2014] feito com 225 pesquisas identificou uma melhora de 6% nos testes acadêmicos em turmas que utilizaram aprendizagem ativa e 50% mais reprovações em turmas sem o uso deste método.

Podem ser definidas como qualquer método de ensino que exija do estudante tanto a execução de atividades significativas quanto uma reflexão sobre o que foi feito [BONWELL; EISON, 1991], evitando assim a passividade do recebimento de informações muito comum em aulas tradicionais. Possuem as seguintes características principais [VILLARINI, 1998] [BONWELL; EISON, 1991]:

- O estudante está engajado no processo, ou seja, a aprendizagem é significativa para ele;

- O estudante não é um agente passivo, ele porta-se de forma ativa e reflexiva;
- O ambiente é necessariamente colaborativo, havendo trocas de informações, discussões e trabalho em conjunto com outros estudantes;
- Está ligada com a realidade, ajudando a desenvolver competências e habilidades que permanecem com o decorrer do tempo.

O objetivo principal da aprendizagem ativa é o de engajar os estudantes em atividades que os façam pensar criticamente sobre o conteúdo estudado. Por meio de um comportamento ativo, os estudantes são preparados para lidar com questões e desafios profissionais, que normalmente exigem habilidades de avaliação e resolução de problemas. Existem vários métodos com diferentes abordagens mas com características comuns: o impacto ativo do aprendiz e seu envolvimento no processo de ensino-aprendizagem [NIEMI, 2002]. Alguns dos métodos de aprendizagem ativa que possuem alguma similaridade com o CBL são:

- **Problem Based Learning (PBL)**: utiliza problemas da vida real para engajar os estudantes em um processo de busca de conhecimento e de soluções. Os professores atuam como facilitadores do processo, nunca oferecendo respostas prontas. [BOUD; FELETTI, 1999]
- **Project Based Learning (PjBL)**: inicia com um produto que requer determinados conhecimentos para o seu entendimento, fornecendo assim um contexto prático para engajar os estudantes no estudo. [MOALOSI et al., 2012]
- **Sala de Aula Invertida**: é a inversão de uma aula tradicional, ou seja, os estudantes estudam os materiais fora da sala de aula enquanto as horas de aula são utilizadas para assimilar os conhecimentos por meio de resolução de problemas e discussões. [TUCKER, 2012]

Outros métodos instrucionais são usados total ou parcialmente pelos definidos acima (incluindo o CBL): aprendizagem colaborativa, onde estudantes trabalham juntos para alcançar um objetivo comum; e aprendizagem cooperativa que se assemelha ao conceito anterior mas com uma diferença importante: a avaliação individual [PRINCE, 2004].

Aparentemente, metodologias de aprendizagem ativa podem ser utilizadas para ensinar qualquer tipo de conteúdo, devido ao grau de motivação que proporcionam [FREEMAN, 2014]. Pelo seu caráter eminentemente prático, o ensino de programação e desenvolvimento de software pode se beneficiar da sistematização e organização que uma metodologia oferece. A próxima seção apresenta o uso de metodologias não tradicionais para o ensino de programação de computadores.

2.2 O método CBL

Conforme visto no Capítulo 1, o projeto deveria ser desenvolvido utilizando-se o método de aprendizagem ativa CBL. Ele surgiu como subproduto de uma iniciativa colaborativa de larga escala que começou em 2008, denominada *Apple Classroom of Tomorrow – Today*, cujo objetivo principal foi o de identificar os princípios essenciais para projetar ambientes educacionais no século XXI [JOHNSON; ADAMS, 2011].

Pode ser definido como uma abordagem motivadora, colaborativa e multidisciplinar que incentiva o uso de tecnologias comuns para aquisição de conhecimento e solução de problemas do mundo real.

Os princípios norteadores do método são [JOHNSON et al., 2009]:

- Um ambiente para reflexão investigativa sobre ensino e aprendizagem;
- Framework flexível com múltiplos pontos de entrada;
- Modelo escalável não proprietário;
- Foco em desafios globais com soluções locais;
- Conexão autêntica entre disciplinas acadêmicas e experiência do mundo real;
- Framework para desenvolver habilidades do século XXI;
- Oportunidade para os estudantes fazerem algo importante agora, antes de terminarem sua educação formal;
- Um processo que coloca o estudante como responsável pelo seu aprendizado.
- Requer que os estudantes desenvolvam e implementem soluções em um ambiente autêntico (real).

2.2.1 Etapas do CBL

O método CBL, representado na Figura 2-1, tem como principal foco os estudantes, a comunidade de ensino-aprendizagem e as habilidades comuns do século XXI (produção de vídeo, presença em redes sociais, uso de *smartphones* para compartilhamento, entre outros).

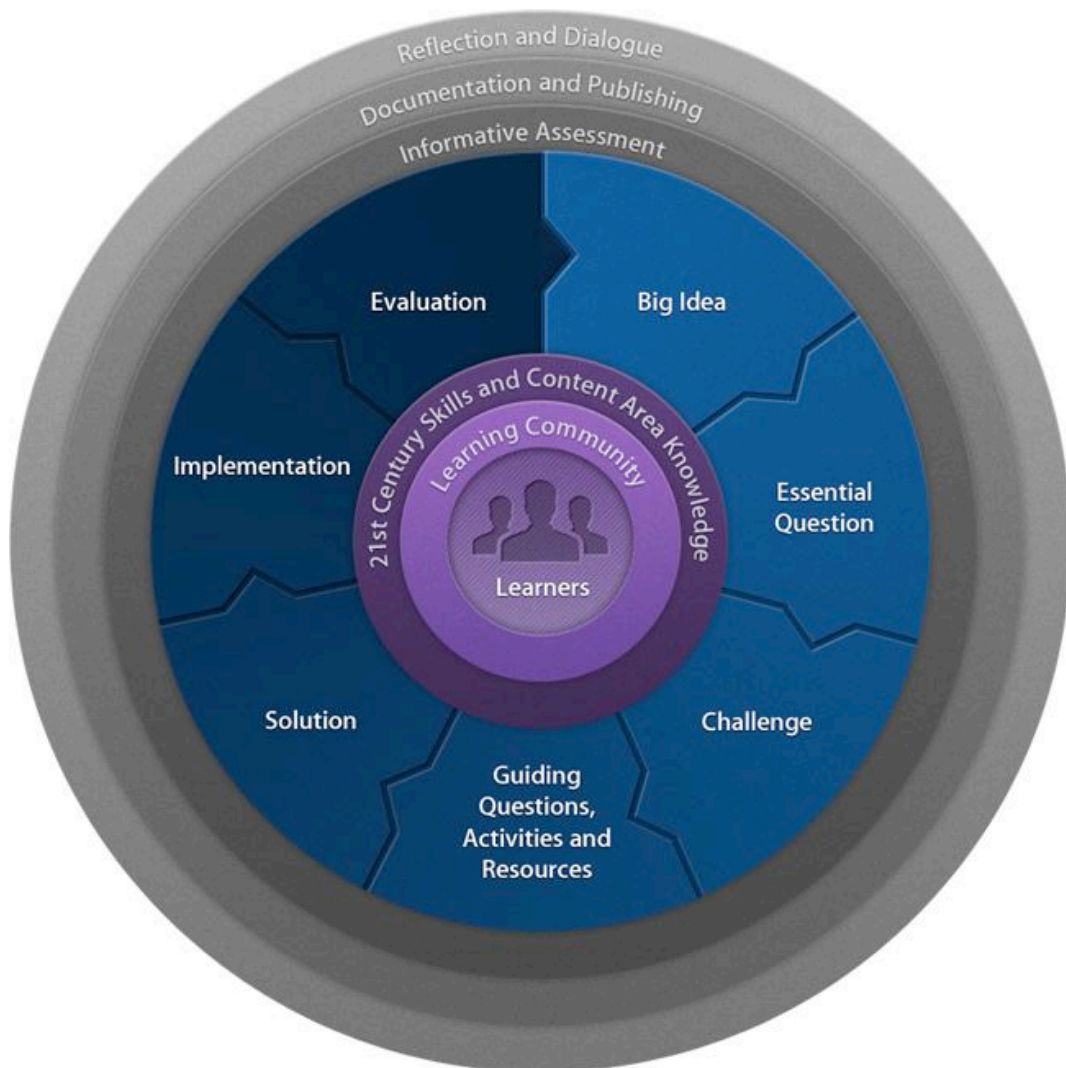


Figura 2-1: O Método *Challenge Based Learning* [CBL; 2011].

O método possui um conjunto de etapas cujo objetivo principal é auxiliar a resolver um problema crítico global por meio de ações locais. Durante todas as etapas são executadas ações de avaliação, documentação e reflexão.

As etapas do método são executadas sequencialmente, mas é possível iniciar o processo em qualquer uma das quatro primeiras etapas. Como o método foi criado para ser utilizado de forma colaborativa, devem ser montadas equipes antes de iniciar o trabalho. As etapas são as seguintes: [JOHNSON et al., 2009]

- a) **Grande Ideia (*Big idea*):** É o tema do futuro projeto, um conceito geral, que pode ser explorado de várias formas. Pode representar tanto um campo da ciência, como “Saúde”, quanto um conceito mais abstrato como “Honestidade”. Este conceito é utilizado em todas as reflexões seguintes e representa o alvo de qual área será afetada pelo projeto.
- b) **Questão Essencial (*Essential Question*):** Após a definição do tema (Grande Ideia), um conjunto de questões que refletem problemas correntes globais referentes ao tema são elaboradas. Após reflexão e diálogo, os membros da equipe devem reduzir essas questões a apenas uma questão principal que se pretende resolver, por exemplo, “como é possível auxiliar o diagnóstico de doenças oculares?”
- c) **Desafio (*Challenge*):** Trata-se da articulação de um desafio local derivado dos problemas globais identificados na etapa anterior. Ou seja, da articulação do problema em uma frase simples, clara e que possa servir de guia para a equipe. Por exemplo, “auxiliar no diagnóstico de doenças oculares” ou “melhorar o atendimento em restaurantes”.
- d) **Questões Guias, Atividades Guias e Recursos (*Guiding Questions, Guiding Activities and Resources*):** Uma vez que o desafio já é claro e definido, levanta-se um conjunto de questões guias (*guiding questions*) que visam entender melhor o problema e quais habilidades são necessárias para resolvê-lo. Por exemplo “quais são os tipos mais comuns de doenças oculares?” ou “quais são os fatores de satisfação de clientes de restaurantes?” ou ainda “como são diagnosticadas as doenças oculares?”. As questões podem ser levantadas numa sessão de brainstorming, entre todos os membros da equipe. Deve-se ressaltar que essas questões devem cobrir todo o espectro de conhecimento necessário, tanto para o entendimento do problema, quanto para o entendimento do que é necessário para resolvê-lo. Com base nessas questões, um conjunto de Atividades Guia é elaborado. Tratam-se de atividades necessárias para a realização do projeto, ou que auxiliam a responder questões que, mesmo após pesquisa, permanecem em aberto. Por exemplo, a questão “quais são os fatores de satisfação de clientes de restaurantes” pode levar as atividades guia de entender melhor quais segmentos de restaurante existem e de elaborar pesquisas com os clientes desses locais. Finalmente, levantam-se quais são os recursos necessários

para que as atividades sejam realizadas, como por exemplo, livros, cursos, profissionais especialistas, experimentos, desenvolvimento de protótipos, etc.

- e) **Solução (*Solution*)**: Conceituação da solução. Deve-se descrever como a solução funcionará, qual ponto específico do problema ela resolve e quais ela não resolve. É importante que a solução seja pensada, concreta e aplicável a uma comunidade. Aspectos como formas específicas de cobrança ou aspectos de negócio não precisam estar completamente definidos nessa etapa. A solução precisa estar bem definida, por exemplo: “Para melhorar o diagnóstico de doenças do fundo do olho, desenvolveremos uma lente que será acoplada ao iPhone. Usando a câmera e uma aplicação desenvolvida por nós, a coleta será realizada em comunidade remota e os dados ficarão armazenados no telefone, até que possam ser enviados a um médico. A aplicação também deverá processar as imagens e gerar parâmetros, que ajudem o médico a realizar um diagnóstico mais preciso”.

- f) **Implementação (*Implementation*)**: Construção da solução. A solução deve ser construída por quaisquer meios necessários e aplicada em um ambiente real. A solução não é necessariamente um produto de larga escala com longo tempo de construção, podem ser pequenas atividades executadas em um curto espaço de tempo, dependendo do escopo e dos objetivos do projeto.

- g) **Avaliação (*Evaluation*)**: Avaliação da solução. Aqui a solução construída deve ser avaliada por métodos diversos que estejam de acordo com a solução construída, como mensurações, entrevistas, gravação de vídeos, etc.

Essas etapas são constantemente descritas, monitoradas e analisadas a partir de três aspectos opcionais: [JOHNSON et al., 2009]

- a) **Avaliação Informativa (*Informative Assesment*)**: tanto o processo de aprendizagem quanto os produtos gerados a cada etapa devem ser avaliados para se confirmar se o aprendizado está sendo efetivo.

- b) **Documentação e Publicação (*Documentation and Publishing*)**: a cada etapa do processo, os estudantes devem ser incentivados a documentar e publicar suas reflexões e informações sobre sua experiência.

- c) **Reflexão e Diálogo (*Reflection and Dialogue*)**: também a cada etapa os estudantes devem refletir sobre o que aprenderam e a forma como aprenderam, tentando entender como se realizou o processo de aprendizado. Além disso, deve-se conversar e trocar ideias com outros estudantes.

Assim como em outras metodologias ativas, essas etapas mostram que o método CBL utiliza estratégias focadas principalmente na autonomia do estudante. Todas elas convergem para a solução do desafio identificado e elaborado pelo próprio estudante.

O método CBL tem sido usado recentemente com relativo sucesso para ensinar tópicos diversos: composição de materiais para fabricação de aviões [O'MAHONY et al, 2012] e inglês como língua materna [CATALINA et al., 2013], bioquímica [GABRIEL, 2014], programação [SANTOS et al, 2015] e enfermagem [CHENG, 2016].

O caráter flexível e aberto do CBL gera uma necessidade de integração com outros métodos e técnicas. Essa integração pode ser feita em qualquer uma de suas etapas e, mesmo não sendo obrigatória, é encorajada pelos criadores do método. [JOHNSON; ADAMS, 2011] Resultados bem sucedidos foram reportados na integração com o Design Thinking [CATALINA, 2013] com o objetivo de facilitar a geração de ideias durante as primeiras etapas e com SCRUM [SANTOS et al, 2015] para prover uma forma de gerenciamento de projetos durante a etapa de elaboração da solução. Como será visto nos próximos capítulos, no presente trabalho o CBL foi integrado com o método de pesquisa-ação para a realização da pesquisa; com um processo de desenvolvimento de software adaptado de métodos ágeis e cascata para guiar os estudantes durante suas atividades de produção de software.

Além métodos formais, mudanças pontuais também são possíveis, tais como reordenação de suas etapas iniciais e direcionamento no formato do estudo [GABRIEL, 2014]. No presente estudo também foram feitas diversas mudanças pontuais, tais como pesquisa extensiva de trabalhos semelhantes, workshops de assuntos específicos, aulas de reforço, entre outros.

2.3 Metodologias Ativas no Ensino de Programação

2.3.1 Uso de Metodologias não Tradicionais no Ensino de Programação

A motivação para se utilizar metodologias não tradicionais vem da observação e experimentação do comportamento de estudantes em aulas tradicionais. Existe uma relação entre o tipo de estímulo recebido e o engajamento dos estudantes durante uma aula. Isso é comprovado pela experiência em sala de aula onde os estudantes normalmente preferem fazer do que escutar [ROSIENE; ROSIENE, 2015].

Metodologias não tradicionais de ensino têm sido usadas há vários anos em cursos de programação com resultados mistos, principalmente as metodologias PBL [KINNUNEN; MALMI, 2005], sala de aula invertida [ROSIENE; ROSIENE, 2015] e ambos simultaneamente [FASSBINDER et al., 2015].

Pelo lado positivo, vários fatores apareceram em tais pesquisas: a motivação dos estudantes e conseqüente engajamento, mantém-se alta durante a aula; a possibilidade dos estudantes seguirem seu próprio ritmo; a colaboração entre os estudantes acontece naturalmente devido à natureza orgânica e dinâmica das atividades práticas; a proximidade entre o professor e os estudantes; e finalmente, o tempo de aula melhor aproveitado, sendo usado para orientação e saneamento de dúvidas enquanto conteúdos cuja assimilação depende de uma simples leitura são estudados fora da aula.

Os aspectos negativos encontrados orbitam entre o comportamento dos estudantes e o esforço necessário para que as aulas sejam produtivas: os estudantes têm necessariamente que se preparar para as aulas; os estudantes devem fazer um esforço adicional, pois há muito mais trabalho a ser realizado; os estudantes devem ser auto motivados [BERBEL, 2012]; o professor também tem mais trabalho do que teria se as mesmas aulas fossem ministradas com métodos tradicionais [NIEMI, 2002], tanto na preparação pessoal, quanto na preparação de materiais e acompanhamento dos estudantes durante as aulas.

Embora o método CBL tenha sido uma exigência da empresa parceira, é necessário entender sua adequação ao ensino de programação. Sua natureza flexível e dinâmica e sua relativa similaridade com as metodologias ativas que já foram utilizadas para ensino de programação são evidências de que o CBL poderia

ser usado no ensino de programação. A próxima seção apresenta um panorama dos desafios de desenvolvimento de software para dispositivos móveis.

2.3.2 Desafios no Desenvolvimento de Software para Dispositivos Móveis

O desenvolvimento de aplicativos para dispositivos móveis é um processo relativamente novo, normalmente possuindo um ciclo de vida menor do que softwares para computadores do tipo desktop/laptop ou web e tendo como forma de distribuição a pré-instalação no dispositivo ou o *download* de uma loja de aplicativos [FLORA; CHANDE, 2013].

Desde a sua criação em 2007, a venda de dispositivos móveis do tipo *smartphone* vem tendo crescimento todos os anos conforme mostra a Figura 2-2, sendo estimado que em 2017, 34% da população mundial possuirá um *smartphone* [STATISTA, 2016].

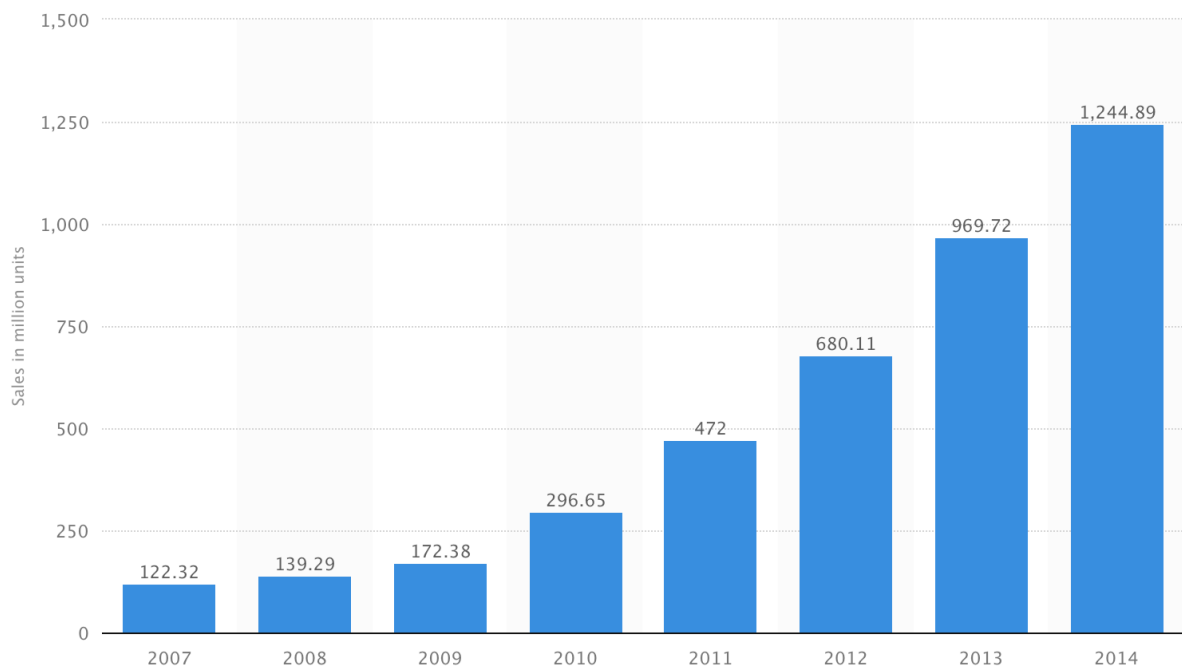


Figura 2-2: Número de dispositivos móveis vendidos por ano. Fonte: Statista, 2016.

Devido a esse crescimento, as aplicações para dispositivos móveis estão se tornando uma parte importante do portfolio de software de missão crítica das empresas [LEWIS et al., 2013]. Por isso, a demanda pelo desenvolvimento de aplicativos específicos e a formação de desenvolvedores para esses aplicativos continuará a ser crescer nos próximos anos.

O principal desafio para os cursos de desenvolvimento de aplicativos para dispositivos móveis é como preparar os estudantes para entrarem neste mercado em crescimento [NAISMITH et al., 2013]. Atualmente são necessários um conjunto de habilidades e competências que vão muito além do desenvolvimento de software [HAWTHORNE, 2005].

Estes desafios vêm ao encontro ao objetivo da pesquisa do presente trabalho, no qual a formação dos estudantes foi pensada de maneira completa, iniciando com o processo seletivo, uso de uma metodologia ativa de ensino (CBL), imersão em um processo de desenvolvimento de software e culminando com a criação de *startups* alavancadas pelo aplicativo produzido. A próxima seção apresenta um comparativo com o objetivo de identificar as vantagens e desvantagens do método CBL.

2.3.3 Comparativo Metodologias Ativas x CBL

Uma das principais diferenças entre o método CBL e outras metodologias ativas é o objetivo do estudo, ou seja, qual é o resultado final desejado pela aplicação do conjunto de atividades propostas em sala.

O CBL tem como foco, a aquisição de determinadas habilidades por meio do desenvolvimento de uma solução para um problema real **identificado e proposto pelo estudante**. Os conteúdos estudados surgem das necessidades do desafio proposto, ou seja, conteúdos que não serão úteis para o desafio, não serão estudados.

Nas outras metodologias, usualmente são propostos conteúdos (sala de aula invertida)[GILBOY et al, 2015] ou problemas associados a determinados conteúdos (PBL e PjBL) [ROBINSON, 2013] e em seguida atividades práticas que envolvam tais conteúdos. Ou seja, enquanto no CBL a atividade prática (o desafio) define o conteúdo que deve ser estudado, nas outras metodologias ocorre o contrário, é definido primeiro o conteúdo de estudo e as atividades práticas são relacionadas a este conteúdo.

Considerando-se que em cursos que utilizem o método CBL, os estudantes definem os próprios projetos, pode ocorrer de estudantes diferentes estarem estudando tópicos diferentes para resolverem seus desafios.

O Quadro 2-1 apresenta um comparativo entre as características do CBL e dos outros métodos citados neste trabalho: PBL, PjBL e sala de aula invertida. O critério para a escolha das características baseou-se nas características mais

relevantes para o ensino de programação de acordo com a observação feita pelo pesquisador no caso do CBL e na análise dos estudos de metodologias ativas aplicadas em programação no caso dos outros métodos.

Quadro 2-1: Comparativo CBL x Outras Metodologias Ativas. O autor.

Características / Métodos	CBL	PBL	PjBL	Sala de Aula Invertida
Estudo Antecipado				x
Todo o conteúdo é pré-definido		x	x	x
Desafio pré-definido		x	x	x
Aprendizado no próprio ritmo	x	x	x	x
Incentivo à Colaboração	x	x	x	x
Incentivo à Autonomia	x	x	x	x
Requer Pensamento Crítico	x	x	x	x
Requer Resolução de Problemas	x	x	x	x
Requer <i>Feedback</i>	x	x	x	x
Requer Revisão	x	x	x	x
Tema e Desafio definidos pelo Estudante	x			
Necessidade de Ambiente Específico	x			
Implantação da Solução Desenvolvida	x			
Requer Autorreflexão Frequente	x			

Dentre as características levantadas, sete delas aparecem em todos os métodos e representam uma reação ao que é normalmente criticado em métodos tradicionais de ensino: autonomia, colaboração, pensamento crítico, resolução de problemas, aprendizado no ritmo do estudante, *feedback* e revisão. É interessante notar que, com exceção do item relativo ao ritmo do estudante, são justamente algumas das características necessárias para o trabalho de um desenvolvedor de software:

- Autonomia: para se manter atualizado com as mais recentes novidades em termos de desenvolvimento, linguagens de programação e novos dispositivos.
- Colaboração: desenvolver software não é normalmente uma tarefa solitária, exigindo interações com pessoas de diversas funções.
- Pensamento Crítico: necessário em vários aspectos do desenvolvimento de software, como por exemplo, qual algoritmo usar, quais as melhores decisões arquiteturais, etc.
- Resolução de problemas: a construção de um software é em si uma forma de resolver problemas.

- *Feedback*: sempre é importante receber críticas e sugestões sobre o estado atual do software desenvolvido.
- *Revisão*: boas práticas de programação sugerem que a revisão do código é uma boa forma de se encontrar erros.

Ou seja, pode ser considerado que tais métodos de ensino estão alinhados com várias habilidades necessárias aos desenvolvedores de software.

Os aspectos únicos do método CBL são: tema e desafio definidos pelo estudante; necessidade de um ambiente específico; autorreflexão frequente; e implantação da solução desenvolvida. Com relação ao ensino de programação, nenhuma dessas características seria essencial para a formação dos estudantes. Entretanto, todas apresentam vantagens no que diz respeito à motivação e experiência dos estudantes no desenvolvimento de software.

O fato do estudante poder escolher seu tema e seu desafio é um fator de motivação a mais, pois incentiva a liberdade e, principalmente, os possíveis projetos pessoais que muitos estudantes possuem ao iniciar um curso.

O ambiente específico é composto de sala de aula flexível, laboratórios com ilhas de trabalho em equipe, espaços para reflexão individual, capacidade de projeção de telas do computador em televisões de alta qualidade. É um ambiente diferenciado, agradável, de alta qualidade e altamente motivador, fatos comprovados pelos testemunhos e pela presença maciça de estudantes fora dos horários de aula durante a execução desta pesquisa.

A autorreflexão frequente possibilita que o desenvolvedor de software encontre caminhos melhores para a solução de determinados problemas. A análise dos projetos finalizados é uma prática comum em alguns nichos de desenvolvimento, notadamente na produção de jogos digitais, como pode ser facilmente identificado pela leitura das principais revistas da área e das sessões de apresentação de trabalhos das principais conferências acadêmicas/profissionais de jogos, como SBGames no Brasil e GDC nos Estados Unidos.

E finalmente, a necessidade da implantação da solução desenvolvida, obriga o estudante a se preocupar com aspectos relativos à produção de software profissional que não são levados em consideração em cursos comuns. Dentre estes aspectos, podemos citar: qualidade do software como um todo, usabilidade e interface e necessidade de manutenção após a publicação do software.

Essa necessidade de implantação é apresentada pelo método CBL de forma livre e abstrata, sem detalhamento para situações específicas, como é o caso da presente pesquisa, no ensino de programação. Em um curso de programação, essa implantação é bastante desafiadora, envolvendo diversas necessidades de difícil assimilação para estudantes novatos. A próxima seção discute alguns destes desafios encontrados no desenvolvimento de aplicativos móveis.

2.4 Trabalhos Relacionados

O presente trabalho possui diversas características que, quando consideradas em conjunto, o tornam complexo: engloba o tripé fundamental do sistema universitário brasileiro (ensino, pesquisa e extensão), curso de desenvolvimento para aplicativos móveis de longa duração (800 horas no primeiro ciclo, 600 horas no segundo ciclo); curso com aulas complementares sobre assuntos de apoio (criatividade, usabilidade, desenvolvimento orientado a testes, desenvolvimento de servidores web); estudantes de diversos níveis de escolaridade estudando o mesmo assunto em uma mesma classe: ensino médio técnico, graduação, especialização, mestrado e doutorado; utilização de uma metodologia de aprendizagem ativa; imersão em um processo de desenvolvimento de software; bolsa de estudos com valor superior ao praticado por agências de fomento científico e tecnológico; cessão de equipamentos de última geração para trabalhos em sala ou fora dela; quinze horas semanais disponibilizadas exclusivamente para o curso; desenvolvimento e publicação de um aplicativo relevante; equipes de apoio para produção de recursos adicionais e prestação de serviços: interface e usabilidade, arte gráfica em geral, sonorização, dublagem e trilhas sonoras, inovação e empreendedorismo; possibilidade de contratação de serviços externos para outras necessidades não cobertas pelas equipes de apoio: localização, criação de vídeos, fotografia e outros; professores disponíveis para orientação em tempo integral; canal direto de comunicação com profissionais especializados em aplicativos para dispositivos móveis.

Foram realizados estudos com semelhança em alguns dos aspectos listados acima, mas com escala e escopo menores.

Em um trabalho de pesquisa com origem similar, o método CBL foi integrado ao SCRUM [SANTOS et al, 2015] com o objetivo de melhorar a qualidade das apps

desenvolvidas pelos estudantes. Foi realizado um curso de desenvolvimento de aplicativos para iOS com duração de 6 meses para alunos de graduação de diversas áreas. Questionários foram respondidos pelos estudantes em vários momentos do curso para avaliar o uso de CBL e SCRUM.

O resultado principal da pesquisa foi que um ambiente baseado em uma combinação de CBL com SCRUM é uma forma efetiva de ensinar, em um curto espaço de tempo, o desenvolvimento de aplicações para dispositivos móveis. Os dois métodos se completam, SCRUM ajuda na execução dos desafios enquanto o CBL ajuda na melhoria da qualidade dos softwares desenvolvidos. Também foi confirmado que o CBL é efetivo em ambientes tecnologicamente ricos e que as reflexões periódicas de professores e alunos contribuíram para a melhoria do processo de aprendizado.

Dentre as semelhanças com a presente pesquisa estão o uso do método CBL para ensinar programação para dispositivos móveis, preocupação em complementar o CBL com outro método (SCRUM) para melhorar algum aspecto do desenvolvimento de software.

As principais diferenças com o presente trabalho são: tempo de execução menor, foco no curso e não no processo inteiro (seleção dos estudantes, preparação de professores, criação de *startups*), apenas estudantes de graduação participaram do curso, os estudantes não foram imersos em um processo de desenvolvimento de software e, finalmente, avaliação com menor escopo, já que a avaliação foi feita apenas por estudantes de uma turma e não foi feita avaliação com os professores envolvidos.

Outro trabalho foi o curso “Desenvolvimento de Aplicativos para Telefones Móveis e Empreendedorismo” realizado na Universidade Pace [SCHARFF; VERMA, 2010].

As características do curso relacionadas ao presente trabalho foram: seis semanas de duração com dois encontros semanais; o estudante já deveria conhecer programação em linguagem Java; além dos tópicos técnicos relacionados ao desenvolvimento de aplicativos, foi apresentado um panorama geral dos aplicativos baseados em redes sociais com impacto social; o componente de empreendedorismo era composto de uma descrição do mercado dos EUA, e de alguns países emergentes da África e América do Sul e a definição de inovação e empreendedorismo no contexto global; uso do método Scrum de desenvolvimento

em conjunto com as atividades de ensino; e a criação de um aplicativo para resolver um problema real, no caso um sistema de pedidos de restaurantes.

Os objetivos principais do estudo [SCHARFF; VERMA, 2010] eram entender as dificuldades encontradas pelos estudantes no aprendizado de desenvolvimento para aplicativos móveis e explorar a experiência geral dos estudantes com relação ao Scrum: curva de aprendizado, percepção e dificuldades.

O resultado foi considerado positivo por todos os envolvidos no processo: estudantes e instrutores que foram divididos em dois papéis: Scrum Master e cliente. Foram produzidos três versões completas e funcionais do sistema proposto (controle de pedidos de restaurantes), sendo que o grupo considerou o uso do Scrum e o comprometimento dos estudantes como principais fatores para este sucesso. Para os pesquisadores foi demonstrado que o método Scrum pode ser utilizado dentro de uma experiência de sala de aula com estudantes novatos com o objetivo de aumentar as chances de se produzir um software de qualidade e no prazo estipulado. Uma contribuição prática da pesquisa foi um guia para introdução de Scrum em cursos de ensino de programação. [SCHARFF; VERMA, 2010]

Dentre as semelhanças com o presente trabalho, destacam-se a preocupação dos autores em integrar um método de desenvolvimento de software ao curso, o desenvolvimento de um software completamente funcional e a necessidade de se dar noções básicas de empreendedorismo e inovação para os estudantes. Apesar de todos os estudantes já terem um background em desenvolvimento de software, o curto espaço de tempo foi um grande limitador para todos esses aspectos, incluindo a superficialidade das noções de empreendedorismo e o pequeno escopo do projeto.

Com relação aos aspectos diferentes, o presente estudo apresentou as seguintes características principais: tempo maior de curso; projetos escolhidos pelos próprios estudantes; aplicações publicadas e disponibilizadas para o mercado; ensino de programação básica; uso de uma metodologia de ensino não tradicional; recursos produzidos pelas equipes de apoio; noções mais aprofundadas de empreendedorismo; apoio de profissionais especializados; e apoio para criação de *startups*.

2.5 Considerações sobre o capítulo

Este capítulo apresentou um referencial teórico sobre os aspectos ligados a esta pesquisa. Foi apresentado o método CBL, metodologias de aprendizagem ativa, ensino de programação com metodologias ativas, desafios no desenvolvimento de software e trabalhos relacionados.

CAPÍTULO 3 - ESTRUTURAÇÃO DA PESQUISA

*All sorts of things can happen when you're open to new ideas
and playing around with things.*

Stephanie Kwolek

Este capítulo tem por objetivo a definição do método de pesquisa a ser utilizado no presente trabalho. Inicialmente é apresentada a caracterização da pesquisa e em seguida, o método selecionado, bem como suas etapas.

3.1 Caracterização da pesquisa

As características e classificações apresentadas a seguir basearam-se nos estudos de Gil (2002). Esta pesquisa, com relação à natureza, é classificada como aplicada e com relação ao envolvimento do pesquisador é classificada como pesquisa-ação.

Este trabalho é caracterizado como uma pesquisa aplicada, pois define uma abordagem para ambientes complexos que utilizem ensino, inovação e produção de software para dispositivos móveis. Isto gera conhecimento prático dirigido à solução de um problema específico, envolvendo verdades e interesses locais [GIL, 2002].

A pesquisa pretende explorar um ambiente complexo diferenciado, no qual um método de ensino pouco estudado, o CBL, é aplicado especificamente em turmas especiais de desenvolvimento de aplicativos para dispositivos móveis. São chamadas de especiais, pois os estudantes não apenas recebem instrução referente aos conteúdos, mas também são incluídos em um processo de desenvolvimento de software cujo objetivo é produzir (ou seja, desenvolver e publicar) um aplicativo inovador para dispositivos móveis.

Quanto aos procedimentos técnicos, optou-se pela pesquisa-ação devido ao caráter colaborativo do trabalho e à possibilidade do pesquisador atuar de forma concorrente à sua execução [COUGHLAN; COGHLAN, 2002].

3.2 Pesquisa-ação

A pesquisa-ação corresponde a um tipo de pesquisa social com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores estão envolvidos de modo cooperativo ou participativo.

Em Coughlan e Coughlan (2002), os autores descrevem uma série de características da pesquisa-ação, sendo elas:

- Foco na pesquisa em ação: os pesquisadores atuam junto com aqueles que vivenciam o problema. No caso específico do presente trabalho, os próprios pesquisadores também fazem parte da população que vivencia o problema, ou seja, o ensino de desenvolvimento de software em um ambiente complexo de inovação.
- Participativa: o pesquisador participa do processo de mudança junto aos sujeitos (estudantes).
- Pesquisa concorrente à ação: O pesquisador faz reflexões junto com os sujeitos (estudantes) para encontrar soluções para os problemas de aprendizado. As soluções são então aplicadas em seguida, pelos mesmos professores e para os mesmos estudantes.
- Sequência de eventos e abordagem para solução de problemas: a Pesquisa-Ação é tanto um conjunto de ciclos para obtenção de dados quanto uma abordagem ao método científico para análise e experimentação. O resultado da pesquisa não é apenas uma solução para o problema, mas também uma contribuição para o conjunto de conhecimentos da área em estudo.

O pesquisador é um ator dentro do contexto da pesquisa, um agente responsável pela mudança e está imerso no processo que pretende compreender, analisar e modificar.

Estas características devem ser consideradas desde o momento da concepção da pesquisa, compreendendo três fases principais: (i) preliminar; (ii) ciclo de condução; e (iii) monitoramento (COUGHLAN; COGHLAN, 2002). Os mesmos autores sugerem uma implementação iterativa e incremental para a Pesquisa-Ação conforme mostra a Figura 3-1. Nota-se que, embora a etapa de monitoramento esteja sendo considerada como uma terceira fase, na realidade ela não é sequencial

em relação à segunda. Trata-se de um procedimento contínuo, que permeia toda a execução da pesquisa.

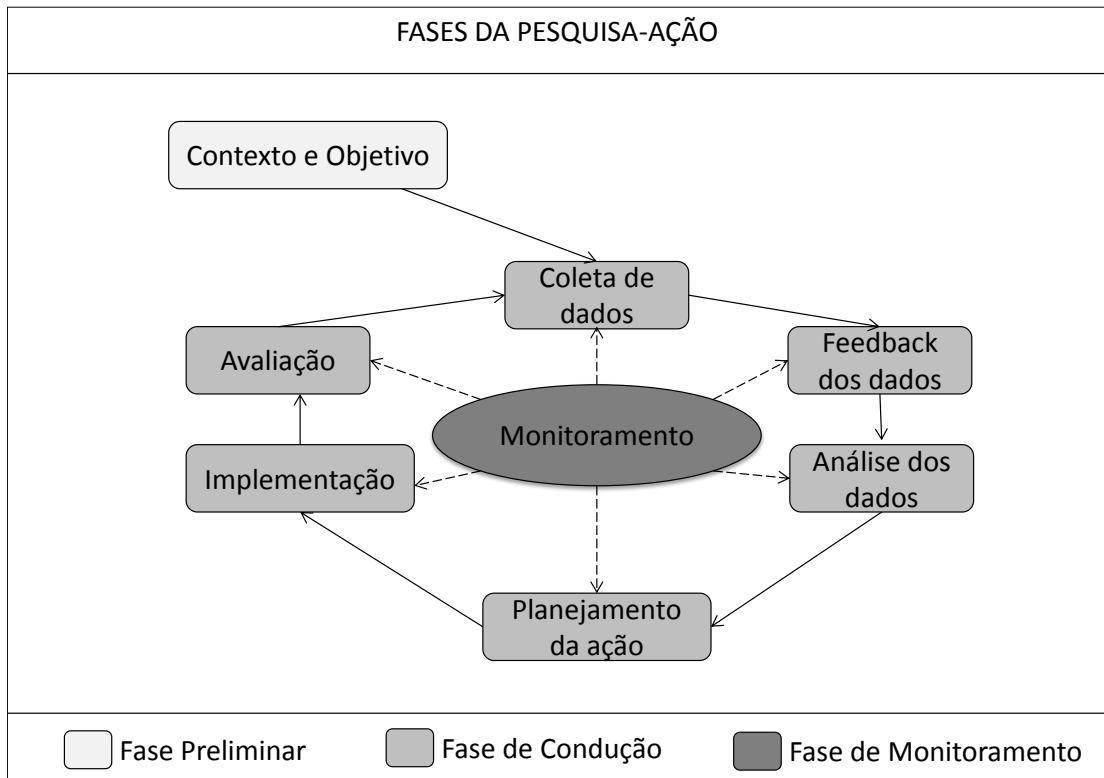


Figura 3-1: Fases da Pesquisa-Ação. Adaptado de Coughlan e Coughlan (2002).

A primeira fase da Pesquisa-Ação é a **Fase Preliminar** que corresponde a um entendimento sobre o contexto no qual a pesquisa será realizada, bem como a definição do objetivo. Essa fase envolve o estabelecimento das justificativas para a ação requerida, além das justificativas para a condução da pesquisa. A partir desta compreensão são planejados os ciclos de realização da pesquisa, que podem ser um ou mais.

Na segunda fase da Pesquisa-Ação estão representados os passos do ciclo de condução da pesquisa propriamente dita, que envolve:

- (i) Coleta de Dados;
- (ii) Feedback dos Dados;
- (iii) Análise dos Dados;
- (iv) Planejamento da Ação;
- (v) Implementação da Ação, e;
- (vi) Avaliação.

Segundo Coughlan e Coughlan (2002), na pesquisa-ação, os dados são gerados por meio do envolvimento com o processo organizacional. Assim, os dados são resultado da participação e observação das equipes de trabalho na execução de suas atividades.

A **Coleta de Dados** pode ocorrer de diversas maneiras, podendo-se obter dados qualitativos, por meio de observação direta, discussões ou entrevistas; ou dados quantitativos, como aqueles obtidos em medições, relatórios e questionários.

Após a Coleta de Dados o pesquisador fornece o **Feedback dos Dados** para a organização analisá-los, o que pode ser feito por meio de relatórios ou reuniões de feedback.

A **Análise dos Dados** é realizada de forma conjunta, ou seja, pesquisador e membros da organização. Esta análise crítica envolve a participação ativa dos membros da organização com o pesquisador, onde cada parte contribui com a sua visão, seja acerca das questões internas da empresa, seja acerca da interação com os conhecimentos do pesquisador.

Na sequência ocorre o **Planejamento das Ações**, que também deve ser realizado em conjunto entre a equipe da organização e o pesquisador. Nesta atividade é estabelecido o que vai ser feito e o prazo. No trabalho de Coughlan e Coughlan (2002), os autores citam questões-chaves que devem ser respondidas, tais como:

- O que precisa mudar?
- Em que partes da organização?
- Que tipos de mudanças são necessárias?
- Qual o apoio é necessário?
- Como compromisso pode ser obtido?
- Como a resistência pode ser gerenciada?

Uma vez que a ação tenha sido definida, ela é então implementada (etapa de Implementação), visando promover as mudanças planejadas. Esta atividade envolve fazer as alterações planejadas em colaboração com os principais membros da organização.

Por fim, a **Avaliação** é o último passo desta segunda fase e é a chave para a aprendizagem. Segundo Coughlan e Coughlan (2002), a avaliação é uma reflexão dos resultados esperados ou não, decorrentes da implementação da ação, visando melhorias para o ciclo seguinte.

Embora, de certa forma, possa parecer que o **Monitoramento** constitua uma terceira fase, na realidade ele é uma atividade contínua, que ocorre simultaneamente às demais etapas. O objetivo é monitorar se as atividades planejadas para o ciclo estão sendo realizadas e se o seu resultado atinge os propósitos definidos.

A pesquisa-ação e metodologias ativas em geral (e o método CBL em particular) possuem alguns aspectos em comum, o que facilitou a sua integração neste trabalho. Duas etapas auxiliares do CBL (reflexão e diálogo; avaliação informativa) podem ser diretamente relacionadas com Coleta e Análise de Dados e o seu Monitoramento.

3.3 Estratégia de pesquisa

Uma vez selecionado o método de pesquisa, foi estabelecida a estratégia a ser utilizada. Inicialmente, foi estabelecido que seriam utilizados dois ciclos de aplicação da pesquisa-ação. Diz-se inicialmente, pois em uma pesquisa-ação, dada a natureza iterativa e interativa do processo, novos ciclos podem se fazer necessários, na medida em que a ação vai sendo desenrolada. Desta forma, os seguintes ciclos foram identificados no primeiro momento:

- Fase Preliminar: identificação e mapeamento da situação da organização
- Ciclo 1 – Turma 2013-2014
- Ciclo 2 – Turma 2015
- Monitoramento: presente em todas as etapas da pesquisa

3.3.1 Fase Preliminar – Contexto e Objetivo

Conforme descrito no Capítulo 1, o cenário de inserção da pesquisa foi um projeto de inovação tecnológica que visava à produção de aplicativos para dispositivos móveis dentro de um ambiente universitário.

A primeira premissa era de que as atividades do projeto deveriam ser orientadas por professores da área de computação, com habilidades e competências para o desenvolvimento de aplicativos desta natureza e deveriam ser desenvolvidas por estudantes de cursos técnicos, de graduação, de especialização, de mestrado e doutorado.

A segunda premissa era que o método de ensino a ser utilizado deveria ser o CBL (*Challenge Based Learning*) e que o ambiente físico deveria ser adaptado para comportar o uso deste método. No primeiro momento este ambiente ainda não estava totalmente adaptado, o que gerou algumas questões que serão discutidas mais adiante.

A Figura 3-2 apresenta a abordagem inicial definida para a organização. Como se pode observar, ela era composta por três fases: Seleção dos Estudantes, Preparação dos Professores e Execução do Projeto.

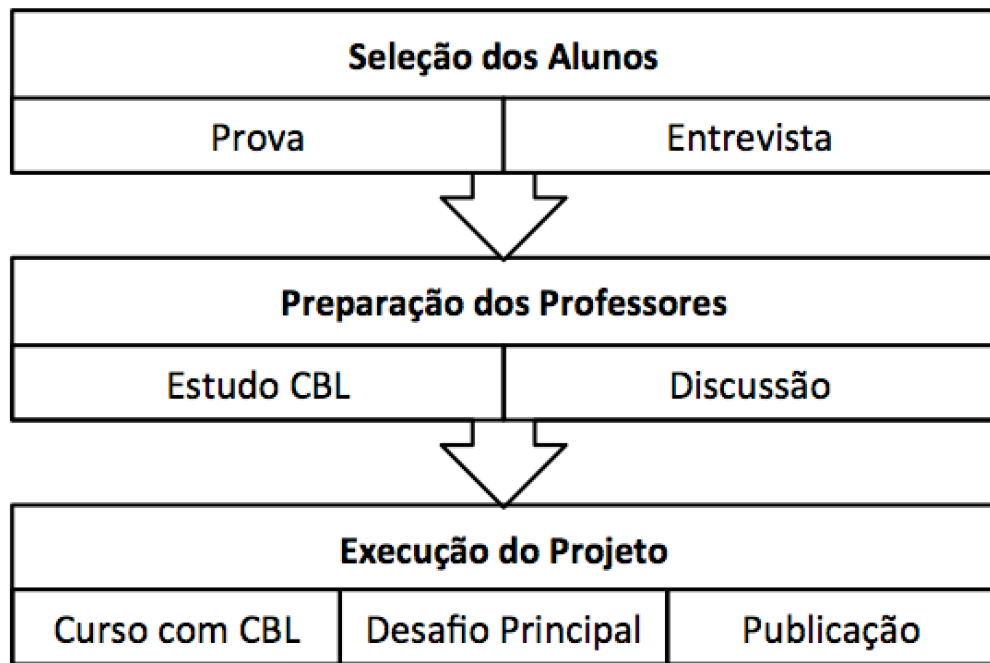


Figura 3-2: Versão Inicial da Abordagem Proposta.

Seleção dos Estudantes

Na primeira fase da abordagem, Seleção dos Estudantes, o objetivo era aplicar técnicas de seleção para a escolha do perfil mais adequado de estudantes para o desenvolvimento do projeto.

Foram aplicadas técnicas convencionais de seleção como Prova (técnica) e Entrevista (básica). A premissa inicial era de que os estudantes deveriam ter boas habilidades em programação básica, o que foi o foco da prova técnica aplicada, conforme apresentado no Apêndice H. A prova foi dividida em duas partes:

programação e raciocínio lógico (constantes, variáveis, condições, repetições e vetores unidimensionais). O conteúdo foi equilibrado no sentido de evitar que estudantes mais avançados tivessem alguma vantagem sobre os demais, eliminando, portanto, possíveis talentos ainda não tão experientes. Foram criadas questões abertas de nível médio e difícil, apenas utilizando os conceitos iniciais de um curso regular de programação. Em todas as questões era solicitado que o estudante elaborasse uma solução com o menor número possível de linhas. Foram atribuídos graus diferentes para respostas corretas, dependendo do critério de número de linhas. A prova foi teórica, ou seja, os algoritmos foram apresentados no papel e não no computador. Desta forma, mesmo não chegando à solução completa, era possível avaliar o raciocínio utilizado. Inscreveram-se nesta etapa 398 candidatos e foram selecionados os 150 melhores classificados de acordo com a nota obtida na prova.

A entrevista objetivava conhecer um pouco mais do perfil do estudante e foi realizada presencialmente com todos os 150 candidatos aprovados na prova técnica. Seu roteiro encontra-se no Apêndice I. Foram aprovados diretamente 72 candidatos. Para os demais 78, devido ao desempenho na prova, foi oferecido um curso de nivelamento na linguagem C++. O objetivo do curso foi nivelar e avaliar melhor os candidatos que não atingiram a nota mínima na prova teórica. Ao término do curso de nivelamento foram considerados aptos 74 candidatos. Foram classificados pela nota no curso de nivelamento e selecionados 38 candidatos, totalizando 110 estudantes para a primeira turma.

Os estudantes tinham origem em diversos cursos, incluindo graduação, pós-graduação lato sensu stricto sensu e ensino técnico, conforme mostra a Tabela 3-1. A maioria destes estudantes era de computação, nos diversos níveis (ensino médio ao doutorado) e apenas 3 estudantes (2,7%) eram de cursos sem disciplinas referentes a programação de computadores.

Tabela 3-1: Estudantes selecionados para a primeira turma.

Curso	Total	%
Ensino Médio Técnico Informática	1	0,9%
Graduação Jogos Digitais	23	20,9%
Graduação Sistemas de Informação	22	20,0%
Graduação Engenharia de Controle e Automação	14	12,7%
Graduação Ciência da Computação	13	11,8%

Graduação Engenharia da Computação	11	10,0%
Graduação Engenharia Elétrica	4	3,6%
Graduação Design Digital	1	0,9%
Graduação Engenharia Civil	1	0,9%
Graduação Engenharia de Produção	1	0,9%
Graduação Medicina	1	0,9%
Especialização Desenvolvimento Web	1	0,9%
Especialização Desenvolvimento de Jogos Digitais	10	9,1%
Especialização Engenharia de Software	7	6,4%
Doutorado Informática	1	0,9%

Estes 110 estudantes foram divididos em cinco classes: três durante a semana nos turnos matutino, vespertino e noturno e duas no sábado no período vespertino. Todas as classes tinham 6 horas de aula por semana e a obrigação de trabalharem até 9 horas fora do horário de aula. Até meados de julho de 2014 foram usadas duas salas de aula especiais. Após este período, com a conclusão do espaço físico definitivo para o projeto, foi utilizado o prédio concebido especialmente para este fim.

Preparação dos Professores

Na segunda fase da abordagem, Preparação dos Professores, o objetivo era capacitar os docentes da instituição que haviam sido escolhidos para participar do projeto, no método CBL. A seleção foi feita por meio da análise do currículo e das avaliações dos professores interessados, sendo escolhidos aqueles com avaliações positivas e com prévia experiência no desenvolvimento de aplicativos para dispositivos móveis. A preparação foi realizada por meio de cursos e leitura dos materiais disponíveis, além de discussões em grupo. Esta atividade foi de vital importância, pois nenhum dos professores envolvidos possuía conhecimento prévio de metodologias ativas em geral e do método CBL em particular. Os cinco professores da PUCPR que foram alocados tinham formação na área de computação: um com doutorado, dois com mestrado, um com especialização e o outro com graduação. Quatro dos professores eram da área de jogos digitais e um da área de animação digital.

Essa preparação foi composta de duas partes: estudo aprofundado do método e discussão de sua aplicação para o ensino de programação (onde se buscou o consenso dos professores para a montagem do plano de ensino).

Os professores realizaram um estudo prévio sobre o método CBL, lendo e analisando os principais artigos do site oficial do método, entre eles o relatório de implementação do método [JOHNSON; ADAMS, 2011]. Em seguida, os mesmos professores fizeram um curso com duração de dois dias, onde o método foi aplicado duas vezes: a primeira pelo próprio instrutor do curso (um dos colaboradores da criação do método) e a segunda em forma de exercício para os grupos formados pelos estudantes participantes. Foi destacado pelo instrutor que o CBL nunca havia sido utilizado de forma contínua e relevante no ensino de programação. A dificuldade principal no aprendizado foi em relação à forma com que o método deveria ser aplicado: Seria para todos os assuntos? Seria apenas nos trabalhos principais? Deveria ser mesclado com algum outro método de ensino?

A próxima parte da fase de preparação foi composta de discussões entre os professores com o objetivo de identificar as melhores práticas para aplicar o método CBL no ensino de programação. Foram feitas reuniões com o intuito de identificar possíveis problemas relacionados ao perfil dos estudantes selecionados. Melhores práticas foram definidas e um plano de ensino preliminar foi elaborado. O plano de ensino foi composto de objetivos gerais sem uma ementa rígida pré-definida. Essa flexibilização foi feita devido a uma das características do CBL que dita ao estudante a definição dos tópicos de estudo, o que é feito usando os assuntos necessários ao projeto que estiver sendo realizado.

Embora houvesse a possibilidade de se utilizar o método CBL sem qualquer tipo de aula ministrada por professores, optou-se por continuar com aulas expositivas por dois motivos. O primeiro refere à importância do papel do professor em despertar interesse nos estudantes, mesmo quando métodos de aprendizagem ativa são utilizados [ROTGANS; ADAMS, 2011]. O segundo é que formas diferenciadas de ensino são necessárias para que o estudante atinja níveis complexos de pensamento e comprometimento [BERBEL, 2012].

Execução do Projeto

Na terceira atividade da abordagem, Execução do Projeto, o objetivo era a preparação dos estudantes por meio do Curso Técnico com CBL, a definição do Desafio Principal e o desenvolvimento e publicação do aplicativo. Ao iniciar a execução da terceira parte, alguns desafios foram identificados, relativos à

complexidade do ambiente, o que motivou a realização desta pesquisa, conforme já descrito anteriormente:

- dificuldade dos estudantes com metodologias ativas, notadamente o método CBL, que nenhum deles havia usado anteriormente;
- ausência de habilidades técnicas específicas como: desenvolvimento para dispositivos móveis; concepção e execução de interfaces gráficas mais elaboradas; utilização de recursos de sonorização, ilustração, animação e modelagem 3D;
- dificuldade dos estudantes com a geração de aplicativos inéditos e relevantes para a comunidade de usuários de aplicativos para dispositivos móveis;
- ausência de experiência anterior com processos de desenvolvimento de software complexos.

Estes desafios foram identificados tanto nas reflexões semanais dos estudantes e dos professores quanto nas observações do andamento das atividades em sala de aula. A partir destas constatações, foram definidos os planos para a realização dos Ciclos 1 e 2 da Pesquisa-Ação, conforme se verá nas próximas seções.

3.3.2 Ciclo 1 – Turma 2013-2014

O Ciclo 1 da Pesquisa-Ação teve início no final de outubro/2013, quando os estudantes já haviam sido selecionados e os professores já haviam sido treinados. Neste momento, as dificuldades já haviam sido mapeadas, ou seja, os passos iniciais da Fase 2 do Ciclo 1 da Pesquisa-Ação, representados pela Coleta de Dados (estudo do método CBL, discussões sobre as melhores práticas e perfil dos estudantes selecionados), Feedback dos Dados e Análise dos Dados já haviam sido realizados com base na execução das primeiras atividades da etapa Preliminar.

Ao executar a atividade de Execução do Projeto da abordagem proposta, conforme apresentado na seção anterior, foi possível identificar que esta etapa, em especial, deveria sofrer alterações significativas, de modo a alinhar as expectativas do projeto com a realidade identificada no ambiente. Isto foi possível pelo Planejamento da Ação que se constituiu de uma proposta de alteração na abordagem proposta, mais especificamente na atividade de Execução.

A realização deste Ciclo 1 foi encerrada em dezembro/2014, com a entrega e publicação dos aplicativos desenvolvidos e a análise crítica do ciclo.

Este Ciclo será apresentado em detalhes no Capítulo 4 desta tese.

3.3.3 Ciclo 2 – Turma 2015

A saída da etapa de Avaliação do Ciclo 1 forneceu os insumos necessários para identificar que outras fases do Processo precisariam ser alteradas, uma vez que os desafios encontrados durante a Fase Preliminar (Contexto e Objetivo) não puderam ser completamente solucionados por meio apenas das alterações na atividade de Execução do Projeto da abordagem. As outras atividades da abordagem também deveriam passar por adaptações, especialmente a atividade de Seleção dos Estudantes.

O Ciclo 2 teve início com a preparação para a entrada da segunda turma de estudantes, ocorrida em fevereiro/2015. O Ciclo foi encerrado em dezembro/2015, com a publicação dos aplicativos desenvolvidos e a análise crítica do ciclo.

Este Ciclo será apresentado em detalhes no Capítulo 5 desta tese.

3.4 Considerações sobre o capítulo

O presente capítulo apresentou os conceitos da pesquisa utilizada, bem como as etapas que foram criadas para a estrutura da pesquisa. As etapas foram descritas de forma resumida para mostrar como cada uma foi executada. As etapas são explicadas em detalhe nos próximos dois capítulos: o capítulo quatro trata da execução da pesquisa-ação durante o primeiro ciclo e sua avaliação; e o capítulo cinco apresenta a execução da pesquisa-ação durante o segundo ciclo bem como sua avaliação.

CAPÍTULO 4 - CICLO 1 – TURMA 2013-2014

Don't be afraid of hard work. Nothing worthwhile comes easily.

Gertrude B. Elion

O Ciclo 1, conforme relatado na seção 3.3.2, foi motivado pelos desafios encontrados na Fase Preliminar, que apontavam para um distanciamento entre o perfil dos estudantes que foram selecionados e as exigências impostas por este tipo de projeto (inovação, método de ensino-aprendizagem, tecnologias etc.). O seu planejamento e sua realização estão representados na Figura 4-1 e serão detalhados nas próximas seções.

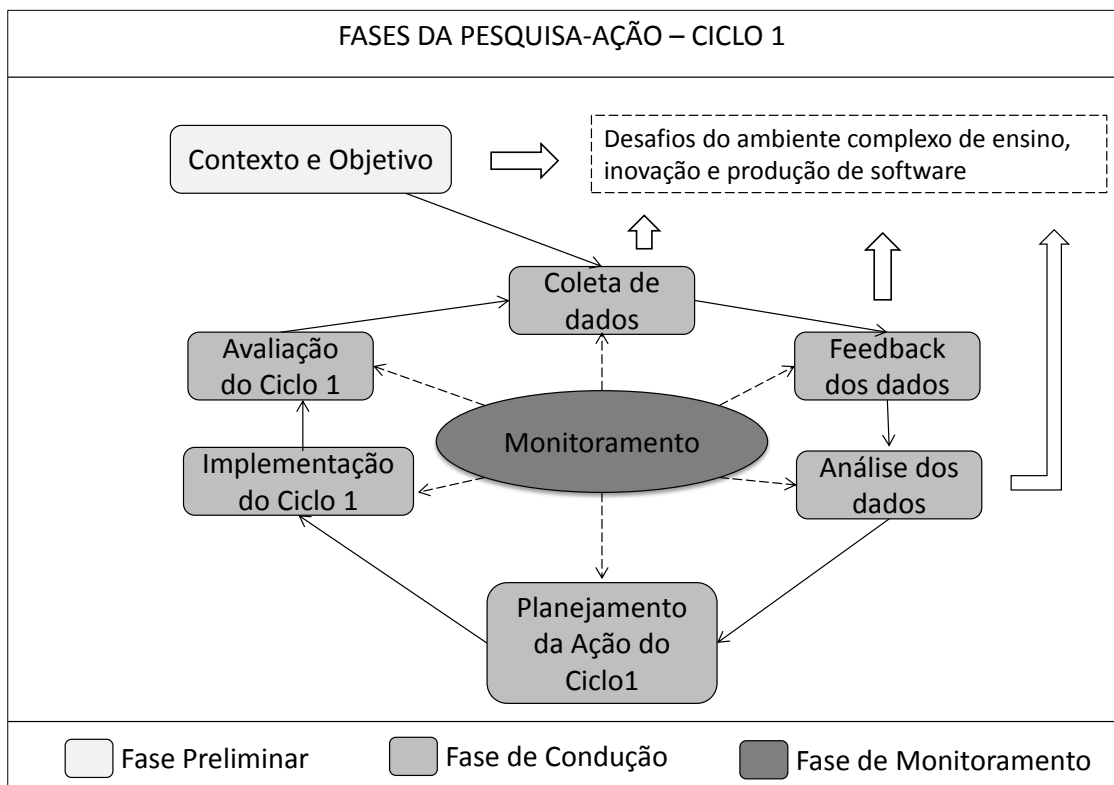


Figura 4-1: Fases da Pesquisa-Ação – Ciclo 1.

4.1 Planejamento da Ação do Ciclo 1

O Planejamento da Ação resultou em uma necessidade de alteração da atividade de Execução do Projeto, que havia sido originalmente proposta como sendo desdobrada em Curso com CBL, Desafio Principal e Publicação (Figura 3-2). A alteração foi motivada pela dificuldade dos estudantes em partir diretamente do curso para o Desafio Principal. Essa dificuldade foi identificada na execução dos exercícios propostos pelos professores. Observou-se a dificuldade dos estudantes em desenvolver programas que exigissem soluções não triviais. O resultado é a abordagem sequencial revisada apresentada na Figura 4-2.

Como se pode observar, a atividade de Execução do Projeto passou a ser dividida em Desafio Inicial, Mini Desafios, Desafio Principal e uma tarefa transversal de Apoio. A publicação do produto desenvolvido que estava identificada explicitamente no primeiro ciclo foi incorporado na atividade Desafio Principal, pois não é uma atividade separada. A tarefa de Apoio representa todas as atividades necessárias para a produção do software que não envolvam programação: interface e usabilidade, arte 2D, arte 3D, animação, sonorização, criação de trilha sonora e mentoria para a criação de *startups*.

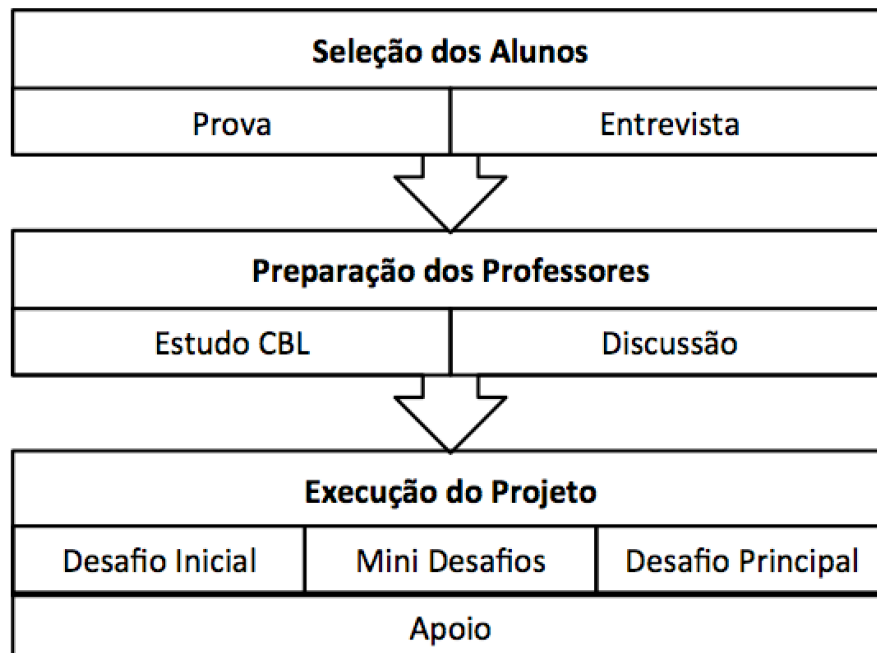


Figura 4-2: Versão ajustada da abordagem proposta

4.2 Implementação do Ciclo 1

Observou-se que várias melhorias eram necessárias para o ensino de programação. Algumas delas puderam ser feitas durante a própria atividade de Execução do Projeto, por exemplo, a integração de um modelo de processo de software ao CBL. Isto foi realizado, porém de forma ainda embrionária.

Nesta etapa, o CBL foi aplicado de três formas distintas: a primeira, com o objetivo de ensinar como ele deveria ser aplicado na elaboração da ideia de um aplicativo móvel para resolver um problema relevante (Desafio Inicial); em seguida, várias vezes no desenvolvimento efetivo de aplicativos para dispositivos móveis (Mini Desafios), com o objetivo de tornar os estudantes fluentes no método; e finalmente, no trabalho final, sob o nome de Desafio Principal onde os estudantes desenvolveram e publicaram um aplicativo relevante. Em cada uma dessas aplicações foram feitas avaliações e anotações com o objetivo de melhorar o método logo na aplicação seguinte, ou seja, pequenas melhorias incrementais foram sendo discutidas e aplicadas durante a própria execução.

Desafio Inicial

A primeira aplicação do método CBL consistiu em uma aula rápida sobre características de Apps (aplicativos para dispositivos móveis) relevantes e bem sucedidas e explicação do método CBL, seguida do Desafio Inicial. Este desafio foi incluído pois oferece ao estudante escolhas significativas sobre o que e como estudar, sendo esta uma das formas de se aumentar o engajamento e interesse em sala de aula (SCHRAW et al; 2001). Ele consistia em definir uma ideia de aplicativo e concebê-lo conceitualmente, gerando um vídeo, não sendo necessário desenvolver o aplicativo propriamente dito. Os estudantes seguiram os critérios utilizados para identificar Apps relevantes e bem sucedidas retirados do guia para projetar Apps grandiosas (GELEYNSE, 2012). Este guia sugere que Apps relevantes e bem sucedidas (denominadas Apps grandiosas) possuem duas ou mais das seguintes oito características:

- **Agradável:** convidativa (deixa uma excelente primeira impressão), intuitiva (fácil de entender e usar), cativante (faz os usuários voltarem a usar a App várias vezes), *enabling* (no sentido de permitir que uma pessoa possa realizar tarefas que não conseguiria sem a App), entusiástica (geradora de

entusiasmo entre os usuários), memorável (mostra um significado importante para o usuário imediatamente), atende a uma necessidade (resolve um problema do seu cliente), funciona como esperado (sem bugs críticos e bem representada na sua descrição e *screenshots* da loja de aplicativos);

- **Inovadora:** única (não há outra igual), criadora de categoria (uma nova categoria de Apps é criada a partir dela), inspiradora (inspira a criação de outras apps semelhantes).
- **Estado da Arte:** aproveita-se das características técnicas dos dispositivos e sistemas operacionais mais recentes, integra-se com novas tecnologias e usa as mais recentes técnicas de desenvolvimento. Além disso, é rápida e responsiva.
- **Conectada:** usa hardware e software que permitem conexão, integração e comunicação com outros dispositivos.
- **Acessível:** utiliza-se de características do hardware e software para alcançar todos os grupos de pessoas, incluindo aquelas com dificuldades de visão, audição e aprendizado.
- **Localizada:** oferece suporte a múltiplos idiomas.
- **Projetada:** usabilidade eficiente, belo design visual, reflete o que é familiar aos usuários e modelada a partir do mundo real.
- **Disruptiva:** tem grande impacto, deixa uma forte impressão, muda a maneira de pensar e inicia tendências.

Os estudantes se reuniram em grupos de três ou quatro participantes e tiveram três horas para terminar o desafio. O tempo para o desafio foi definido com base na baixa complexidade da atividade. Ao término da atividade cada equipe apresentou seu trabalho para todos os estudantes. Em seguida os estudantes foram convidados a gravar um áudio de reflexão individual de dois minutos que consistia na complementação das seguintes frases:

- Eu aprendi...
- Eu gostaria de ter aprendido...
- Eu ainda preciso aprender...

A análise destes primeiros áudios e a observação de todas as aulas por parte do pesquisador mostrou que o método CBL deveria ser aplicado de forma mais

gradativa, principalmente pela falta de experiência dos estudantes no uso de métodos ativos, bem como do espaço físico que estava sendo utilizado. A falta de experiência ficou bem clara, pois a maioria dos estudantes colocou na última frase (Eu ainda preciso aprender...) aspectos que poderiam ter sido encontrados em uma pesquisa rápida na Internet. O espaço utilizado foi uma sala de aula comum do tipo anfiteatro. Esse tipo de espaço dificultou a execução do trabalho em grupo.

Como os estudantes selecionados não possuíam experiência neste tipo de método ativo e como o espaço definitivo necessário para a aplicação correta do método não estava pronto, optou-se por aplicar o método CBL completo apenas em determinadas situações específicas.

Os Mini Desafios consistiam em tarefas de construção de um aplicativo que resolvesse um problema real, porém de pequeno porte, uma vez que o tempo destinado a esta atividade era de apenas duas semanas. Duas aplicações chegaram ao estágio de Publicação, conforme apresentado no Apêndice K¹. No mini desafio, a equipe deve fazer uma pesquisa inicial sobre o tema proposto utilizando o método CBL. Em seguida deve desenvolver e publicar uma App relevante para o problema escolhida. Durante este processo, os estudantes são responsáveis por buscar auxílio com os professores referentes às questões de desenvolvimento de software e solicitar aos grupos de apoio os recursos necessários para a produção.

O Desafio Principal foi desenvolvido durante um período de 6 meses e foi avaliado em três momentos diferentes: protótipo, versão beta e versão final. Todos os estudantes passaram por uma banca de avaliação. A banca da última avaliação contou com a presença do pesquisador e mais três especialistas em aplicativos para dispositivos móveis de sucesso, todos colaboradores da empresa parceira.

Cada aplicação do método CBL (Desafio Inicial, Mini Desafios e Desafio Principal) consistiu em 4 etapas: Definir o Desafio, Adquirir Conhecimento, Desenvolver a Solução e Avaliar o Processo. Esta etapa de Avaliar o Processo foi feita por meio de uma avaliação com os estudantes utilizando análise de vídeos e áudios de reflexão, além de conversas com as turmas.

Mini Desafios

¹ Um destes aplicativos foi premiado com o Apple Design Award 2015 na categoria de melhor aplicação de estudante.

Com o intuito de adequar o comportamento de aprendizagem dos estudantes de forma gradativa, decidiu-se pela continuidade da utilização de aulas expositivas e práticas em determinadas situações. Para as aulas normais decidiu-se usar uma mistura de aulas expositivas e práticas; nos desafios foi decidido usar o método CBL original. As aulas expositivas foram planejadas com o objetivo de fornecer uma visão geral do assunto em questão e como preparação para uso do método CBL. A aplicação de aulas cujo conteúdo era apenas a visão geral de um tópico ao invés de sua explicação completa teve como objetivo instigar nos estudantes a necessidade de estudo complementar para melhorar o entendimento do assunto, sempre tendo em vista a futura aplicação do método CBL.

Após uma semana de aulas executadas neste formato (duas aulas de 3 horas por semana, sendo uma hora de aula expositiva e duas horas de prática), foi pedida novamente a gravação de um áudio de reflexão individual para os estudantes nos mesmos moldes do áudio anterior. A análise destes áudios demonstrou que a resposta foi positiva para 60% dos estudantes com relação ao novo formato de aulas. Seguiu-se dessa forma por mais duas semanas, sempre se pedindo a gravação individual de um áudio ou vídeo de reflexão ao final de cada semana. Após esse tempo, percebeu-se um amadurecimento dos estudantes com relação ao estudo, refletido na qualidade geral dos projetos de programação semanais que sempre envolviam o entendimento de tópicos que não haviam sido explicados na aula expositiva, ou haviam sido explicados de forma superficial.

Essa mudança de postura levou a mais uma alteração no formato de aulas em dois aspectos: quantidade e tempo das aulas expositivas. Passou-se de duas aulas expositivas para uma e decidiu-se pela diminuição gradativa do seu tempo, iniciando em 50 minutos e diminuindo 10 minutos a cada semana (duas aulas por semana) até um mínimo de 10 minutos. A gravação semanal de áudios ou vídeos individuais de reflexão continuou com uma pequena mudança: os estudantes agora poderiam falar sobre o que quisessem, sem ter um conjunto de frases para guiá-los.

A avaliação das reflexões dessas cinco semanas mostrou que a mudança foi positiva para a maioria dos estudantes (67%). As respostas dos demais indicavam a dificuldade em estudar sozinhos os tópicos sugeridos nas aulas expositivas, principalmente por falta de uma base sólida no desenvolvimento de software. Entretanto, a observação constante de tais estudantes mostrou um pouco de falta de comprometimento com o estudo e atividades sugeridas.

Devido à resposta positiva, decidiu-se pela aplicação do método CBL em um Mini Desafio para avaliar os efeitos do curso no desenvolvimento de um software para publicação: os estudantes deveriam desenvolver uma App relevante e pronta para publicação no período de duas semanas. Esse desafio serviu para avaliar a postura dos estudantes com relação ao método, assim como para avaliar de que forma o método poderia ser usado como parte de um processo de desenvolvimento de software. O curto espaço de tempo e a necessidade de desenvolver algo completo e relevante serviram como incentivo para o planejamento e o uso do método CBL.

Os estudantes tiveram a liberdade de escolher suas equipes que deveriam ser compostas de 2 a 4 estudantes, o que resultou em 39 equipes. Ao final das duas semanas foram finalizados 12 projetos dos 39 propostos. Desses 12 projetos apenas 2 deles foram publicados na Loja de Aplicativos para Dispositivos Móveis e efetivamente utilizados pelo público alvo. As reflexões individuais acusaram três problemas principais: o tempo curto, a inexperiência no desenvolvimento de um produto completo (para o caso dos estudantes que não possuíam experiência anterior em desenvolvimento de software) e a inadequação do método CBL para o desenvolvimento de software (para os estudantes que já possuíam experiência anterior em desenvolvimento de software).

Esse resultado serviu como base para as seguintes ações:

- Criação do conceito de Workshops: conjuntos de aulas de cerca de até 15 horas para tópicos específicos. Os dois primeiros foram um reforço sobre os assuntos estudados para os estudantes com dificuldades e processos ágeis de desenvolvimento de software.
- Integração do método CBL dentro de um processo de desenvolvimento de software. O processo foi desenvolvido com os objetivos de simplicidade e utilidade. A simplicidade devido ao fato de que muitos estudantes tinham pouca ou nenhuma experiência no desenvolvimento de software profissional. A utilidade para que os estudantes enxergassem o processo como algo que lhes ajudasse e não mais um obstáculo ou um conjunto de documentos a preencher para satisfazer os requisitos de determinada atividade. Procurou-se selecionar aspectos vantajosos de métodos diferentes em um processo pequeno e simples: método cascata, ideal para projetos pequenos e com melhorias

incrementais; método ágil, adaptabilidade a contingências, favorecimento a inovações.

Desafio Principal

Devido à necessidade de se desenvolver e publicar uma App ao final do curso, decidiu-se que os estudantes deveriam participar ativamente de um método de desenvolvimento de software. Como muitos estudantes eram novatos na área de programação e nunca haviam desenvolvido software profissionalmente, foi escolhido um método híbrido que une características e técnicas tradicionais (cascata ou *waterfall*) e ágeis (*agile*), algumas vezes chamado de “AgileFall”. Os objetivos dessa abordagem foram:

- Levar os estudantes a estabelecer um escopo de projeto factível para o tempo e recursos disponíveis no curso;
- Dar visibilidade aos estudantes do andamento e possíveis atrasos dos seus projetos;
- Detectar atrasos e problemas de desempenho de estudantes o mais rápido possível, possibilitando a execução de correções adequadas.;
- Fornecer uma visão macro-gerencial do andamento dos projetos do curso como um todo, facilitando tomadas de decisão e correções de rota do curso no ano atual e subsequente .

Esse método foi utilizado no desafio principal em todos os 60 projetos propostos. Estes projetos foram divididos igualmente entre os cinco professores orientadores que se reuniam semanalmente para relatar a situação de cada equipe.

Ao término do curso, foram finalizados e apresentados 54 projetos, sendo 41 aprovados para publicação, cuja lista pode ser encontrada no Apêndice F. Os estudantes entregaram a documentação, fontes e Apps instaladas em dispositivos próprios, além de um vídeo de reflexão final com tema livre.

4.3 Avaliação do Ciclo 1

Devido ao caráter prático do curso e também às evidências que o uso de metodologias ativas afetam de forma mais direta o comportamento e atitudes dos estudantes do que seu rendimento em testes padronizados [PRINCE, 2004], optou-se por não fazer uma avaliação de desempenho acadêmico dos estudantes. Assim,

a avaliação do Ciclo 1 foi realizada por meio de questionários aplicados aos estudantes (Apêndice A) e entrevistas semiestruturadas realizadas com os professores da própria PUCPR e também de outras Universidades que estavam aplicando o CBL em ambientes similares (Apêndice B).

4.3.1 Avaliação dos Estudantes

Com o objetivo de avaliar a aplicação do método CBL em diversos aspectos, foi aplicado um questionário (Apêndice A) para os estudantes que finalizaram o primeiro ciclo com sucesso. Dos 110 estudantes originalmente selecionados, 84 finalizaram o curso, sendo que 78 destes responderam o questionário. A turma foi composta de 57,7% de estudantes novatos, ou seja, estudantes com menos de 1 ano de experiência em programação e 15,4% de veteranos, aqueles com mais de 3 anos de experiência em programação.

Tabela 4-1: Egressos da primeira turma

Curso	Total	%
Graduação Jogos Digitais	15	17,44%
Graduação Sistemas de Informação	17	19,77%
Graduação Engenharia de Controle e Automação	9	10,47%
Graduação Ciência da Computação	10	11,63%
Graduação Engenharia da Computação	10	11,63%
Graduação Engenharia Elétrica	4	4,65%
Graduação Engenharia Civil	1	1,16%
Graduação Engenharia de Produção	1	1,16%
Graduação Medicina	1	1,16%
Especialização Desenvolvimento Web	1	1,16%
Especialização Desenvolvimento de Jogos Digitais	10	11,63%
Especialização Engenharia de Software	6	6,98%
Doutorado Informática	1	1,16%

Os estudantes foram efetivamente obrigados a utilizar o método CBL apenas três vezes durante o curso. Entretanto, por iniciativa própria, 71,8% dos estudantes o utilizaram mais do que três vezes como mostra a Figura 4-3, indicando que houve um interesse por parte da maioria dos estudantes em aplicar o método, mesmo sem a sua exigência. Este resultado está de acordo com a pergunta que tratava do fator motivacional do método, onde 70,5% dos estudantes sentiram-se motivados a resolver o problema utilizando o método CBL.

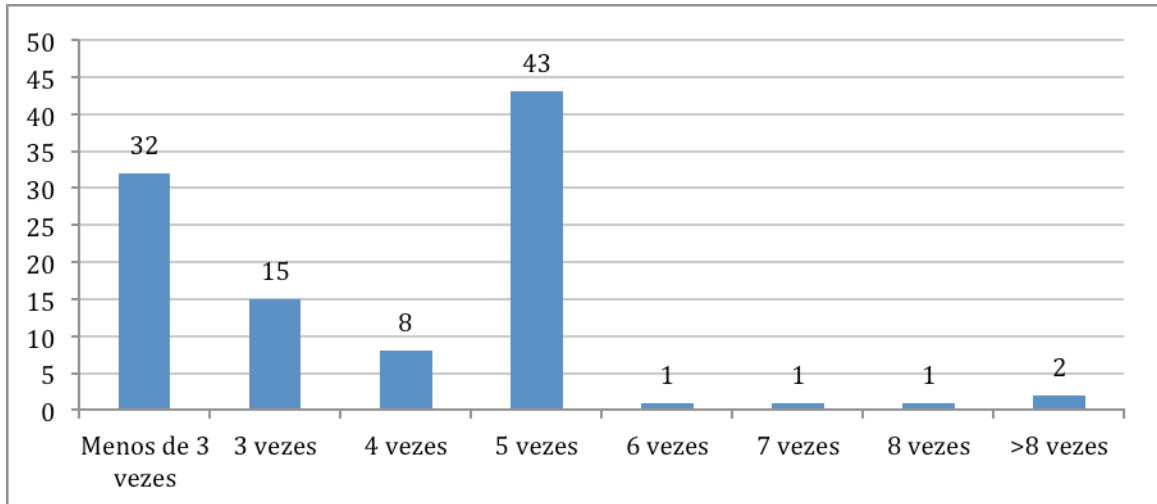


Figura 4-3: Número de vezes que os estudantes usaram o CBL no Ciclo 1.

Em suas fases iniciais, o CBL tem como um dos objetivos auxiliar no entendimento do problema ou desafio. A maioria dos estudantes (74,4%) concorda que o método os ajudou nesse sentido conforme mostra a Figura 4-4. Além disso, 39,7% dos estudantes escolheram as opções Concordo Parcial ou Totalmente para representar o grau com que o método os ajudou a resolver o problema.

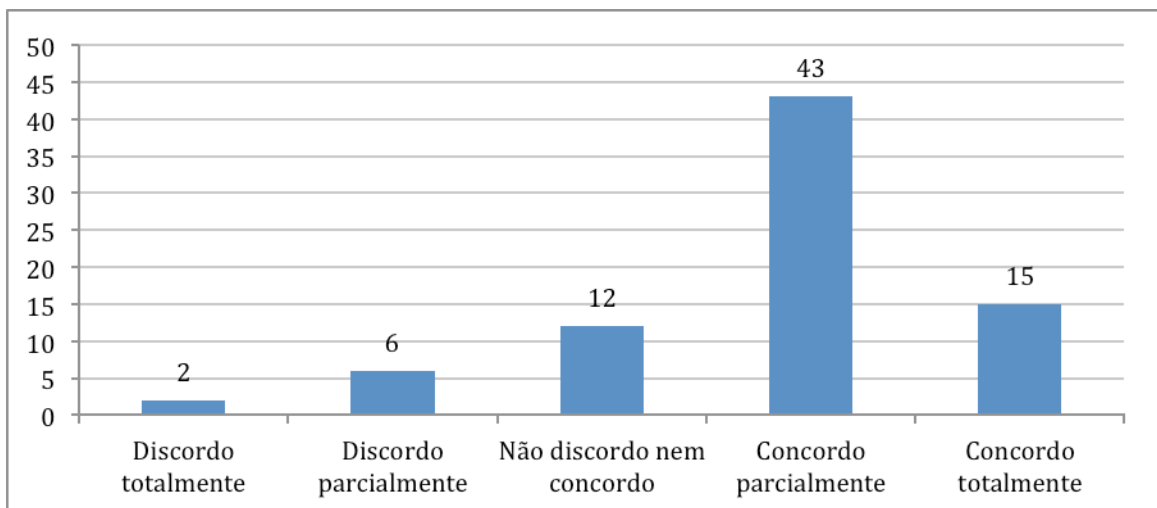


Figura 4-4: Efetividade do CBL no Entendimento dos Desafios Propostos no Ciclo 1.

A efetividade do método na elaboração de uma solução foi considerada como positiva para 70,5% dos estudantes e 82,1% consideraram que as questões guias, atividades e recursos ajudaram no desenvolvimento da App, conforme demonstra a Figura 4-5. Esse resultado era esperado, visto que a maioria dos estudantes não

tinha experiência anterior em desenvolvimento de software e naturalmente teria dificuldade em desenvolver uma App sem nenhum tipo de auxílio sistematizado. Esse auxílio deu-se pela elaboração das questões relacionadas ao problema e suas respectivas respostas. Essas respostas foram úteis, não apenas para reforçar o entendimento do problema, mas também acabaram servindo de guia para o desenvolvimento, principalmente para os estudantes com pouca ou nenhuma experiência anterior na produção de software. De forma geral, 82,1% consideraram o método efetivo para a estruturação da solução.

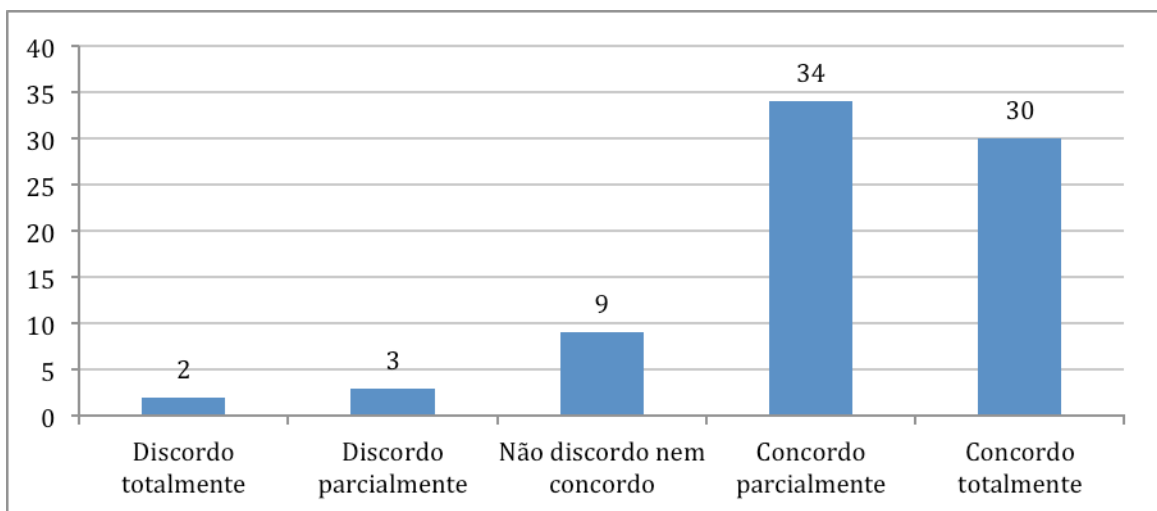


Figura 4-5: Efetividade das Questões Guias na Construção das Soluções do Ciclo 1.

Ainda com relação ao desenvolvimento de uma solução, 48,7% dos estudantes consideraram que o método ajudou a encontrar uma solução inovadora. Esse resultado está de acordo com a opinião da banca de avaliadores que classificou metade dos projetos apresentados como sendo inovadores e com grande potencial de mercado ou relevância para nichos específicos.

A recepção do método pelos estudantes pode ser considerada como positiva, pois 76,9% dos estudantes responderam que podem usá-lo em outras situações e 87,2% acham que ele pode contribuir para o ensino de outros conteúdos que não sejam de programação e desenvolvimento de software.

Outros resultados positivos dizem respeito à melhoria nas habilidades de programação e ao fator motivacional do uso do método. Uma boa parcela dos estudantes com experiência em programação (mais do que três anos), 58,3% consideraram que o método contribuiu para a melhoria das habilidades de

programação enquanto apenas 8,3% responderam negativamente (Figura 4-6). Na mesma figura e levando-se em conta apenas estudantes novatos (menos de um ano de experiência em programação), 35,6% concordaram Parcial ou Totalmente com a pergunta e 22,2% discordaram Parcial ou Totalmente.

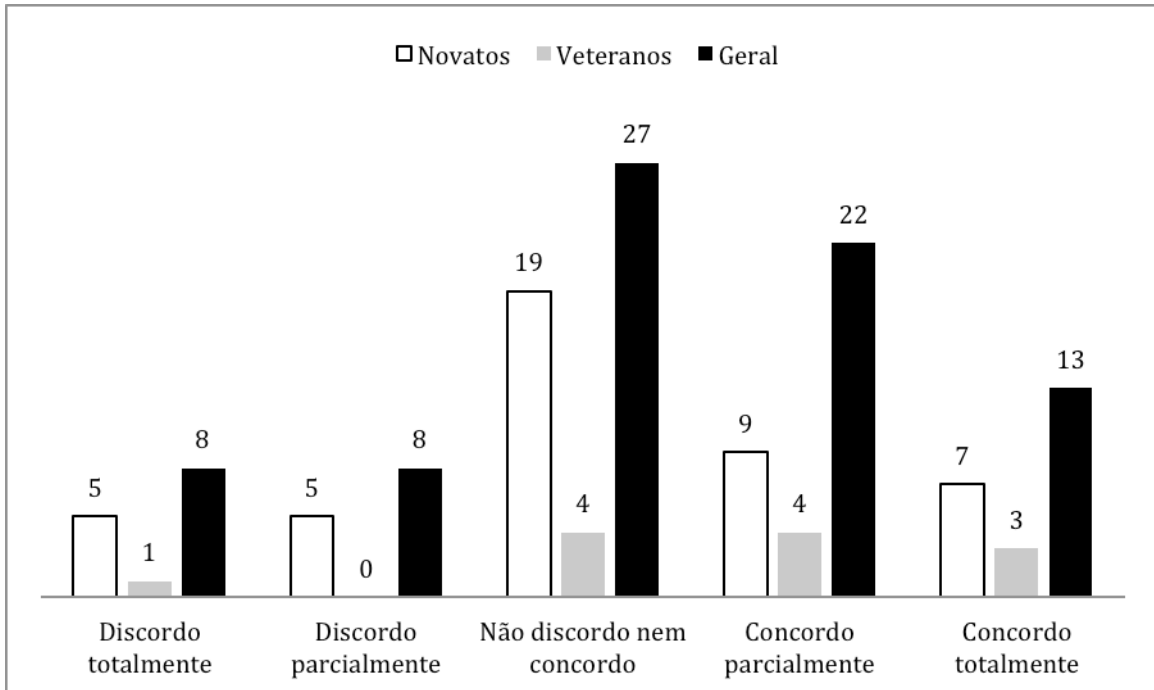


Figura 4-6: Efetividade do CBL na Melhoria das Habilidades de Programação no Ciclo 1.

No espaço para comentários livres, 26,9% dos estudantes escreveram algum tipo de consideração sobre o método CBL. O que mais se destacou nesses comentários foi a necessidade de se integrar o CBL a um método de desenvolvimento de software e o fato de que o CBL auxilia bastante no entendimento do problema a ser resolvido e na sua solução conforme comentários abaixo:

Como uma pessoa com dificuldade em programação, o CBL me ajudou a estruturar meus trabalhos e meu app, solucionou alguns problemas sozinho e colaborou para a solução de outros. É um sistema prático que facilita a elaboração de projetos e com certeza usarei futuramente. (A37)

O método CBL ajudou muito em organizar a linha de pensamento no momento da concepção e planejamento dos projetos. (A39)

Outros comentários relevantes que resultaram em algum tipo de ação para a ciclo seguinte foram: a insegurança com relação à viabilidade do projeto, principalmente na identificação do público alvo, dificuldade em ter uma ideia original para o desafio final, excesso de documentação/pesquisa exigidos e a ausência de

auxílio para estudantes que pretendem criar empresas apoiadas na App desenvolvida.

4.3.2 Avaliação dos Professores

Foram realizadas avaliações com 23 professores de oito universidades brasileiras onde estavam acontecendo cursos de iOS similares e com a utilização do método CBL. Para esta avaliação foi utilizado o questionário apresentado no Apêndice B, composto de questões objetivas e subjetivas. Todas as entrevistas foram gravadas, transcritas, revisadas e analisadas quantitativa e qualitativamente.

O perfil dos professores foi levantado com base nos aspectos: tempo de experiência no desenvolvimento de software, no ensino de programação e no ensino em geral, uso de métodos tradicionais de ensino e conhecimento de métodos de desenvolvimento de software. Para a realização das entrevistas foram escolhidos professores que tivessem trabalhado com o método CBL por pelo menos seis meses e o tivessem aplicado no mínimo duas vezes em cursos de desenvolvimento de aplicativos para dispositivos móveis.

Os professores envolvidos possuíam uma ampla experiência no desenvolvimento de software (média de 14,0 anos, desvio padrão de 7,7, mediana 15) e uma razoável experiência no ensino, tanto de programação, com média de 7,1 anos (desvio padrão 6,0, mediana 5), quanto no ensino em geral, com média de 9,0 anos (desvio padrão 6,3, mediana 8).

As técnicas tradicionais de ensino (anteriores ao uso do CBL) mais utilizadas em sala de aula por estes professores foram aulas expositivas e aulas práticas com 100% de respostas positivas; e tutorial guiado, com 50% de respostas positivas dos entrevistados. Tutorial guiado foi definido como sendo uma atividade na qual um problema de programação é proposto e o professor resolve-o junto com os estudantes, programando diretamente na ferramenta adequada e mostrando todo o processo de desenvolvimento e a linha de raciocínio.

O conhecimento profissional de métodos de desenvolvimento de software ficou assim dividido: 56,5% dos professores com alguma experiência em métodos tradicionais; 47,8% com experiência em métodos iterativos e incrementais; e 73,9% com experiência em métodos ágeis. Esta questão permitia múltiplas respostas.

Corroborando a opinião geral dos estudantes, 95,6% dos professores concordaram total ou parcialmente com a afirmação: “O método CBL motivou mais

os estudantes do que outros métodos que você conhece”. Aqueles que concordaram parcialmente justificaram a escolha afirmando que houve realmente uma motivação maior, mas não sabiam se essa motivação era decorrente do método CBL ou do aspecto de novidade, de utilizar uma metodologia de ensino diferente das tradicionais.

As opções possivelmente ou certamente, foram escolhidas por 95,6% dos professores quando perguntados se utilizarão o método CBL para outras classes de programação enquanto 91,3% pretendem utilizar o método para ensinar outros conteúdos diferentes de programação. Entre os conteúdos citados estão tópicos comuns em cursos de computação, mas também design digital e administração.

O uso do método CBL para o ensino de programação foi tratado na questão: A qualidade do software desenvolvido pelos estudantes utilizando o método CBL é melhor do que a qualidade do software desenvolvido utilizando outros métodos que você tem experiência?” Nesta questão foi explicado aos professores que o termo qualidade refere-se ao conjunto das seguintes características: software pronto para publicação, relevante, sem erros críticos e sem problemas de usabilidade. Diferente das outras questões, nesta o resultado não foi significativamente positivo. Concordaram total ou parcialmente, 65,2% dos professores e os demais escolheram a opção “não concordo nem discordo”. A justificativa destes foi o fato de que não haviam trabalhado o suficiente com outros métodos para embasar uma opinião positiva.

A parte subjetiva da entrevista foi composta de três tipos: comentários espontâneos sobre qualquer resposta objetiva que o entrevistado julgasse ser necessário justificar, quais alterações foram efetuadas em cada etapa do método CBL e opinião geral sobre o método CBL.

A análise de redes temáticas [ATTRIDE-STIRLING, 2001] da parte subjetiva das entrevistas gerou o tema principal denominado “Proposta de mudança”, composto por cinco subtemas: i) aspectos positivos; ii) aspectos negativos; iii) aspectos inconclusivos; iv) sugestões de outros métodos e; v) sugestões de melhoria.

As relações entre esses temas são apresentados nas redes temáticas da Figura 4-7: Rede Temática de Aspectos Positivos, Negativos e Inconclusivos do Ciclo 1.e Figura 4-8: Rede Temática de Sugestões de Melhoria e de Outros Métodos

no Ciclo 1. O detalhamento dos dados tabulados que geraram essas redes temáticas pode ser consultado no Apêndice M.

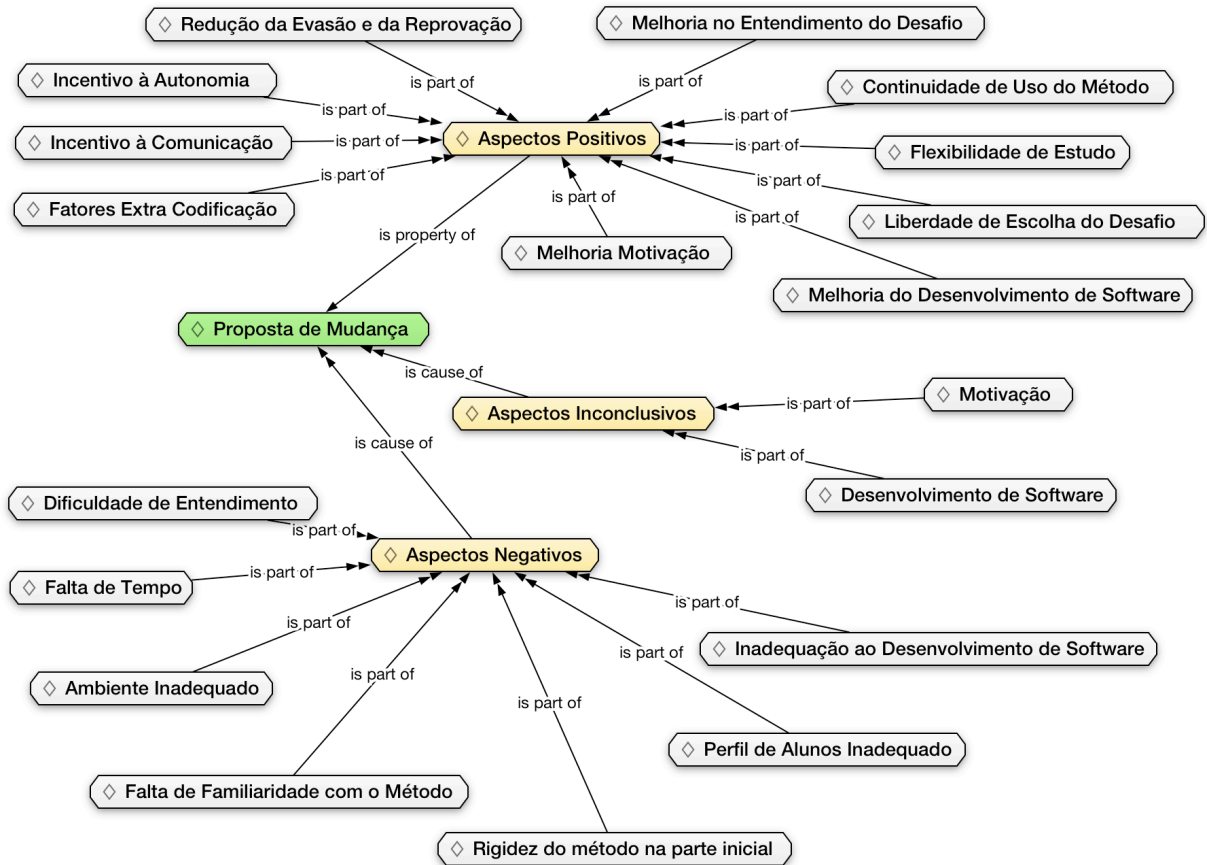


Figura 4-7: Rede Temática de Aspectos Positivos, Negativos e Inconclusivos do Ciclo 1.

Conforme pode ser observado na Figura 4-7, segundo os entrevistados, a aplicação do método CBL teve diversos aspectos positivos, sendo os mais citados: intenção de continuar a usar o método em outras situações de ensino; melhoria da motivação dos estudantes para estudar e executar as atividades propostas; incentivo à autonomia do estudante; e melhoria do entendimento dos estudantes sobre os problemas a serem resolvidos.

Uma constatação interessante dos professores foi que, a flexibilidade de estudo do método (não ter ementa pré-definida) promoveu o desenvolvimento de algumas habilidades nos estudantes, como por exemplo, a autonomia. Para execução do projeto da disciplina os estudantes têm a liberdade de escolha do desafio, ponto de partida do método, e utilizam de diferentes formas de aprendizagem (cursos, livros, monitoria, buscas na internet) para adquirir

conhecimento necessário para resolver o problema. Surge também a necessidade de se preocupar com outros fatores que vão além da codificação, como usabilidade, viabilidade econômica, etc. Outro ponto mencionado como positivo foi o incentivo à comunicação, pois o método é eminentemente colaborativo. Os estudantes ao longo do desenvolvimento do projeto buscam os professores e os estudantes mais experientes para ajudarem na resolução de seus problemas. Segundo um dos entrevistados, estes aspectos contribuíram para a redução da evasão e da reprovação em uma turma de programação.

Apesar do uso do método ter sido considerado pela totalidade dos entrevistados como bem sucedido, alguns pontos negativos foram identificados. O problema mais citado foi a inadequação do método para determinados perfis de estudantes. Foi percebido pelos professores que o perfil inadequado dos estudantes, tem relação com a falta de comprometimento, falta de base teórica e técnica, bem como a resistência à autonomia. Esta mesma percepção foi identificada pelo pesquisador nas reflexões dos estudantes e na observação do comportamento deles durante a execução das atividades propostas.

Outro problema diz respeito ao ambiente de ensino. O método CBL requer um ambiente com determinadas características essenciais para uma aplicação adequada. A falta deste ambiente dificulta, ou até mesmo impossibilita, o seu uso.

Também foi apontada pelos entrevistados a dificuldade de entendimento do método por parte dos estudantes e a falta de familiaridade dos professores com a sua aplicação. Foi percebido que é necessário um tempo maior tanto para alunos quanto professores. Consequentemente, a falta de tempo para a sua aplicação prejudica tanto o entendimento quanto a execução.

Percebeu-se que uma aparente rigidez inicial foi motivo de resistência por parte de alguns estudantes que preferiam partir logo para a solução, ignorando a necessidade de entendimento do problema.

Em relação ao tema denominado “Aspectos inconclusivos”: o fator motivacional, ou seja, o quanto o método motiva o aprendizado, se mostrou carente de maiores estudos, pois a aplicação do método foi acompanhada de uma série de outras medidas que podem ser consideradas altamente motivadoras (bolsa de estudos com valor superior ao praticado na universidade, professores disponíveis presencialmente em tempo integral, fornecimento de equipamentos lançados mais

recentemente, entre outros). Devido a estes fatores, muitos entrevistados consideraram impossível afirmar que o método por si só seja claramente motivador.

O aspecto adequação do método CBL para o desenvolvimento de software mostrou-se inconclusivo nas entrevistas, pois apareceram tanto opiniões negativas, quanto positivas e neutras.

O método não foi criado especificamente para o ensino de programação, sendo considerado como um framework educacional de uso geral, portanto, foi um tanto inesperada a resposta dos 13% de entrevistados que informaram que o método é por si só adequado ao ensino de desenvolvimento de software.

A partir da análise do perfil dos entrevistados foi possível identificar que 33% dos professores que opinaram dessa forma, consideraram o CBL adequado ao desenvolvimento apenas na etapa de levantamento de requisitos, o que é perfeitamente natural visto que uma de suas etapas tem justamente o objetivo de melhorar o entendimento do problema. Os 66% restantes não eram da área de computação ou similar, tendo portanto, uma visão diferente sobre desenvolvimento de software como pode ser constatado a partir do comentário de um deles sobre este assunto:

... na outra experiência que eu tive a gente usou outros conceitos que eram levemente parecidos em termos das pesquisas, [...] foi utilizado na época na prévia a gente usou o design thinking, a gente usou alguns conceitos de user experience de pesquisas que são relativamente parecidos e os resultados foram muito interessantes também, mas eu acho que nesse primeiro caso foi uma junção de métodos, de técnicas e o CBL é por si só é mais completo, [...]. O CBL em si engloba tudo o que tinha sido planejado na experiência que eu tive, anterior, onde tive que juntar diferentes métodos. (P22)

A Figura 4-8 apresenta outra dimensão capturada nas entrevistas e diz respeito a sugestões de melhorias e de outros métodos similares. Apesar do número de citações dessas melhorias ser relativamente pequeno (13,0%) e o seu impacto no método CBL ser restrito principalmente a mudanças pontuais, isso demonstra que foram identificadas oportunidades e necessidades de alterações durante a aplicação do CBL.

As principais sugestões de melhorias podem ser agrupadas em duas categorias: pontuais e globais. As sugestões pontuais são aquelas em que a alteração ocorreu apenas em uma fase do método CBL e globais, para alterações que abrangem mais de uma fase.

As alterações pontuais mais citadas foram o uso de questões guias para definir a questão essencial, oferecendo assim um rumo pré-definido para o estudante; e iniciar a aplicação do método pelo desafio, normalmente quando o estudante já tem a ideia do que vai fazer, eliminando-se assim a necessidade de se definir a grande ideia e questão essencial.

As alterações globais foram sugeridas no sentido de enriquecer o método CBL na etapa de elaboração da solução. Por exemplo, a inclusão do método ágil de desenvolvimento Scrum com o objetivo de sistematizar a programação do aplicativo após as fases iniciais do CBL terem ajudado a entender o problema e a estruturar uma solução.

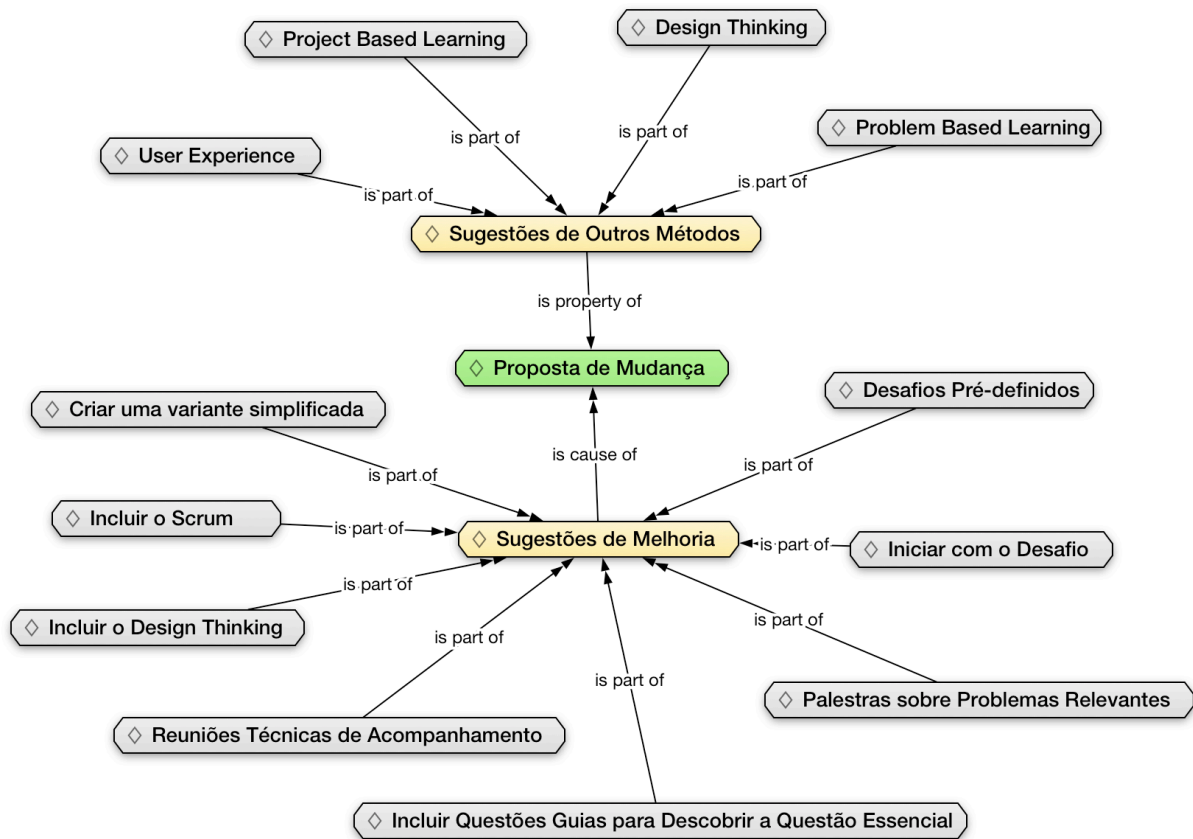


Figura 4-8: Rede Temática de Sugestões de Melhoria e de Outros Métodos no Ciclo 1.

No tema denominado de sugestões de outros métodos com alguma similaridade, os professores citaram: Project Based Learning e Problem Based Learning. O motivo de tais citações foi a de que o método CBL aparenta ser uma

evolução, apresentando algumas novidades expressivas, como liberdade de escolha do estudante, flexibilização do estudo e integração ao ambiente de ensino.

A análise qualitativa das entrevistas dos professores contribuiu para comparar a similaridade de problemas encontrados pela observação em sala e análise das reflexões e questionários dos estudantes, corroborando a sua importância. Outro ponto relevante foi a identificação da necessidade de se integrar um método de desenvolvimento de software com o CBL, com o objetivo de viabilizar a construção de Apps relevantes para publicação.

4.4 Considerações sobre o Capítulo

Neste capítulo foram detalhados os passos da execução do primeiro ciclo da pesquisa-ação. Foram descritas as etapas da pesquisa bem como a avaliação feita com estudantes e professores.

Os principais problemas encontrados durante a execução deste primeiro ciclo, por meio da observação, reflexões gravadas pelos estudantes, questionário aplicado aos estudantes e entrevista com os professores foram:

- Dificuldades no acompanhamento do método CBL;
- Dificuldades no desenvolvimento de aplicações devido à falta de experiência;
- Dificuldades na integração do método CBL em um processo de desenvolvimento de software;
- Necessidade de escolher os estudantes com perfil diferenciado para que o método possa ser aplicado de forma eficiente;
- Dificuldade em identificar problemas relevantes que possam ser resolvidos por Apps;
- Ausência de auxílio na área de inovação e empreendedorismo: monetização, pesquisa de mercado, viabilidade econômica, identificação de público alvo, criação de *startups*, etc.

As ações tomadas durante o primeiro ciclo para a resolução dos problemas acima foram:

- Inclusão de apresentações de problemas atuais em diversas áreas distintas: medicina, arquitetura, negócios, música, etc.

- Inclusão de projetos externos, tanto de professores pesquisadores quanto de empreendedores para apreciação dos estudantes e escolha do desafio final.
- Inclusão de aulas expositivas regulares durante o curso;
- Inclusão de aulas expositivas de reforço;
- Integração de um processo de desenvolvimento de software com o método CBL;
- Mudança no processo seletivo com o objetivo de escolher estudantes com perfil adequado às exigências do método CBL.
- Integração do processo com atividades relacionadas a empreendedorismo e inovação com o objetivo de viabilizar a criação de empresas a partir das Apps desenvolvidas. Essa integração foi realizada em conjunto com a aceleradora de *startups* da universidades.

O resultado das alterações acima foi compilado em uma abordagem para ambientes complexos de ensino, inovação e desenvolvimento de software, proposta nesta tese, denominada IDEAS que foi utilizada na execução do segundo ciclo da pesquisa-ação descrita no capítulo 5.

CAPÍTULO 5 - CICLO 2 – TURMA 2015

Be less curious about people and more curious about ideas.

Marie Curie

Os resultados das avaliações do Ciclo 1, correspondentes à primeira turma de estudantes (2013-2014), foram analisados e diversos problemas e oportunidades de melhoria foram identificados. Este capítulo descreve o Planejamento da Ação decorrente destas constatações, bem como sua Implementação e Avaliação, conforme apresentado na Figura 5-1.

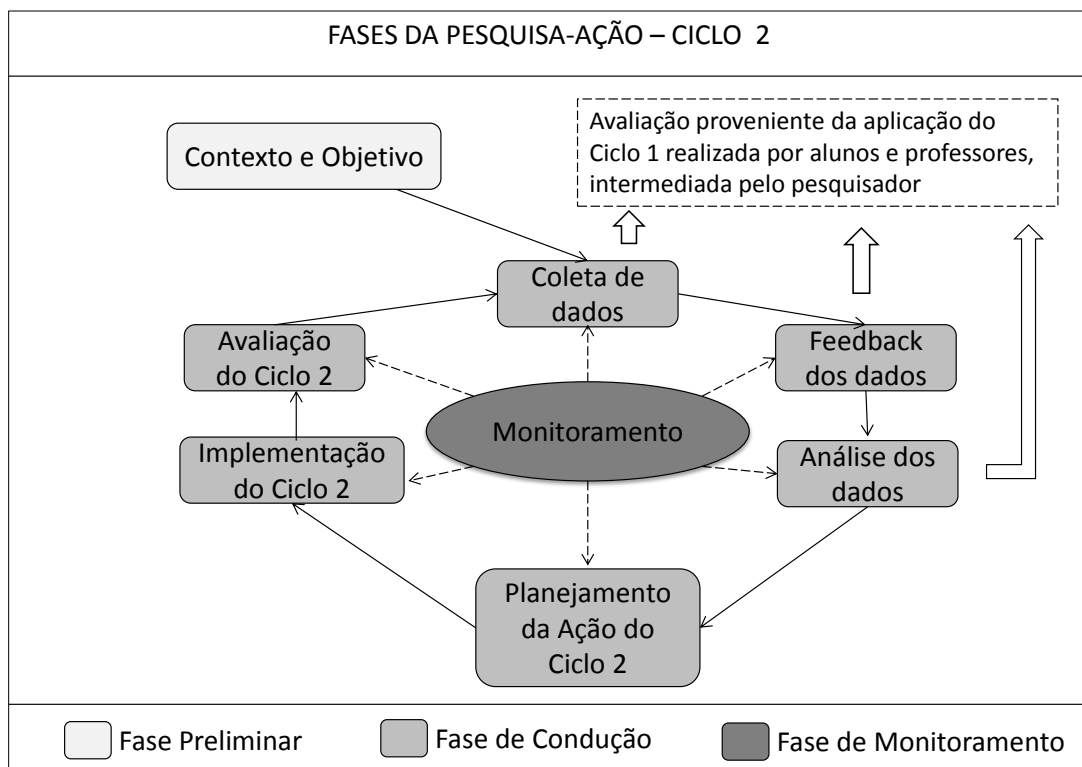


Figura 5-1: Ciclo 2 da Pesquisa-Ação.

5.1 Planejamento da Ação do Ciclo 2

Os seguintes aspectos foram objeto da mudança realizada para o Ciclo 2: melhoria da seleção dos estudantes de acordo com o perfil desejado; aplicação gradativa do método CBL; imersão dos estudantes em um processo de desenvolvimento de software; e, orientação para a criação de *startups*. Essas ações de mudança resultaram em uma abordagem denominada IDEAS – Inovação, Desenvolvimento e Ensino em Ambientes Complexos de Software.

A criação da abordagem IDEAS levou em conta os resultados da avaliação apresentada no Capítulo 4, referentes ao Ciclo 1, bem como as diversas dimensões envolvidas no curso de formação de desenvolvedores iOS:

- Utilização do método CBL de forma simplificada durante as aulas e de maneira completa em pelo menos três situações reais de produção de software.
- Ensino de programação das linguagens base para o iOS: Swift e Objective-C.
- Ensino de desenvolvimento de Apps para iOS, desde a concepção da ideia até a publicação e possível criação de uma empresa.
- Imersão gradativa dos estudantes em um processo de desenvolvimento de software durante o seu aprendizado de programação. Essa imersão possibilita a vivência em boas práticas de programação e ambientação do estudante com as atividades e regras inerentes ao desenvolvimento de software de uma empresa.
- Colaboração dos estudantes com as quatro equipes de apoio ao desenvolvimento e publicação nas áreas de: usabilidade e interface; trilhas musicais e efeitos sonoros; ilustração, modelagem e animação; e empreendedorismo (*startups*). Essas equipes de apoio foram formadas por professores e estudantes da própria instituição. O objetivo das equipes foi o de fornecer os recursos complementares (não ligados à programação de software) necessários para a construção de apps profissionais.
- Workshops de formação auxiliar para necessidades dinâmicas e específicas dos projetos dos estudantes, tais como: desenvolvimento de sistemas web, utilização de bancos de dados, entre outros.

- Garantia de um projeto de App relevante para o desafio final, seja por escolha e criação dos próprios estudantes ou por meio das apresentações de projetos externos de professores.

Na Figura 5-2 é apresentada a visão geral da abordagem IDEAS e suas etapas: Selecionar perfil, Executar Curso, Produzir Software e Criar Startup. As próximas seções detalham cada uma das etapas e a sua aplicação.

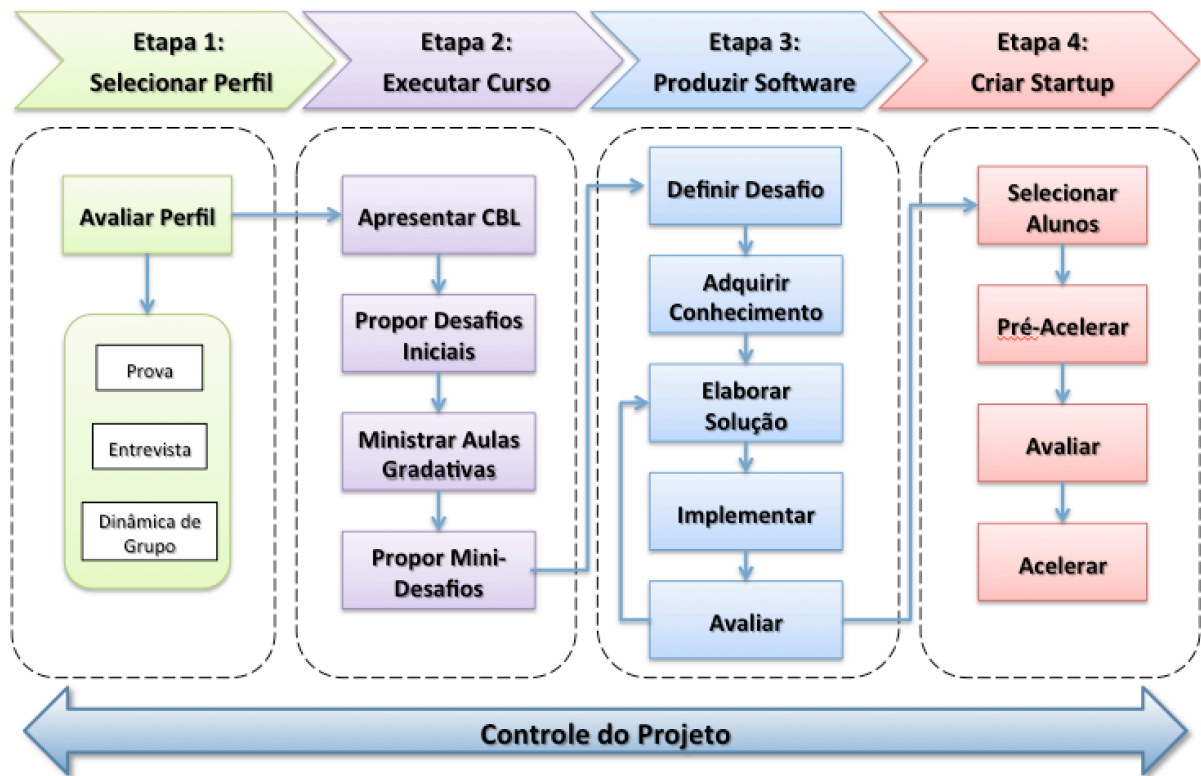


Figura 5-2: Abordagem IDEAS.

5.2 Implementação do Ciclo 2

5.2.1 Etapa 1: Selecionar Perfil

Conforme explicado no Capítulo 4, percebeu-se a necessidade de melhorar o processo para selecionar estudantes com o perfil específico necessário que envolvia: raciocínio lógico, autonomia, comprometimento, pró-atividade e colaboração. Para satisfazer a essa exigência foi ampliado o processo de seleção utilizado no Ciclo 1. Além da Prova (técnica) e da Entrevista, foi introduzida uma

Dinâmica de Grupo denominada Bootcamp. A Entrevista também foi ampliada, visando obter mais subsídios para a seleção.

A primeira parte da seleção é constituída de uma prova teórica composta de duas partes: programação básica (constantes, variáveis, condições, repetições e vetores unidimensionais) e raciocínio lógico:

- O objetivo da prova de programação é avaliar a capacidade de resolver problemas de programação e não medir conhecimento profundo em tópicos mais avançados. Dessa forma foram escolhidos apenas temas básicos de cursos de programação para evitar que estudantes iniciantes tivessem alguma desvantagem injusta com relação a estudantes mais experientes que conhecem tópicos avançados, mas que não são necessariamente excelentes programadores.
- A prova de raciocínio lógico tem o objetivo de identificar potenciais talentos de outras áreas diferentes de computação. Estudantes que conseguem uma boa avaliação na prova de raciocínio lógico ($\geq 90\%$), mas uma avaliação ruim na prova de programação ($< 60\%$) devem ser analisados com mais detalhes na Entrevista e no Bootcamp, conforme explicação posterior. Esse critério tem por objetivo identificar estudantes com pouca ou nenhuma experiência em programação, mas que tenham potencial para aprender programação durante o tempo do curso. Uma consequência direta desse critério é a não eliminação de bons candidatos com pouca ou nenhuma experiência em programação, mas com destaque em outros aspectos.

A segunda parte da seleção consiste na realização de uma Entrevista de até 30 minutos por estudante selecionado na Prova, utilizando-se como apoio um questionário (Apêndice C) e a técnica de entrevista de histórias de vida [ATKINSON, 2002]. Essa técnica consiste na enumeração dos eventos positivos e negativos pelo entrevistado e tem como objetivo principal a identificação dos seus interesses pessoais e do seu perfil.

A entrevista é composta de questões de aceite obrigatório em que são explicadas as condições do curso e questões sobre o histórico escolar, oportunidades para o estudante falar sobre sua vida escolar do ano atual e explicar eventuais exames finais ou dependências em disciplinas. Em seguida o estudante é

convidado a traçar sua linha de vida, identificando e explicando pontos importantes positivos e negativos de sua vida.

A última parte do processo é chamada de Bootcamp, que é uma dinâmica na qual os estudantes são divididos em equipes de 4 pessoas e têm 36 horas para elaborar uma proposta de solução para um problema real que não tem solução fácil e imediata. O objetivo dessa atividade é observar e avaliar os estudantes em uma situação crítica de trabalho. Como o método CBL exige um certo conjunto de atitudes por parte do estudante, essas atitudes são avaliadas nessa dinâmica. Os estudantes são avaliados em cinco aspectos: colaboração, liderança, pró-atividade, criatividade e autonomia.

O processo seletivo foi realizado com cerca de 900 candidatos. Após a correção da prova, foram selecionados 246 estudantes para a entrevista. A entrevista foi conduzida pelo autor, utilizando-se as perguntas definidas no Apêndice C. Foram selecionados 220 estudantes para a fase seguinte, uma dinâmica de grupo extensa chamada de Bootcamp. Nela os estudantes foram divididos em grupos de quatro pessoas e tiveram cerca de 36 horas para desenvolver uma solução para um problema atual e de difícil solução: encontrar alternativas para viabilizar a continuidade da mídia impressa em conjunto com o mundo digital. Os estudantes foram avaliados com base nos critérios: criatividade, colaboração, produtividade, adaptabilidade e comprometimento. Cada estudante foi avaliado por no mínimo três avaliadores (escolhidos entre criadores de *startups* bem sucedidas), sendo que as avaliações foram executadas e armazenadas em um aplicativo para dispositivos móveis desenvolvido exclusivamente para este fim. Após essa etapa foram escolhidos 100 estudantes para fazerem parte da segunda turma, originários dos cursos apresentados na Tabela 5-1.

Tabela 5-1: Estudantes x Cursos Ciclo 2 - Turma 2015.

Curso	Total	%
Ensino Médio Técnico em Informática	1	1%
Graduação Administração	1	1%
Graduação Análise de Sistemas	1	1%
Graduação Arquitetura e Urbanismo	1	1%
Graduação Ciência da Computação	11	11%
Graduação Ciências Econômicas	2	2%
Graduação Design Digital	1	1%
Graduação Design Gráfico	1	1%

Graduação Engenharia Ambiental	1	1%
Graduação Engenharia da Computação	14	14%
Graduação Engenharia de Controle e Automação	7	7%
Graduação Engenharia Elétrica/Eletrônica	16	16%
Graduação Engenharia Logística e Operações	1	1%
Graduação Engenharia Mecânica	1	1%
Graduação Engenharia Química	1	1%
Graduação Jogos Digitais	15	15%
Graduação Licenciatura em Física	1	1%
Graduação Licenciatura em Matemática	1	1%
Graduação Medicina	1	1%
Graduação Sistemas de Informação	10	10%
Especialização App Development	2	2%
Especialização Arte e Animação para Jogos Digitais	1	1%
Especialização Gestão da Qualidade	1	1%
Especialização Redes e Segurança	2	2%
Mestrado em Bioinformática	1	1%
Mestrado em Engenharia Elétrica	2	2%
Mestrado em Informática	2	2%
Doutorado em Informática	1	1%

A comparação entre a Tabela 3-1 e Tabela 5-1 mostra uma das consequências do processo seletivo alterado: no ciclo anterior 2,7% de estudantes eram de cursos com nenhuma disciplina de programação, enquanto no ciclo atual esse número subiu para 11%.

5.2.2 Etapa 2: Executar Curso

Esta etapa representa a formação dos estudantes tanto no uso do método CBL, quanto nos conteúdos específicos de programação e formação adicional (workshops).

Inicialmente o método CBL é apresentado em seu formato simplificado, ou seja, da grande ideia até as questões guias, atividades e recursos. Com o objetivo de tornar o método atrativo e motivador, são apresentados casos de estudo bem sucedidos do método, destacando-se sua característica flexível (ele possui vários pontos de entrada) e seu caráter dinâmico e colaborativo. Pretende-se com isso evitar a impressão de que o método é burocrático e com documentação desnecessária.

O método deve então ser executado uma primeira vez para uma tarefa rápida (usualmente menos de 20 minutos) mas de difícil execução para quem não é

especialista, como por exemplo, melhorar o áudio de um vídeo com gravação cheia de ruídos.

É de extrema importância que todas as equipes apresentem seus resultados e que se faça uma demonstração de como o resultado poderia ser obtido por meio de uma pesquisa que responda às perguntas certas (questões guias). Essa demonstração tem como objetivo tentar convencer os estudantes que preferem começar a tarefa com pouco ou nenhum entendimento do desafio.

Em seguida é proposta uma atividade mais extensa, mas que ainda não envolva o desenvolvimento de software. Nela os estudantes devem elaborar a ideia de uma App relevante e completa, da grande ideia até a “solução” que no caso é a produção de um vídeo de apresentação do produto final. O objetivo dessa atividade é a percepção de que o modelo pode ser usado para buscar novas ideias e transformá-las em produtos.

Um ponto muito importante que foi levantado por vários estudantes do primeiro ciclo foi o excesso de documentação e de pesquisa exigidos pelo método antes de se começar o desenvolvimento.

A terceira atividade proposta para os estudantes é a criação de um vídeo explicativo sobre um determinado assunto, orientação a objetos, por exemplo. Nesse tipo de atividade é essencial que se faça uma pesquisa sobre o conteúdo e que se reflita sobre as melhores maneiras de se explicá-lo.

Um dos objetivos do curso é que o estudante consiga uma autonomia completa, que ele se torne o próprio professor. Entretanto, como os estudantes estão acostumados ao modelo tradicional de aulas expositivas, uma aplicação gradativa se faz necessária. A partir de uma ementa básica são ministradas aulas curtas de até 50 minutos uma ou duas vezes por semana, com diminuição gradativa do tempo de aula (10 minutos) e respectivo aumento gradativo do tempo de atividades práticas. Os estudantes são então desafiados a desenvolverem aplicativos semanais, sempre em relação ao tópico corrente. Esses aplicativos devem ser sempre finalizados e apresentados até o início da semana seguinte. Após quatro ou cinco semanas acontece o primeiro Mini Desafio.

O Mini Desafio tem por objetivo colocar o estudante em uma situação de prazos críticos, pois ele deve desenvolver (em equipes de até quatro pessoas) um aplicativo relevante para publicação em duas semanas, tendo mais uma semana para realizar eventuais ajustes. Além disso, o estudante começa a receber noções

de um processo de desenvolvimento de software para ajudá-lo a cumprir a tarefa e preparando-o para o desafio final, no qual o processo de desenvolvimento de software estará totalmente integrado ao método CBL.

Os estudantes têm dois dias para definir o desafio, dois dias para executar a etapa de questões guias, atividades e recursos e no quinto dia eles são orientados por um professor sobre o planejamento, divisão de tarefas e cronograma. Durante esse período os estudantes são observados e acompanhados pelos professores, pois apesar da autonomia ser incentivada, muitas vezes um problema crítico pode ser resolvido mais rapidamente com o auxílio de um professor. Ao final de duas semanas todas as equipes apresentam seus produtos junto com um vídeo promocional e um vídeo de reflexão individual.

A partir deste ponto devem ser oferecidas aulas de reforço para os estudantes que tiveram desempenho abaixo do esperado nas atividades semanais e no primeiro Mini Desafio. Os demais tópicos da ementa são comentados rapidamente e logo em seguida os estudantes devem realizar estudos autônomos sempre com o objetivo de desenvolver algum aplicativo.

Após mais quatro semanas os estudantes devem encarar o segundo Mini Desafio onde mais uma vez deverão desenvolver uma App relevante para publicação. O objetivo do segundo desafio é consolidar as habilidades de desenvolvimento de software sob pressões de prazo. Esse Mini Desafio segue os mesmos moldes do anterior, também com prazo de duas semanas e mais uma de ajustes, apresentação e publicação.

5.2.3 Etapa 3: Produzir Software

Esta etapa tem por objetivo completar o desafio final, ou seja, desenvolver e publicar uma App relevante. Com base na avaliação do Ciclo 1 que identificou a dificuldade dos alunos em desenvolver os desafios de programação, foi feita a integração do método CBL com um processo de desenvolvimento de software, conforme pode ser observado na Figura 5-3. As caixas com fundo escuro representam etapas existentes do método CBL e as caixas com fundo branco representam as etapas do processo de desenvolvimento integrado. Não estão contempladas na figura as interações com as equipes de apoio de produção de conteúdo (usabilidade, artes e sonorização) e nem as reuniões de acompanhamento. As quatro primeiras etapas do CBL foram inseridas dentro de

atividades maiores devido à necessidade de se realizar outras tarefas complementares. As outras três etapas do CBL (solução, implementação e avaliação) possuem, originalmente, definição bem abstrata e sem detalhamento. Foi realizado o detalhamento de cada uma com tarefas relacionadas a desenvolvimento de software, como será explicado a seguir.

A integração é necessária para contemplar a obrigatoriedade de capacitar estudantes de diferentes níveis de conhecimento em desenvolvimento de software a produzir um aplicativo. Os estudantes são imersos em um processo complexo de desenvolvimento de software, semelhante a uma fábrica de software onde vários produtos são desenvolvidos (no caso os projetos de cada equipe) de forma sistemática, com prazos bem definidos e utilizando recursos comuns a todos, representados nesta abordagem pelas equipes de apoio de produção de conteúdo e pelos professores que atuam tanto como gestores dos projetos quanto consultores em questões técnicas.

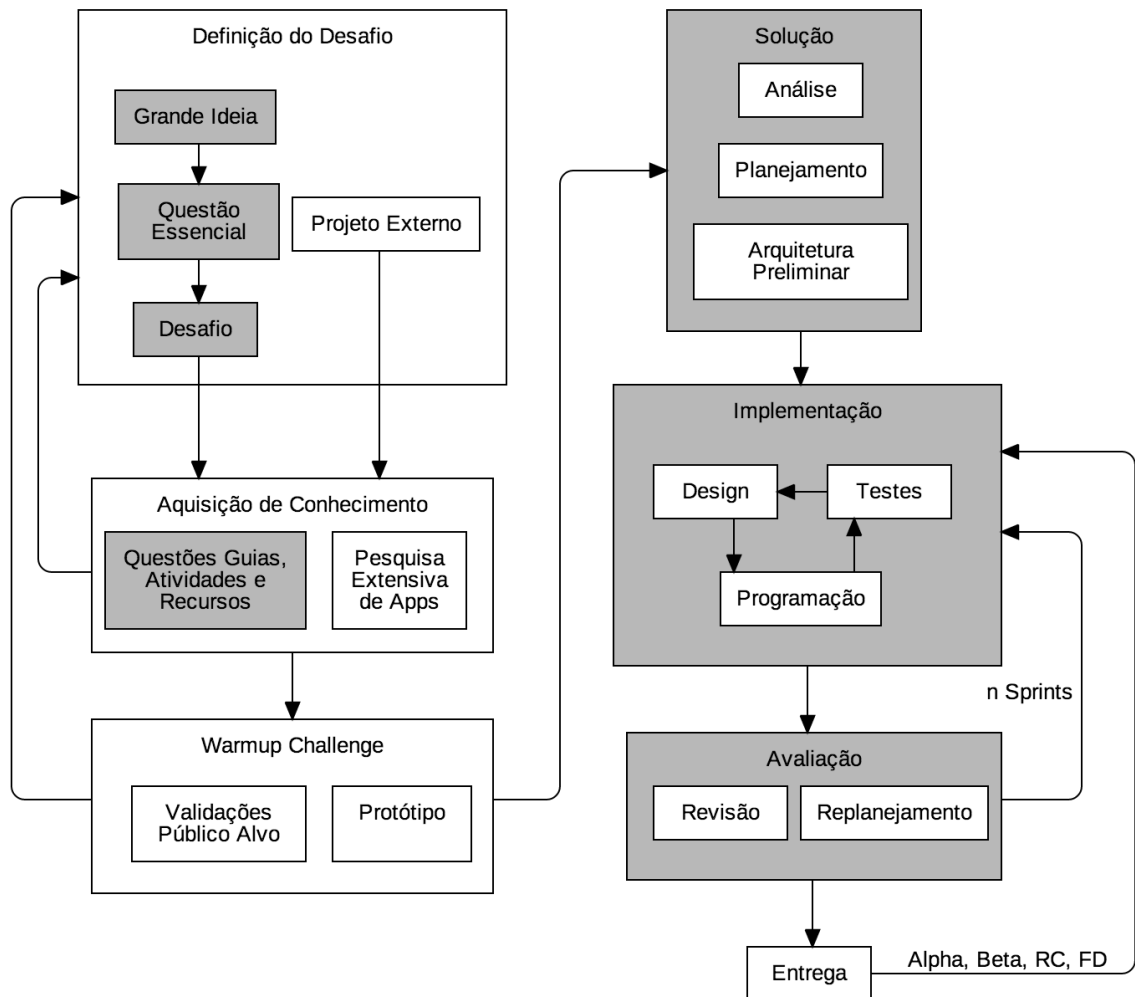


Figura 5-3: Detalhamento da Etapa 3 da Abordagem IDEAS: Produzir Software

A primeira atividade dos estudantes é a definição do desafio que pode ser feita com a utilização das etapas iniciais do método CBL (Grande Ideia, Questão Essencial, Desafio) quanto pela escolha de um projeto externo.

A atividade seguinte, denominada aquisição de conhecimento tem por objetivo a busca pelo entendimento do desafio e é realizada usando-se a etapa Questões Guias, Atividades e Recursos do método CBL aliada a uma pesquisa extensiva por aplicativos similares. Caso seja identificada alguma questão que inviabilize o projeto, volta-se para a primeira atividade.

A etapa seguinte, chamada de Warmup Challenge tem por objetivo a validação adicional do software, antes do comprometimento sobre a sua produção. Dois caminhos podem ser seguidos: caso haja a intenção de abertura de uma *startup*, são realizadas as validações de público alvo; ou então é desenvolvido um

protótipo de viabilidade onde um ou mais pontos críticos técnicos do projeto são implementados e validados.

As três atividades seguintes (solução, implementação e avaliação) estão definidas no método CBL de uma forma abstrata, sem detalhamento sobre como devem ser executadas. Para contemplar o objetivo de produzir um software foram incluídos nestas atividades, passos comuns de processos de desenvolvimento de software conforme será explicado adiante.

A última atividade é a entrega que representa as quatro entregas realizadas pelos estudantes: versão alfa, versão beta, release candidate e final delivery. Em cada uma delas os estudantes entregam uma série de artefatos e apresentam seus resultados para os professores. Nas próximas seções são detalhadas as atividades aqui resumidas.

Definição do Desafio

Para a escolha da grande ideia os estudantes têm duas opções: escolher um projeto externo ou buscar uma ideia própria a partir das três primeiras atividades do CBL: (i) grande ideia ou definição do tema; (ii) definição da questão essencial e; (iii) definição do desafio.

O projeto externo pode ser escolhido a partir das apresentações de projetos propostos por professores externos ao curso mediante preenchimento de formulário específico (Apêndice E) e análise pela equipe de professores. O formulário foi dividido em 4 partes:

- Dados pessoais: identificação do professor e em que curso está lotado.
- Dados do projeto: identificação do projeto, como Nome do Projeto, Área (saúde, educação, segurança), Descrição, Perspectiva de Uso, Público Alvo e tipo de retorno esperado (financeiro, portfólio, social, etc.). Parte dessas perguntas tentam aproximar o professor do método CBL de maneira indireta.
- Análise de Forças, Fraquezas, Oportunidades e Ameaças (SWOT): Optou-se pela metodologia SWOT pois suas reflexões são um ótimo ponto de partida para a composição das questões essenciais.
- Recursos necessários: Tentando mapear recursos humanos e tecnológicos, além de referências para aplicações similares.

Caso o estudante prefira buscar sua própria ideia, deve começar pela grande ideia ou definição do tema que consiste na escolha do problema que pretende-se resolver. Nesta atividade, existe o mínimo de intervenção possível para que o estudante desenvolva sua percepção e autonomia. Para isto, são realizadas as seguintes atividades: (i) estratégias de definição de tema; (ii) construção de lista de problemas críticos e; (iii) avaliação do professor.

As estratégias de definição do tema são atividades de auxílio e direcionamento para escolha do tema. Por exemplo, a realização de brainstorm; busca de Apps (softwares criados para dispositivos móveis) atuais de interesse; leitura de um livro; consulta a profissionais.

O resultado da realização de uma ou mais destas estratégias é uma lista de problemas críticos relacionados ao tema central. Em seguida, o estudante faz uma pesquisa exploratória sobre estes problemas críticos e escolhe um dos problemas de acordo com seu interesse.

Finalmente, a escolha do tema passa por um processo de avaliação do professor que decide se o tema é relevante ou não. São utilizados os critérios:

- Inovação: Procura-se identificar se a solução proposta realmente possui algum diferencial, ou se simplesmente replica algum modelo de negócio já consolidado;
- Escopo: Identifica se o escopo era adequado para o tempo de projeto;
- Inadequação às Regras: Verifica se existe algo que vá contra as normas vigentes de publicação na loja virtual.

Após a escolha do tema central, o estudante tem que definir a questão essencial. A questão essencial é uma pergunta que o estudante escolheu dentre os problemas críticos do tema selecionado.

A partir da questão essencial é definido o desafio que se pretende enfrentar e resolver por meio do desenvolvimento de uma App.

Aquisição de Conhecimento

Esta etapa consiste em aprofundar o conhecimento sobre o desafio, ou seja, são levantadas as características sobre o problema, bem como uma pesquisa extensiva sobre as possíveis soluções já identificadas. Para isto, é oferecida ao

estudante a Quadro 5-2 composta por questões guias, atividades, recursos, o que foi aprendido e avaliação.

Quadro 5-2: Guia para a Etapa: Questões Guias, Atividades e Recursos.

Questões Guias	Atividades e Recursos Guias	O que Aprendi?	Método de Avaliação
Liste todas as questões que você precisa responder para encontrar a melhor solução para o seu desafio.	Liste as atividades e recursos que você utilizará para responder cada questão guia.	Após completar a atividade, o que você aprendeu sobre a questão guia?	Como você vai avaliar se a questão guia foi respondida efetivamente?

As questões guias são uma série de perguntas desenvolvidas pelos estudantes que identificam e representam o conhecimento e as habilidades necessárias para o desenvolvimento da solução. Devem ser feitas de forma a cobrir todo o escopo de conhecimento necessário para entender o problema e elaborar uma solução bem sucedida.

Para responder as questões guias o estudante tem que definir atividades, recursos e uma avaliação crítica das respostas. As atividades são as ações que devem ser executadas a partir de um recurso para responder as questões guias.

A atividade pode ser qualquer forma de aquisição de conhecimento, por exemplo, a realização de entrevistas com especialistas sobre o tema, pesquisas bibliográficas, assistir palestras, fazer um curso, etc.

O recurso é qualquer tipo de fonte de conhecimento necessária para responder a pergunta, por exemplo: sites, livros, artigos, cursos, especialistas, etc. Após, a realização da atividade o estudante tem que realizar uma avaliação crítica se a resposta atendeu a questão guia.

A avaliação é definida e executada pelos estudantes e depende do tipo da atividade. Este processo é iterativo, sendo executado até que seja alcançado o objetivo da questão guia.

Embora o processo da CBL seja suficiente para uma ampla gama de projetos, houve necessidade de impor documentação adicional para atender a requisitos específicos do curso:

- Pesquisa extensiva de apps semelhantes: Um requisito específico do curso é a criação de projetos originais ou pouco explorados. No mínimo, o estudante deve ter uma visão clara de qual diferencial está oferecendo com seu aplicativo, e o que o tornará único em relação ao que existe na loja.
- Mockup² de telas: os estudantes devem elaborar um *mockup* inicial das telas com o objetivo de traçar uma visão geral do projeto. Projetos pequenos podem ser aceitos somente se a ideia for extremamente inovadora e original. Projetos muito ambiciosos devem ter o escopo discutido com o estudante. O mockup também mostra-se valioso pois ele auxilia a equipe a ter uma visão mais concreta do aplicativo e ter uma noção mais detalhada de que trechos da sua aplicação ainda não estão bem definidos.

Warmup Challenge

Nesse ponto o estudante deve decidir se há interesse na criação de uma *startup* para o seu futuro produto ou não. Caso opte pela *startup* deve ser criado um canvas [OSTERWALDER; PIGNEUR, 2010] e realizada uma série de validações seguindo técnicas de Design Thinking [POLAINE et al., 2013] junto com a equipe de apoio de empreendedorismo.

Caso contrário, com base na experiência dos professores, tenta-se levantar, para cada projeto, um ou mais pontos considerados críticos. Trata-se de um desafio de projeto complexo ou pouco conhecido - preferencialmente um fator que possa representar alto risco para o projeto. As equipes devem elaborar um protótipo de viabilidade técnica usando os pontos críticos levantados. Esse protótipo não deve ter qualquer relação com o produto final e deve ser descartado após análise dos professores. O objetivo aqui é evitar que os estudantes (principalmente os mais inexperientes) utilizem esse protótipo como versão inicial de seu produto.

Dependendo do projeto, pode ser necessário realizar ambas as atividades, tanto a da criação da *startup* quanto a do desenvolvimento do protótipo de viabilidade.

² Modelos de interfaces de tela sem a funcionalidade subjacente.

Esta etapa, que possui uma duração máxima de duas semanas, tem como objetivo oferecer uma oportunidade para o estudante avaliar a viabilidade de seu projeto e definir melhor o desafio com o qual vai trabalhar. O Quadro 5-3 mostra exemplos de alguns *Warmup Challenges* propostos durante a Turma 2015.

Quadro 5-3: Exemplos de Projetos Propostos para o Desafio Final da Turma 2015.

Projeto	Desafio	Protótipo proposto	Objetivos
Underground survivors	Game utilizando física. O estudante já havia feito um game similar em um dos mini desafios.	Testar implementação com o framework Metal, pouco conhecido e com documentação escassa.	Identificar se é melhor investir na nova tecnologia gráfica (Metal) ou se é melhor manter uma tecnologia já dominada (OpenGL).
Cointer	Contar moedas	Tentar identificar moedas de 1 real.	Aprender sobre a integração da OpenCV, uso da câmera e processamento de imagem.
Ready 2 Order	Fazer pedidos em restaurantes sem garçom	Estudar modos de cobrança de cartão.	Traçar uma estratégia de negócio para realizar a cobrança dos clientes, estudar o modelo da loja e de brokers de cartão.

Solução

Segundo a definição do método CBL, “a solução (ou soluções) deve ser bem pensada, concreta, claramente articulada e passível de execução” [CBL, 2011]. Considerando que o levantamento de requisitos foi realizado na etapa de aquisição do conhecimento, essa definição de solução foi detalhada na abordagem IDEAS como sendo a parte da análise do sistema posterior ao levantamento de requisitos, definição da arquitetura inicial e planejamento.

A análise inicia com a equipe de estudantes realizando a tarefa de divisão do projeto em módulos menores por meio da criação de um diagrama de Work Breakdown Structure (WBS), como recomendado pelas práticas do Project Management Body of Knowledge (PMBOK). A partir desse diagrama deve ser

elaborado o planejamento estimado do projeto por meio de um diagrama de Gantt para representar o cronograma previsto.

Sugeriu-se o uso da ferramenta Freemind³ para diagramação do WBS na forma de um Mind Map. A Figura 5-4 mostra parte de um exemplo de WBS escolhido dentre os projetos apresentados na segunda turma.

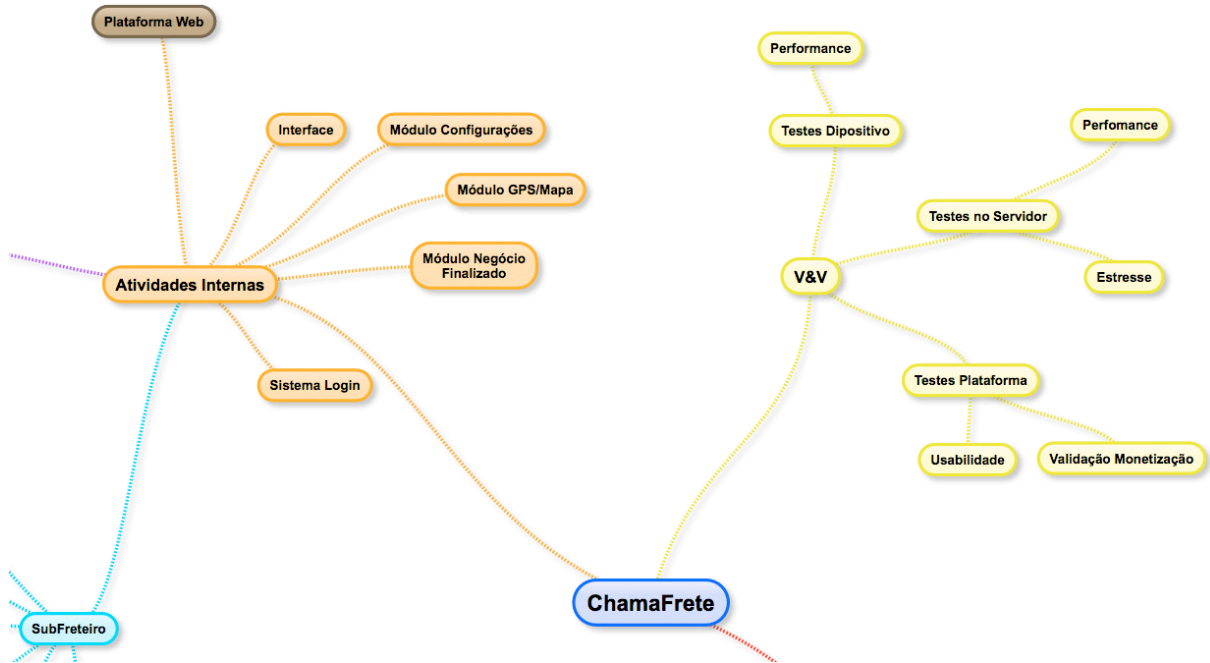


Figura 5-4: Trecho de WBS do projeto Chama Frete.

Para o planejamento do projeto e criação do diagrama de Gantt foi sugerida a ferramenta Gantt for Google Drive. A Figura 5-5 mostra o diagrama de Gantt do mesmo projeto Chama Frete.

³ Software para criação de mapas mentais.

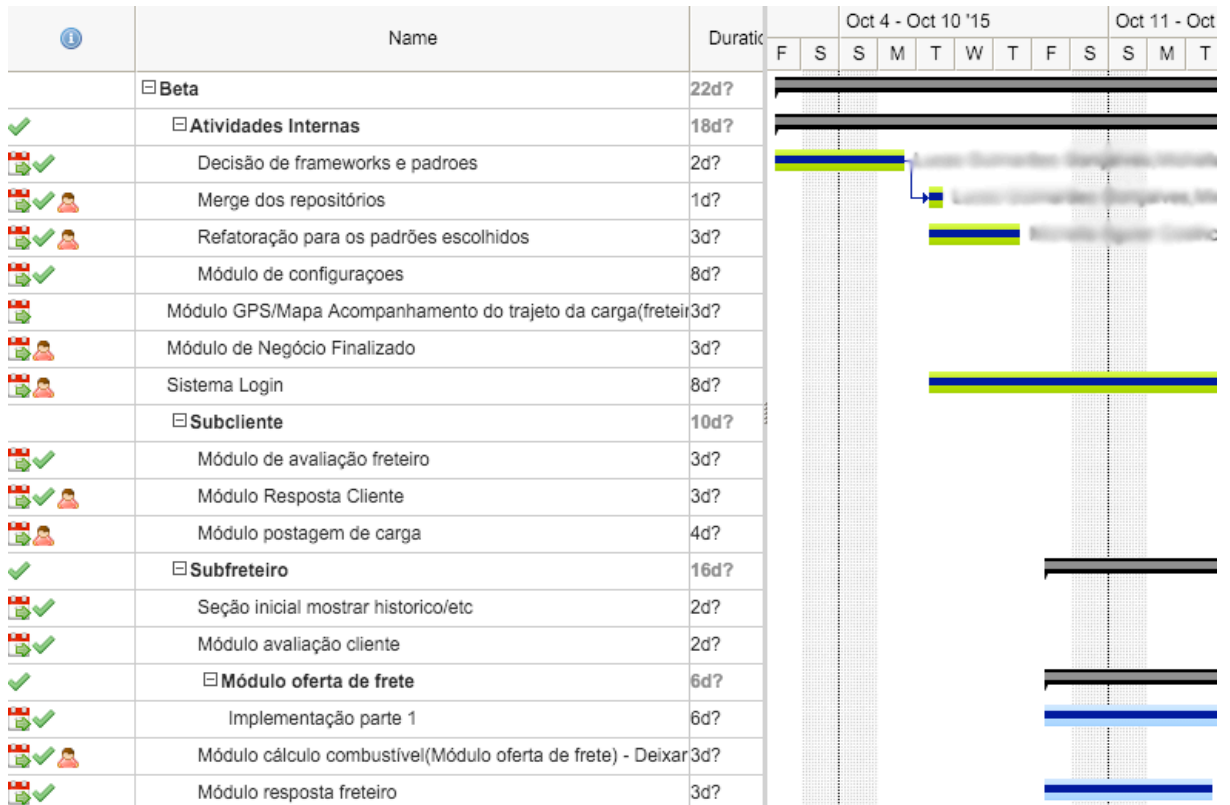


Figura 5-5: Trecho de Diagrama Gantt do Projeto ChamaFrete.

A maior parte das dificuldades encontradas durante a fase de planejamento se deram devido a inexperiência dos estudantes e a diversidade dos projetos. O Quadro 5-4 mostra essas dificuldades, as ações corretivas tomadas nesse ano e ações corretivas sugeridas para o próximo ano:

Quadro 5-4: Dificuldades x Ações Tomadas no Desafio Final da Turma 2015.

Dificuldade	Ação corretiva	Ação preventiva
Inexperiência dos estudantes na divisão de tarefas	Nenhuma.	Realizar um pequeno ciclo de planejamento nos mini-desafios.
Inexperiência dos estudantes na elaboração do Gantt	Reunião com estudantes caso-a-caso	Aulas e workshops sobre gestão de projetos. Estimular que os estudantes estimem prazos e usem o gantt em um dos mini-desafios.
Projetos de cunho científico, cujas tarefas eram difíceis ou impossíveis de se estimar	Mudança da perspectiva do Gantt de atividades para ciclos (inspeção de técnicas, refinamento, produto final).	Estudar metodologia específica para gestão desse tipo de projeto. Orientação mais próxima aos estudantes.

Equipes onde nenhum membro cursava faculdade de informática	Não houve.	Aumentar o rigor para desestimular equipes com esse tipo de formação.
Falta de percepção de valor dos estudantes no método proposto	Imposição da documentação associada a nota	Workshops práticos de gestão de projetos, onde seriam mostrados exemplos de como as práticas auxiliaram projetos anteriores.

Implementação

Foi definido para esta etapa um ciclo iterativo de design, programação e testes. Devido à necessidade de uma interface robusta e requisitos de usabilidade dinâmicos foi definido um processo onde a equipe de usabilidade fornecia o design da interface, em seguida os estudantes executavam a programação e os testes com representantes do público alvo. Os resultados eram repassados para a equipe de usabilidade e o ciclo se repetia até que não houvessem mais problemas críticos de usabilidade.

Para o acompanhamento semanal de atividades, sugeriu-se o uso de um quadro de atividades similar ao do Scrum, porém, construído por meio do uso de uma ferramenta digital, para possibilitar o acompanhamento pelo professor responsável. A ferramenta sugerida foi o Trello⁴ cujo projeto Chama Frete está apresentado na Figura 5-6.

⁴ Software usado no gerenciamento de projetos.

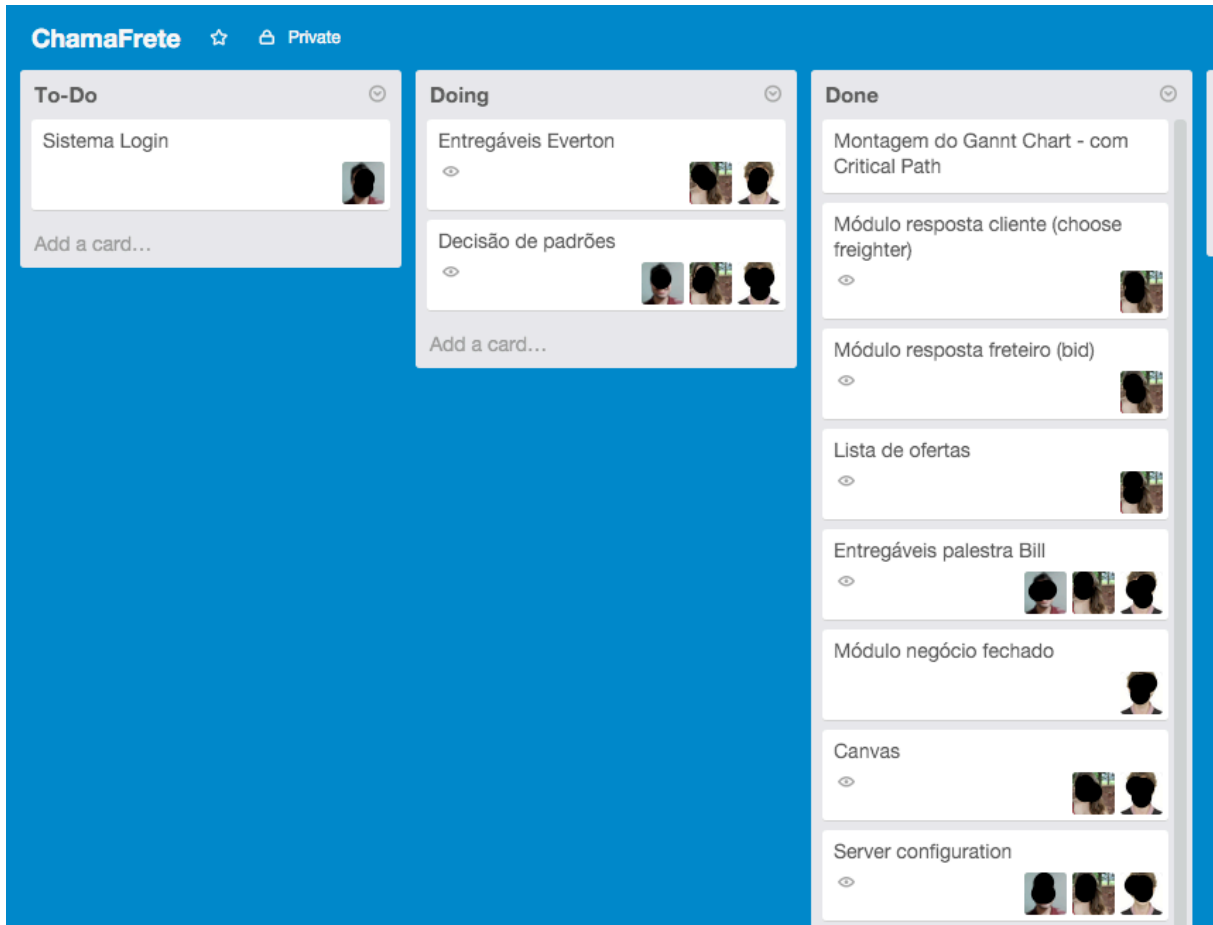


Figura 5-6: Quadro SCRUMM do Projeto ChamaFrete.

Foi deixado a cargo da equipe decidir como os membros se comunicariam e se gerenciariam entre si. As equipes também tinham liberdade de produzir outros diagramas que julgassem necessários - tais como mockups, diagramas de fluxo de dados, etc.

Foi imposto às equipes o uso da ferramenta de controle de versão Git. A maior parte dos estudantes optou pelo Bitbucket⁵ como repositório do seu projeto. Os professores receberam um login e senha, para que pudessem acompanhar a evolução do código.

A ferramenta Gitinspector, desenvolvida pela Universidade de Tecnologia de Chalmers e pela Universidade de Gothenburg foi utilizada para resolver os casos de conflito de versão. A ferramenta permite visualizar qual estudante contribuiu com o maior número de linhas de código no projeto e em que semana enviou conteúdo

⁵ Serviço de hospedagem para projetos que necessitam de um controle de versionamento.

para o repositório. A Figura 5-7 mostra o trecho relevante do relatório do gitinspector para o projeto Gems:



Figura 5-7: Trecho de Relatório do Gitinspector

O primeiro gráfico do relatório mostra a contribuição de cada participante da equipe durante todo o decorrer do projeto, o relatório é obtido contando-se todas as linhas submetidas pelo membro.

O segundo gráfico mostra a contribuição do participante no projeto, levando em conta apenas as linhas submetidas pelo membro que ainda estejam no repositório. Dessa forma, linhas apagadas não serão contadas.

O terceiro gráfico mostrou-se o mais útil no acompanhamento. Mostra as linhas inseridas e removidas por membro do projeto a cada semana.

É importante ressaltar que, como o controle do projeto ficou a cargo da equipe, o relatório não foi usado com fim fiscalizatório. Ele foi exibido nas primeiras reuniões, para que os estudantes soubessem de sua existência e que a equipe de professores possuía a capacidade de retirá-lo.

Demais gerações do relatório ocorreram apenas sob demanda ou em caso de conflito na equipe. Algumas equipes estavam perfeitamente confortáveis com um dos membros cuidando de tarefas que não envolvessem código, tais como integração com núcleos de apoio, geração de conteúdo, controle de tarefas, design etc.

A Quadro 5-5 mostra as principais dificuldades encontradas durante a etapa de Implementação, as ações corretivas tomadas durante esse projeto e as ações preventivas cogitadas para futuros projetos:

Quadro 5-5: Dificuldades na Etapa de Implementação do Desafio Final da Turma 2015.

Dificuldade	Ação corretiva	Ação preventiva
Falta de conhecimento técnico dos estudantes que optaram por aplicativos com servidor.	Workshop de Ruby on Rails	Formalização do workshop de uma tecnologia de servidor Estimular estudantes a criar apps que não envolvam servidores
Quebra de projetos devido atualização do XCode	Nenhuma	Orientar estudantes a não realizarem atualizações antes de datas críticas.
Desistência de estudante sem formação de TI em projeto individual	Relocação do estudante	Desestimular ao máximo essa situação.
Membro da equipe improdutivo	Reuniões e conversas com o membro, exibindo o relatório de gitinspector.	Uso mais efetivo do gitinspector. Acompanhamento mais próximo das atividades realizadas por membro.
Atraso dos núcleos de apoio	Reuniões com os responsáveis pelo núcleo. Contratação de profissionais externos.	Melhor organização e transparência do cronograma dos núcleos de apoio.

Controle do Projeto

O planejamento do controle e acompanhamento do Ciclo 2 (turma 2015) foi realizado com base em lições aprendidas do ano de 2014:

- O acompanhamento dos projetos entre todos os professores do grupo tornou a gestão complexa, pois era difícil manter unicidade de metodologia entre tantos professores diferentes;
- Diferente dos demais grupos de apoio, a gestão do núcleo de artes (NAJA) sobrecarrega um dos professores do curso, e demanda atenção especial;

- Evitar sobrecarga de estudantes com atividades burocráticas durante a execução do projeto.

Assim, decidiu-se que o planejamento e controle dos 47 projetos seria realizado por apenas dois professores, ambos com experiência empresarial em gerência de projetos. A divisão foi realizada da seguinte forma:

- Professor 1: Acompanhamento de 18 projetos (40% do total), além da gestão do grupo de apoio de ilustração NAJA.
- Professor 2: Acompanhamento dos demais 29 projetos.

Como consequência dessa divisão, foi priorizado ao Professor 1 a maior parte dos projetos que possuíam maior necessidade de artes, tais como os games e os projetos educacionais. Os demais professores ficariam disponíveis para tirar dúvidas técnicas dos estudantes, além de atividades burocráticas e de gestão do curso como um todo.

Foi elaborado um diagrama de Gantt macro, contendo todas as equipes de determinado turno e seu status durante as etapas Alpha, Beta e Release Candidate. A Figura 5-8 mostra parte do diagrama de Gantt do período noturno. O objetivo desse diagrama era identificar equipes em atraso, além de servir como uma forma de comunicação rápida.

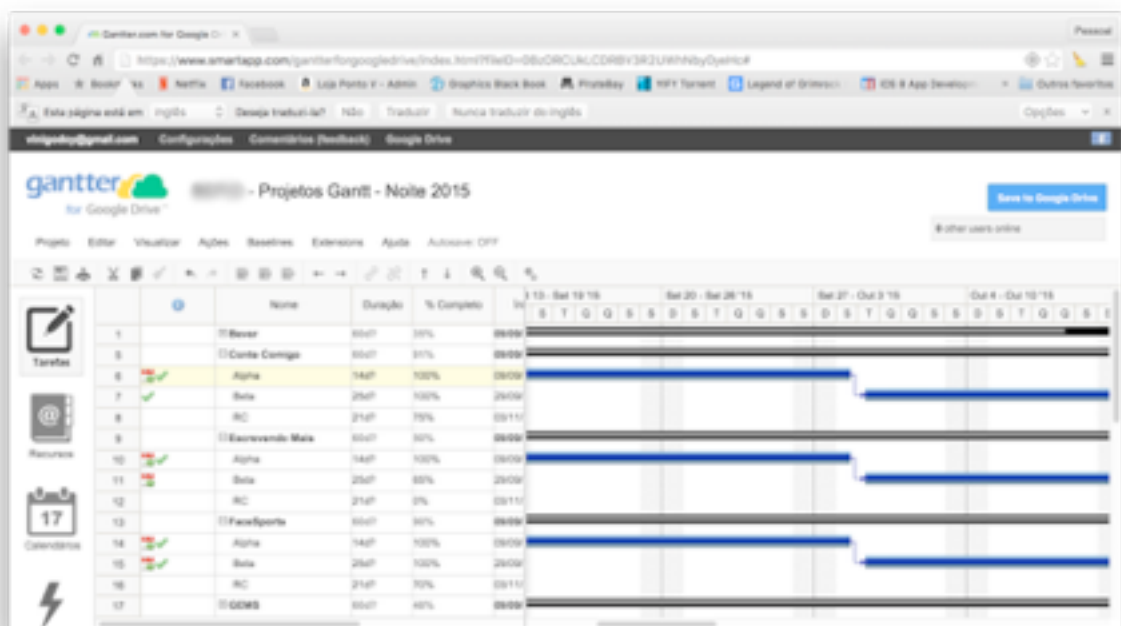


Figura 5-8: Exemplo de Diagrama Gantt - Classe da Noite da Turma 2015.

O acompanhamento das equipes seria feito por meio de reuniões semanais, de 15 minutos cada, onde seriam discutidos:

- Avanço percentual do projeto
- Atividades realizadas na última semana por membro da equipe
- Dificuldades encontradas
- Atividades a serem realizadas na próxima semana
- Atendimento pelos núcleos de apoio
- Riscos identificados

As reuniões foram registradas em um documento chamado “History Log”. Este documento foi feito pelo professor, na própria reunião, evitando sobrecarga para os estudantes. O tempo da reunião mostrou-se adequado para o acompanhamento geral. É importante ressaltar que não era papel da reunião, ou dos professores, dar sugestões sobre o conteúdo dos projetos ou aprofundar-se nas relações entre os estudantes. A reunião tinha como objetivo primário identificar problemas para que pudessem ser reportados para os demais professores ou núcleos de apoio.

Entrega

Durante a etapa de planejamento, dividem-se os projetos em blocos menores e estima-se o tempo de desenvolvimento. São definidos quatro prazos principais de entregas conforme o Quadro 5-6, descritos abaixo:

- a) **Alfa version:** Prazo de um mês. O objetivo deste *milestone* é a disponibilização de uma versão com as funcionalidades principais da aplicação. O design final da aplicação não é necessário.
- b) **Beta version:** Prazo de um mês e meio a partir da entrega da versão alfa. Todas as funcionalidades principais planejadas já devem estar implementados. Não é necessário que todo o design esteja aplicado ao software, mas é desejável que pelo menos parte dele já esteja funcionando. Além disso, fluxos alternativos para a mesma funcionalidade não precisam estar implementados.
- c) **Release candidate (RC):** O mês final de desenvolvimento deve ser reservado para a aplicação do design final, polimento, testes e correções

de bugs. Ao final do período, uma versão release candidate do aplicativo deve ser disponibilizada para apresentação final em banca.

- d) **Final delivery (FD):** Entrega final do aplicativo, um mês após a RC. É considerado entregue o aplicativo submetido e com o aceite da loja. Para que o projeto fosse considerado finalizado, exigiu-se dos estudantes a organização do seguinte pacote de entrega:
- Código fonte da aplicação
 - Materiais desenvolvidos pelos núcleos de apoio
 - Documentos da CBL, incluindo um vídeo de reflexão final
 - Pasta com a descrição e screenshots da aplicação que seriam enviados para a loja
 - Vídeo institucional do produto final
 - Vídeo de reflexão final do curso

Além disso, todas as equipes estavam obrigadas não só a enviar o app desenvolvido para a loja como também obter sua aprovação. A etapa de aprovação pode exigir diversos ajustes e reenvios até que o app seja aprovado. Após a aprovação, o app não precisaria ser publicado, podendo ser mantido em status “on hold”.

Quadro 5-6: Cronograma de Execução dos Projetos da Turma 2015.

	Setembro					Outubro				Novembro					Dezembro	
	01	07	14	21	28	05	12	19	26	02	09	16	23	30	07	14
Alpha	■	■	■	■												
Beta					■	■	■	■	■							
RC										■	■	■	■	■		
FD															■	■

5.2.4 Etapa 4: Criar Startup

A criação de *startups* foi feita em parceria com a HotMilk, aceleradora de empresas da PUCPR. Foi reformado um local dentro da HotMilk com 44 postos de trabalho exclusivos para *startups* de egressos do curso.

Esta etapa foi composta de quatro fases: seleção dos estudantes, pré-aceleração, avaliação e aceleração, seguindo as definições de [HUIJGEVOORT, 2012]. Até a publicação deste documento, apenas a fase de seleção havia sido realizada.

O processo seletivo foi iniciado no final da turma de 2015 e estava aberto exclusivamente para os estudantes que haviam demonstrado interesse na criação de *startups* durante o *Warmup Challenge*. Os egressos foram selecionados pelo pesquisador e pelo professor responsável pela equipe de apoio ao empreendedorismo dentre os inscritos para esta fase com o devido preenchimento do formulário disponível no Apêndice D. O critério para seleção foi o desempenho dos estudantes durante o curso e a produção do software. Excepcionalmente, pelo fato de que todos os inscritos haviam tido um excelente desempenho durante o curso, todos foram selecionados para a pré-aceleração.

O programa de pré-aceleração dura em média 2 meses e prepara o futuro empreendedor para gerenciar o seu negócio por meio de minicursos e orientação individual (mentoria). Normalmente, parte-se da ideia e chega-se ao *minimum viable product* (MVP). Entretanto, no caso dos egressos do curso, todos já tem um produto pronto, seja em versão definitiva ou em uma versão com um mínimo de funcionalidades que o tornem útil mas que ainda lhe faltem algumas funcionalidades adicionais. Eles são considerados como *makers*, ou seja, empreendedores que efetivamente desenvolvem o próprio produto. A pré-aceleração os prepara para a vida de empreendedor por meio do entendimento do modelo de negócio, marketing, finanças e potenciais desafios e dificuldades futuras. Também os ajuda a ajustar o produto desenvolvido para o mercado.

A seleção para a aceleração é a próxima etapa, na qual normalmente procura-se identificar quais sócios de *startups* são, ao mesmo tempo *makers* e empreendedores. Como os egressos do curso são todos *makers*, a seleção resume-se a encontrar aqueles que tem perfil empreendedor. O critério é basicamente o alcance das metas traçadas pelos mentores, indicando um comprometimento com a cultura empreendedora. Ou seja, na seleção o importante não é a ideia ou mesmo o projeto, mas sim o comportamento dos egressos diante das dificuldades comuns encontradas por empreendedores.

Na etapa de aceleração que pode durar entre seis meses e um ano, o produto já está no mercado, sendo utilizado por clientes, gerando engajamento e atração e

possivelmente produzindo faturamento. Nesta etapa são dois os objetivos: conexão e alavancagem. A aceleradora ajuda as *startups* a entrarem em uma cadeia de relacionamento, com possibilidades de parceria, sociedade, indicação de clientes e até mesmo investimento, alavancando assim o negócio.

5.3 Avaliação do Ciclo 2

Foram realizados diversos tipos de avaliações sobre a abordagem IDEAS durante e ao final do Ciclo 2 (Turma 2015):

- Entrevistas de acompanhamento gravadas em vídeo com nove estudantes de acordo com sua posição no processo seletivo: os três primeiros, os três últimos e os três do meio. Cada estudante foi entrevistado duas vezes: uma no início e outra após o término da parte teórica do curso (quatro meses).
- Reflexões semanais dos estudantes gravadas em áudio e vídeo, com o objetivo de verificar eventuais problemas e avaliar sugestões e críticas pontuais.
- Reflexões periódicas dos professores, escritas ou gravadas em áudio e vídeo, com os mesmos objetivos das reflexões dos estudantes.
- Questionário final para os estudantes com questões objetivas e subjetivas com o objetivo de verificar os mesmos itens avaliados no primeiro ciclo com o acréscimo de três perguntas (Apêndice A).
- Aplicação de uma prova de avaliação de conhecimentos de programação para os egressos que obtiveram grau inferior a 50% na prova do processo seletivo.
- Entrevista final com os professores com questões objetivas e subjetivas com o objetivo de verificar os mesmos itens avaliados no segundo ciclo com o acréscimo de perguntas sobre a abordagem IDEAS apenas para os professores da PUCPR (Apêndice L).

O Ciclo 2 (Turma 2015) apresentou alguns resultados diretos, ou seja, resultados positivos mensuráveis que independem de avaliações específicas.

O principal resultado positivo mensurável foi o percentual de egressos das turmas: 76,3% na primeira turma e 95% na segunda turma. A primeira turma teve 18,2% dos estudantes excluídos por problemas de desempenho e comprometimento

enquanto a segunda turma não teve nenhum. Levando-se em conta que não houve mudança nos critérios de exclusão, no prazo e complexidade dos trabalhos, nos materiais e equipamentos fornecidos e na disponibilidade dos professores para atendimento fora do horário de aula, é seguro considerar que as mudanças efetuadas no segundo ciclo, principalmente aquelas referentes ao processo seletivo foram bem sucedidas. Esse resultado pode ser atribuído à qualidade geral dos estudantes, fruto da melhoria no processo seletivo.

Outro resultado positivo foram as avaliações dos projetos finais realizadas por uma banca composta por pelo menos um professor do curso e por três profissionais especializados na avaliação de Apps. Na primeira turma, 46,7% dos projetos tiveram avaliação excelente (igual ou superior a 80%) enquanto no segundo ciclo, 88,9% dos projetos alcançaram tal avaliação. A imersão dos estudantes dentro de um processo de desenvolvimento de software integrado ao método CBL possivelmente ajudou neste resultado.

Um terceiro resultado expressivo foi com relação aos estudantes selecionados para a pré-aceleração de suas *startups* na aceleradora da PUCPR: 14,2% dos egressos no primeiro ciclo e 38,9% no segundo ciclo. A inclusão de uma equipe de apoio ao empreendedorismo, workshops de monetização e avaliação de público alvo, palestra com um dos nomes mais respeitados no mundo das *startups* e o incentivo aos projetos comerciais foram os responsáveis por essa melhoria.

E finalmente, a quantidade de Apps aceitas e publicadas na loja de aplicativos: 41 publicadas no primeiro ciclo e 84 no segundo. Essa diferença pode ser explicada pela melhoria da abordagem em geral, mas principalmente pelas alterações no processo seletivo e na integração do processo de desenvolvimento de software com o método CBL.

As avaliações finais sobre a abordagem IDEAS consistiram de um questionário com questões objetivas e subjetivas para os estudantes e uma entrevista semiestruturada, gravada e composta de questões objetivas e subjetivas para os professores.

Com relação ao questionário, dos 95 estudantes que finalizaram o curso do segundo ciclo, 89 responderam a avaliação. Dos respondentes, 57,3% eram estudantes novatos (menos de um ano de experiência prévia em programação) e 19,1% eram estudantes veteranos (mais de três anos de experiência prévia em

programação). Ou seja, são percentuais de novatos e veteranos bem semelhantes aos do primeiro ciclo.

A Quadro 5-7 apresenta um resumo comparativo entre os resultados das turmas dos dois ciclos. As linhas representam as perguntas mais significativas para esta comparação, e as colunas os percentuais de respostas positivas, sendo que a primeira coluna mostra o resultado do primeiro ciclo e a segunda coluna, o resultado do segundo ciclo.

Quadro 5-7: Quadro Comparativo dos Ciclos (Questionários dos Estudantes).

Perguntas	Ciclo 1	Ciclo 2
1) Quantas Vezes Usou o Método? (>3)	71,8%	89,9%
2) O Método Ajudou a Entender o Desafio?	74,4%	88,8%
3) O Método Ajudou a Elaborar Solução?	70,5%	82,0%
4) O Método Ajudou a Encontrar Solução Inovadora?	48,7%	49,4%
5) Grau de Motivação do Método?	70,5%	73,0%
6) Pretende Utilizar o Método em Outras Situações?	76,9%	79,8%
7) O Método Contribuiu para Melhoria de Suas Habilidades como Programador? (Novatos)	35,6%	37,3%
8) As Questões Guias, Atividades e Recursos Ajudaram no Desenvolvimento de suas Apps?	82,1%	70,8%

De forma geral a abordagem IDEAS apresentou vantagens com relação a abordagem utilizada no primeiro ciclo. Em 7 dos 8 pontos principais o resultado foi melhor, sendo que em 3 deles (perguntas 1, 2 e 3) as respostas positivas superaram o ano anterior significativamente (mais de 10%). Em apenas um dos itens (pergunta 8) apresentou resultado inferior ao ano anterior, devido principalmente à inclusão de uma etapa de levantamento de requisitos dentro do processo de desenvolvimento de software integrado ao método CBL.

No espaço para os comentários não foi citado no segundo ciclo nenhum dos problemas identificados na avaliação do segundo ciclo: integração com método de desenvolvimento de software, insegurança sobre a viabilidade do projeto, identificação do público alvo e ausência de auxílio para estudantes que pretendem abrir empresas. Entretanto, foram citadas algumas sugestões de melhoria: integração com algum processo de desenvolvimento de jogos digitais, pouco tempo para a realização do desafio final de forma completa e necessidade de aprimoramento de algumas arestas do método.

No que se refere à avaliação dos professores, para ciclo foram realizadas entrevistas com onze professores de três universidades. Os entrevistados foram

escolhidos com base em um critério único: a finalização de sua segunda turma de desenvolvimento iOS usando o CBL em 2015. A entrevista seguiu o mesmo roteiro do primeiro ciclo, exceto para os professores da PUCPR que responderam algumas perguntas adicionais sobre a aplicação da abordagem IDEAS. A diferença no número de universidades escolhidas com relação ao Ciclo 1 deveu-se ao fato de que as outras universidades ainda não haviam atingido o Ciclo 2 no momento da execução da avaliação.

Os professores envolvidos possuíam uma ampla experiência no desenvolvimento de software (média de 15,9 anos, desvio padrão 7,7 e mediana 13,5) e uma ótima experiência no ensino, tanto de programação com média de 9,7 anos (desvio padrão 5,01, mediana 10,5), quanto no ensino em geral com média de 8,6 anos (desvio padrão 6,0 e mediana 10,0).

As técnicas tradicionais de ensino (anteriores ao uso do CBL) mais utilizadas em sala de aula por estes professores foram aulas expositivas com 100% de respostas positivas; aulas práticas com 88,9% de respostas positivas; e tutorial guiado, com 66,7% de respostas positivas dos entrevistados.

O conhecimento profissional de métodos de desenvolvimento de software ficou assim dividido: 55,5% dos professores com alguma experiência em métodos tradicionais; 44,5% com experiência em métodos iterativos e incrementais; e 66,7% com experiência em métodos ágeis.

A partir das entrevistas com os professores foram geradas duas redes temáticas, uma referente aos professores da PUCPR e a seguinte, aos professores das outras universidades. O objetivo daquela foi fazer uma avaliação da abordagem IDEAS e desta, uma verificação sobre a utilização do CBL sem as alterações propostas nesta tese.

A análise da primeira rede temática, resultado da entrevista com os professores da PUCPR, pode ser visualizada na Figura 5-9. Ela gerou o tema principal Melhorias IDEAS e os subtemas: Aspectos Negativos IDEAS e Aspectos Positivos IDEAS.

Os aspectos positivos mais lembrados foram basicamente aqueles que sofreram algum tipo de alteração com relação ao primeiro ciclo. Essa associação direta pode ser facilmente explicada pelo fato de que esses professores entrevistados atuaram no curso em ambas as turmas.

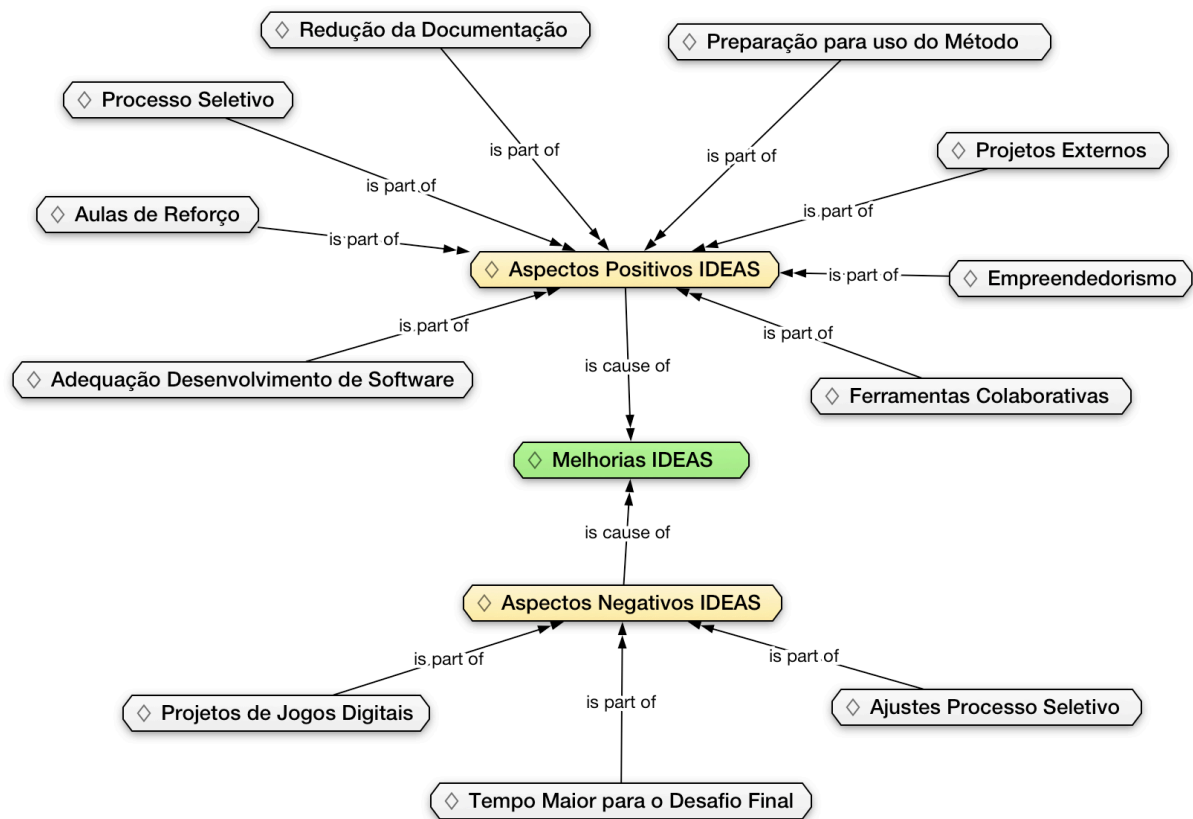


Figura 5-9: Rede Temática de Sugestões de Melhorias dos Docentes da PUCPR - Ciclo 2.

Os aspectos negativos que devem ser tratados em uma próxima edição do curso foram: tempo maior para o desafio final, integração com um método de desenvolvimento de jogos digitais e ajustes no processo seletivo.

A Figura 5-10 apresenta a rede temática resultante da análise das entrevistas do segundo ciclo de outras universidades. Percebe-se uma série de aspectos negativos e inconclusivos iguais, ou semelhantes, aos encontrados na análise das entrevistas do primeiro ciclo. Uma diferença relevante foi o subtema “alterações CBL” representando alterações feitas por esses professores na execução da segunda turma. Destacam-se o uso de um CBL Simplificado (da grande ideia até as questões guias) e a integração do método de desenvolvimento ágil Scrum na fase de questões guias, atividades e recursos, principalmente para facilitar o gerenciamento dos projetos.

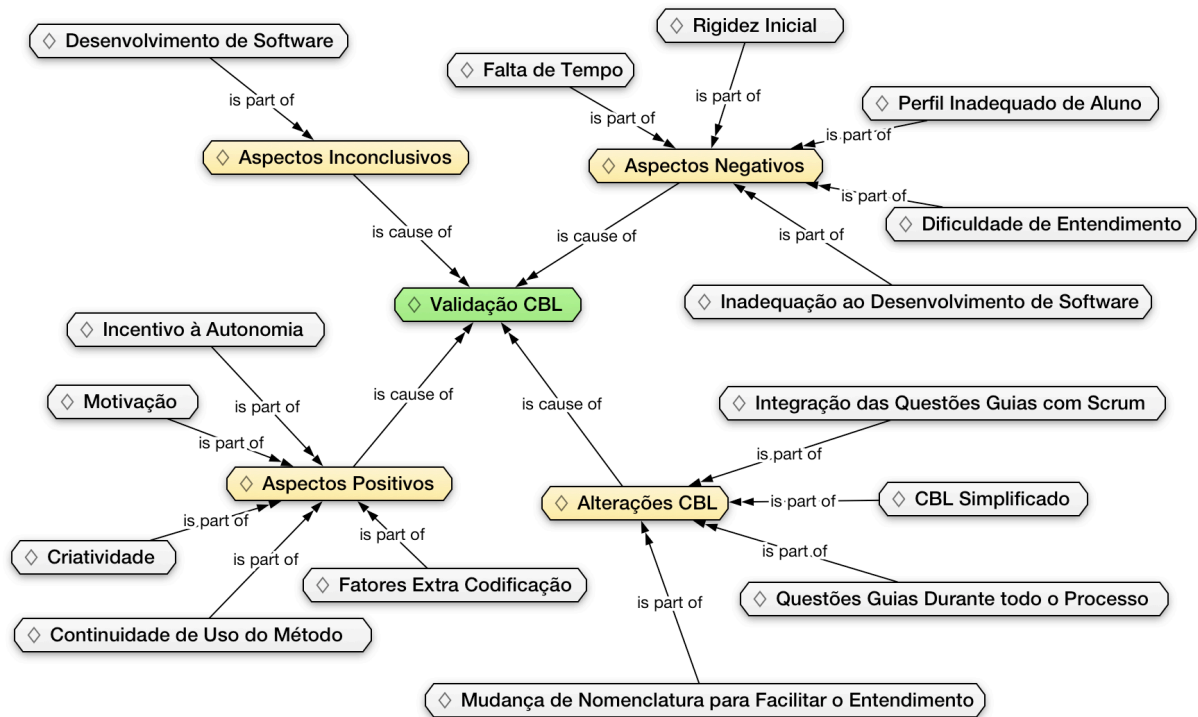


Figura 5-10: Rede Temática dos Docentes de Outras Universidades - Ciclo 2.

O mesmo curso com estrutura similar, mas com autonomia pedagógica para realizar modificações foi executado em dez instituições de ensino do Brasil. Todos os cursos são avaliados pela empresa parceira por meio das reflexões dos professores, entrevistas com os nove estudantes selecionados, reuniões semanais e avaliação das Apps após sua publicação.

A empresa parceira considerou um sucesso o formato de aceleração de *startups* proposto pela abordagem IDEAS. Devido a esta avaliação positiva vai replicar o mesmo modelo nas outras nove universidades.

O resultado da avaliação foi uma série de ajustes pontuais na abordagem (principalmente no processo seletivo) e do aumento da carga horária do curso de um ano (600 horas letivas) para dois anos (1200 horas letivas). Essa alteração de carga horária foi necessária devido ao aumento do conteúdo teórico para contemplar assuntos referentes a dois novos tipos de dispositivos (Smart Watch e Smart TV) e pela necessidade de se aumentar o tempo de desenvolvimento do desafio final com o objetivo de produzir um software mais robusto. Dessa forma o conteúdo teórico foi organizado todo no primeiro ano enquanto o segundo ano foi reservado inteiramente para o desafio final.

5.4 Considerações sobre o Capítulo

Neste capítulo foi apresentada a abordagem IDEAS bem como uma descrição de sua aplicação no segundo ciclo. Foram detalhadas as etapas, seu funcionamento e apresentados exemplos reais de como a abordagem foi utilizada.

As avaliações de estudantes e professores e os resultados diretos indicaram que a abordagem foi bem sucedida em vários aspectos: aumento do número de egressos, aumento do número de Apps publicadas, satisfação geral dos estudantes e professores, entre outros. Em praticamente todos os itens avaliados os resultados do Ciclo 2 foram superiores aos obtidos no Ciclo 1.

Também foram identificadas problemas e oportunidades de melhoria que serão discutidas no capítulo 6.

CAPÍTULO 6 - CONSIDERAÇÕES FINAIS

Above all, don't fear difficult moments.

The best comes from them.

Rita Levi-Montalcini

Este capítulo apresenta considerações sobre relevância do estudo, contribuições, limitações da pesquisa e trabalhos futuros.

6.1 Relevância do estudo

O desenvolvimento de software e o ensino de programação, para apoiar os ambientes de desenvolvimento de software, são temas explorados há muitos anos pela Engenharia de Software e outras áreas de pesquisa.

Na literatura são encontrados vários trabalhos que discutem diferentes tipos de abordagens para o ensino de programação, assim como diferentes metodologias que podem ser exploradas nos ambientes de desenvolvimento de software.

Porém, com o advento dos dispositivos móveis, o olhar para a programação teve que se modificar, se atualizar, e os projetos começaram a envolver cada dia mais inovação e a se tornarem mais desafiadores para todos os envolvidos em todo o ciclo de desenvolvimento. Isso ocorre porque, além da interface e usabilidade completamente diferentes, há um ciclo de vida menor do que softwares para computadores do tipo desktop/laptop ou web, assim como existe uma forma de distribuição diferente, que é a pré-instalação no dispositivo ou o *download* de uma loja de aplicativos.

Essa demanda tem como consequência direta, a necessidade crescente por uma formação sólida de desenvolvedores de aplicativos para dispositivos móveis (conhecidos por Apps), que ofereça tanto o ensino de programação básica quanto o ensino de desenvolvimento de aplicativos e produção de software para publicação.

Entretanto, cursos de programação apresentam grandes dificuldades, com alta evasão, os trabalhos propostos são eminentemente acadêmicos, não voltados

ao mercado. Os desafios normalmente encontrados em equipes de desenvolvimento de software permanecem escondidos em um ambiente acadêmico.

Assim, combinar o uso de metodologias de aprendizagem ativas, que colocam o foco no estudante, aumentando a motivação e consequente engajamento pareceu uma das maneiras de contribuir para os desafios em ambientes complexos.

Nesta tese optou-se pela adaptação do método CBL, por ter como foco o desenvolvimento de uma solução para um problema real, onde os conteúdos estudados surgem das necessidades do desafio proposto. A pesquisa explorou um ambiente complexo diferenciado, no qual o CBL foi aplicado especificamente em turmas especiais de desenvolvimento de aplicativos para dispositivos móveis, sendo realizada uma pesquisa-ação devido ao caráter colaborativo do trabalho e à possibilidade do pesquisador atuar de forma concorrente à sua execução.

Com os resultados obtidos e apresentados, foi demonstrado que o estudo contribui no assunto em questão, de forma particular, em mostrar uma abordagem distinta para apoiar ambientes complexos que envolvem ensino, inovação e produção de software para dispositivos móveis, gerando conhecimento prático dirigido à solução de um problema específico, envolvendo verdades e interesses locais.

6.2 Contribuições da pesquisa

A principal contribuição da pesquisa foi a criação da abordagem IDEAS que pode ser utilizada em situações similares que envolvam cursos de programação cujo requisito final seja a produção de um software inovador e, possivelmente, um novo negócio relacionado a este software.

A abordagem IDEAS foi elaborada a partir dos resultados de uma pesquisa-ação realizada em dois ciclos, inserida dentro de um projeto de inovação tecnológica em parceria com uma empresa da área de tecnologia. A abordagem possui quatro etapas, todas necessárias para se atingir o objetivo de formação dos desenvolvedores: seleção dos estudantes, execução do curso, desenvolvimento do software para publicação e criação de uma *startup*. Ou seja, a partir dos candidatos, são selecionados os estudantes mais aptos, esses estudantes são educados para o desenvolvimento de software, em seguida um software profissional é produzido a

partir de uma ideia dos próprios estudantes e, finalmente, uma *startup* é criada em torno do software desenvolvido.

Dentre os resultados obtidos, foi observado durante o primeiro ciclo uma série de problemas: dificuldade no entendimento do método CBL, falta de experiência para o desenvolvimento de aplicativos complexos, perfil inadequado de estudantes, dificuldade em identificar problemas relevantes que necessitem de Apps para sua resolução e ausência de auxílio para a criação de empresas. A partir destas indicações, foram tomadas várias ações com o intuito de sanar esses problemas, que resultaram na abordagem IDEAS. Durante o segundo ciclo a abordagem foi testada em sua totalidade o que permitiu o seu amadurecimento.

O processo seletivo foi melhorado com a inclusão de uma técnica de entrevista da área de psicologia, histórias de vida e com a execução de uma dinâmica de grupo, todas com o objetivo de entender melhor quais candidatos poderiam se adequar ao método CBL. A aplicação do método CBL foi melhorada com a sua apresentação de forma simplificada e gradual. A produção do desafio final foi feita com o método CBL integrado a um método de desenvolvimento de software, no qual as equipes foram imersas no dia a dia de um ambiente de desenvolvimento de software, encarando e resolvendo os desafios comuns a este tipo de situação. Para os estudantes interessados na criação de *startups* foram criados o núcleo de apoio e um processo que se iniciou simultaneamente ao da definição do desafio, possibilitando a continuação do trabalho após o término do curso.

Dentre os resultados positivos secundários da abordagem IDEAS, destacam-se os seguintes:

- A avaliação do método CBL em um ambiente de ensino de programação. Foram feitas avaliações nos dois ciclos, tanto com estudantes quanto com seus respectivos professores. As avaliações mostraram os pontos positivos e negativos do método, sugestões de melhorias e abordagens alternativas.
- Criação de um processo seletivo diferenciado para uso em ambientes similares que vai muito além da aplicação de uma prova de conhecimentos de programação.
- Seleção de *makers* para criação de *startups*, facilitando o processo de pré-aceleração. Em geral, uma das etapas de um processo de pré-

aceleração é justamente a identificação de quem são os *makers*. A abordagem IDEAS já forma essa categoria de profissional, permitindo o corte de uma etapa da pré-aceleração.

- Uma forma de imersão de estudantes novatos em processo de desenvolvimento de software com a consequente publicação de uma App no mercado.
- Aumento do engajamento dos alunos em seus respectivos cursos, com a consequente melhoria de seu desempenho e diminuição da evasão. Os egressos de ambos os ciclos continuaram ativos em seus respectivos cursos.
- Por questões de autonomia de cada universidade, a abordagem IDEAS foi aplicada e avaliada apenas na PUCPR, entretanto, alguns dos resultados aqui alcançados serão replicados nas outras universidades a partir de 2016.

6.3 Limitações da pesquisa

Uma das limitações desta pesquisa é que a parte inicial do primeiro ciclo (toda a parte anterior ao desafio final) foi executada em salas de aula normais, do tipo anfiteatro. Essa necessidade aconteceu devido ao atraso das obras de reforma do espaço definitivo. Como o método CBL exige um conjunto de locais para funcionar corretamente, a percepção dos estudantes com relação a ele com certeza foi prejudicada. A avaliação do segundo ciclo com relação ao método foi bem mais positiva, conforme apresentado no capítulo 5.

Outra limitação foi em relação ao processo seletivo da primeira turma, que foi realizado de forma bastante apressada, devido ao prazo curto que foi exigido pela empresa parceira para seu início. Não houve tempo suficiente para se compreender o método CBL com o objetivo de preparar um processo de seleção mais adequado. Devido a isso, decidiu-se pela aplicação de um modo tradicional de seleção com prova de conhecimentos de programação e raciocínio lógico e uma entrevista com questões comuns.

6.4 Trabalhos futuros

Por fim, são identificadas e apresentadas as perspectivas de trabalhos futuros, algumas já em planejamento para serem executadas:

- Avaliar a abordagem IDEAS nas outras universidades que estão realizando o curso de formação de desenvolvedores iOS. Sob a tutela de grupos diferentes de professores, outros problemas e respectivas melhorias poderão ser identificadas.
- Avaliar a abordagem IDEAS em condições normais de cursos de graduação, ou seja, sem bolsas de estudo, sem professores disponíveis em tempo integral, com os equipamentos usuais e sem quaisquer outras formas adicionais de motivação. Dessa forma, dois aspectos poderão ser avaliados: se a abordagem é efetiva por si só e se é possível aplicá-la em situações comuns decorrentes da realidade econômica das regiões, nas quais não se encontram facilmente recursos suficientes para alavancar um projeto da magnitude desta pesquisa.
- Fazer ajustes no sentido de melhorar as questões de inovação e de melhoria das habilidades de programação. Ambos serão realizados pelo próprio pesquisador nos próximos dois ciclos de pesquisa que se iniciarão, respectivamente em março de 2016 e no início de 2017. O aspecto de inovação será tratado no terceiro ciclo por meio de um estudo das Apps mais populares e bem sucedidas e por workshops de inovação, ministrados por um professor que seja profissional atuante nesta área. A melhoria das habilidades de programação será tratada no quarto ciclo, com planejamento previsto para o final do terceiro ciclo.

REFERÊNCIAS BIBLIOGRÁFICAS

- (ATKINSON, 2002) Atkinson, R. **The Life Story in Interview.** *Handbook of Interview Research: Context & Method*. P. 1a. Edição, Londres, Sage Publications. 121-140.
- (ATTRIDE-STIRLING, 2001) ATTRIDE-STIRLING, Jennifer. **Thematic networks: an analytic tool for qualitative research.** *Qualitative research*, v. 1, n. 3, p. 385-405, 2001.
- (BERBEL, 2012) BERBEL, N. **As metodologias ativas e a promoção da autonomia de estudantes.** In: *Semina: Ciências Sociais e Humanas* 32.1, 2012, p. 25-40.
- (BLIGH, 1998) Bligh, Donald A. **What's the Use of Lectures?**. Intellect books, 1998.
- (BONWELL; EISON, 1991) BONWELL, C.; EISON, J. **Active Learning: Creating Excitement in the Classroom.** *1991 ASHE-ERIC Higher Education Reports*, Washington, 1991.
- (BOUD; FELETTI, 1999) BOUD, D.; FELETTI, G. **Changing problem-based learning: introduction to second edition.** In: BOUD, D.; FELETTI G. (Eds.). *The challenge of problem-based learning*. Londres: Kogan Page, 1999. p. 1-14.
- (CANALETA et al., 2014) CANALETA, X.; VERNET, D.; VICENT, L.; MONTERO, J. **Master in teacher training: A real implementation of active learning.** In: *Computers in human behavior* 31, 2014, p. 651-658.
- (CATALINA et al., 2013) CATALINA, M.; HARGIS, J.; CAVANAUGH, C. **iPad Learning Ecosystem: Developing Challenge-Based Learning Using Design Thinking.** In: *Turkish Online Journal of Distance Education* 14.2, 2013.
- (CBL, 2011) The Challenge Based Learning White Paper. Disponível em <http://www.challengebasedlearning.org/public/admin/docs/CBL_Paper_October_2011.pdf> Acesso em 7 fev. 2016.
- (CHENG, 2016) CHENG, W. **Application Of Challenge-Based Learning In Nursing Education.** *Nurse Education Today*, 2016.
- (COUGHLAN; COUGHLAN, 2002) COUGHLAN, P.; COUGHLAN, D. **Action research for operations management.** *International Journal of Operations and Production Management*, 2002, v.22, n.2. Academic Research Library, 2002, p. 220- 240.
- (DREW; MACKIE, 2011) DREW, V. e MACKIE, L. **Extending the constructs of active learning: implications for teachers' pedagogy and practice.** In: *Curriculum Journal* 22.4, 2011, p. 451-467.
- (FASSBINDER et al., 2015) FASSBINDER, A; MARTINS, R; BOTELHO, T.; BARBOSA, E. **Applying flipped classroom and problem-based learning in a CS1**

course. In: Frontiers in Education Conference, 2015, El Paso, NM .**Anais...**, 2015, p. 803-805.

(FLORA; CHANDE, 2013) FLORA, Harleen K.; CHANDE, Swati V. **A review and analysis on mobile application development processes using agile methodologies.** In: **International Journal of Research in Computer Science**, v. 3, n. 4, p. 9, 2013.

(FREEMAN, 2014) FREEMAN, S; EDDY, S; MCDONOUGH, M.; SMITH, M.; OKOROAFOR, N; HORDT, H; WENDEROTH, M. **Active learning increases student performance in science, engineering, and mathematics.** In: Proceedings of the National Academy of Sciences 111.23, 2014, p. 8410-8415.

(GABRIEL, 2014) GABRIEL, S. **A Modified Challenge-Based Learning Approach in a Capstone Course to Improve Student Satisfaction and Engagement.** In: Journal of microbiology & biology education 15.2, 2014, p. 316.

(GELEYNSE, 2012) Geleynse, John. **Ingredients of Great Apps.** WWDC 2012. Disponível em < <https://www.youtube.com/watch?v=9-7NRU5jXZ0>> Acesso em 4 fev. 2016.

(GIL, 2002) GIL, A. C. **Como elaborar projetos de pesquisa.** 4a. ed. São Paulo: Atlas, 2002.

(GILBOY et al, 2015) Gilboy, M.; HEINERICHS, S.; e PAZZAGLIA, G. **Enhancing student engagement using the flipped classroom.** In: Journal of nutrition education and behavior 47.1, 2015, p. 109-114.

(HAWTHORNE; PERRY, 2005) HAWTHORNE, Matthew J.; PERRY, Dewayne E. **Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities.** In: **Software Engineering Education in the Modern Age.** Springer Berlin Heidelberg, 2005. p. 166-185.

(HUIJGEVOORT, 2012) HUIJGEVOORT, T. **The 'Business Accelerator': Just a Different Name for a Business Incubator.** Bachelor Thesis, Utrecht School of Economics, Utrecht, The Netherlands, 2012.

(JOHNSON; ADAMS, 2011) Johnson, L. e Adams, S. **Challenge Based Learning: The Report from the Implementation Project.** Austin, Texas: The New Media Consortium, 2011, 42p. Disponível em: <challengebasedlearning.org/public/admin/docs/NMC_CBLi_Report_Oct_2011> Acesso em 26 dez. 2015.

(JOHNSON et al., 2009) JOHNSON, L. F.; SMITH, R. S.; SMYTHE, J.; VARON, R. K.. **Challenge-based learning: An approach for our time.** Technical report, The New Media Consortium, Austin, TX, 2009.

(KINNUNEN; MALMI , 2005) KINNUNEN, Päivi; MALMI, Lauri. **Problems in Problem-Based Learning-Experiences, Analysis and Lessons Learned on an Introductory Programming Course.** In: Informatics in Education, v. 4, n. 2, p. 193-214, 2005.

(LEWIS et al., 2013) LEWIS, G. A.; NAGAPPAN, N.; GRAY, J.; ROSENBLUM, D.; MUCCINI, H.; SHIHAB, E. **Report of the 2013 ICSE: 1st international workshop on engineering mobile-enabled systems**. 2013.

(O'MAHONY et al., 2012) O'MAHONY, T. K.; VYE, N. J.; BRANSFORD, J. D.; SANDERS, E. A.; STEVENS, R.; STEPHENS, R. D.; RICHEY, M. C.; LIN, K. Y.; SOLEIMAN, M. **A comparison of lecture-based and challenge-based learning in a workplace setting: Course designs, patterns of interactivity, and learning outcomes**. *Journal of the Learning Sciences*, 21, 2012, p. 182–206.

(MICHAEL, 2006) MICHAEL, J. **Where's the evidence that active learning works?** In: *Advances in physiology education* 30.4, 2006, p. 159-167.

(MOALOSI et al., 2012) MOALOSI, R., MOLOKWANE, S., MOTHIBEDI, G. (2012). **Using a Design-orientated Project to Attain Graduate Attributes**, In: *Design and Technology Education*, v. 17 . 1, p. 30-43.

(NAISMITH et al., 2013) NAISMITH, L.; LONSDALE, P.; VAVOULA, G.; SHARPLES, M. **Literature review in mobile technologies and learning**. Technical report, University of Birmingham, 2013.

(NIEMI, 2002) NIEMI, H. **Active learning—a cultural change needed in teacher education and schools**. In: *Teaching and teacher education* 18.7, 2002, p. 763-780.

(OSTERWALDER; PIGNEUR, 2010) OSTERWALDER, A. e PIGNEUR, Y. **Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers**. 1ª edição. John Wiley and Sons, 2010. 288p.

(POLAINE et al; 2013) POLAINE, A.; LOVLIE, L; REASON, B. **Service Design: from Insight to Implementation**. 1ª edição. Rosenfeld Media LLC, 2013. 202 p.

(PRINCE, 2004) Prince, M. **Does active learning work? A review of the research**. In: *Journal of engineering education* 93.3, 2004, p. 223-231.

(ROBINSON, 2013) ROBINSON, J. **Project-based learning: improving student engagement and performance in the laboratory**. In: *Analytical and bioanalytical chemistry*, 2013, p. 1-7.

(ROSIENE; ROSIENE, 2015) ROSIENE, Carolyn, P; ROSIENE, Joel A. **Flipping a Programming Course: the Good, the Bad, and the Ugly**. In: *Frontiers in Education Conference, 2015, El Paso, NM .Anais...*, 2015, p. 803-805.

(ROTGANS; SCHMIDT, 2011) ROTGANS, J., e SCHMIDT, H. **The role of teachers in facilitating situational interest in an active-learning classroom**. In: *Teaching and Teacher Education* 27.1, 2011, p. 37-42.

(SANTOS et al, 2015) SANTOS, A; SALES, A; FERNANDES, P; NICHOLS, M. **Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development**. In: *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 2015.

(SCHARFF; VERMA, 2010) SCHARFF, C.; VERMA, R. **Scrum to support mobile application development projects in a just-in-time learning context.** In: Proceedings of the 32nd ICSE, 2010, p. 25–31.

(SCHRAW et al; 2001) SCHRAW, G., FLOWERDAY, T., & LEHMAN, S. **Increasing situational interest in the classroom.** In: Educational Psychology Review, 2001, p. 211-224.

(STATISTA, 2016) **Number of smartphones sold to end users worldwide from 2007 to 2014 (in million units).** Disponível em <<http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>> Acesso em 18 fev. 2016.

(TUCKER, 2012) TUCKER, Bill. **The flipped classroom.** In: **Education Next**, v. 12, n. 1, 2012.

GLOSSÁRIO

App	Software desenvolvido para dispositivos móveis ou portáteis.
Ciclo	Cada Ciclo da Pesquisa-Ação. Corresponde ao período que vai da coleta de dados até o término do curso.
Classe	Conjunto de estudantes pertencentes a um determinado período. Classe da manhã, Classe de sábado, etc.
Turma	Conjunto total de estudantes que foram aprovados para o curso em um determinado ano. Turma de 2013-2014 e Turma de 2015.

APÊNDICES

Apêndice A - QUESTIONÁRIO PARA ESTUDANTES

Com o objetivo de realizar uma avaliação sobre os resultados finais do primeiro ciclo, foi criado o questionário abaixo no Google Forms. A aplicação do questionário ocorreu em um laboratório em três turnos, sendo que as respostas foram individuais. Responderam ao questionário 78 estudantes da PUCPR de um total de 84 estudantes formandos.

No segundo ciclo um questionário semelhante foi aplicado para 89 estudantes da PUCPR de um total de 95 estudantes formados. As novas questões encontram-se ao final do questionário abaixo:

1) Quantos anos de estudo de programação você tinha antes de iniciar o curso?

2) Quantos anos de experiência em programação para dispositivos móveis você tinha antes de iniciar o curso?

Cite as plataformas.

3) Quantos anos de experiência em desenvolvimento de software como programador você tinha antes de iniciar o curso?

4) Quantos anos de experiência em desenvolvimento de software como analista de sistemas / analista de negócios / arquiteto de software você tinha antes de iniciar o curso?

5) Quantos anos de experiência em desenvolvimento de software como gestor de projetos você tinha antes de iniciar o curso?

6) Já cursou disciplina referente a processos de desenvolvimento de software ou gestão de projetos?

Sim ou Não

7) Quantas vezes você usou o método CBL?

8) O método CBL ajudou a entender o problema.

Discordo totalmente, Discordo parcialmente, Não discordo nem concordo, Concordo parcialmente, Concordo totalmente.

9) O método CBL ajudou a elaborar uma solução que resolveu o problema.

Discordo totalmente, Discordo parcialmente, Não discordo nem concordo, Concordo parcialmente, Concordo totalmente.

10) O método CBL ajudou a elaborar uma solução inovadora.

Discordo totalmente, Discordo parcialmente, Não discordo nem concordo, Concordo parcialmente, Concordo totalmente.

11) Em que grau o método CBL contribuiu para a sua motivação em resolver o problema?

Muito pouco, Pouco, De forma razoável, Muito, Totalmente.

12) Quais tecnologias o método CBL o incentivou a utilizar como ao aprendizado e solução do problema?

Criação de Vídeos ,Tablets, Smartphones, Redes Sociais ,Outros.

13) As questões-guia ajudaram no desenvolvimento de suas apps?

Discordo totalmente, Discordo parcialmente, Não discordo nem concordo, Concordo parcialmente, Concordo totalmente.

14) Após o término do projeto atual você pretende usar o método CBL em outras situações?

Nunca, Provavelmente não, Talvez, Provavelmente sim, Certamente.

Em caso afirmativo, onde?

15) O método CBL contribuiu para a melhoria de suas habilidades como programador?

Discordo totalmente, Discordo parcialmente, Não discordo nem concordo, Concordo parcialmente, Concordo totalmente.

16) Em que grau o método CBL contribuiu para o entendimento do problema.

Nada, Pouco, De forma razoável, Muito, Totalmente.

17) Em que grau o método CBL contribuiu para a estruturação da solução.

Nada, Pouco, De forma razoável, Muito, Totalmente.

18) Com que frequência você utilizou o método CBL durante as aulas?

Nenhuma vez, Poucas vezes, Algumas vezes, Muitas vezes, Todas as vezes.

19) Em que grau você acha que o método CBL contribuiria para o ensino de outros conteúdos que não sejam de programação?

Nada, Pouco, De forma razoável, Muito, Totalmente.

20) Algum comentário a acrescentar?

Questões Adicionais para os estudantes da Segunda Turma:

Defina o Challenge Based Learning (CBL).

Dê um exemplo de como você usou o CBL.

Você considera o CBL um bom método para o desenvolvimento de aplicativos?

Apêndice B - QUESTIONÁRIO PARA PROFESSORES DA 1ª TURMA

O questionário abaixo foi aplicado para vinte e três professores de oito universidades que haviam aplicado uma vez ou estavam aplicando pela primeira vez o método CBL. Foram entrevistas semi-estruturadas individuais, gravadas em áudio, transcritas e analisadas usando redes temáticas.

1) Quanto tempo de experiência você tem no desenvolvimento de software?

2) Quanto tempo de experiência você tem no ensino?

3) Quanto tempo de experiência você tem no ensino de programação?

4) Com que frequência você utilizou os métodos de ensino de programação abaixo? (Nunca, Poucas vezes, Algumas vezes, Muitas vezes, Sempre)

Aula Expositiva

Aula Prática (Exercício Proposto)

Tutorial Guiado (Professor programa exemplos, estudantes acompanham)

Codificação DOJO

5) Qual é o seu grau de experiência com os seguintes métodos de Desenvolvimento de Software? (Não tenho experiência, Muito pouco, Pouco, Razoável, Muito)

Tradicional

Processo Iterativo e Incremental (RUP por exemplo)

Métodos Ágeis (Scrum, XP, etc.)

6) Quantas vezes você aplicou o método CBL?

7) O método CBL motivou mais os estudantes do que outros métodos que você conhece.

Discordo totalmente, Discordo parcialmente, Não concordo nem discordo, Concordo parcialmente, Concordo totalmente.

8) Após o término do projeto você pretende utilizar o método CBL em outras turmas de programação?

Certamente, Provavelmente sim, Talvez, Provavelmente não, Nunca

9) Após o término do projeto você pretende utilizar o método CBL para ensinar outros conteúdos que não sejam de programação?

Certamente, Provavelmente sim, Talvez, Provavelmente não, Nunca

10) Você já utilizou o método CBL em outras situações fora do projeto?

Sim, Não

11) Caso tenha respondido positivamente à questão anterior, liste as disciplinas em que usou/pretende usar o método CBL.

12) Caso tenha respondido negativamente, liste os motivos.

13) A qualidade do software desenvolvido pelos estudantes utilizando o método CBL é melhor do que a qualidade do software desenvolvido utilizando outros métodos que você tem experiência.

Discordo totalmente, Discordo parcialmente, Não concordo nem discordo, Concordo parcialmente, Concordo totalmente

14) O método CBL incentiva os estudantes a utilizarem as seguintes tecnologias modernas?

Vídeos, Tablets, Smartphones, Redes Sociais, Outros

15) Você promoveu alguma alteração no método CBL durante sua aplicação?

Sim, Não

Alterações no método CBL

Abaixo estão listadas as etapas do método. Caso não tenha feita alterações em uma determinada etapa, deixe a resposta em branco.

16) Que alterações você promoveu na etapa "Big Idea"?

17) Que alterações você promoveu na etapa "Essential Question"?

18) Que alterações você promoveu na etapa "The Challenge"?

19) Que alterações você promoveu na etapa "Guiding Questions, Activities and Resources"?

20) Que alterações você promoveu na etapa "Solution"?

21) Que alterações você promoveu na etapa "Implementation"?

22) Que alterações você promoveu na etapa "Evaluation"?

23) Descreva outras alterações que você fez no método CBL.

24) Qual é a sua opinião sobre o método CBL?

Apêndice C - ENTREVISTA DOS CANDIDATOS DA 2ª. TURMA

O questionário abaixo foi aplicado por um professor para cada um dos candidatos selecionados na prova teórica. As entrevistas duraram de 20 a 30 minutos e os dados foram coletados em sistema próprio desenvolvido para este fim. As respostas da técnica de histórias de vida foram escritas no quadro branco do local da entrevista, fotografadas pelo professor e também armazenadas no sistema.

Perguntas de Aceite Obrigatório

Informações sobre a disponibilidade:

- a) Serão 3 horas por dia em um só turno
- b) Não serão permitidos turnos quebrados

Compromisso assumido - Decisão IMEDIATA

Disponível: Sim Não

Informações sobre a concordância do processo:

- a) Será um trabalho por semana
- b) Mini-challenges
- c) Filosofia do 10 vale 7
- d) Avaliações mensais, com possibilidade de recuperação
- e) Obrigatoriedade de presença - duas faltas semanais passível de exclusão
- f) Obrigatoriedade de ter um app publicado ao final do curso
- g) Apoio das equipes de arte e música
- h) Para quem tirou 4 ou menos acertos em programação:
 - a. Vai ter que correr atrás da parte de programação
 - b. Poderá contar com nosso suporte (não haverá nivelamento)

Aceite: Sim Não

Análise do Histórico Escolar

Analisar as seguintes informações:

- a) Pegou dependências? Quantas?

b) Pegou finais? Quantas?

Está OK? Sim Não

Observações:

Linha da Vida

Informações: Trace seus pontos altos e baixos

Lembre-se de pontos que representam pontos altos e baixos em sua vida e trace-os em um cronograma que remonta a tão longe quanto você possa lembrar.

O eixo vertical representa prazer e/ou entusiasmo; o eixo horizontal representa o tempo.

"Pontos Altos" e "Pontos Baixos" são:

- a) Eventos específicos, importantes em sua vida: bons ou ruins, pessoais ou profissionais - relacionados ao trabalho, vida social, amor, hobbies, meio acadêmico, buscas espirituais, ou outras áreas
- b) Marcos ou fatos de que se lembra claramente e estão associados a sentimentos fortes
- c) Principais mudanças de carreira, tanto positivas quanto negativas. Inicie na extremidade esquerda da linha com o mais distante ponto alto ou baixo que lembrar e, em seguida, trabalhe até o presente. Quando tiver traçado cerca de 15 eventos desenhe uma linha ligando todos os pontos.

Imagem:

Anotações que considerar importante sobre os pontos altos e baixos.

Comentários gerais sobre o candidato.

Apêndice D - FORMULÁRIO DE INSCRIÇÃO PARA A HOTMILK

Para participar do processo de seleção para a HotMilk, os estudantes interessados deveriam preencher o formulário abaixo logo após o término do curso em dezembro de 2015. Havia dois pré-requisitos: ter demonstrado interesse em abrir uma *startup* no início da etapa de produção de software e ter conseguido aceitação da loja de aplicativos para a App desenvolvida.

1. Nome da Startup
2. Descreva a dor que a sua Startup está resolvendo.
3. Quem são os clientes?
4. Como chegará até seus clientes?
5. Que tipo de serviço prestará?
6. De quem você depende financeiramente?
7. Escreva o nome e função de cada sócio. Indique também se é egresso do curso e qual é o percentual da empresa que cada um possui.

Apêndice E - FORMULÁRIO DE CAPTAÇÃO DE PROJETOS

O formulário abaixo foi disponibilizado para professores da PUCPR apresentarem projetos para o desenvolvimento dentro do curso. Os projetos foram avaliados pela equipe de professores, seguindo os critérios: viabilidade técnica, viabilidade de prazo, viabilidade de custos e originalidade.

Dados Pessoais

Nome Completo*

Email*

Telefone*

Curso de Lotação na PUCPR*

Dados do Projeto

Nome do Projeto*

Área do Projeto*

Indique a área geral de sua aplicação. Ex.: Saúde, Comunidades, Engenharia, Segurança, Educação, etc.

Descrição*

Descreva em um parágrafo curto o que é a app.

Descreva seu app com mais detalhes. Preferencialmente sobre a perspectiva de uso.*

(o que o usuário pode fazer com o app?)

Como imagina que será a vida das pessoas depois da existência da sua app?*

Você acredita que seu projeto é inovador ?*

Sim ou Não

Porque ?

Justifique a resposta à questão anterior.

Público Alvo : Faixa etária*

Indique a faixa etária aproximada dos usuários do app.

0 - 5

5 - 10

10 - 15

15 - 20

20 - 30

30 - 40

40 - 50

50 - 60

acima de 60

Publico Alvo : Generalista X Especialista*

Quem é o público predominante a usar o seu app? Um especialista da área ou um leigo?

Generalista

Especialista

Retorno Esperado*

financeiro

portfólio

validação e/ou aplicação pesquisa

ajudar pessoas

explorar tecnologia

Análise de Forças, Fraquezas, Oportunidades e Ameaças (SWOT)

Descreva sucintamente o que você percebe como pontos fortes e fracos do projeto, oportunidades e ameaças.

PONTOS FORTES*

(ex. "O aplicativo pode salvar vidas", "Nova forma de visualizar os dados", "O estilo do jogo é um dos mais vendidos no mundo", ...)

PONTOS FRACOS*

(ex. "Existem alguns concorrentes mais conhecidos", "Base de dados cara para ser adquirida", "Pesquisa ainda inicial",...)

OPORTUNIDADES*

(ex. "O aplicativo vai ser lançado próximo à Olimpíada", "Mudança da legislação exige investimento na área", "Pesquisas em ubiqüidade estão apontando a convergência de mídias", ...)

AMEAÇAS*

(ex. "O aplicativo depende do input intenso da comunidade", "a tecnologia necessária para ele funcionar ainda não foi testada", ...)

Recursos necessários**Humanos**

(ex. sonoplastia, especialista em usabilidade, desenhistas, etc...)

Tecnológicos

(ex. Leitores de RFid, aparelho de captura de movimentos, etc...)

Referências*

Numere e liste as referências, como textos, artigos e softwares similares, que possam ser úteis para explicar seu app.

Justificativas das referências

Numere de acordo com a resposta acima e liste por que elas são relevantes para seu projeto

Apêndice F - LISTA DE PROJETOS FINALIZADOS NA 1ª TURMA

Segue abaixo a lista com todos os 40 projetos do desafio principal finalizados da primeira turma:

- +Cook
- Acesso Fácil
- Angle
- Aprendendo
- Aut's Adventures
- Baladero
- BodyFitness
- Compact
- Crowdesty
- Readditive
- Dream World
- Dysgame
- Funfy
- GO
- Hepeo
- Hotel Já
- iAdd
- Landscape
- LocDoc
- Mail Mate
- Mapper
- Mimics
- Minha Família
- Mini Mental
- Multitela
- O que é que tem?
- Omega Pad
- Online Parenting
- Prompe
- SpaceCore
- Take Colors
- Take Notes
- Together
- Vicci
- War Bot
- Where To
- World Cup Striker
- World Traveler
- Wyz
- YouUnderstand

Apêndice G - LISTA DE PROJETOS FINALIZADOS NA 2ª TURMA

Segue abaixo a lista com todos os 89 projetos finalizados da segunda turma:

- Alarmers
- Allan
- Allthors
- Analyser
- Angels & Devils
- Anyone's Life
- Ashhh
- Avatar
- Aventuras Alfa
- Azienda
- Beautify
- Bento App
- Bever
- Career planning
- Cativa
- ChamaFrete
- Changier
- Click
- Cointer
- Colobixo
- Compre +
- Conte Comigo
- CopyNow
- Crushes
- Darkness
- Dash
- DiseaseFundoscopy
- Dog Keeper
- E-SportsApp
- Element Rush
- Escrevendo mais
- Eternal
- Eternal - Rede social para casamentos
- FaceSports
- Festas

- Follow-UP
- Food Fighters
- Futebot
- GEMS
- Gestão Geradores
- HeatKit
- Imagens para WhatsApp
- Julie & Jetbear
- Kubo Education

- LogLocation
- Masters of Maori
- Matchly
- Mental Health
- Meu Ambiente
- Movie Quiz
- Movie2Comic
- My Math
- My Wish List
- nGym
- Nível de Anjo
- O som dos bichos 2
- Os Ludis
- PhotoCopy
- PhotoCopy 2.0
- Porta Documentos
- Projeto Futebot
- Projeto Holmies
- PUC GUIDE
- PUCPR Minhas Notas
- Rabbet
- Ready To Order (R2O)
- RockMath
- ROI Finder
- Roundy
- Routine
- Selfie Checker
- Sharpblow
- SizeGuide
- SkyShop
- Sociedade RPG
- Space Cowting
- Spider Beats
- Survival Squad
- Taggr - Smart Tag
- Taro Kai
- Tembo
- Timeer
- Toy Stock
- Training Analyser
- Travel Lines
- Underground
- Vamos aprender?
- Wooly
- XPrize app1

Apêndice H - PROVA DA 1ª TURMA

A prova abaixo é composta de 3 questões de raciocínio lógico e 7 questões de programação básica (variáveis, condições, repetições e vetores). Todas as questões são abertas para possibilitar um olhar mais cuidadoso na correção. Foi aplicada em junho de 2013 para 300 estudantes.

Processo Seletivo: Projeto iOS PUCPR

IMPORTANTE: REGRAS PARA REALIZAR A PROVA

- Desligue o celular
- A prova deve ser feita a caneta
- É proibido consultar qualquer tipo de material
- A interpretação das questões faz parte da prova. Não há nenhum enunciado incorreto.
- Cola de qualquer tipo acarretará a exclusão do autor de todos os processos seletivos do Projeto iOS!

Raciocínio Lógico

1. Duas crianças, um menino e uma menina, estão sentados em um banco. A criança de cabelo vermelho afirma: “Eu sou menino”, enquanto a criança de cabelo loiro diz: “Eu sou menina“. Sabendo que pelo menos uma delas está mentindo, defina o sexo de cada uma delas.
2. Três homens, Luís, Carlos e Paulo, são casados com Lúcia, Patrícia e Maria, mas não sabemos quem é casado com quem. Eles trabalham com engenharia, advocacia e medicina, mas também não sabemos quem faz o quê. Com base nas dicas abaixo, tente descobrir o nome da esposa e a profissão de cada um.
 - a) O médico é casado com Maria.
 - b) Paulo é advogado.
 - c) Patrícia não é casada com Paulo
 - d) Carlos não é médico.
3. Você está no escuro. No chão há seis sapatos de três cores e uma pilha de vinte e quatro meias nas cores preto e marrom. Quantas meias e quantos sapatos você deve pegar e levar até a luz para ter certeza de que você possui um par de meias e um par de sapatos de mesma cor?

Linguagem de Programação

Atenção:

- a) nas questões abaixo, as tabelas de entrada e saída são exemplos de execuções corretas do programa, mas não são as únicas entradas e saídas possíveis.
- b) **todos** os programas **podem** ser resolvidos sem o uso de estruturas de dados avançadas.
- c) **sugestão**: resolva as questões em português para não ter que se preocupar com sintaxe.
- d) **Não serão aceitas soluções escritas em texto discursivo.**
- e) Soluções do tipo força bruta receberão uma pontuação menor. Soluções elegantes e criativas receberão mais pontos.

4. Dona Neuza, uma senhora muito simpática de 90 anos, já perdeu a conta de quantos netos possui. Outro dia, perguntada sobre qual a idade do mais novo e do mais velho a pobre Dona Neuza não soube responder.

Ajude Dona Neuza a responder esta questão elaborando um programa que receba como entrada a quantidade de netos e a idade de cada um. Como saída, o programa deve dizer qual a idade do neto mais novo e qual a idade do neto mais velho.

Exemplo:

Entrada	Saída
7	
1 10 99 5 19 23 17	Menor 1 Maior 99

5. No famoso e polêmico “jogo do bicho” existem 25 animais a serem apostados. Como os números sorteados são em formato de milhar, são divididos 4 números sequenciais para cada animal (1 a 4 – animal numero 1, 5 a 8 – animal numero 2, etc) usando os dois últimos algarismos.

Elabore um programa que, dada uma determinada milhar (0 – 9999), retorne o número do bicho correspondente (1 a 25) de acordo com os dados acima.

6. Dois países A e B possuem, respectivamente, H_1 e H_2 número de habitantes, sendo T_1 e T_2 as suas taxas de crescimento anual em porcentagem. Levando em consideração que $H_1 < H_2$ e $T_1 > T_2$, obrigatoriamente, elabore um programa que receba como entrada na primeira linha os valores de população H_1 e H_2 e na segunda linha os valores das taxas T_1 e T_2 , ambos separados por um espaço. A saída deverá mostrar em quantos anos o país A passará o país B em termos populacionais.

Exemplo:

Entrada	Saída
60000 70000	16
3 2	

7. Bob é um estudante do ensino médio que gosta de matemática. Na última aula aprendeu o que são *Progressões Aritméticas (PAs)* e ficou fascinado por elas. Pelo que Bob entendeu, PA são seqüências de números nas quais a diferença entre dois elementos consecutivos é sempre igual a uma constante r , chamada de razão da PA.

Um exemplo de PA de razão 2 é -1, 1, 3, 5. Além disso, toda seqüência com um ou dois elementos é sempre uma PA. Por outro lado, 5, 6, 8, 9, 10 não é uma PA porque a diferença entre elementos consecutivos não é constante: a diferença entre os dois primeiros elementos é $6 - 5 = 1$, enquanto a diferença entre o terceiro e o segundo elementos é $8 - 6 = 2$.

Bob percebeu que qualquer seqüência, mesmo que não seja uma PA, pode ser quebrada em seqüências menores que são PAs. Por exemplo, vimos que a seqüência 5, 6, 8, 9, 10 não é uma PA, mas podemos quebrar ela entre o 6 e o 8 para obtermos as seqüências 5, 6 e 8, 9, 10, que são PAs. Note que não tem como quebrar a seqüência em menos partes se quisermos ter apenas PAs no fim do processo.

Bob não está conseguindo descobrir como quebrar seqüências muito grandes de um jeito eficiente; por isso, pediu que você escrevesse um programa para, dada uma seqüência qualquer, imprimir o número mínimo de partes em que precisamos quebrar a seqüência para termos apenas PAs ao término do processo. Caso a seqüência original já seja uma PA, podemos terminar o processo com uma única parte, e a resposta é 1.

Entrada

A entrada é composta por um inteiro N indicando o a quantidade de elementos da sequência, seguido dos números inteiros a_i , os elementos da sequência.

Saída

A saída contém o número mínimo de partes em que Bob precisa quebrar a sequência original para que ele termine apenas com PAs.

Restrições

- $1 \leq N \leq 10^5$
- $-10^5 \leq a_i \leq 10^5$

Exemplo:

Entrada	Saída
5 1 2 3 4 5	1
7 -2 0 2 3 3 4 6	3

8. Há muitos jogos divertidos que usam pequenas varetas coloridas. A variante usada neste problema envolve a construção de retângulos. O jogo consiste em, dado um conjunto de varetas de comprimentos variados, desenhar retângulos no chão utilizando as varetas como lados dos retângulos, sendo que cada uma pode ser utilizada em apenas um retângulo, e cada lado de um retângulo é formado por uma única vareta. Nesse jogo, duas crianças recebem dois conjuntos iguais de varetas. Ganha o jogo a criança que desenhar o maior número de retângulos.

Dado um conjunto de varetas de comprimentos inteiros, você deve escrever um programa para determinar o maior número de retângulos que é possível desenhar.

Entrada

A entrada contém vários casos de teste. A primeira linha contém um inteiro N que indica o número de diferentes comprimentos de varetas ($1 \leq N \leq 1.000$) no conjunto. Cada uma das N linhas seguintes contém dois números inteiros C_i e V_i , representando, respectivamente, um comprimento ($1 \leq C_i \leq 10.000$) e o número de varetas com esse comprimento ($1 \leq V_i \leq 1.000$). Cada comprimento de vareta aparece no máximo uma vez em um conjunto de testes (os valores C_i são distintos). O final da entrada é indicado por $N = 0$.

Saída

Para cada caso de teste seu programa deve produzir uma única linha na saída, contendo um número inteiro indicando o número máximo de retângulos que podem ser formados com o conjunto de varetas dado.

Exemplo:

Entrada	Saída
1	1
10 7	3
4	2
50 2	
40 2	
30 4	
60 4	
5	
15 3	
6 3	
12 3	
70 5	
71 1	
0	

9. Uma Copa do Mundo de futebol está sendo realizada com times de todo o mundo. A classificação é baseada no número de pontos ganhos pelos times, e a distribuição de pontos é feita da forma usual: quando um time ganha um jogo, ele recebe 3 pontos; se o jogo termina empatado, ambos os times recebem 1 ponto; e o perdedor não recebe nenhum ponto.

Dada a classificação atual dos times, o número de times participantes e o número de jogos já realizados na Copa do Mundo, sua tarefa é determinar quantos jogos terminaram empatados até o momento.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois inteiros T e N indicando, respectivamente, o número de times participantes ($2 \leq T \leq 200$) e o número de partidas jogadas ($0 \leq N \leq 10^4$). Cada uma das T linhas seguintes contém o nome de um time (uma cadeia de máximo 10 letras e dígitos) seguido de um espaço em branco do número de pontos que o time obteve até o momento. O final da entrada é indicado por uma linha que contém apenas o número zero.

Saída

Para cada um dos casos de teste de entrada seu programa deve imprimir uma única linha contendo um número inteiro, representando a quantidade de jogos que terminaram empatados até o momento.

Exemplo:

Entrada	Saída
3 3	0
Espanha 3	2
Uruguai 3	
Nigéria 3	
3 3	
Brasil 5	
Japão 1	
México 1	
0 0	

10. A Rede Cubo é a maior rede de televisão de Banânia, um pequeno país situado a leste da América do Sul (ou sul da América do Leste). O esporte mais popular em Banânia, obviamente, é o futebol.

A Rede Cubo recebe muitos pedidos para repassar lances polêmicos de impedimento. Um jogador atacante está impedido se ele está mais próximo da linha do gol do oponente do que o penúltimo adversário. Um jogador não está impedido se:

- ele está na mesma linha que o penúltimo adversário ou
- ele está na mesma linha que os dois últimos adversários.

Através do uso de computação gráfica, a Rede Cubo consegue tirar uma foto do campo e determinar as distâncias dos jogadores até a linha do gol do time defensor, mas eles ainda precisam de um programa que, dadas essas distâncias, decida se um jogador está impedido.

Entrada

Os dados de entrada contêm vários casos de teste. A primeira linha de cada caso de teste contém dois inteiros A e D separados por um espaço indicando, respectivamente, o número de jogadores atacantes e defensores envolvidos na jogada ($2 \leq A, D \leq 11$). A próxima linha contém A inteiros B_i separados por um espaço, indicando as distâncias dos jogadores atacantes até a linha do gol ($1 \leq B_i \leq 10^4$). A próxima linha contém D inteiros C_i separados por um espaço, indicando as distâncias dos defensores até a linha do gol ($1 \leq C_i \leq 10^4$). O final da entrada é dado por $A = D = 0$.

Saída

Para cada caso de teste na entrada imprima uma linha contendo um único caractere: "S" (maiúsculo) se existe um jogador atacante impedido, e "N" (maiúsculo) caso contrário.

Exemplo:

Entrada	Saída
2 3	N
500 700	S
700 500 500	N
2 2	
200 400	
200 1000	
3 4	
530 510 490	
480 470 50 310	
0 0	

Apêndice I - ENTREVISTA DA 1ª TURMA

O roteiro de entrevista apresentado abaixo foi aplicado aos 150 classificados na prova e foi montado com os seguintes objetivos: verificação do desempenho escolar, familiaridade com as plataformas Mac OS X e iOS, quantidade de tempo livre e conhecimento de metodologias de aprendizagem ativa. O estudante deveria comparecer à entrevista munido do crachá da PUCPR e do seu histórico escolar da graduação atualizado ou, no caso de estudantes do ensino técnico, histórico escolar do ensino médio.

1. Nome Completo
2. Curso Atual
3. Após análise do Histórico Escolar: Quais os motivos para suas dependências?
4. Quais são as suas atividades semanais?
5. Quantas horas pode dispor ao projeto por semana?
6. Utiliza equipamentos que rodem Mac OS X? Com qual frequência?
7. Utiliza equipamentos que rodem iOS? Com qual frequência?
8. Após explicação sobre o que são metodologias não tradicionais de ensino, principalmente metodologias de aprendizagem ativa: Já participou de alguma atividade que utilizou metodologias não tradicionais de ensino? Descreva sua experiência.
9. Possui algum projeto pessoal que gostaria de realizar durante o curso? Descreva-o.

Apêndice J - PROVA DA 2ª TURMA

O grande número de candidatos (~900) para a segunda turma inviabilizou a prova escrita subjetiva. Optou-se por automatizar o processo o que incluiu um software para entrada e resolução das provas e uma base de dados com cerca de 200 questões de níveis fácil, médio e difícil. As provas foram geradas para cada estudante um dia antes das datas agendadas com 6 questões de raciocínio lógico e 10 questões de programação. Segue abaixo uma das provas geradas:

01.
Sobre o algoritmo:

```
int funcao( char * sFrase )
{
    int ilnicio, iFim;
    ilnicio = 0;
    iFim = strlen(sFrase)-1;
    while( ilnicio < iFim )
    {
        if( sFrase[iFim] != sFrase[ilnicio] )
        {
            return 0;
        }
        iFim--;
        ilnicio++;
    }
    return 1;
}
```

ele ira retornar 0 caso a entrada seja "mussum"
 ele ira retornar 1 caso a entrada seja "mussum"
 ele ira retornar 1 caso a entrada seja "bepid"
 ele sempre irá retornar 1
 ele sempre irá retornar 0

02. Três atletas - Marcelo, Juninho e Raul - frequentam uma mesma academia sendo que, em um dado período do dia, cada um realiza um determinado exercício. Considere o que segue:

- I. Os exercício por eles realizados são: bicicleta, esteira e supino;
- II. Os períodos em que os exercícios foram realizados foram: das 8h00 às 9h00, das 10h00 às 11h00 e das 14h00 às 15h00;
- III. Raul exercitou-se na bicicleta;
- IV. O atleta que correu na esteira o fez das 8h00 às 9h00;
- V. Marcelo realizou o seu exercício à tarde.

Nessas condições, é correto afirmar que:

- Marcelo correu na esteira.
- Raul realizou seu exercício das 8 às 9 horas.
- Juninho correu na esteira.
- Marcelo não fez supino.
- Juninho exercitou-se no supino.

03. Dado um vetor com n elementos do tipo inteiro, com $n > 1$, se deseja criar um algoritmo que, usando uma única estrutura de repetição, itera o referido vetor buscando encontrar dentre seus elementos aquele que tem o maior valor. Pergunta: qual o número mínimo necessário de iterações dentro do laço para garantir que o elemento de maior valor seja encontrado?

```

n*n
n
(n-1)*(n-1)
n-1
n+1

```

```

04.   int sum=0;
      int k=0;
      for (k=1 ; k<=9; k+=4)
      {
        sum = sum +k;
      }

```

que valores q k tem durante a execução desse código?

```

1,5,9
1,5,9,13
0,1,5,9,13
0,1,5,9
0,4,8,12

```

05. Dado algoritmo abaixo, responda:

```

aux = 0

for(var i = 0; i++ < 10; i++){
  aux += i;
}

```

aux tem valor de 24
 aux tem valor de 20
 o loop roda 6 vezes e aux tem valor de 35
 o loop roda 5 vezes e aux tem valor de 25
 o loop roda 10 vezes e aux tem valor de 45

06.

Em uma repartição com cinco funcionários, um deles cometeu um erro grave. Todos eles sabem quem foi o autor desse erro. Esses funcionários têm uma característica muito interessante: quatro deles sempre falam a verdade em qualquer situação, e um deles, às vezes, mente. Um auditor, ao interrogá-los, obteve as seguintes respostas: Funcionário 1: "sou inocente". Funcionário 2: "o funcionário 3 mentiu". Funcionário 3: "o funcionário 4 é o culpado". Funcionário 4: "o funcionário 2 é o culpado". Funcionário 5: "o funcionário 1 disse a verdade".

É CORRETO afirmar que o culpado é o:

```

Funcionário 1
Funcionário 2
Funcionário 3
Funcionário 4
Funcionário 5

```

07. Três rapazes e duas moças vão ao cinema e desejam sentar-se, os cinco, lado a lado, na mesma fila. O número de maneiras pelas quais eles podem distribuir-se nos assentos de modo que as duas moças fiquem juntas, uma ao lado da outra, é igual a:

```

2
4
24
48
120

```

08.

```

for( iL = 0; iL < QTD_L; iL++ )
{

```



```

for( iC = 0; iC < QTD_C; iC++)
{
    mC[iL][iC] = (mA[iL][iC] == mB[iL][iC]);
}
}

```

sendo mC, mA e mB arrays multidimensionais e contando que o algoritmo rode sem problema, selecione a resposta verdadeira

caso o vetor mA tenha valores 2,4,6 e B tenha valores 2,5,8 o vetor C terá os valores
2,5,8

caso o vetor mA tenha valores 2,4,6 e B tenha valores 2,5,8 o vetor C terá os valores
1,0,0

o vetor C terá sempre o valor 1

o vetor C terá sempre o valor 0

caso o vetor mA tenha valores 2,4,6 e B tenha valores 2,5,8 o vetor C terá os valores
0,1,1

09. No trecho do código abaixo, tem-se um procedimento para tratar uma variável B, que é recebida como atributo. O valor desse atributo é a string:

```
B := '>-1.78;+2.34;+3.76;-4.98;+5.65;+6.34<'
```

Tem-se uma variável A que recebe os dados tratados.

Algoritmo Alg1(b string)

início

vet a[1..6]:real

aux, s: string

i,j: inteiro

i ← 1

j ← 1

aux ← ""

enquanto b[i] <> '<' faça

 i ← i+1

 repita

 se b[i] = ','

 então b[i] ← '>'

 aux ← aux + b[i]

 i ← i + 1

 fim_se

 até (b[i] = ';' ou b[i] = '<')

 a[j] ← string2Real(aux)

 se b[i] <> '<'

 então aux ← ""

 j ← j+1

 fim_se

 fim_enquanto

 s ← Real2String(a[1]+a[2]+a[3]+a[4]+a[5]+a[6])

fim

Marque a alternativa que indica corretamente o valor da variável S :

11,33

>-1.78;+2.34;+3.76;-4.98;+5.65;+6.34<

-1.78;+2.34;+3.76;-4.98;+5.65;+6.34

-1.78;+2.34;+3.76;-4.98;+5.65;+6.34<

-1,78+2,34+3,76-4,98+5,65+6,34

10.

Em relação a um código de cinco letras, sabe-se que:

- TREVO e GLERO não têm letras em comum com ele;
- PRELO tem uma letra em comum, que está na posição correta;
- PARVO, CONTO e SENAL têm, cada um, duas letras comuns com o código, uma que se encontra na mesma posição, a outra não;
- MUNCA tem com ele três letras comuns, que se encontram na mesma posição;
- TIROL tem uma letra em comum, que está na posição correta.

O código a que se refere o enunciado da questão é

MIECA
PUNCI
PINAI
PANCI
PINCA

11.

Considere as seguintes afirmações: "Nenhuma estudante é modelo", "Nenhuma fotógrafa é dançarina" e "Todas as jogadoras são estudantes". Com base nessas informações, é correto afirmar que:

Alguma fotógrafa é jogadora e alguma estudante não é modelo.

Toda fotógrafa é jogadora.

Alguma jogadora é fotógrafa.

Nenhuma jogadora é modelo.

Nenhuma modelo é dançarina.

12. Dado o algoritmo, o valor guardado na variável contador é:

```
Contador = 0
for(i=0;i<5000;i+=2)
{
  if(1==i%2)
    Contador += i*2
}
1000
12500
12495000
0
6247500
```

13. Dado o algoritmo a seguir:

```
var
  A, B, C, D: inteiro;
inicio
  escreva "Insira quatro números naturais: "
  leia A, B, C, D
  se ((A > C) && (D < (C - B))) então
    exiba (A + B)
  senão se ((B < (C - D)) || ((B + D) > C)) então
    exiba (A + C)
  senão se ((D < (A - C)) || (B < A)) então
    exiba (A + D)
```

```

senão se ((A + B) < (C + D)) então
  exiba (A + C + D)
senão
  exiba (A + B + C + D)
fim se
fim

```

Qual será a saída caso A = 20, B = 12, C = 30 e D = 14?

32
50
34
64
76

```

14.   int x = 0;
if (x == 1)
if (x == 0)
printf("dentro if\n");
else
printf("dentro else if\n");
else
printf("dentro else\n");

```

dado o algoritmo acima, ele vai imprimir:

nada
dentro if
dentro else if
dentro else
o código tem erro e não roda

15. Em um restaurante, João, Pedro e Rodrigo pediram pratos de carne, frango e peixe, não necessariamente nessa ordem, mas cada um pediu um único prato. As cores de suas camisas eram azul, branco e verde; Pedro usava camisa azul; a pessoa de camisa verde pediu carne e Rodrigo não pediu frango. Essas informações podem ser visualizadas na tabela T abaixo, em que, no cruzamento de uma linha (exibida com o índice entre colchetes após o "T") com uma coluna, V corresponde a fato verdadeiro e F, a fato falso:

```

T[0] = "? , ? , ? , ? , V , ?"
T[1] = "? , ? , ? , ? , ? , ?"
T[2] = "V , ? , ? , ? , ? , ?"
T[1] = "? , ? , ? , ? , ? , ?"
T[1] = "? , F , ? , ? , ? , ?"

```

Na montagem desta tabela, considere os seguintes cabeçalhos de colunas:
Carne, Frango, Peixe, João, Pedro, Rodrigo

E os seguintes títulos de linhas:
Azul, Branca, Verde, João, Pedro, Rodrigo

Considerando a situação apresentada, marque a opção correta:

Se João pediu peixe, então Rodrigo usava camisa Branca.
Se João pediu peixe, então Rodrigo pediu Frango.

Se João pediu carne, então Rodrigo usava camisa Branca.

Se João pediu peixe, então Pedro pediu Carne.
Se João pediu carne, então Pedro pediu peixe.

16. Dado o algoritmo, o valor guardado na variável contador é:

```
contador = 10;  
for(var i = contador-1;i>0;i--){  
  contador *= i  
}  
10000  
30000  
3628800  
3678822  
45000
```

17. Questão Opcional. Descreva de forma resumida uma ou mais aplicações para iOS que você pretende desenvolver durante o curso.

Apêndice K - PUBLICAÇÕES DO MINI DESAFIO DA 1ª TURMA

O primeiro mini-desafio foi proposto no final de 2013 e tinha como objetivo o desenvolvimento de uma aplicação para resolver um problema real. Os estudantes deveriam se reunir em equipes de 3 ou 4 componentes e tinham um prazo de duas semanas para completar a aplicação. Das 12 aplicações desenvolvidas neste período, duas foram publicadas:

- **Força Verde:** App desenvolvida para a polícia ambiental do Paraná cujo objetivo é auxiliar na localização de problemas ambientais.
- **jump-O:** Jogo casual minimalista do tipo puzzle. Foi destaque na Loja de Aplicativos para Dispositivos Móveis duas vezes e ganhou o Principal Prêmio de Design de Dispositivos Móveis em 2015, na categoria de melhor jogo feito por estudantes. Em menos de dois anos o jogo passou a marca de 100 mil downloads na Loja de Aplicativos para Dispositivos Móveis.

Apêndice L - QUESTIONÁRIO PARA PROFESSORES DA 2ª TURMA

O questionário abaixo foi aplicado para dez professores de três universidades que haviam aplicado duas vezes ou estavam aplicando pela segunda vez o método CBL. Foram entrevistas semi-estruturadas individuais, gravadas em áudio, transcritas e analisadas usando redes temáticas.

Experiência Pessoal

1) Quanto tempo de experiência você tem no desenvolvimento de software?

2) Quanto tempo de experiência você tem no ensino?

3) Quanto tempo de experiência você tem no ensino de programação?

4) Com que frequência você utilizou os métodos de ensino de programação abaixo? (Nunca, Poucas vezes, Algumas vezes, Muitas vezes, Sempre)

Aula Expositiva

Aula Prática (Exercício Proposto)

Tutorial Guiado (Professor programa exemplos, estudantes acompanham)

Codificação DOJO

5) Qual é o seu grau de experiência com os seguintes métodos de Desenvolvimento de Software? (Não tenho experiência, Muito pouco, Pouco, Razoável, Muito)

Tradicional

Processo Iterativo e Incremental (RUP por exemplo)

Métodos Ágeis (Scrum, XP, etc.)

Uso do CBL

6) Quantas vezes você aplicou o método CBL?

7) O método CBL motivou mais os estudantes do que outros métodos que você conhece.

Discordo totalmente, Discordo parcialmente, Não concordo nem discordo, Concordo parcialmente, Concordo totalmente.

8) Após o término do projeto você pretende utilizar o método CBL em outras turmas de programação?

Certamente, Provavelmente sim, Talvez, Provavelmente não, Nunca

9) Após o término do projeto você pretende utilizar o método CBL para ensinar outros conteúdos que não sejam de programação?

Certamente, Provavelmente sim, Talvez, Provavelmente não, Nunca

10) Você já utilizou o método CBL em outras situações fora do projeto?

Sim, Não

11) Caso tenha respondido positivamente à questão anterior, liste as disciplinas em que usou/pretende usar o método CBL.

12) Caso tenha respondido negativamente, liste os motivos.

13) A qualidade do software desenvolvido pelos estudantes utilizando o método CBL é melhor do que a qualidade do software desenvolvido utilizando outros métodos que você tem experiência.

Discordo totalmente, Discordo parcialmente, Não concordo nem discordo, Concordo parcialmente, Concordo totalmente

14) O método CBL incentiva os estudantes a utilizarem as seguintes tecnologias modernas?

Vídeos, Tablets, Smartphones, Redes Sociais, Outros

15) Você promoveu alguma alteração no método CBL durante sua aplicação?

Sim, Não

Alterações no método CBL

Abaixo estão listadas as etapas do método. Caso não tenha feita alterações em uma determinada etapa, deixe a resposta em branco.

16) Que alterações você promoveu na etapa "Big Idea"?

17) Que alterações você promoveu na etapa "Essential Question"?

18) Que alterações você promoveu na etapa "The Challenge"?

19) Que alterações você promoveu na etapa "Guiding Questions, Activities and Resources"?

20) Que alterações você promoveu na etapa "Solution"?

21) Que alterações você promoveu na etapa "Implementation"?

22) Que alterações você promoveu na etapa "Evaluation"?

23) Descreva outras alterações que você fez no método CBL.

24) Qual é a sua opinião sobre o método CBL?

25) Apenas PUCPR: Qual é a sua opinião sobre as alterações feitas na primeira turma?

26) Apenas PUCPR: Qual é a sua opinião sobre as alterações feitas na segunda turma?

Apêndice M - TABULAÇÃO DAS ENTREVISTAS DA 1ª TURMA

A tabela M-1 foi gerada pela ferramenta Atlas.TI com base na análise das entrevistas dos professores do primeiro ciclo. A qualificação das respostas dos professores foi feita com base principalmente nas justificativas apresentadas. As justificativas foram encorajadas em todas as respostas, deixando-se o professor livre durante suas explicações.

Tabela M-1: Temas Básicos x Temas Organizadores.

Temas Básicos / Temas Organizadores	Aspectos Inconclusivos	Aspectos Negativos	Aspectos Positivos	Sugestões de Melhoria	Sugestões de Outros Métodos
Ambiente Inadequado	0	3	0	0	0
Continuidade de Uso do Método	0	0	14	0	0
Criar uma variante simplificada	0	0	0	1	0
Desafios Pré-definidos	0	0	0	1	0
Desenvolvimento de Software	5	0	0	0	0
Design Thinking	0	0	0	0	1
Dificuldade de Entendimento	0	2	0	0	0
Falta de Familiaridade com o Método	0	2	0	0	0
Falta de Tempo	0	1	0	0	0
Fatores Extra Codificação	0	0	1	0	0
Flexibilidade de Estudo	0	0	2	0	0
Inadequação ao Desenvolvimento de Software	0	2	0	0	0
Incentivo à Autonomia	0	0	6	0	0
Incentivo à Comunicação	0	0	1	0	0
Incluir o Design Thinking	0	0	0	1	0
Incluir o Scrum	0	0	0	1	0
Incluir Questões Guias para Descobrir a Questão Essencial	0	0	0	3	0
Iniciar com o Desafio	0	0	0	2	0
Liberdade de Escolha do Desafio	0	0	1	0	0
Melhoria do Desenvolvimento de Software	0	0	1	0	0
Melhoria Motivação	0	0	13	0	0
Melhoria no Entendimento do Desafio	0	0	6	0	0
Motivação	3	0	0	0	0
Palestras sobre Problemas Relevantes	0	0	0	1	0
Perfil de Estudantes Inadequado	0	8	0	0	0
Problem Based Learning	0	0	0	0	3
Project Based Learning	0	0	0	0	2
Redução da Evasão e da Reprovação	0	0	1	0	0
Reuniões Técnicas de Acompanhamento	0	0	0	1	0
Rigidez do método na parte inicial	0	1	0	0	0
User Experience	0	0	0	0	1

Apêndice N - REFLEXÕES DOS PROFESSORES NA SEGUNDA TURMA

As perguntas abaixo foram respondidas pelos professores em forma de áudio gravado (até 2 minutos) e realizadas em inglês devido a exigências da empresa parceira. O texto inicial entre parênteses indica a fase do curso na qual a reflexão deveria ser feita.

1.1 (early phase) What are my teaching goals this year? What are my learning goals this year?

1.2 (early phase) How did we establish a sense that this class was not going to be “business as usual”?

1.3 (early phase) The most valuable part of the CBL process so far is . . . Why?

1.4 (early phase) What are my key take aways from the weekly Fuze meetings?

2.1 (middle phase) What was the most interesting event I witnessed during the mini challenges? What caused the event?

2.2 (middle phase) What soft skills are the students learning and how are they learning them?

2.3 (middle phase) Are their emerging skills demonstrating competency in iOS development? How am I measuring them?

2.4 (middle phase) Are the students successfully applying their iOS development knowledge? Where are they strong? Weak?

2.5 (middle phase) At this stage, what support do I need from the Support team and why?

3.1 (final challenge phase) What have I learned about teaching so far?

3.2 (final challenge phase) Are the skills demonstrating competency in iOS development? How are you measuring?

3.3 (final challenge phase) What soft skills are the students learning and how are they learning them?

3.4 (final challenge phase) What is the most challenging part of the final challenge phase?

3.5 (final challenge phase) Are my students comfortable with the CBL process? How is this demonstrated?

3.6 (final challenge phase) At this stage, what support do I need from the Support team and why?

4.1 (post phase) This past year taught me . . .

4.2 (post phase) My highlight of the last year was . . .

4.3 (post phase) My biggest struggle of the last year was . . .

4.4 (post phase) Our programs most innovative contribution to BEPiD is . . .

4.5 (post phase) CBL is . . .

4.6 (post phase) I believe my students will be successful because . . .

4.7 (post phase) I think my students may struggle in the following areas . . .

Why?

4.8 (post phase) Next time I will improve on... How?

4.9 (post phase) The Support team could better support me by...

Apêndice O – CRONOGRAMA DA PESQUISA-AÇÃO

A Figura O-1 mostra o cronograma de eventos da pesquisa-ação.

Atividades/Período	2013						2014						2015																	
	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Ciclo 1	[Barra preta]																													
Seleção dos Alunos	[Barra preta]																													
Curso de Nivelamento	[Barra preta]																													
Preparação dos Professores	[Barra preta]																													
Desafio Inicial	[Barra preta]																													
Estudo dos Tópicos	[Barra preta]																													
Mini-Desafio 1	[Barra preta]																													
Desafio Final	[Barra preta]																													
Criação de Startups	[Barra preta]																													
Ciclo 2	[Barra preta]																													
Seleção dos Alunos	[Barra preta]																													
Estudo dos Tópicos	[Barra preta]																													
Desafio Inicial	[Barra preta]																													
Mini-Desafio 1	[Barra preta]																													
Mini-Desafio 2	[Barra preta]																													
Desafio Final	[Barra preta]																													
Seleção das Startups	[Barra preta]																													

Figura O-1: Cronograma da Pesquisa-Ação