



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Design Sprint e Aprendizagem Baseada em Projetos:
um modelo para o alcance de autenticidade nos
projetos acadêmicos**

Vinícius Gomes Ferreira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientadora
Prof.a Dr.a Edna Dias Canedo

Brasília
2019

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

GG633d Gomes Ferreira, Vinícius
Design Sprint e Aprendizagem Baseada em Projetos: um
modelo para o alcance de autenticidade aos projetos
semestrais / Vinícius Gomes Ferreira; orientador Edna Dias
Canedo. -- Brasília, 2019.
124 p.

Dissertação (Mestrado - Mestrado Profissional em
Computação Aplicada) -- Universidade de Brasília, 2019.

1. Aprendizagem Baseada em Projetos. 2. Design Sprint.
3. Experiência do Usuário. 4. Autenticidade. 5. Educação em
Engenharia de Software. I. Dias Canedo, Edna, orient. II.
Título.



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Design Sprint e Aprendizagem Baseada em Projetos:
um modelo para o alcance de autenticidade nos
projetos acadêmicos**

Vinícius Gomes Ferreira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof.a Dr.a Edna Dias Canedo (Orientadora)
CIC/UnB

Prof. Dr.a Letícia Lopes Leite Dr. Sérgio Crespo Coelho da Silva Pinto
Universidade de Brasília (UnB) Universidade Federal Fluminense (UFF)

Prof.a Dr.a Aletéia Patrícia Favacho de Araújo
Coordenadora do Programa de Pós-graduação em Computação Aplicada

Brasília, 22 de Agosto de 2019

Dedicatória

Dedico esse trabalho primeiramente à Deus, que me concedeu forças e capacidade para estar onde estou hoje, à minha esposa, que me assistiu e suportou durante esse tempo de mestrado (só ela e Deus sabem o quanto precisei do seu apoio) e à minha mãe, senhora Maurett, que me criou tão bem, motivo pelo qual sou quem sou hoje.

Agradecimentos

Presto meus agradecimentos à Universidade de Brasília e ao Instituto Federal de Goiás, especialmente ao campus Formosa, instituições que ajudam na minha formação como profissional, cidadão e pessoa. Não devo também esquecer de minha orientadora Edna Dias Canedo, que foi um divisor de águas na minha vida, me transformando em um acadêmico muito melhor e de meus colegas de mestrado, todos muito solícitos ante aos meus pedidos de socorro.

Resumo

A aplicação do conteúdo didático somente através das aulas em formato expositivo pode não ser suficiente para o ensino de engenharia de software nesta nova era. Muito disso se deve à mudança de foco do produto para o usuário que o usa que influencia a comunidade de software nos tempos atuais. Como resultado disso, a experiência do usuário (UX) tornou-se requisito essencial no processo de desenvolvimento de software e uma atividade necessária a todos os profissionais da área de Tecnologia da Informação (TI) que buscam construir produtos de qualidade. No entanto, ainda existem poucas iniciativas que proporcionam a experimentação de técnicas de desenvolvimento de softwares voltados para a UX na grade curricular dos cursos de graduação na área de TI durante a formação dos estudantes. A *Design Sprint* (DS) aliada à Aprendizagem Baseada em Projetos (ABP) se apresenta como uma maneira capaz de mitigar esse problema. Uma das principais características do ABP para desenvolvimento de software é a possibilidade de gerar artefatos autênticos, ou seja, fundamentados em um contexto de mundo real. Adicionalmente, a *Design Sprint* funciona como um bom processo para criação de produtos voltados para o usuário. Este trabalho apresenta uma revisão sistemática de literatura (RSL) cujo objetivo é investigar como comumente são executadas as unidades de ABP que produzem softwares autênticos. Além disso, são relatados dois estudos de caso que visam explorar como a *Design Sprint* se comporta em um contexto de sala de aula. Os resultados da RSL indicaram que o *Scrum* é o processo de desenvolvimento de software mais utilizado em unidades ABP que geram softwares autênticos, que os alunos são normalmente divididos em grupos de 2 a 5 pessoas durante as atividades e que os monitores exercem um papel importante na qualidade dos softwares produzidos. Já os resultados dos estudos de caso indicam que o tempo limitado para a condução da ABP é um fator de complicação na adaptação da *Design Sprint* para o contexto educacional, que os alunos julgam de maneira positiva sua própria aprendizagem e participação e que a *Design Sprint* fornece *insights* interessantes que podem ser úteis para a especificação dos requisitos do software a ser desenvolvido Aprendizagem Baseada em Projetos. O agrupamento desses resultados permitiu a criação de um formato de *Design Sprint* adaptado para a inserção de ideação da ABP.

Palavras-chave: Aprendizagem Baseada em Projetos, *Design Sprint*, Experiência do Usuário, Autenticidade, Educação em Engenharia de Software.

Abstract

Apply didactic content only through lessons in expository format may not be sufficient for the teaching software engineering in this new era. Much of this is due to the change of focus of product to user that influences the software community in current times. As a result, user experience (UX) has become an essential requirement in the software development process and a necessary activity for all professionals in the Information Technology (IT) area who seek to build quality products. However, there are still few initiatives that provide the experimentation of software development techniques focused on UX in curriculum of undergraduate courses in the IT area. *Design Sprint* (DS) combined with Project-Based Learning (PBL) appear as a way to mitigate this problem. One of the main characteristics of the PBL for software development is the possibility of generating authentic artifacts, that is, artifacts based on a real world context. Additionally, Design Sprint works as a good process for creating user-oriented products. This work presents a systematic literature review (SRL) whose objective is to investigate how PBL units that produce authentic software are commonly executed. In addition, two case studies are reported that aims to explore how Design Sprint behaves into a classroom context. The SRL results indicated that Scrum is the most widely used software development process in ABP units that generate authentic software, that students are usually divided into groups of 2 to 5 people during the activities and that monitors play an important role in the quality of the software produced. The results of the case studies indicate that the limited time to conduct the Design Sprint unit is a complicating factor in its adaptation to educational context, that students judge positively their own learning and participation and that Design Sprint provides interesting insights that can be useful for specifying the requirements of the software to be developed in PBL unit. The grouping of these results allowed the creating a Design Sprint adapted for insertion in PBL's ideation stage.

Keywords: Project-Based Learning, Design Sprint, User Experience, Authenticity, Software Engineering Education.

Sumário

1	Introdução	1
1.1	Problema de Pesquisa	3
1.2	Justificativa	3
1.3	Objetivos	5
1.3.1	Objetivo Geral	5
1.3.2	Objetivos Específicos	5
1.4	Resultados Esperados	6
1.5	Metodologia de Pesquisa	6
1.6	Estrutura do Trabalho	7
2	Embasamento Teórico	8
2.1	As influências psicopedagógicas da Aprendizagem Baseada em Projetos . . .	8
2.1.1	As raízes das teorias da aprendizagem: do <i>behaviorismo</i> ao cognitivismo	9
2.1.2	Construtivismo	11
2.1.3	“ <i>Learning by doing</i> ”: o pragmatismo de John Dewey e o método de projeto de William Kilpatrick	14
2.1.4	Construcionismo: aprender por meio da construção de artefatos	16
2.1.5	Aprendizagem Baseada em Projetos	16
2.1.6	A autenticidade na aprendizagem	20
2.2	Experiência do Usuário (UX)	23
2.2.1	Os artefatos e sua capacidade de prover experiência	24
2.2.2	<i>Design Thinking</i> e <i>Design Sprint</i> como processos de UX	28
2.3	Trabalhos Correlatos	34
2.4	Síntese do Capítulo	36
3	Revisão Sistemática de Literatura	37
3.1	Planejamento	39
3.1.1	Questões de Pesquisa	39
3.1.2	Estratégias de Pesquisa	40

3.1.3	Critérios de Seleção	42
3.1.4	Critérios de Qualidade	43
3.1.5	Validação do Protocolo	44
3.2	Condução	44
3.2.1	Seleção dos Estudos Primários	44
3.2.2	Extração de Dados	46
3.3	Resultados	46
3.3.1	QS.1 - Como o conteúdo teórico e técnico é ensinado aos alunos enquanto eles desenvolvem os projetos propostos nas disciplinas?	50
3.3.2	QS.2 - Qual o tempo disponível utilizado nas aplicações de aprendizagem baseada em projetos para criação de softwares que serão ou podem ser usados em um contexto real?	52
3.3.3	QS.3 - Quais modelos e/ou técnicas de engenharia de software são usados no projeto?	54
3.3.4	QS.4 - Como são formados os grupos de alunos que participam do desenvolvimento de software em sala de aula?	55
3.3.5	QS.5 - Houve mudança no rendimento acadêmico dos alunos após participarem das atividades práticas de aprendizagem propostas durante a unidade letiva?	56
3.3.6	QP - Como aplicações de <i>Aprendizagem Baseada em Projetos</i> para <i>projetos de desenvolvimento de software durante os cursos de graduação</i> tornam um <i>software autêntico</i> ?	57
4	Estudos de Caso Exploratórios	59
4.1	Estudo de caso 1: ABP usando <i>Design Sprint</i> em uma disciplina de Engenharia de Requisitos	59
4.1.1	Pré-Sprint	61
4.1.2	Entender	62
4.1.3	Decisão	63
4.1.4	Prototipar	64
4.1.5	Pós-Sprint	64
4.1.6	Questionário do Estudo de Caso 1	65
4.1.7	Considerações sobre primeiro estudo de caso	67
4.2	Estudo de caso 2: ABP usando <i>Design Sprint</i> em uma disciplina de Engenharia de Software	69
4.2.1	Pré-Sprint	71
4.2.2	Aula 1 - Questão âncora e imersão	71
4.2.3	Aula 2 - Demonstrações relâmpago e ideiação	72

4.2.4 Aula 3 - Decisão	73
4.2.5 Aula 4 - Validação	74
4.2.6 Questionário do Estudo de Caso 2	74
4.2.7 Considerações Sobre o Estudo de Caso 2	76
5 Proposta	79
5.1 Identificar o Problema	80
5.2 Convidar o Especialista do Negócio	81
5.3 Desenhar a Jornada do Usuário	81
5.4 Aplicar a <i>Design Sprint as a Framework</i> - DSaaF	82
5.5 Aplicar o <i>eduScrum</i>	86
5.6 Realizar a Apresentação Pública	87
6 Conclusão	89
Referências	92
Apêndice	103
A Roteiro das Aulas da Design Sprint para Aprendizagem Baseada em Projetos	104

Lista de Figuras

2.1 Duplo Diamante. Adaptado de [1]	30
2.2 Passos da <i>Design Sprint</i> . Adaptado de [2]	32
3.1 Protocolo da Revisão Sistemática da Literatura	38
3.2 Quantidade de artigos selecionados por biblioteca digital.	45
3.3 Proporção de artigos por fonte de busca	46
3.4 Quantidade de artigos por etapa da RSL	47
4.1 Áreas de interesse dos alunos	66
4.2 Aprendizado e comprometimento dos alunos	67
4.3 Avaliação geral da aplicação da <i>Design Sprint</i>	68
4.4 Áreas de interesse dos alunos	75
4.5 Aprendizado e comprometimento dos alunos	76
4.6 Avaliação geral da aplicação da <i>Design Sprint</i>	77
5.1 Modelo macro da proposta	79
5.2 <i>Design Sprint as a Framework</i>	82
5.3 Mapa de Empatia	85
5.4 <i>Job Story</i> de Experiência do Usuário	85

Lista de Tabelas

3.1	Questões secundárias de pesquisa (QS)	40
3.2	Conferências e Periódicos consultados na Busca Manual	41
3.3	Artigos selecionados na RSL	48
3.4	Tempo programado para a unidade de Aprendizagem Baseada em Projetos .	53
3.5	Uso dos processos de desenvolvimento de software	58
4.1	Notas do How Might We (HMW)	63

Capítulo 1

Introdução

A característica que separa os cursos de tecnologia de outros cursos de graduação (bacharelado e licenciatura) é o foco nas questões práticas [3], preparando o aluno muito mais para o mercado de trabalho do que para a academia, como fazem os cursos de bacharelado e licenciatura. A matriz curricular desses cursos sugerida pelo catálogo de cursos do Ministério da Educação (MEC) é desenhada para atender esse objetivo [4].

A discussão que cerca o poder que os cursos de graduação têm em produzir profissionais para suprir as demandas da indústria tem sido feita há algum tempo. Schön [5] discute o comprometimento das ciências em geral, inclusive das aplicadas, como uma epistemologia particular afastada das competências práticas dos profissionais, quando na verdade deveria pautar seus métodos de ensino no exercício da profissão. Problemas muito similares cercam também a educação em desenvolvimento de software e sustentam o distanciamento entre o que se espera dos egressos e o que eles realmente conseguem fazer quando saem da sala de aula.

Um dos aspectos mais críticos do processo de desenvolvimento de software é a tradução de requisitos explícitos e tácitos dos usuários em funcionalidades que façam sentido para eles [6]. É esperado de um egresso de um curso de desenvolvimento de sistemas, ou áreas afins, que ele tenha, entre outras coisas, as habilidades de [7]:

- Escolher entre alternativas de soluções de problemas, avaliando seus riscos, custos e benefícios, em conjunto com os *stakeholders* dos projetos de software;
- Lidar com os clientes, principalmente no tocante ao levantamento e gerenciamento dos requisitos destes para o software que esperam receber;
- Tomar decisões de projeto e pensar em termos de resolução de problemas para a construção de soluções que sejam úteis e eficientes para seus usuários finais.

Lidar com usuários tem sido um desafio para o profissional de desenvolvimento de software desde sempre e, nos dias atuais, tem ganhado uma dimensão ainda mais importante.

Uma das tendências dos últimos anos está na mudança de foco dos aspectos funcionais de um sistema, como é tratado pela área da usabilidade e da interação humano computador (IHC), para outro que considere as características hedônicas dos usuários, tais como alegria, satisfação, prazer, etc. A esse campo de estudo, dá-se o nome de Experiência do Usuário (UX) [8].

Uma das maneiras mais efetivas de se lidar com a educação dos novos profissionais da área, principalmente no que tange às novas tendências e aos conhecimentos críticos da profissão, é ensiná-los por meio de exercícios práticos que simulem as situações que os profissionais encontram na indústria [9, 10, 11]. Nessa linha, as pesquisas em educação prática para o desenvolvimento de software propõe que metodologias construtivistas de ensino ocupem esse papel [12, 13, 14, 15], sendo a Aprendizagem Baseada em Projetos (ABP) uma das mais utilizadas para este fim [16, 17, 18]. A ideia por trás da ABP está na possibilidade do aluno desenvolver seu conhecimento por meio de construção de artefatos que possam ser percebidos e avaliados por outros [19, 20, 21], não mais aguardando que o professor lhe preencha com os saberes da humanidade, mas agindo ativamente em busca de informações, testando hipóteses, criando artefatos e lidando com problemas do mundo real, que são comumente relacionados à autenticidade da unidade pedagógica.

Uma unidade letiva autêntica é aquela, que além de ser muito próxima de situações que acontecem no mundo real, também envolve algum nível de qualidade do artefato produzido em sala de aula [22, 23, 24]. Neste sentido, é razoável considerar que, para a ABP aplicada no contexto da educação em desenvolvimento de software, a qualidade da experiência que o usuário final terá durante o uso do artefato deve ser considerada como um aspecto importante de sua autenticidade. Isso pode trazer vantagem competitiva para os artefatos desenvolvidos pelos alunos em comparação àqueles produzidos por profissionais formados diante da aceitação e uso deles, uma vez que para fazer que os softwares desenvolvidos pelos alunos não fiquem engavetados, é necessário que eles sejam minimamente comparáveis aos seus concorrentes feitos por profissionais, que já começam a considerar a UX como preocupação chave de um software de sucesso.

Embora as atividades na ABP, no geral, sejam bem próximas das que os alunos encontrarão em sua vida laboral e alguns resultados de sua aplicação reportados na literatura sejam animadores [25, 26, 27], não há muitos trabalhos que se proponham a estudar maneiras de incluir explicitamente a preocupação com UX no escopo das atividades de desenvolvimento de software de unidades de ABP. A atividade mais próxima desse objetivo é o *brainstorming* para criação de protótipos [28], sem um foco em exercícios de empatia, como os modelos de *design* em UX propõem que seja feito [29].

Um substituto para o *brainstorming* que é usado projetos da indústria de software quando há alguma necessidade de empatia é a *Design Sprint* [2], um processo baseado

em Design Thinking, orientado por exercícios de criatividade focados e limitados por tempo *brainstorming* apresenta [2, 30]. Portanto, guiado pela ideia de que um protótipo derivado de um processo de UX é capaz de incluir traços de um produto melhor avaliado pelos usuários [31], esse trabalho propõe a substituição do *brainstorming* convencional pela *Design Sprint* para criação da proposta do artefato de software a ser construído na unidade de ABP.

Entendendo que a qualidade de uso de um artefato de ABP está ligado a seus atributos de autenticidade, este trabalho apresenta uma Revisão Sistemática da Literatura (RSL) sobre a ABP com o propósito de identificar como a autenticidade tem sido alcançada em que envolvem a construção de software. O conhecimento obtido com essa pesquisa se une ao conhecimento empírico de estudos de caso exploratórios para a confecção de um modelo de *Design Sprint* que permita a construção de propostas de software voltados à UX a serem desenvolvidos em unidades de ABP.

1.1 Problema de Pesquisa

Os softwares desenvolvidos pelos alunos do curso de Análise e Desenvolvimento de Sistemas, seja em disciplinas regulares, em projetos de iniciação científica e tecnológica ou em projetos de conclusão de curso, frequentemente ficam inutilizados quando concorrem com outros softwares que são comerciais. Um dos motivos para isso, se deve a um reconhecimento de que esses softwares não são úteis no dia a dia das pessoas, podendo mais atrapalhar do que ajudar. Em outras palavras, as propostas de softwares feita pelos alunos carecem de um olhar voltado para a satisfação de variáveis hedônicas dos usuários para terem algum apelo e chance de concorrência com os sistemas feitos por profissionais.

1.2 Justificativa

O Instituto Federal de Goiás (IFG) possui uma estrutura multi campi espalhada pelo Estado de Goiás, tendo sua reitoria na cidade de Goiânia, onde suas principais funções de administração e governança são centralizadas. Desta maneira, os campi não possuem seus próprios desenvolvedores e suas demandas são supridas apenas quando elas precisam ser supridas por todos os outros campi. Em virtude do acúmulo de trabalho para a equipe de desenvolvimento, os novos sistemas demoram ser lançados.

Como a maioria dos campi ficam em outras cidades, à exceção do campus Goiânia e do campus Goiânia Oeste, a distância entre a reitoria e esses campi torna inviável a presença dos membros da equipe de desenvolvimento em todos eles para manter relação estreita com os servidores que serão afetados pelos sistemas desenvolvidos e conduzir testes de

usabilidade e experiência de usuário. Em especial o campus Formosa, que está entre um dos campus mais distantes da Capital. Uma solução imediata a este problema é o aproveitamento da mão de obra discente de cursos de Tecnologia em Análise e Desenvolvimento de Sistemas dos campi para prover softwares que supram demandas pontuais. Uma abordagem adequada para a viabilização desta estratégia é a adoção da ABP em disciplinas do curso ou projetos extracurriculares de desenvolvimento de software. Atualmente o curso de Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do IFG Campus Formosa passa por um processo de reformulação de seu Projeto Pedagógico de Curso e considera incluir uma série de disciplinas de ABP para desenvolvimento de softwares que sejam úteis à comunidade acadêmica que o cerca. No entanto, para tornar justificável a adoção de softwares desenvolvidos pelos alunos é necessário que esses softwares sejam fáceis e prazerosos de usar, de maneira que traga real valor às tarefas do dia a dia dos funcionários do IFG.

Ter softwares autênticos, principalmente em relação à experiência do usuário que o utilizará, produzidos pelos alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas no IFG, poderá, entre outras coisas, alcançar os seguintes benefícios a longo prazo:

- Aceleração do suprimento de demandas específicas dos campi;
- Maior atenção às necessidades dos usuários que trabalham nos campi e precisam usar os sistemas da instituição;
- Possibilidade de redução de custos com a diminuição de gastos com empresas que mantêm softwares proprietários;
- Criação de portfólios de sistemas para os alunos que agregarão em seus currículos como desenvolvedores;
- Possibilidade de proporcionar aos alunos experiências reais de desenvolvimento de software;
- Aumento da motivação do aluno nas unidades de ABP em virtude da autenticidade do trabalho realizado;
- Ensino do valor que o *design* de experiência do usuário possui em um projeto de software.

No entanto, há uma série de críticas contundentes à falta de estrutura das sessões conduzidas nesse formato, à falta de espaço destinado para análise das ideias e maior consideração das soluções e a uma possível influência das respostas dadas pelos participantes mais extrovertidos nas respostas dadas pelos participantes mais introvertidos [2, 30] o

que prejudica bastante o exercício de criatividade de um processo de *design* de UX [29]. Para o desenvolvimento de soluções que serão úteis é necessário que a proposta de solução seja bem definida e explicitamente voltados para os problemas dos usuários. Portanto, o foco desse trabalho reside na definição de um modelo que seja capaz de lidar com essas dificuldades que os modelos convencionais de ABP oferecem, com principal enfoque em cursos de graduação contemplados pelo ensino da engenharia de software.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é redesenhar o formato de *Design Sprint* para ser usado no estágio de ideação, prototipação e validação de ideias da Aprendizagem Baseada em Projetos com vistas à formulação de propostas mais aceitáveis do ponto de vista do usuário.

1.3.2 Objetivos Específicos

Com o intuito de atingir o objetivo geral, alguns objetivos específicos foram definidos:

- Verificar como a *Design Sprint* se comporta quando inserida em sala de aula, com restrições de tempo e um conjunto de indivíduos homogêneo, diferente do que o proposto por seus idealizadores [2], por meio de estudo de casos descritivos;
- Identificar qual a visão dos alunos do IFG - Campus Formosa em relação ao uso da *Design Sprint* com propósitos educacionais;
- Conhecer como a autenticidade dos artefatos produzidos por alunos dos cursos de Tecnologia (Ciência da Computação, Engenharia de Software e Sistema de Informação) tem sido alcançada durante a ABP através de uma revisão sistemática da literatura, em especial às aplicações da abordagem em disciplinas com foco em processo como Engenharia de Software e Gerência de Projetos;
- Ajustar a *Design Sprint* iterativamente às necessidades específicas de uma sala de aula conforme os aprendizados obtidos pela revisão da literatura e em cada estudo de caso exploratório;
- Validar o modelo final da *Design Sprint* em uma típica sala de aula com alunos de graduação de um curso da área da computação.

1.4 Resultados Esperados

Se espera com este trabalho os seguintes resultados:

- Conhecer a viabilidade da aplicação da *Design Sprint* em um contexto educacional;
- Contribuir para o corpo de conhecimento da ABP e suas variantes no tocante à autenticidade dos artefatos produzidos pelos alunos;
- Providenciar um meio para que os produtos de software desenvolvidos pelos alunos em unidades de ABP alcancem requisitos de UX e possam ser utilizados pelos funcionários do IFG;
- Obter a percepção dos alunos de um curso de Tecnologia em Análise e Desenvolvimento de Sistemas sobre a *Design Sprint* com propósitos educacionais;
- Abrir um caminho para que as demandas pontuais dos campus das cidades do interior do Instituto Federal de Goiás possam ser supridas pelos softwares desenvolvidos pelos próprios alunos;
- Tornar os alunos do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do IFG - Campus Formosa conscientes da prática do processo de desenvolvimento de software e do entendimento das necessidades de desenvolvimento centrado no usuário.

1.5 Metodologia de Pesquisa

Este trabalho faz uso da Revisão Sistemática de Literatura (RSL) definida por Kitchenham [32] para exploração bibliográfica a fim de que forneça *insights* e respostas a questões relativas ao alcance do atributo autenticidade nos softwares desenvolvidos dentro do contexto da Aprendizagem Baseada em Projetos.

Os estudos de caso exploratórios foram definidos conforme a recomendação de Gil [33] e compõem a etapa de estudo da *Design Sprint* para fins educacionais. Os dados coletados por meio observação e por meio dos questionários respondidos pelos alunos se unirão aos dados coletados da RSL para modelar a proposta de *Design Sprint* que funcione bem dentro das restrições de um ambiente educacional

Esta pesquisa teve todas as suas etapas realizadas no âmbito do Instituto Federal de Goiás - Campus Formosa. O primeiro estudo de caso foi realizado nas aulas da disciplina de Engenharia de Requisitos, que possui carga horária semanal de 3 horas, mas possuiu uma etapa anterior e uma etapa posterior que complementou as horas delimitadas para a execução das principais atividades do estudo. O segundo estudo de caso foi realizado

na disciplina de Engenharia de Software, durante 4 semanas, usando 5 horas de sua carga horária semanal. O detalhamento dos estudos de casos realizados são apresentados no Capítulo 4.

1.6 Estrutura do Trabalho

Este trabalho está organizado em 5 capítulos, além deste, consistindo em:

- **Capítulo 2:** apresenta os conceitos teóricos necessários para o entendimento deste trabalho, bem como os trabalhos correlatos existentes na literatura.
- **Capítulo 3:** descreve a metodologia de pesquisa adotada, o protocolo utilizado e o resultado da Revisão Sistemática de Literatura, bem como as respostas às questões de pesquisas que foram definidas neste trabalho.
- **Capítulo 4:** apresenta os estudos de casos realizados para a alimentação da proposta final da *Design Sprint* para ABP.
- **Capítulo 5:** apresenta uma proposta da *Design Sprint* para ABP.
- **Capítulo 6:** Este capítulo apresenta a conclusão do trabalho e apresenta os trabalhos futuros.

Capítulo 2

Embasamento Teórico

Este capítulo apresenta os conceitos necessários para o entendimento desta pesquisa. Tendo ela um caráter multidisciplinar que usa a psicologia, a pedagogia e a engenharia de software como fontes, desenha-se, portanto, discussões nessas três áreas. Por parte da psicologia e da pedagogia, na seção 2.1 são feitas considerações partindo dos princípios epistemológicos da psicologia da aprendizagem e de correntes de pensamento desta área que conduziram às epistemologias que influenciaram diretamente a Aprendizagem Baseada em Projetos (ABP), nomeadamente o *learning by doing* (Seção 2.1.3) e o construcionismo (Seção 2.1.4). Ainda dentro do escopo da pedagogia, é desenvolvida uma discussão sobre o conceito de autenticidade mantendo enfoque nas considerações sobre a autenticidade na construção de artefatos, que é o que, de fato, interessa a esta pesquisa. E, finalmente, por parte da Engenharia de Software, a discussão alcança os conceitos de experiência de usuário (UX), *Design Thinking* e descreve em detalhes a *Design Sprint* como ferramenta de inovação inspirada por estes dois últimos. Além disso, trabalhos correlatos ao que se propõe nessa pesquisa são apresentados e discutidos na Seção 2.3.

2.1 As influências psicopedagógicas da Aprendizagem Baseada em Projetos

A área da psicologia tem influenciado a educação desde o século XIX, quando se sobrepôs à filosofia na função de sistematizar este corpo de conhecimento e de propor métodos e teorias para que lhe servissem de base [34] e não seria diferente com a prática pedagógica da Aprendizagem Baseada em Projetos. Diz-se que a ABP é uma abordagem, ao mesmo tempo, construtivista [35, 36] e construcionista [37], que permite que os alunos acumulem conhecimento por meio do “aprender fazendo” (*learning by doing*) [38, 18].

Importantes trabalhos na afirmação da Psicologia da Educação como área em si mesma, apartada da mera concepção de psicologia aplicada à educação, partiram das ideias de Edward L. Thorndike [39, 40, 41, 42, 34] e da contraposição ideológica presente nos trabalhos de Charles H. Judd [43, 44]. A partir de então, muitas teorias de como o conhecimento é adquirido tem sido propostas, competindo entre si e influenciando a prática escolar em suas concepções, crenças e execuções. Uma corrente em especial, o construtivismo, ganhou proeminência e trouxe alternativas aos modelos de ensino centrados no professor, deslocando o aluno para o centro da aprendizagem [45, 46]. Partiu dela, então, muitos métodos pedagógicos progressistas que se ramificaram em um variado número de epistemologias que viriam a embasar os métodos modernos de ensino. Esses têm como pressuposto o empoderamento do aluno em detrimento da mera transmissão de conhecimento do professor a uma sala de aula que permanece passiva [47].

As próximas subseções tratarão de expor a evolução das teorias da aprendizagem, conduzindo uma discussão com foco no nascimento e difusão das ideias do ensino construcionista e do aprendizado por meio da prática. Cabe mencionar que as subseções seguintes não têm por objetivo esgotar a discussão sobre as teorias pedagógicas construtivistas, mas delinear um apanhado de conceitos que justificam a prática pedagógica de construção de artefatos, que está diretamente ligada ao propósito dessa pesquisa.

2.1.1 As raízes das teorias da aprendizagem: do *behaviorismo* ao cognitivismo

A epistemologia é de interesse da filosofia desde os tempos antigos e dela surgem as grandes teorias de como o ser humano aprende e acumula conhecimentos [48], o que influencia diretamente a pedagogia e dá suporte a variados métodos de ensino que estão sendo explorados nos dias atuais. Embora a epistemologia já exista há bastante tempo, somente nos dois últimos séculos que as grandes teorias científicas começaram a surgir, dentre as quais o *behaviorismo* foi a primeira [49, 34].

O *behaviorismo*, ou comportamentalismo, foi, portanto, o ponto de partida de uma sistematização de teorias de aprendizagem. John Watson e Frederick Skinner são dois dos principais nomes na escola da psicologia *behaviorista*, sendo o primeiro deles representante do *Behaviorismo* Metodológico e o segundo representante do *behaviorismo* Radical [49, 46]. O *Behaviorismo* Metodológico parte do pressuposto que como as emoções não podem ser observadas, elas também não podem ser estudadas. Desta forma, o interesse dessa escola se fixa na simples relação entre o estímulo e a ação involuntária, comumente chamada de ação reflexa, julgando que a introspecção não deveria ser considerada prática científica [49, 46, 50]. Por outro lado, o *Behaviorismo* Radical não desconsidera as emoções no estudo do

comportamento, mas as tratam como fenômenos de natureza material e partes integrantes do comportamento assim como toda reação fisiológica é [45, 51]. Por consequência, o *Behaviorismo* Radical entende que as emoções também podem ser medidas assim como o comportamento fisiológico [50].

Por ser interessado no estudo das ações reflexas, o *Behaviorismo* Metodológico considera que é possível direcionar respostas (*outputs*) por intermédio de entradas (*inputs*), sendo essa a única forma de extrair um tipo de desejado de comportamento de um indivíduo [49]. No entanto, o *Behaviorismo* Radical entende que existem outros tipos de ações não explicadas pela simples entradas de estímulos e espera de respostas na forma de ações reflexas [45]. Essas ações são, por natureza, mais complexas, não sendo apenas as reações involuntárias que ocorrem dado um determinado estímulo, mas compõem quase que a totalidade das ações que os seres humanos fazem. Carregam ainda uma incerteza em relação à sua manifestação, mas são calculadas com base na probabilidade ou não de acontecerem depois de o indivíduo que as produz ter sido, de alguma forma, recompensado pelo exercício de alguma ação anterior que ele tenha feito [45]. Por exemplo, se alguém conta uma piada e as pessoas ao seu redor riem dela, é mais provável que o comportamento de contar piadas volte a acontecer.

Em comum entre essas duas vertentes do *behaviorismo* está a busca pela compreensão do comportamento observável e manifesto, enfatizando as relações entre o comportamento e o ambiente. Ou seja, o foco do *behaviorismo* está no papel e na influência dos estímulos ambientais, na determinação das ações do indivíduo e longe das práticas de introspecção e estudo dos fenômenos mentais como forma de explicação das aprendizagens do indivíduo [49].

Quase na mesma época, a teoria da *Gestalt* surgiu para contrapor as premissas de que todo o aprendizado humano é explicado por meio de estudos de comportamentos [49]. Sua filosofia reside na ideia de que o que percebemos são relações e não sensações, afinal as partes, sob o risco de terem seu significado destruído, devem ser estudadas em conjunto e não isoladamente. Segundo a *Gestalt*, o todo é maior e também diferente do resultado da soma das partes [49]. Há, portanto, uma camada de interpretação que transforma o conjunto das partes nesse todo compreensível [50], o que inevitavelmente direciona o foco dos estudos da psicologia da aprendizagem para os aspectos mentais do indivíduo.

Essa maneira de pensar gerada pela *Gestalt*, aliada às contribuições da Teoria dos Esquemas desembocariam posteriormente em uma espécie de revolução cognitiva da psicologia [52]. Segundo a Teoria dos Esquemas as nossas experiências de vida são organizadas em esquemas cognitivos, que são inter relacionados entre si dentro de uma parte da memória (longo prazo), deixando a outra parte (memória de curto prazo) responsável por processar os pensamentos que são conscientes e imediatos [52]. Esse tipo de ponto de

vista sobre o que acontece entre o estímulo recebido e a resposta produzida pelo indivíduo em termos de comportamento deu base para a consideração da cognição como variável importante de processamento existente entre o estímulo e a resposta do indivíduo [49].

A discordância com o *modus operandi* essencialmente materialista e empirista de desconsideração de qualquer variável intermediária entre o estímulo e a resposta levou alguns psicólogos contemporâneos de Skinner a formular outro modelo. Edward Tolman, que estava entre eles, trouxe o *behaviorismo* Cognitivo, sendo um dos primeiros psicólogos a se interessar pelo estudo da mente em sua época [53], e Robert Gagné incluiu aos pressupostos behavioristas o estudo do que acontece dentro da cabeça do indivíduo que aprende, criando teorias sobre como a informação é processada [50], inspirado no modelo Atkinson-Shiffrin de funcionamento da mente, que a considera como se fosse uma espécie de computador que possui divisão de memória de longo e de curto prazo [54].

O crescimento do cognitivismo foi o que lançou as raízes para as teorias de aprendizagem baseadas no construtivismo. Elas inspiraram formulações como o “*learning by doing*” de John Dewey (ver seção 2.1.3) e o construcionismo (ver seção 2.1.4), que, como discutido anteriormente, influenciaram e trouxeram justificativas para algumas práticas usadas na Aprendizagem Baseada em Projetos [37]. A Subseção 2.1.2 discutirá alguns dos mais influentes teóricos do construtivismo e suas devidas contribuições.

2.1.2 Construtivismo

Para o construtivismo, tudo o que o indivíduo sabe sobre o mundo é resultado de seus pontos de vistas e este indivíduo é capaz de construir seu próprio conhecimento e não somente absorvê-lo [45]. No construtivismo, o indivíduo é educado pela interação com o ambiente, tirando seus aprendizados (ou seja, construindo significados) de cada experiência pela qual ele passa. Dessa forma, esse indivíduo (sujeito) modifica o ambiente (meio) e o ambiente, em contrapartida, o modifica também fazendo com que ele aprenda coisas novas [49].

Há pelo menos três vertentes do que se conhece hoje por construtivismo [47]. Dentre elas estão a psicogênese de Piaget [55], o socio-construtivismo de Vygotsky [56] e o construtivismo de Bruner [45].

A psicogênese defende que os sujeitos aprendam interagindo com aquilo que desejam compreender, sendo essa interação entre indivíduo e ambiente um dos principais fatores que determinam o desenvolvimento cognitivo do indivíduo [49]. Embora não seja a única epistemologia da psicologia construtivista aplicada ao aprendizado, o construtivismo piagetiano é, sem dúvida, uma das teorias mais influentes na área da educação em todo o mundo [45]. Ela, em conjunto com outras teorias de Piaget, supõe não haver estruturas cognitivas inatas, mas uma construção delas (ou seja, um movimento de passagem de um

estado anterior, que é menos desenvolvido, para um estado posterior mais desenvolvido [49]) para complementação de um processo de desenvolvimento cognitivo dividido em etapas [49]. Segundo essas teorias, o desenvolvimento cognitivo se orienta para o equilíbrio evolutivo, onde este indivíduo sempre busca se acomodar a novas situações que tenham desorganizado seus esquemas mentais anteriores [49]. Pode-se dizer que as pessoas se sentem melhores quando entendem o mundo que as cerca, sendo desequilibradas quando se deparam com algo que não conhecem. A esse fenômeno, dá-se o nome de *equilíbrio majorante*.

A assimilação e a acomodação dos novos conhecimentos são os dois movimentos possíveis dentro do processo de *equilíbrio majorante*. Na assimilação o indivíduo adquire uma fácil compreensão de um novo aprendizado por causa dos esquemas mentais que concordam com a nova informação e na acomodação há uma modificação dos esquemas prévios para que o indivíduo dê conta das novas informações com as quais está se deparando [55, 49]. Por meio desses mecanismos é possível trabalhar para aguçar a curiosidade dos alunos. Um exemplo disso, é quando o professor provê atividades que tenham como ponto de partida algo que já é conhecido pelo aluno (assimilação), mas prossegue com atividades que contenham elementos desconhecidos e desafiadores para o aluno, colocando-o em uma posição de busca de respostas (acomodação) [45, 49]. A admissão dos conceitos da pedagogia construtivista piagetiana inevitavelmente leva o professor a projetar atividades de pesquisa e de exploração, que são atividades onde os estudantes possuem papel ativo na busca pelo conhecimento [49].

O construtivismo de Bruner traz consigo a ideia do ensino estruturado, da aprendizagem em espiral e da aprendizagem por descoberta [45], três métodos pedagógicos com alguma influência na ABP. Eles ditam que o professor deve organizar os conteúdos em uma sequência que favoreça a aprendizagem do aluno, lançando mão de um encadeamento de estudo, pesquisa e avaliação das aprendizagens deles. Há, portanto, foco no bom planejamento da matriz de conteúdos levando em consideração a organização em pequenos passos, o enfoque pedagógico sobre os conceitos centrais de cada assunto [45] e a atuação ativa do aluno por intermédio de construção de hipóteses e realização de pesquisas motivado por sua curiosidade e auxiliado pelo professor [45].

Já o socio-construtivismo de Vygotsky tem como premissa básica a ideia de que o ser humano não traz consigo as aptidões naturais para determinadas aprendizagens através de sua herança biológica [45]. Para Vygotsky a aprendizagem é construída ao longo do tempo, sendo possível ao ser humano aprender qualquer coisa. Ao contrário da teoria psicogenética, em que há a sugestão de que as pessoas podem pensar de maneiras mais complexas de forma natural, o que inevitavelmente relega o papel social a uma posição secundária, a teoria socio-construtivista, que é influenciada pelo materialismo dialético

de Karl Marx, crê que é a interação social que torna as pessoas capazes de raciocinar de maneira cada vez mais complexa. [46, 57].

Nessa teoria, a cultura e os meios sociais são grandes depósitos de um conhecimento acumulado pela humanidade durante a sua existência e cabe ao ser humano aprendiz, com ajuda de outro ser humano mais experiente, se apropriar desse conhecimento e fazer uso dele conforme as regras e costumes sociais determinam que seja feito [58]. Existe um conhecimento do lado de fora impregnado na cultura, nos objetos, na forma como as pessoas vivem e quando alguém adota esse conhecimento, ele é reinterpretado com base na sua história de vida [46, 58, 59]. Por exemplo, um humano aprendiz, em determinado momento da sua vida, entra em contato com uma colher, mas esse ser humano apenas poderá fazer o uso correto dessa colher se outro ser humano mais experiente em seu uso lhe transmitir o conhecimento que possui sobre como usá-la, conhecimento esse que veio sendo transmitido geração por geração [46].

A Zona de Desenvolvimento Proximal é uma das inovações mais importantes trazidas pelo pensamento de Vygotsky [60]. A principal ideia contida nela é a de que existem coisas que o ser humano consegue fazer individualmente, coisas que ele pode aprender com a ajuda de outros e coisas que ele não pode aprender, nem mesmo com ajuda [46]. O que o ser humano já sabe fazer individualmente pertence à Zona de Desenvolvimento Real, enquanto que aquilo que ele pode aprender com ajuda de outros mais experientes pertence à Zona de Desenvolvimento Proximal. Aquilo que ainda é impossível ao aprendiz daquele ser humano se localiza fora da Zona de Desenvolvimento Proximal, devendo ser trazido para dentro dela para que possa ser aprendida com auxílio e finalmente possa passar para a Zona de Desenvolvimento Real. Neste sentido, a presença de um professor como sendo essa figura mais experiente é essencial no processo de ensino-aprendizagem sob a ótica do pensamento de Vygotsky [46, 60].

No construtivismo em geral (isto é, nessas três vertentes), o trabalho em equipe exerce importância crucial, em razão dos efeitos que a interação com o outro pode produzir nos estudantes. Ao trabalhar em grupo, os alunos discutem entre si suas próprias ideias e negociam pontos de vistas, promovendo a percepção de que existem muitas maneiras diferentes pelo qual um problema pode ser interpretado [49]. Além disso, os fatores que os ligam estão relacionados à disposição ativa do aluno para o aprendizado em detrimento da maneira passiva com que devem se comportar em aulas expositivas. Agir ativamente leva o estudante à interação com o meio e a construção de conhecimento, enquanto esperar pelo conhecimento transmitido apenas faz com que o aluno receba esse conhecimento, sem qualquer espécie de construção [47].

O aluno passa, então, a ser um indivíduo empoderado dentro de sala de aula. Isso significa que não cabe mais ao professor a exclusividade na transmissão do conhecimento

enquanto o aluno aguarda passivo pela sua retenção. Mudar esse ponto de vista permitiu que outras formulações epistemológicas trabalhassem com a ideia de um aluno ativo, que pode aprender pelo exercício de sua própria vontade e praticar conceitos teóricos abstratos por meio da construção de objetos de conhecimentos que podem ser visualizados pelos outros.

2.1.3 “*Learning by doing*”: o pragmatismo de John Dewey e o método de projeto de William Kilpatrick

Dewey, que é a referência intelectual da educação progressiva americana, acredita que a educação deve estar focada no estudante, embora não desmereça a importância dos conteúdos escolares ensinados por um adulto mais experiente. Postula, então, que a melhor forma de aprender é pela experiência, ou seja, aprender fazendo (“*learning by doing*”), opondo-se à aprendizagem rotineira baseada em memorização, cópia e repetição [61]. Ele é frequentemente referenciado como um dos teóricos que lançou as sementes da ABP [62, 63, 38, 64]. Neste sentido, suas principais contribuições ao modelo de ABP atual é a ideia de que a educação deve ser vivida e não justificada por propósitos imateriais e permanentes atribuídos à natureza humana [65].

Por ser um defensor aguerrido da democracia (que para Dewey é baseada em condições de participação igualitária em todas as esferas da sociedade e não em processos governamentais ritualísticos [66]), Dewey entendeu a educação como ferramenta importante para a construção do mundo democrático [66] e isso influenciou seu modo de pensar sobre como educação deveria ser conduzida. Para ele a sala de aula poderia servir como uma sociedade embriônica [65], o que fez com que sua proposta para a educação fosse a de transformá-la em um ambiente de construção de um mundo mais igualitário e de participação ativa dos alunos por intermédio de seus próprios interesses. Basicamente, influenciado pelo pragmatismo, ele entendia que conceitos filosóficos abstratos só teriam significado na medida em que pudessem ser utilizados de forma prática [66]. Repudiava, portanto, toda e qualquer idealização e crença em “-ismos” filosóficos (ou seja, em princípios absolutos justificadores de modos de vida ideológicos) [65].

É possível encontrar resquícios de suas teorias psicológicas sobre seu funcionalismo (uma reinterpretação do conceito de reação a estímulos presente no *behaviorismo* como coordenação de ações encadeadas) no movimento da escola progressiva [61]. Dewey indicava que ao estudar o comportamento não devemos dividi-lo em processos sensoriais, processos cerebrais e respostas motoras, mas ver esses elementos como um sistema integrado cujo objetivo final é a sobrevivência do organismo [67]. Na educação, isso implica levar em consideração o que a criança já tinha de bagagem, opondo-se à prática de tratar

a criança como página em branco onde seriam escritos os saberes da humanidade [61], entendimento compartilhado pelo construtivismo como discutido na seção 2.1.2.

Embora não sejam teorias de aprendizagem propriamente ditas, tanto o pragmatismo quanto o funcionalismo de Dewey viriam influenciar suas proposições pedagógicas [66] e a de outros autores importantes para a educação. William Kilpatrick, em especial, foi um dos que se sentiram muito atraídos pelas premissas da educação progressiva e para a democracia. A partir delas, mais especificamente, das teorias da experiência e educação, Kilpatrick desenvolveu o chamado método de projeto [11].

Embora Kilpatrick tenha popularizado a ideia de uso do projeto como ferramenta pedagógica, ele não foi o precursor da ideia. Há evidências de que a prática de projeto já existia no século XVI dentro do movimento de educação em arquitetura e engenharia na Itália, muito ligada ao ensino de profissões [68]. Mesmo nos Estados Unidos, haviam proposições anteriores à de Kilpatrick, como é o caso do “*home project plan*” de Rufus W. Stimson, onde as crianças eram incentivadas a cultivar feijões, cenouras e ervilhas nas fazendas de seus pais depois de terem sido apresentadas a conhecimento teóricos sobre vegetais [68]. A inovação de Kilpatrick foi em relação ao significado dado à palavra “projeto” dentro do contexto pedagógico. Para ele, o projeto se tratava de um ato intencional e isso significava dar total liberdade ao aluno dentro da sala de aula para resolver problemas práticos em situações sociais que fossem guiados por suas próprias motivações [68, 69].

Dessa maneira, pode-se dizer que haviam tipos distintos de projetos com as quais o professor podia trabalhar [69]. Esses podiam ir da construção de artefatos a partir de algum plano ou ideia seguindo processo composto pelas etapas de perspectivar, planejar, executar e avaliar até o aprendizado de vocabulários ao som de uma música clássica durante o pôr do sol [69, 68].

O uso de projeto na pedagogia foi um grande salto para a composição do método de ABP como a conhecemos hoje (mais detalhes na seção 2.1.5). Kilpatrick havia notado uma inadequação do modelo educacional americano de sua época em preparar os jovens para a vida adulta, principalmente pelo foco em aquisição de conhecimentos pré-formulados com base na memorização [69] e isso se relacionava profundamente com as ideias de Dewey. No entanto, Dewey criticou fortemente o método de Kilpatrick no que tange à diminuição da importância da figura do professor como um dos requisitos da execução do ato intencional [68]. Isso fez com que o método de projeto estancasse seu avanço nos Estados Unidos, embora tenha ganhando proeminência na Europa, principalmente por influência do comunismo [68].

2.1.4 Construcionismo: aprender por meio da construção de artefatos

A ideia do “*learning by doing*” pregada por Dewey é encontrada também na filosofia construcionista, grande influenciadora epistemológica do formato da ABP. Segundo o construcionismo (e dessa forma, bem próximo das proposições do construtivismo), o aprendiz aprende em interação direta com o ambiente, construindo objetos de conhecimento que podem ser submetidos ao julgamento, análise e crítica de outros. Tal semelhança não existe por acaso, afinal Seymour Papert, idealizador do construcionismo, era próximo de Piaget [21]. O construcionismo trouxe consigo uma metáfora sobre como os seres humanos aprendem com mais efetividade. Isso acontece através da construção de um modelo, de uma reflexão sobre ele, de sua depuração e de seu posterior compartilhamento. Tal exercício foi impulsionado pelo uso das tecnologias digitais e das linguagens de programação como ferramentas de construção [19, 20].

No entanto, o construcionismo afastou-se de Piaget no que tange à concepção sobre como o processo de aprendizagem por construção ocorre, partindo de um movimento de dentro para fora, surgindo da evolução mental dos indivíduos, para um movimento de fora para dentro, em um sentido mais situado, onde o indivíduo deve primeiro entrar em contato com as situações que geram o aprendizado para que depois o aprendizado aconteça [21].

Essa ideia de aprendizagem situada tem um papel muito importante dentro do modelo de aprendizagem construcionista chegando a justificar a necessidade de fazer com que os alunos sejam aprendizes construtores [21]. Assumir que a aprendizagem é situada significa assumir que o conhecimento agregado não é independente, mas um produto da cultura, atividade e contexto onde ele é desenvolvido [9]. Cria, portanto, um senso de autenticidade, por meio do qual o que o aluno faz em sala de aula se assemelha ao que o profissional faz no exercício de sua profissão [23]. Tratar a aprendizagem por meio desse delineamento é fundamentalmente o que a ABP faz e o que faz com que, tanto a maneira como ela deve ser conduzida, quanto os produtos de sua condução, sejam autênticos [26].

2.1.5 Aprendizagem Baseada em Projetos

A Aprendizagem Baseada em Projetos (ABP) não é um modelo tão coeso quanto a adoção de um termo que o define faz parecer que é. Há muitas abordagens que recebem o nome de Aprendizagem Baseada em Projetos, mas são muito diferentes entre si e há tantas outras que não levam o nome de Aprendizagem Baseada em Projetos, mas possuem muitas similaridades com o que a maioria das pessoas entendem por ABP [26]. Em uma definição

mais generalista, a Aprendizagem Baseada em Projetos é uma abordagem que usa a construção de projetos como ferramenta pedagógica [26, 70, 22].

Morgan delinea um conceito para projetos baseados na ABP por meio de três tipos: os *exercícios de projetos*, que são aqueles onde a centralidade do ensino está no professor, os *componentes de projeto* e *orientações ao projeto*, que são aqueles onde uma maior autonomia é dada ao aluno [38]. O principal representante do primeiro tipo (exercícios de projetos), são os projetos que comumente chamamos de Trabalho de Conclusão de Curso (TCC). Estes estão mais consolidados. No entanto, são os outros dois tipos (componentes de projeto e orientações ao projeto) que são tratados como pertencentes ao movimento progressista de educação fundamentado nos princípios construtivistas, que advoga por abordagens que sejam orientadas ao aluno e produzem as chamadas “habilidades para o século 21” [28, 27]. A diferença entre eles é que os componentes de projeto são um tipo interdisciplinar de projeto e a resolução de problemas que acontece em paralelo com outras disciplinas regulares e as orientações ao projeto são mais agressivas, afinal tratam-se de abordagens que envolvem toda a comunidade escolar/acadêmica [38].

Embora, Morgan argumente que haja três tipos de projetos, há uma corrente de trabalhos que tende a fixar limites para a prática da ABP estabelecendo características necessárias para que uma unidade de ensino se encaixe nessa definição. Por exemplo, as revisões de Thomas [26] e Condliffe et al. [27] reúnem uma série de características que delimitam um escopo do que poderia ser considerado projeto com base em muitos outros trabalhos que já se debruçaram sobre o assunto. Segundo estes dois trabalhos, uma abordagem para ser considerada como ABP deve seguir os seguintes critérios:

- Ter projetos que são centrais ao plano de ensino, não periféricos;
- Ser guiada por questões motrizes que podem motivar os alunos enquanto eles se envolvem na descoberta de conceitos centrais e princípios de uma determinada disciplina;
- Permitir que os alunos trabalhem em uma investigação construtivista para transformação e construção de conhecimento;
- Não se basear em projetos que são “empacotados” pelos professores, ao passo que mais poder de escolha seja dado ao aluno;
- Trabalhar com projetos ligados à questões do mundo real;
- Prover um ambiente propício à colaboração;
- Ser feita com base em uma estratégia de andaimes (*scaffolding*), partindo de mais apoio a menos apoio aos alunos ao longo do tempo;

- Ter como alvo objetivos de aprendizagem significantes, que ensinem os alunos habilidades úteis para sua vida profissional, mas não percam de vista os conteúdos ementários.

A centralidade do projeto é um dos principais limites que determinam se uma abordagem pedagógica centrada no aluno poderá ser chamada de ABP ou não [38]. E, como se não bastasse a porção de características que um projeto deve ter para que a unidade letiva no qual ele está inserido possa ser chamada de ABP, o papel do professor também é colocado como um aspecto importante nessa caracterização. Em virtude da resignificação do termo “projeto” por Kilpatrick [68], o poder de escolha e a motivação do aluno passaram a serem vistos como molas propulsoras para a aprendizagem por meio do “fazer” porque são elas que ativam a motivação dos alunos para o aprendizado. Por causa disso, o papel de um professor dentro de uma unidade ABP não deve ser o de um expositor de conteúdos, mas o de um facilitador, tutor ou *coach* que guia o aluno por meio das já referidas técnicas de *scaffolding* [35, 71].

Sem essas características, a execução do “aprender fazendo” dentro de uma sala de aula pode ser chamado apenas de execução de tarefas e não de projetos educacionais [68, 28, 72, 73, 74, 70, 22, 36, 75], embora não haja uma veemência no como devem (ou se devem) ser executadas estratégias especiais de *scaffolding*, avaliação, autonomia dos alunos e trabalho em grupo [27].

A grande questão é que nem todas as abordagens que apresentam essas características podem ser exclusivamente chamadas de ABP [26]. Por conta disso, passados tantos anos desde que o método de projeto foi formalizado, ainda há debates sobre o que de fato é a Aprendizagem Baseada em Projetos e o que não é. Há muita semelhança entre a ABP e a Aprendizagem Baseada em Problemas, a Aprendizagem Expedicionária, a Aprendizagem Situada, Aprendizagem Experiencial, Aprendizagem Intencional, *Design Experiments*, Aprendizagem Baseada no Trabalho e Aprendizagem de Serviço [22, 26]. Porque estas abordagens partem dos mesmos princípios psicopedagógicos de aprendizagem pragmática [10, 68, 56, 76, 22], as separatrizes entre a ABP e essas outras abordagens não são muito nítidas. Portanto, é necessário que algo separe a ABP das outras abordagens, ou pelo menos leve a entender que algumas abordagens são apenas a ABP com outro nome. Dos pontos estabelecidos por Thomas [26], o que certamente mais ajuda a constituir essa linha de separação é a forte presença das ideias do construcionismo na ABP [70]. O foco de uma ABP está no desenvolvimento das habilidades sociais, cognitivas e metacognitivas por intermédio de tarefas de construção de artefatos e não somente da imersão dos alunos nos ambientes autênticos da prática profissional. É certo que abordagens como a Aprendizagem de Serviço e a Aprendizagem Baseada no Trabalho também permitem

a construção de artefatos, mas pelo menos é possível encontrar nelas semelhanças que permitam aceitar que elas são abordagens que também podem ser chamadas de ABP.

A Construção de artefatos dentro de uma unidade de ABP

Os pontos em comum reunidos na maioria das abordagens que chamamos de ABP torna possível a construção de modelos genéricos que sirvam de um plano de execução para que os alunos construam artefatos dentro de uma unidade pedagógica. Essas abordagens são balizadas por *práticas de investigação* feitas pelos estudantes, por meio de *colaboração* entre eles e pelo *uso de tecnologias novas* para que seja possível responder *questões motrizes* [77]. A partir disso, os alunos criarão artefatos autênticos que representem o entendimento deles [77]. Embora haja inúmeras versões de ABP (ou de abordagens que poderiam ser chamadas de ABP), estes são pilares-chaves de unidades de ABP já registrados pela literatura, onde há mais consenso [27]. Os pilares são :

- **Questão motriz:** Os projetos educacionais são essencialmente diferentes das outras atividades escolares por causa de uma questão que guia a investigação [70, 28]. A presença de uma questão motriz é o motivo pelo qual as unidades de ABP não são desconectadas dos contextos de mundo real [77] e o motivo pelo qual o trabalho executado pelos alunos possui significado para além da sala de aula [70]. As questões motrizes devem, na medida do possível, ser escolhidas pelos próprios alunos, pois, o método do projeto de Kilpatrick pressupõe que a liberdade dos alunos beneficia seus processos de aprendizagem [70, 28]. Ter uma questão motriz é um fator de muita importância para a aplicação bem sucedida de uma ABP. Os alunos que foram submetidos à abordagem guiada por uma questão motriz foram mais capazes de formular suas próprias questões, além de entenderem melhor o que eles próprios estavam aprendendo e saberem usar com mais eficiência os métodos científicos [64]. Há três características essenciais para que uma questão seja considerada uma boa questão motriz de uma unidade de ABP. Em primeiro lugar, ela precisa conduzir os alunos ao aprendizado dos conteúdos dos padrões curriculares existentes. Ela também precisa englobar um problema do mundo real que seja significativa para os alunos, a fim de que eles sejam motivados para trabalhar nela. E, por último, as questões devem ser passíveis de serem respondidas por meio das investigações dos alunos [77].
- **Investigações:** As investigações são estimuladas pela questão motriz e fornecem as respostas para ela em um movimento de retroalimentação [77]. No geral, elas são inspiradas em modelos práticos de trabalhos dos profissionais que buscam conhecimento sobre alguma coisa no ato de resolução de algum problema [23, 77]. Buscar

respostas à questão por meio destas investigações levam os alunos a organizarem e elaborarem seus pensamentos na intenção de respondê-las [77], além de permitir que eles adquiram habilidades específicas de processo e passem por experiências cognitivas mais profundas [70]. Algo da natureza das investigações que as diferenciam das tarefas de laboratórios, é que nas investigações as questões são abertas e os passos não são predefinidos anteriormente. Nestas características, reside o poder que a investigação possui de pôr os alunos em situações de desafio [77].

- **Ferramentas tecnológicas:** A tecnologia web é, na opinião de muitos estudiosos, uma ferramenta chave na ABP [70], também importante para os métodos fundamentados na epistemologia construcionista. As ferramentas tecnológicas apoiam os alunos na condução da investigação e também na produção dos artefatos [77]. Com acesso à Internet, os alunos têm diante de si uma vasta biblioteca de informações que os ajudarão a aprender conceitos e métodos novos em sua tarefa de pesquisa. É possível também, que a internet suporte a interação dos alunos com alunos de outros lugares promovendo, assim, um aprendizado social e colaborativo [64]. Além disso, os computadores permitem que artefatos digitais sejam criados, entre os quais podem estar, inclusive, os protótipos resultantes da investigação dos alunos. Os computadores tornam a confecção de artefatos mais barata e possibilitam que os artefatos sejam colocados na internet para benefício de outros.
- **Colaboração:** Fazer projetos, conduzir investigações e também aprender é uma atividade social. Um dos grandes desafios que os alunos enfrentarão na vida profissional é lidar com grupos e realizar trabalhos dentro deles. Estar em grupo exige muitas habilidades que precisam ser desenvolvidas como, por exemplo, negociar, trabalhar em equipe, discutir ideias, entre outros [64]. A colaboração é um aspecto tão importante em uma unidade de ABP que deve ser explicitamente incentivada pelos professores [77].

Em virtude da crucialidade da construção de artefatos autênticos para a caracterização de uma ABP, este trabalho se propõe a estudá-la. A seção 2.1.6 irá detalhar o significado do que é ser autêntico dentro do escopo específico da ABP, de modo a dar a fundamentação necessária à conexão conceitual da ABP com a Experiência do Usuário (UX) e discutir como é possível reforçar a autenticidade dos projetos de software executados em sala de aula ao unir seus conceitos fundamentais.

2.1.6 A autenticidade na aprendizagem

Embora o conceito de autenticidade seja de interesse da psicologia e da educação, ele ainda é um conceito difícil de se definir e até controverso [24, 23]. Archibald e Newmann trazem

à tona a discussão sobre aprendizagem autêntica e, inclusive, definem critérios para que ela seja classificada como tal [78, 79]. Em seu trabalho, Newmann define três parâmetros para a classificação da autenticidade: (1) a construção de conhecimento, (2) a investigação e (3) uma similaridade com o que acontece no mundo de fora da escola [78, 79]. De acordo com a filosofia construtivista, a pedagogia autêntica de Newmann dita que os alunos devem utilizar seu conhecimento prévio para se envolver com problemas, tarefas e desafios que se conectam com o mundo além da sala de aula. Significa que a aprendizagem autêntica parte de problemas autênticos e um problema autêntico é um problema fundamentado em um contexto do mundo real, uma concepção que ele detinha que influenciou outros que vieram depois dele [63, 70, 70, 22, 28].

Essa associação do termo autenticidade a atividades do mundo real ou alguma demonstração prática é uma percepção comum [24]. No entanto, o mero fato de se apoiar em contextos do mundo real (ou de fora da escola) é criticado por Splitter [23], que entende que a dicotomia entre o mundo de fora da escola e de dentro da escola é um erro, ao passo que, se apoiando na escola progressiva de Dewey, afirma que o processo de aprendizado de um aluno é sequencial. Sua defesa final é a de que a autenticidade do processo de aprendizado reside não no que os outros fazem fora da escola, mas no que deveriam fazer. Embora Splitter critique a transformação do termo autenticidade em um definidor exclusivo da prática de fazer coisas que sejam similares àquelas de fora da escola, ele faz coro à construção ideológica de Newmann sobre autenticidade e suas relações com as práticas exemplares do mundo do trabalho, desde que não relegue a escola a uma posição de separação do que as pessoas costumam chamar de mundo real [23].

Dado que a autenticidade dentro das práticas pedagógicas se relaciona com epistemologia da aprendizagem situada [26], é recomendável que a instrução seja realizada em um contexto de resolução de problemas, tal como preconizam as teorias da *Gestalt* [26]. Além disso, sendo a ABP um modelo construtivista [36], a autenticidade exerce nela uma grande importância. É por meio da autenticidade que se argumenta que é necessário que os alunos dominem as técnicas e ferramentas dos profissionais da área para o qual estão se preparando, desde que a realização das atividades até a construção de objetos de conhecimento [24].

Os projetos desenvolvidos em uma unidade de ABP incorporam várias características que são responsáveis pela definição de autenticidade que damos a eles. Estas características podem incluir o tema, as tarefas, os papéis que os alunos desempenham, o contexto em que o trabalho do projeto é realizado e os colaboradores que trabalham com os alunos no projeto [26]. E do ponto de vista da qualidade do que os alunos constroem, a autenticidade também reside nos próprios artefatos que são produzidos, no público para os produtos do projeto e até nos critérios pelos quais os produtos são julgados [26].

Na busca filosófica de Splitter sobre uma definição adequada para a autenticidade, ele afirma que o que é produzido no contexto pedagógico pode ser considerado autêntico se comparado a um arquétipo (um conjunto mínimo de características) de um exemplar original [23]. Isso significa que o que conta como autêntico para uma pessoa pode estar longe de ser autêntico para outra [23]. Partindo desse pressuposto, é possível argumentar que a autenticidade não está alicerçada na simples produção de algo, mas na produção de algo útil, adequado ao conjunto de características mínimas definidoras de algum objeto de referência criado por especialistas profissionais [80]. E neste sentido, é usar os problemas de outras pessoas como pontos de referência para diminuição dos efeitos da subjetividade. Ou seja, para que um artefato seja considerado autêntico, deve haver nele algum valor estético, utilitário ou pessoal para o público que receberá este artefato para uso [26, 23].

Uma das práticas da ABP que reforça o entendimento de que o que é produzido pelos alunos deve transcender a utilidade meramente didática, é o fato de que os artefatos construídos em unidades de ABP devem ser submetidos à crítica externa [70, 64, 62, 24], característica inerente à epistemologia construcionista [21]. Quando a transferência do sucesso de uma unidade autêntica de aprendizagem se traduz na disponibilização dos artefatos para uso da comunidade externa, até essa comunidade se beneficia ao receber um objeto útil e valioso para uso e mitigação de suas dores. Então, o aluno começa a vivenciar a experiência de ser útil e a entender que o papel profissional, em última instância, é o de sanar as dores de outros. A ideia de que o artefato gerado pela unidade de ABP pode resolver problemas reais causa efeitos positivos na motivação dos alunos, que é um dos elementos que, se não forem bem trabalhados durante a execução dos projetos, pode levá-los ao fracasso [70]. E, dessa maneira, como criar condições nas quais as questões que os alunos buscam e os artefatos que eles criam não sejam do tipo de “trabalhos de escola” é uma questão que merece ser explorada [81].

Este trabalho mantém o foco da exploração da autenticidade no objeto de conhecimento construído em uma unidade de ABP para o ensino do desenvolvimento de software e não no processo ou na similaridade da prática (ou seja, nos modelos de engenharia de software) para construção do objeto, embora este último possua uma parcela grande de efeitos na qualidade do que é produzido, como discutido nos resultados da Revisão Sistemática da Literatura descrita na seção 3. Os problemas residem no mundo real e as pessoas sofrem com ele. Tentar resolvê-los não deve significar trazer mais problemas para estes usuários, mas levá-los a ter uma experiência recompensadora de uso.

Em busca de uma visão mais objetiva do que pode vir a ser a autenticidade em um software produzido por meio da ABP, destaca-se os requisitos de Experiência do Usuário (UX), entendendo que um artefato para ser útil deve ser bem recebido pelo usuário e usado [8]. Neste contexto, a próxima seção levanta uma discussão sobre como os valores de UX

também são importantes para os artefatos de software produzidos dentro das unidades de ABP.

2.2 Experiência do Usuário (UX)

A experiência do usuário é comumente confundida com a usabilidade. No entanto, embora usabilidade e UX possuam objetivos semelhantes, elas não são, a rigor, exatamente iguais [82]. Enquanto a usabilidade se preocupa principalmente com os aspectos pragmáticos de uma interface de usuário e em como essa interface deve ser desenvolvida para alcançar determinados objetivos [83, 84], a UX extrapola esse escopo e se preocupa com variáveis hedônicas (felicidade, alegria, satisfação, etc.) do usuário no contexto de uso de determinados serviços ou produtos [85, 84, 82, 86]. A UX é vista como uma disciplina separada e mais abrangente que a Interação Humano Computador (IHC) [87], e que extrapola a preocupação com o alcance de objetivos comportamentais para considerar quais tipos de efeitos podem ser alcançados a partir da interação do usuário com o produto interativo [88, 87]. Em termos simples, a UX direciona seus esforços de criação de experiências (que pode ser por meio de pessoas, objetos, lugares, etc.) para os produtos interativos [8].

A base teórica da UX é sustentada por conceitos de áreas como a antropologia cultural, engenharia, jornalismo, negócios, psicologia e *design* gráfico, além dos conceitos importados da IHC e da usabilidade [29, 89], de forma que o resultado da interação de um usuário com uma aplicação resulte em uma experiência positiva [82]. Desta forma, um praticante de UX busca alinhar os objetivos de negócio de qualquer empreendimento (que normalmente giram em torno de motivações econômicas) com as necessidades do usuário [29].

A UX pode ser dividida em duas áreas, pesquisa em UX e *design* de UX [29]. A pesquisa em UX é a sub-área que se preocupa fundamentalmente com as necessidades, objetivo e motivação dos usuários, buscando entendê-las, modelá-las e fornecer informações para o desenvolvimento do produto, enquanto que o *design* de UX envolve o projeto de um produto ou serviço, sejam eles completos ou apenas pedaços de um produto ou serviço maior [29].

Uma das crenças fundacionais do *design* e da pesquisa de UX é a de que os desenvolvedores de produtos/serviços precisam aprender aos poucos com os próprios usuários por meio de um ciclo contínuo de construção, medição e aprendizado [90]. Desta crença se extraem alguns princípios que ajudam a delimitar do que se trata o *design* fundamentado em UX. Um deles é o de que a experiência do usuário é inevitável, onde quer que o usuário esteja e o que quer que ele esteja fazendo, de alguma forma, ele estará vivenciando uma experiência, seja ela boa ou ruim. Portanto, é importante que a experiência desse usuário

seja desenvolvida de forma intencional, ao passo que deixar que ela seja criada de forma acidental pode ser um motivo para o abandono de determinado serviço ou produto por parte dos usuários [29].

Outro princípio diz respeito à própria natureza do que é definido como usuário dentro do contexto do desenvolvimento de um produto. O usuário, para a UX, é alguém que terá uma experiência. Mesmo que os criadores do produto sejam pessoas que executem atividades similares às dos usuários que pertencem ao público-alvo, ele não deve desenvolver a solução com base em suas próprias necessidades. Os *designers* possuem informações técnicas sobre o projeto e o funcionamento do produto ou serviço que o usuário final não tem, eles também não poderão se encaixar nas mesmas limitações e expectativas do usuário, tornando a definição de requisitos de UX irrealis. O usuário está em uma jornada que não é compartilhada pelos membros do time de desenvolvimento [29], significando que há uma série de etapas que precisam ser vencidas pelo usuário para que ele chegue ao final do caminho e tenha seus objetivos alcançados por meio do serviço ou produto desenvolvido. Qualquer que seja a distração, dificuldade ou ruído na experiência deste em relação àquele produto ou serviço pode fazer com que ele abandone a jornada [29].

A criação de dogmas e a assunção de certezas baseadas em suposições é um dos maiores inimigos da pesquisa em UX, ao passo que a filosofia por trás da UX é sustentada na validação de hipóteses mediante o confronto delas com o que acontece no mundo real (ou seja, como de fato os usuários são, o que de fato pensam e o que de fato querem) [90]. Os desenvolvedores de soluções, como seres humanos que são, possuem muitos vieses e tratar esses vieses como sendo verdades absolutas sem as devidas validações elevam muito o risco da solução desenvolvida não satisfazer o conjunto de usuários que ela deveria atender e, por consequência, caírem em desuso [29, 90].

Levando em consideração que os artefatos criados por alunos em um contexto de ABP irá competir com produtos feitos por profissionais formados que são, por convenção, visto como mais competentes na atividades de desenvolvimento de sistemas, projetar para que seus usuários tenham uma experiência de uso pode dar-lhes alguma vantagem e aumentar a chance de serem recebidos pela comunidade para o qual esse artefato foi desenvolvido. Partindo dessa premissa, julga-se pertinente a observação de aspectos de experiência de usuário dentro da característica de autenticidade de um determinado artefato produzido em sala de aula. Chegar a este objetivo é, portanto, condicionado a ter uma boa definição do que é um requisito de experiência de usuário e, além disso, saber como avaliá-lo.

2.2.1 Os artefatos e sua capacidade de prover experiência

É importante ter em mente uma definição clara do que é experiência para especificação da experiência do usuário como um requisito a ser alcançado. Em suma, a experiência é um

acontecimento episódico que pode ter influência em seus momentos anteriores e posteriores [89]. Normalmente, considera-se a experiência como tal depois de seu acontecimento. Neste instante, a essa experiência dar-se-à um nome e determina-se seu começo e fim [89]. É possível também recordar-se dela e de todas as coisas que as cercam e que lhes dão significado (sons, visões, cheiros, sensações, pensamentos, etc.). [8].

McCarthy e Wright, referindo-se a John Dewey, define a emoção como o principal elemento subjacente à experiência [91]. Hassenzahl e Carroll [8] reforçam essa afirmação, acrescentando que a verdade acerca disso reside no fato de que a emoção, a cognição, a motivação e a ação estão intrinsecamente interlaçadas [8].

As experiências são fenômenos ao mesmo tempo únicos e emergentes dos elementos que a compõem [8]. Elas compreendem tanto aspectos fisiológicos, quanto culturais subsidiados pelo conhecimento que o indivíduo tem sobre o mundo e são construtivistas no sentido de que são capazes de alterar esse conhecimento e as percepções do indivíduo depois de ter acontecido [8]. Embora, McCarthy e Wright recorram ao funcionalismo e pragmatismo de Dewey para advogar em favor da indivisibilidade e imprevisibilidade da experiência e da inutilidade do estudo de suas partes separadas [91], Hassenzahl e Carroll recorrem a Russell [92], para classificá-las como um meta-processo emocional e argumentar que é possível tanto analisar a experiência como um todo irreduzível, quanto cada um de seus elementos, sendo esse o trabalho do *designer* de UX. Pode até não ser possível ao *designer* definir a forma e organização dos elementos de um produto interativo para que se garanta que alguma experiência vá acontecer, mas é possível trabalhar para que a probabilidade de que ela aconteça aumente [8].

Outras formas de conceber a experiência são apresentadas por Forlizzi e Battarbe [89], e uma delas é a concepção da experiência quando ela acontece. Os autores chamam esse instante de “*experiencing*” e o definem como um fluxo de pensamentos e sentimentos que se tem sobre o estado atual das coisas que se vivencia (uma espécie de auto-conversa durante a experiência) [89, 8]. Outra é a concepção da experiência compartilhada, o que é chamado de co-experiência pelos autores, onde de acordo com os autores, a experiência acontece de maneira compartilhada entre indivíduos e possui um aspecto social, onde os indivíduos influenciam uns aos outros em relação à percepção que terão da experiência [89].

Todas essas considerações são importantes para a especificação de requisitos específicos que tratam da experiência do usuário em uma determinada solução. No entanto, é importante considerar que a tecnologia (produto ou serviço desenvolvido) não é a experiência em si, mas o meio de levá-la ao indivíduo. A experiência se tornará experiência do usuário (UX) quando o foco estiver na produção do produto interativo que entrega aquela experiência [8].

Forlizzi e Battarbee [89] agruparam os modelos teóricos sobre a experiência do usuário em três categorias: os modelos centrados no produto, os modelos centrados no usuário e os modelos centrados na interação [89]. Os modelos centrados no produto dão suporte direto ao grupo de *designers* e desenvolvedores de produto na tarefa de desenvolvimento, geralmente providenciando listas de verificação de aspectos importantes para o desenvolvimento de um produto que faça emergir experiências positivas de seu usuário. Os modelos centrados no usuário ajudam os desenvolvedores e *designers* a entender o usuário e como ele se sente, perceber quais são suas expectativas e o que esperam do uso de um produto ou serviço. Finalmente, os modelos centrados na interação ajudam os *designers* e desenvolvedores a explorarem a interação do produto que desenvolvem com o usuário que o usa e extrai significado daquela interação [89].

Para a UX, há também um agrupamento dos instantes em que se deseja intervir ao projetar uma experiência [82]. Além do instante do “*experiencing*” (UX momentâneo), são também relevantes os instantes anteriores, posteriores e a acumulação de experiências ao longo do tempo. Os instantes anteriores (UX antecipada) são onde acontecem as antecipações e as expectativas formadas com relação ao produto ou serviço desenvolvido. Os instantes posteriores (UX episódica) são quando a “*experiencing*” se torna uma experiência de fato e é retomada por intermédio das memórias do usuário. E, finalmente, a acumulação de experiências ao longo do tempo (UX cumulativa) é o resultado de uma série de episódios de uso e não uso do produto ou serviço que moldam a percepção do usuário [82].

Uma maneira, portanto, de tratar a elicitación de requisitos de UX de maneira intencional é considerar experiências e aspectos afetivos do usuário como objetivos da solução que se deseja atingir, bem como o instante em que se deseja que a experiência exerça maior impacto [88, 86, 93, 94]. Em termos gerais, os requisitos de experiência de usuário podem ser alocados debaixo do guarda-chuva dos requisitos não funcionais [82, 94]. No entanto, embora requisitos não funcionais como disponibilidade, performance, usabilidade, segurança e confiabilidade afetem a experiência do usuário, a elicitación deles muitas vezes não é feita intencionalmente com o propósito de aumentar a experiência ou afetar positivamente o estado afetivo do usuário.

Callele et al. [94] propõem um processo de elicitación de requisitos baseado na interação, que ele chama de estímulo-percepção-resposta. Neste processo, a coleta de requisitos de experiência é feita por meio da especificação objetiva de uma resposta do usuário desejada e posterior desenvolvimento de estímulos providos pela aplicação que sejam percebidos pelo usuário, de forma que o leve a gerar a resposta anteriormente definida. Dessa forma, o desenvolvedor consegue afetar o usuário por meio de suas emoções, de seu intelecto ou de seus sentidos [94]. Os autores argumentam ainda que elicitación dos requisitos de

experiência pode prover uma série de benefícios, tais como: (1) identificar elementos de software ou de mídia que devem ser implementados pelo time de produção; (2) priorizar o desenvolvimento de elementos que são críticos na interação do usuário; e (3) facilitar a identificação dos planos de testes que são críticos [94].

Bentley et al. [93] baseando-se na característica do produto, argumentam em favor do uso do fluxo como característica de um produto ou serviço que seja agradável ao uso. A teoria do fluxo, proposta por Csikszentmihaly [95], descreve que o fluxo é a sensação holística que pessoas sentem quando eles agem com total envolvimento. No estado de fluxo, as pessoas continuam a executar uma atividade sem nenhuma recompensa externa, apenas pelo fato de estarem sendo recompensados internamente e pelo fato de estarem na atividade [93, 95]. As características que provocam o efeito de fluxo precisam que o usuário [95]:

- Se sinta capaz de completar a tarefa;
- Deva ter a habilidade de se concentrar na tarefa;
- Reconheça claramente os objetivos da tarefa;
- Receba *feedback* imediato sobre sua performance na execução da tarefa;
- Seja removida das frustrações e das preocupações do mundo exterior;
- Tenha um senso de controle sobre suas ações;
- Perca a consciência de si próprio, mas tenha uma auto-imagem mais forte depois de ter realizado a tarefa; e
- Tenha seu senso de tempo alterado, de forma que horas pareçam minutos.

Algumas dessas heurísticas, ou todas, a depender da solução a ser desenvolvida, podem ser aplicadas como requisitos de experiência de usuário, uma vez que esses elementos quando combinados em uma simples atividade podem levar o usuário a um estado de prazer tão gratificante (isto é, boa experiência) que farão com que eles estejam dispostos a gastar energia pelo simples fato de realizá-la [93].

Dado que o objetivo da Usabilidade e da UX são diferentes, a forma de avaliar os requisitos derivados delas se constitui como outro aspecto que as diferencia. Na usabilidade, as tarefas de medição buscam essencialmente resultados objetivos e pragmáticos, o que, na maioria dos casos não é verdade dentro da avaliação de UX [84]. A usabilidade se preocupa com resultados ligados à performance e ao tempo de sucesso do usuário em determinadas tarefas, enquanto o foco da UX é mais subjetivo, se importando com valores que evidenciem a satisfação e a felicidade do usuário [82].

O trabalho apresentado por Vermeeren [84] presta um grande serviço para a comunidade científica e para os praticantes, sistematizando 96 métodos de avaliação de UX com características e objetivos diversos [84]. Neste conjunto, é possível encontrar métodos que usam usuários e especialistas como fontes de informação para coletas de dados quantitativos ou qualitativos e que podem ser executados no campo, em um laboratório ou através da internet, por meio de produtos totalmente funcionais, protótipos ou ideias conceituais em fases iniciais de *design* [84].

A avaliação da experiência do usuário serve para apoiar a escolha de um bom *design*, para garantir que o desenvolvimento do produto ou serviço está no caminho certo ou para medir e avaliar o produto final e os objetivos de UX originalmente definidos, o que, por consequência, permitirá que os *designers* compreendam e tenham uma visão sobre como os usuários percebem e valorizam os produtos [82]. Construir produtos ou serviços em consonância com os valores da UX que se preocupa com requisitos que visem a experiência do usuário exige uma mudança na forma de pensar que parte do pensamento orientado ao desenvolvimento do produto para o orientado ao desenvolvimento do cliente.

2.2.2 *Design Thinking* e *Design Sprint* como processos de UX

Muito do que se propaga por meio da cultura de UX se baseia nos princípios de empatia contidos no *Design Thinking*. Embora o termo tenha se tornado frequente na literatura nos últimos anos, as pesquisas sobre a forma como o *designer* pensa não é nova [96]. O *Design Thinking* se propõe a criar produtos que equilibram as necessidades dos indivíduos e as necessidades da sociedade como um todo, com um processo que é acessível e que faz uso da habilidade das pessoas para reconhecer padrões e construir ideias que possuam tanto apelo emocional quanto utilidade [97]. Ele o faz usando esboços, protótipos e participação do usuário final como co-criador da solução [96].

Pressman [98] delinea o *Design Thinking* através de uma série de características. De acordo com ele [98], o *Design Thinking* é:

- Um processo que resulta em plano de ação para melhorar alguma situação;
- Uma habilidade que incorpora conscientização situacional e empatia dentro do processo de geração de ideias;
- Uma ferramenta que usa o pensamento analítico e criativo para resolver problemas que consideram contexto, questões logísticas, custo e requisitos e preferência dos *stakeholders*;

- Um *mindset* no qual as ideias são originadas de fontes diversas e, às vezes, discrepantes e são então construídas a fim de gerar progressivamente melhores soluções aos desafios correspondentes;
- Uma série de ações que são estruturadas por um ciclo onde os problemas são definidos, pesquisa e análise são conduzidas, e ideias são propostas e, em seguida, submetidos a realimentação e modificações críticas, que por sua vez, levam à repetição de partes do ciclo que ajudem a refinar ainda mais as ideias.

De acordo com Brown [97], o *Design Thinking* é a aceitação disposta e entusiástica de restrições concorrentes. Para análise dessas restrições, é necessário verificar qual funcionalidade é possível ser construída dentro de um futuro previsível, qual tipo de construção é passível de se tornar parte de um modelo de negócio sustentável e o que irá fazer sentido para as pessoas [97].

Diferente de outras abordagens de resolução de problemas, o *Design Thinking* não é um algoritmo que busca por uma resposta simples e correta, mas aproxima-se mais das abordagens qualitativas das ciências sociais, que consideram múltiplas respostas dentro das possíveis, nenhuma delas sendo a resposta perfeita. A coleta de informações através de análise de contexto e de *stakeholders*, característica marcante das abordagens sociais, é essencial ao *Design Thinking* [98], o que faz com que o *Design Thinking* seja fundamentalmente um processo de exploração [97].

O processo de trabalho do *Design Thinking* procura não ser prescritivo. Nele há um entendimento de que a abordagem prescritiva de processo, conforme definido no modelo cascata da engenharia de software [99], desencoraja a inovação e fornece obstáculos às perspectivas pessoais. A forma como ele é organizado faz com que os blocos lógicos contenham objetivos coesos, mas não determine exatamente como esses objetivos serão alcançados, além de fazer com que eles aconteçam de maneira cíclica, de modo que, a cada nova iteração, se avance no aprendizado que a equipe tem sobre o contexto e as pessoas para quem se destina a solução [97, 98].

Não há consenso na nomeação desses blocos [96], mas muitos autores a diagramam em um modelo chamado de Duplo Diamante ou modelo 4D, conforme apresentado na Figura 2.1, originado do Conselho de *Design* Britânico [1]. Cada bloco lógico do Duplo Diamante representa um verbo ligado a um objetivo particular [1, 96]:

1. **Descubra:** O primeiro bloco do modelo cobre o início do projeto, onde os *designers* buscam olhar para o mundo de uma forma não enviesada, percebendo coisas novas e reunindo *insights*.
2. **Defina:** O segundo bloco representa a fase de definição, onde os *designers* buscam dar sentido a todas as possibilidades identificadas na fase Descubra. Perguntas como

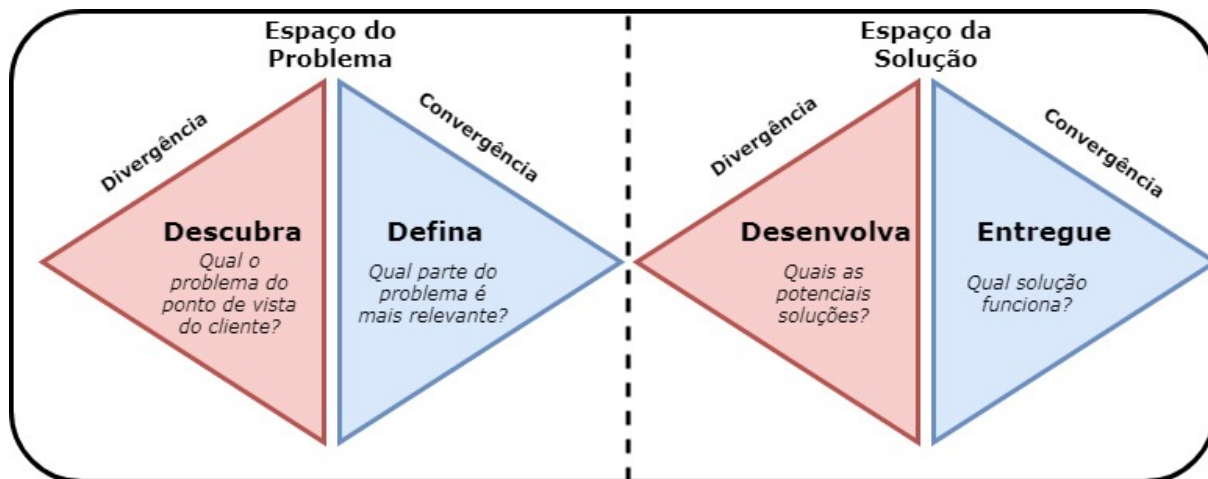


Figura 2.1: Duplo Diamante. Adaptado de [1]

“O que mais importa?” “Em qual parte do problema devemos agir primeiro?” “O que é viável?” são respondidas com o objetivo de criar impulsos criativos focados no desafio fundamental do *design*.

3. **Desenvolva:** O terceiro bloco representa uma fase de construção, onde soluções ou conceitos são criados, prototipados, testados e iterados com base em um processo de tentativa e erro que ajuda os *designers* a melhorarem e refinarem as suas ideias.
4. **Entregue (*Deliver*):** O último bloco do modelo corresponde à fase de entrega, onde o projeto resultante é finalizado, produzido e lançado.

Há uma diferença entre ser um *designer* e pensar como um *designer* [97]. Isso permite que o *Design Thinking* coloque as ferramentas de um *designer* nas mãos de pessoas comuns para que elas as apliquem em um vasto número de problemas [97]. Brown afirma que a natural evolução do *design doing* para o *design thinking* reflete o reconhecimento crescente por parte dos líderes atuais de negócio de que o *design* tornou-se importante demais para ser deixado nas mãos dos *designers* [97].

Essa maneira de pensar leva a crer que mesmo os estudantes de cursos de desenvolvimento de sistemas acostumados com o foco na construção de código podem se beneficiar do *Design Thinking* como paradigma de pensamento e construção. No entanto, o tempo disponível para o exercício desse paradigma em sala de aula se demonstra limitante, dada a natureza dinâmica da criação de um produto que coloca o usuário como centro do processo e se baseia na natureza cíclica da engenharia ágil. Blank [100] combina a engenharia ágil a um método de criação de soluções com uma abordagem centrada no cliente, relata enfaticamente a limitação de uma abordagem que determina uma data de lançamento para o produto [100]. A orientação à data de lançamento de um artefato é o que naturalmente acontece nas unidades de ABP, que são frequentemente limitadas pelo tempo

de uma disciplina curricular. A *Design Sprint*, que leva em consideração o curto tempo disponível para o exercício das dinâmicas propostas pelo *Design Thinking*, parece ser, portanto, uma alternativa promissora para a proposta de tornar a experiência do usuário um requisito importante para a construção de artefatos autênticos.

Design Sprint

Uma das melhores maneiras de se descrever a *Design Sprint* é dizer que ela é uma instanciamento do *Design Thinking*. Seus princípios são os mesmos e o modelo usado para abordar os problemas e fazê-los se transformar em protótipos testáveis também. Até mesmo as ideias de se ter um protótipo para testar no fim do processo possui raízes no modelo de Diamante Duplo do *Design Thinking* [31, 2, 1].

De acordo com Banfield et al. [31], a *Design Sprint* é um exercício de cinco fases restringido por tempo para reduzir os riscos no ato de levar um produto ao mercado. O problema reside no foco desequilibrado nos processos (sejam eles prescritivos ou mesmo ágeis) em detrimento da percepção das necessidades das pessoas. Em particular, os métodos ágeis, relegaram o *design* para uma posição secundária e isso também era parte do problema [31].

Knapp [2] apresentou o método como alternativa aos *brainstormings*. Uma de suas percepções era de que as melhores ideias não saíam das sessões de *brainstormings*, mas de momentos em que essas pessoas pensavam isoladamente sobre os problemas que precisavam resolver [2]. Especialmente em dois projetos em que o autor trabalhou (Caixa prioritária do Gmail e Google Hangouts), mostraram-no que estar debaixo de uma pressão exercida pelo tempo, estar longe do barulho das sessões de *brainstormings* e ter consigo um time multidisciplinar traziam muitos benefícios.

Um dos problemas do modelo proposto por Knapp [2] é juntar pessoas durante uma semana em uma sala. Banfield et al. [31] abordam esse problema e discutem a necessidade de adaptação das atividades da *Design Sprint* dentro dos blocos lógicos do *Design Thinking*, tornando-a tão flexível quanto poderia ser e sugerem diferentes *Sprints* em diferentes quantidades de tempo [31].

Uma versão semanal da *Sprint* divide o trabalho em um dia para compartilhamento de informações e geração de ideias, seguido de quatro dias para desenvolvimento de protótipos, conforme apresentado na Figura 2.2. Essa versão se modificou ao longo do tempo, com base nas experiências de seus praticantes, resultando em uma série de versões e adaptações dessa, sustentadas pelos pilares da multidisciplinaridade, tempo limitado para exercícios de criatividade, desenvolvimento de protótipos e contato com o usuário, tanto para o entendimento de seu problema, quando para a validação do protótipo desenvolvido [31].

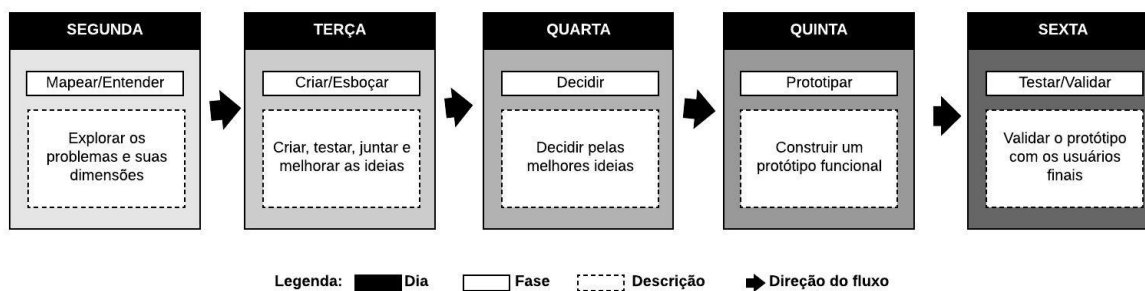


Figura 2.2: Passos da *Design Sprint*. Adaptado de [2]

Mesmo com todas essas variações, é possível pensar na *Design Sprint* como um *framework* ao invés de vê-la como uma estrutura rígida e inflexível [31]. Esse *framework* é dividido em 5 fases, comumente identificadas como dias, embora esses “dias” possam, de fato, caber em mais de um dia ou menos, desde que a estrutura dos blocos lógicos seja respeitada.

A preparação da *Sprint* começa bem antes das fases do *framework*. Em dias anteriores, a pessoa designada para fazer a facilitação deve convidar a equipe, explicá-los sobre o que será feito nos dias da *Sprint*, coletar informações anteriores importantes, tais como, o problema a ser resolvido e o perfil de usuário que se deseja beneficiar com a solução a ser proposta.

As fases da *Design Sprint* são [31]:

Mapear / Entender: É a primeira fase e nela há um diamante completo com divergência e convergência. Nesta fase, tudo o que a equipe multidisciplinar participante da *Sprint* sabe sobre o problema e sobre os usuários envolvidos no sofrimento desse problema é colocado na mesa e todos buscam uma compreensão mais amplificada sobre o problema em si, além de definir qual, de fato, é o verdadeiro problema (na maioria das vezes o problema que é raiz de uma série de outros) a ser atacado durante a *Sprint*. Essa convergência em um ponto específico e crítico permite que o foco e as energias da equipe estejam voltados para um problema realmente relevante. Os exercícios desta fase cumprem esse objetivo ao colocar o participante da *Sprint* para ouvir diferentes opiniões e visões, ver quais outras soluções já existem para aquele problema, definir hipóteses e formas de testá-las, pensar nas características e tarefas das pessoas na jornada deles como o usuário de um produto ou serviço e convergir para a parte mais importante da dessa jornada.

Criar / Esboçar: Nesta fase, as possibilidades são exploradas e muitas soluções são fornecidas. Este é um momento em que a quantidade é mais importante que a qualidade, pois o pressuposto básico é de que quanto mais soluções forem sugeridas, mais provável

será o surgimento de uma grande ideia. Desenhar esboços é a atividade principal dessa fase. Por meio dela, os participantes são levados a colocar no papel a maior quantidade de soluções possíveis para os problemas identificados na fase anterior. Os limites de tempo e as votações em silêncio são cruciais nesse momento, pois são elas que criarão o ambiente para que uma grande quantidade de soluções não enviesadas vá para a fase da convergência.

Decidir: Depois de ter gerado uma grande quantidade de soluções, é necessário convergir novamente, agora não para decidir qual o problema, mas qual solução adotar. Inspirado por tudo que foi feito nos dias anteriores, a equipe irá determinar pontos de conflito entre as propostas de soluções que foram geradas, criar maneiras de testar suas suposições durante o teste do protótipo, desenhar uma jornada de uso da solução por meio de um *storyboard* e determinar a maneira como a interação do teste será feita, incluindo o meio que apresentará o protótipo e um roteiro para a entrevista.

Prototipar: Nesta etapa, o protótipo é desenvolvido pelo time. Embora não haja uma exigência de perfeição para o protótipo e ele, inclusive, possa ser feito por intermédio de *wireframes*, ele precisa ser testável. Por testável, entende-se que este protótipo venha possuir pontos pelos quais o usuário irá poder interagir com ele. Isso é crucial, afinal através da fase de teste, que as suposições feitas serão validadas e os novos caminhos do desenvolvimento serão traçados. O foco do protótipo não são seus elementos estéticos, mas o fluxo e a navegação de um futuro produto.

Testar / Validar: O protótipo é levado a representantes do público alvo para ser testado e avaliado. Cinco é um número adequado de potenciais usuários a serem selecionados para este teste [2]. Essa é uma das etapas mais importantes da *Sprint*, e todo o resto do trabalho feito nas etapas anteriores converge para a execução das atividades dessa fase. Nela os representantes do público alvo podem colocar suas próprias opiniões sobre o trabalho realizado e criticá-lo, no que for necessário, fornecendo à equipe uma porção grande de *feedbacks* e *insights* para a tomada de decisão do trabalho a ser realizado nas etapas seguintes. Os testes são conduzidos por meio de interação do usuário com o protótipo, observação e entrevistas e têm como objetivo principal validar a série de suposições que foram levantadas na execução das fases anteriores da *Sprint*.

De acordo com Banfield et al. [31] são três resultados possíveis da *Sprint*:

- “*A maioria das coisas funcionou*”, quando existirem muitos acertos no desenho do protótipo e, de um modo geral, a maior parte da solução oferecida agradar aos potenciais usuários com os quais foram feitos os testes. Neste cenário, o protótipo estará pronto para ser conduzido à etapa de desenvolvimento;
- “*Algumas grandes questões*”, quando o protótipo é composto de um conjunto equilibrado de pontos fortes e fracos, devendo haver, portanto, manutenção das boas

características da solução e correção das características que falharam. A recomendação é de que, se este for o resultado final da *Sprint*, uma nova iteração seja feita partindo da etapa 2 do *framework* (Divergência).

- “*Tudo explodiu*”, quando a maioria das suposições levantadas falharam. Este é um cenário onde os clientes, por algum motivo, se sentiram perdidos no uso ou demonstraram, por meio de palavras e comportamentos, sua desaprovação com a solução oferecida. Embora neste cenário haja o indicativo de que a solução proposta tenha sido um fracasso total, a equipe não deve se preocupar, afinal terá perdido apenas pouco tempo e poderá refazer a *Sprint* desde a etapa 1 (Entendimento) para descobrir onde erraram e corrigirem seus erros sem que se tenha perdido muitos recursos financeiros e de tempo com um projeto que não iria funcionar.

Estes cinco dias contemplam muito das exigências discutidas na seção 2.2.1 sobre como devem ser especificados os requisitos de um produto de software a ser desenvolvido. Em um prazo bastante reduzido e através de um processo bem estruturado, a proposta de software estuda os aspectos afetivos dos usuários por meio das técnicas de imersão, consideram os vários agrupamentos de instantes de uma experiência e o fluxo da teoria de Csikszentmihalyi [95] por meio do desenho do mapa da jornada do usuário, além de avaliar as reações esperadas dos usuários durante a avaliação do protótipo que representa a proposta de solução. Outro aspecto importante nesse processo é sua identidade convergente com os princípios construtivistas da ABP, representada por meio do uso do trabalho em equipe, da autonomia dos participantes, do processo de investigação e do direcionamento dado por uma questão motriz discutidos na seção 2.1.5.

Outros trabalhos na literatura já se preocuparam com a UX na educação prática, alguns mais ligados às definições do *Design Thinking* outros menos. Entretanto, a definição de um processo estruturado para criação de propostas que alimentem a construção de software na ABP foi pouco explorada. A Seção 2.3 apresenta uma *overview* destes trabalhos, demonstrando quais suas contribuições para este tema específico.

2.3 Trabalhos Correlatos

Dedicar atenção à Experiência do Usuário (UX) por meio da educação é algo que pode trazer benefícios tanto para os alunos, que se tornam mais preparados para uma área em ascensão, quanto para os *stakeholders* que são contemplados por esses projetos e que recebem prestação de serviço, muitas vezes gratuita, sobre como seus produtos e serviços podem ser melhorados tendo como foco a boa experiência de seus clientes [101] e até mesmo para as instituições de ensino que terão um curso atrativo a novos estudantes

[102]. O caráter prático de uma abordagem educativa de UX é essencial. Para ter sucesso, um modelo de Educação em UX requer que os estudantes sejam postos em uma posição de resolução de problemas reais e que eles sejam tratados como aprendizes a caminho da competência [102].

Na literatura é possível encontrar modelos que lidem com a educação prática em UX. Getto e Beecher [102] propuseram um modelo para a introdução da UX na pedagogia de ensino superior, considerando como importante o foco prático da disciplina e destacando os benefícios que um programa de ensino dedicado à UX pode trazer nos dias atuais. As abordagens pedagógicas, segundo o modelo proposto, ressaltam a importância de ter um estágio teórico de introdução aos conceitos principais de UX, em que os alunos observem *designers* experientes no exercício do *designer* e de que os alunos incorporem a diversão no processo de resolução de problemas, inerente ao *design* [102].

MacDonald e Rozaklis [101] trabalharam em um modelo pedagógico onde executaram unidades letivas em um curso de pós-graduação em Biblioteconomia e Ciência da Informação baseadas em um modelo próprio de educação em UX. Nesse trabalho, os autores descrevem como o ensino prático em UX com clientes reais foi conduzido em cursos que se enquadravam em três formatos, sendo um deles embutido no curso, outro que dura o curso inteiro e outro em que os alunos realizaram as atividades fora dos limites de um curso regular [101]. Nesses projetos, os professores coletaram demandas de clientes reais e as atribuíram a grupos de alunos. Esses alunos trabalharam em pesquisas com os usuários dos serviços e em análises de concorrentes dos clientes que demandaram os projetos [101]. Neste trabalho, a proposta criada não tinha como objetivo fomentar o desenvolvimento dentro da unidade de ABP.

Embora, o enfoque nas propostas de software para desenvolvimento em unidades de ABP não seja muito explorado, há autores que fizeram uso ABP como o modelo guia de uma educação prática baseada em UX. Conradie et al. [103] trabalharam com UX para *design* de interfaces tangíveis com alunos de mestrado em um curso de *Design* de Produtos Mecatrônicos. Nesse estudo de caso, uma rede de parceiros da indústria foi montada e os alunos se organizaram em grupos multidisciplinares para ajudá-los com suas demandas [103]. Da mesma forma que o trabalho anteriores a intenção final não foi produzir propostas de desenvolvimento para exploração nas unidades de ABP. Já Gerber et al. [104] fizeram uso da Aprendizagem Baseada em *Design* Extracurricular, uma variação da ABP voltada para a produção de artefatos baseados em design de inovação, para levar os alunos a produzirem soluções que tivessem cunho social para resolução de problemas de uma comunidade local [104], mas não vinculou sua abordagem com experiência de uso dos membros dessa comunidade.

Outra distinção entre os resultados desses trabalhos e a proposta da pesquisa descrita

nessa dissertação é que, embora essas abordagens mencionem a autenticidade, elas mantêm o foco em seu valor pedagógico, relegando a segundo plano a qualidade do artefato como atributo da autenticidade.

2.4 Síntese do Capítulo

Este capítulo apresentou os conceitos necessários para o entendimento do contexto deste trabalho. Foi traçado uma visão geral sobre as teorias psicopedagógicas que influenciaram a prática de construção de artefatos da ABP, mencionando especificamente as contribuições que o construtivismo, o *learning by doing*, o método do projeto, e aprendizagem situada e o construcionismo trouxeram para essa prática. Em seguida, a ABP foi descrita em detalhes, mantendo ainda o foco na criação de artefatos e na autenticidade como característica definidora deles. Além disso, foi discutido os conceitos da Experiência do Usuário, de seus processos e de como ela pode ser tratada como requisito de artefatos digitais. Foi, então, apresentado o *Design Thinking* como principal processo de trabalho com foco em UX e como a *Design Sprint*, uma de suas variações, é capaz alcançar os mesmos objetivos com um *framework* mais coeso e focado, alinhado às necessidades educacionais de uma unidade letiva de desenvolvimento de software. Também discutimos como a educação prática e, mais especificamente, a ABP tem sido reportada na literatura quando o objetivo é o ensino e experimentação dos processos de UX.

Capítulo 3

Revisão Sistemática de Literatura

Este trabalho utiliza a Revisão Sistemática da Literatura (RSL) para exploração de evidências que alimentem a fase empírica da pesquisa. A RSL consiste em uma sequência de passos bem definidos para mapear, encontrar, avaliar criticamente, consolidar e agregar resultados de estudos primários relevantes sobre uma questão de pesquisa específica [32, 105]. Estudos primários são os estudos originais que constituem a base da ciência e alimentam as revisões de literatura, que são chamadas de estudos secundários [32]. Entre os benefícios de uma RSL encontram-se o resumo de evidências empíricas de determinado assunto ou tópico específico, a identificação de lacunas no corpo de conhecimento científico para indicar caminhos para novas investigações, o estabelecimento de um fundo teórico que permita ao pesquisador posicionar de forma adequada novas atividades de pesquisa, a exposição do que já foi feito, resumindo o estado-da-arte de determinada área de estudo para que pesquisadores centralizem seus esforços em pesquisas que sejam de algum modo inovadoras, colocando o pesquisador em contato com uma ampla gama de estudos para que sua própria investigação científica não seja enviesada por estudos com resultados atípicos ou mal reportados, como também permite ao pesquisador entrar em contato com trabalhos relacionados com o seu, a fim de que haja aprendizado com erros e acertos de outros e comparação de resultados de modo a fortalecer evidências [32].

As fases que constituem uma RSL são Planejamento, Condução e Publicação dos Resultados [32]:

- Na fase de **Planejamento**, é definido o objetivo geral, as fontes de pesquisa, as palavras-chave da busca, os critérios de inclusão, exclusão e de qualidade, a estratégia de extração e sintetização dos dados. Essas definições culminam no artefato chamado de *Protocolo de Revisão*, o qual deverá também ser validado antes da realização das buscas [32]. Este protocolo é necessário para guiar e sistematizar o processo de busca de evidências na literatura científica de forma que o viés de interesse do pesquisador possa ser diminuído no decorrer da condução da pesquisa.

- Na fase de **Condução** é realizada a pesquisa. É nesta fase que o pesquisador, munido das declarações constantes no protocolo de pesquisa, efetua as buscas nas bases de dados eletrônicas, sejam científicas ou não, e em outras fontes impressas que sejam relevantes ao estudo. A condução da pesquisa envolve a identificação, a seleção, a avaliação da qualidade e a extração de dados dos estudos primários pertinentes à pesquisa. Os dados destes estudos culminam em uma síntese que facilitará o processo de publicação dos resultados [32].
- Na fase de **Publicação dos Resultados**, os resultados são documentados de acordo com a síntese feita, de forma que eles respondam às questões de pesquisas previamente estabelecidas na fase de planejamento.

O protocolo de pesquisa adotado para a realização da RSL é apresentado na Figura 3.1.

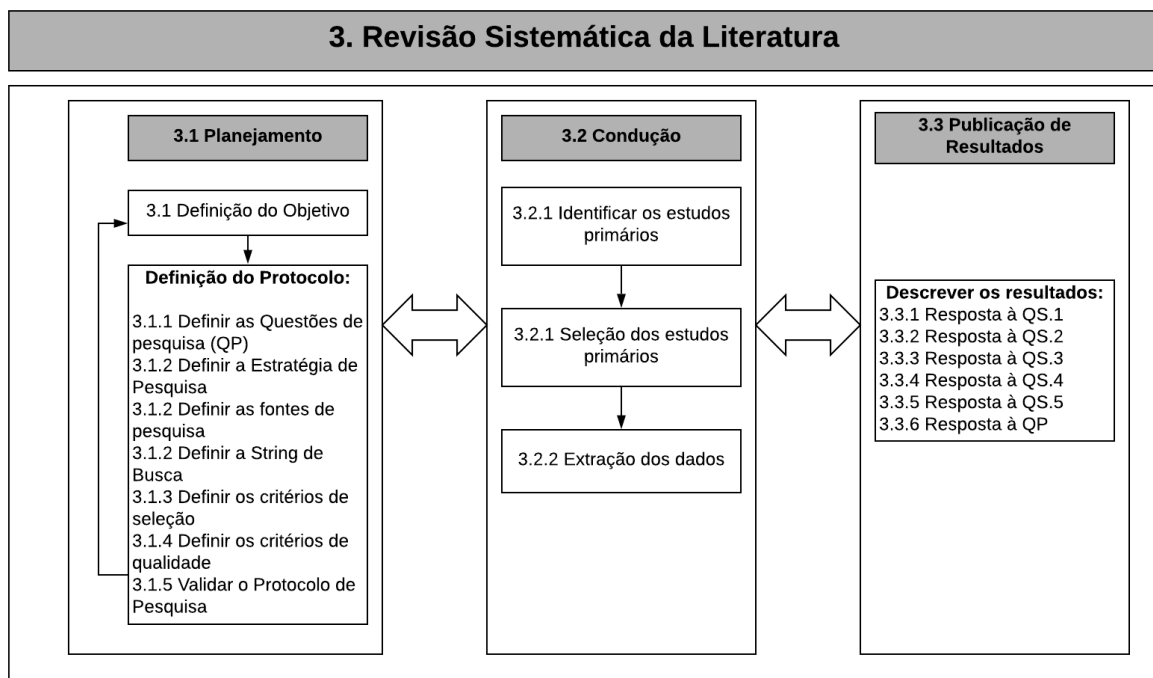


Figura 3.1: Protocolo da Revisão Sistemática da Literatura

Uma RSL pode ser apoiada por softwares que auxiliam o pesquisador no trabalho de organização da grande quantidade de dados gerada pela pesquisa. Neste estudo, o software utilizado foi a ferramenta StArt (*State of the Art through Systematic Review*).

Neste capítulo, a expressão Ciência da Computação refere-se a cursos de graduação e pode significar qualquer curso que tenha como base o ensino de desenvolvimento de software.

3.1 Planejamento

Uma RSL deve começar com uma motivação que justifique sua execução [32], de modo que a proposta feita seja alimentada pelo conhecimento gerado pela RSL e que essa proposta seja inovadora e útil para a resolução do problema da pesquisa. Cabe salientar que, embora o foco deste trabalho esteja no desenvolvimento de um processo que permita formular propostas para projetos de ABP mais adequados à UX, o escopo dessa RSL abrange também as questões pedagógicas e operacionais de uma ABP. Naturalmente, definir uma nova maneira de pensar sobre os artefatos, leva a modificações que devem ser executadas no processo de construí-los. Além disso, é necessário considerar que já existe um significado para o conceito de autenticidade dentro da ABP, sendo, então, importante que a adição de novas características a esse conceito não desconfigure o significado já definido pela literatura (Seção 2.1.6).

Com base nessas questões o seguinte objetivo foi gerado: **Identificar e analisar implementações de aprendizagem baseada em projetos para desenvolvimento de software em contextos educacionais que tenham produzido softwares autênticos**. Durante a condução da RSL não encontramos trabalhos anteriores que possuam este objetivo. Assim, esta revisão mostra-se relevante.

3.1.1 Questões de Pesquisa

Uma RSL sempre busca responder uma questão de pesquisa (QP) que seja importante tanto para o pesquisador quanto para os profissionais que serão beneficiados pelos resultados da pesquisa [32]. Tal questão de pesquisa é melhor definida se for embasada em três pontos de vistas: *população*, *intervenção* e *resultados*. É possível resumir uma questão de pesquisa (QP) para uma RSL no seguinte formato: *Intervenção* para uma *População* com uma *Condição* [32, 106].

Um *template* bastante utilizado para a formação de uma questão de pesquisa é o acrônimo PICO, desenvolvido pela *Cochrane Collaboration*, [106, 107, 108]. As letras deste acrônimo possuem o seguinte significado: a letra P (População) – corresponde a grupos populacionais ou a uma área de aplicação. A letra I (Intervenção) – define as ferramentas, intervenções, técnicas, tecnologias ou métodos usados nos estudos primários. A letra C (Comparação) – é usada quando a pergunta a ser respondida envolve comparação entre métodos. Por fim, a letra O (*Outcomes* em português Resultados) – remete à finalidade dos estudos e seus resultados [107, 108]. Nesta pesquisa, as letras deste acrônimo foram usadas da seguinte maneira:

- **População:** trabalhos que apresentem desenvolvimento de software em unidades letivas de cursos de graduação;

- **Intervenção:** a intervenção a ser estudada é a aprendizagem baseada em projetos;
- **Comparação:** não é intenção desta revisão fazer a comparação entre métodos.
- **Outcomes/Resultados:** os resultados que se buscam encontrar nos estudos se referem ao alcance da autenticidade em softwares criados em sala de aula;

Com base nestas definições foi possível escolher a seguinte questão de pesquisa a ser respondida durante a realização da RSL:

QP - Como aplicações de *Aprendizagem Baseada em Projetos* para *desenvolvimento de software durante os cursos de graduação* tornam um *software autêntico*?

De modo a facilitar o encontro de respostas para a questão norteadora com mais precisão, é possível que ela seja quebrada em questões secundárias e mais específicas [109, 110]. Portanto, as Questões Secundárias (QS) definidas para esta questão de pesquisa são apresentadas na Tabela 3.1.

Tabela 3.1: Questões secundárias de pesquisa (QS)

QS	Descrição
QS.1	Como o conteúdo teórico e técnico é ensinado aos alunos enquanto eles desenvolvem os projetos propostos nas disciplinas?
QS.2	Qual o tempo disponível utilizado nas aplicações de aprendizagem baseada em projetos para criação de softwares que serão ou podem ser usados em um contexto real?
QS.3	Quais modelos e/ou técnicas de engenharia de software são usados no projeto?
QS.4	Como são formados os grupos de alunos que participam do desenvolvimento de software em sala de aula?
QS.5	Houve mudança no rendimento acadêmico dos alunos após participarem das atividades práticas de aprendizagem propostas durante a unidade letiva?

3.1.2 Estratégias de Pesquisa

Como estratégia de pesquisa, foi utilizada a Busca Automática nas bases de dados eletrônicas por intermédio de uma *string* de busca. Como forma de combater o viés de publicação, que acontece por causa da tendência de publicação de trabalhos cujos resultados sejam positivos, e também como forma de encontrar trabalhos que não foram alcançados por intermédio da busca automática, é necessário que se faça também uma consulta à literatura cinzenta (*grey literature*). Por literatura cinzenta, compreende-se os

trabalhos que não foram publicados em revistas e conferências, mas que possam constar nos anais destas. Isso é particularmente importante para este trabalho, uma vez que há uma infinidade de abordagens que possuem características similares à ABP (Seção 2.1.5), sem, no entanto serem denominadas assim [26]. Em virtude dessa diversidade, torna-se impraticável buscar todas as abordagens similares à ABP por meio da identificação das palavras-chaves que as possam representar.

Neste caso, a Busca Manual em anais de revistas/jornais e conferências com reconhecida relevância na área de informática na educação, conforme apresentado na Tabela 3.2, foi também usada como estratégia de pesquisa.

Tabela 3.2: Conferências e Periódicos consultados na Busca Manual

Sigla	Journal / Conferência	Endereço Eletrônico
FIE	IEEE Frontiers in Education Conference	https://ieeexplore.ieee.org
SIGCSE	ACM Technical Symposium on Computer Science Education	https://dl.acm.org/proceedings.cfm
ITiCSE	Conference on Innovation and Technology in Computer Science Education	https://dl.acm.org/proceedings.cfm
ICSE	International Conference on Software Engineering	https://dl.acm.org/proceedings.cfm

***String* de Busca**

É importante que a *string* de busca contenha palavra-chaves que alcancem a maior quantidade de trabalhos relevantes possível [32, 106], durante a Busca Automática. As definições dadas para cada letra do acrônimo PICO orientam a escolha destas palavras-chaves.

- **População:** “software development”, “classroom”;
- **Intervenção:** “project-based learning”, “PBL”, “project based learning”, “ProjBL”, “PjBL”;
- **Comparação:** não se aplica;
- **Outcomes/Resultados:** “authenticity”, “authentic”, “realistic”, “real”, “real world”.

Com base nestas palavras-chave, foi possível desenvolver a seguinte *string* de busca: “software development” AND “classroom” AND (“project-based learning” OR PBL OR “project based learning” OR “ProjBL” OR “PjBL”) AND (“authenticity” OR “authentic” OR “realistic” OR “real” OR “real world”).

3.1.3 Critérios de Seleção

Os seguintes critérios de seleção foram definidos com o objetivo de filtrar os trabalhos a serem utilizados para responder as questões de pesquisa com a realização da RSL:

Critérios de Seleção das Bases

Existe uma infinidade de bases eletrônicas que podem ser escolhidas para a execução de uma RSL, de modo que, dada a limitação de tempo, é impossível consultar todas as bases digitais existentes. Portanto, é necessário que as bases sejam criteriosamente escolhidas observando-se o escopo da pesquisa. Os critérios de seleção das bases para esta pesquisa foram:

- A base deve permitir a exportação em *Bibtex*;
- A base deve permitir fazer consultas automatizadas;
- A base deve poder ser acessada por intermédio do Periódicos Capes no convênio com a Universidade de Brasília (UnB);
- A base deve permitir a filtragem por ano;
- A base deve estar contida na lista de fontes do StArt;
- A base deve contemplar trabalhos de Informática na Educação e/ou Educação em Ciência da Computação.

Desta forma, as bases escolhidas para esta pesquisa foram: Scopus, IEEE Xplore, ACM Digital Library e Web of Science. Alguns critérios são importantes para tornar a pesquisa realizável dentro da limitação de recursos existente e para trazer um conjunto de estudos que consigam responder à questões de pesquisa. Os seguintes critérios de inclusão e exclusão foram definidos desta forma:

Critérios de Inclusão

- O trabalho deve usar a abordagem de aprendizagem baseada em projetos;
- O trabalho deve abordar o desenvolvimento de software autêntico (ou seja, para propósitos além dos educacionais);
- O trabalho deve apresentar a entrega dos artefatos produzidos durante o desenvolvimento do software;

- O modelo deve ter sido validado utilizando um estudo de caso prático e algum usuário deve ter avaliado positivamente o software ou o software deve estar sendo usado em alguma organização.

Foi necessário que o trabalho contemplasse todos os critérios de inclusão para que ele fosse usado na fase de extração de dados.

Critérios de Exclusão

- Trabalhos fora do escopo de aprendizagem baseada em projetos;
- O modelo proposto não foi validado utilizando um estudo de caso prático;
- O software produzido por meio da abordagem não é funcional;
- Não houve aprendizagem baseada em projetos aplicada para educação em Ciência da Computação;
- O artefato desenvolvido pela abordagem não está fundamentada em um contexto do mundo real;
- O trabalho não apresenta um estudo empírico;
- Não foi possível ter acesso ao artigo completo de maneira gratuita através das bases de dados.

O trabalho que foi enquadrado em pelo menos um dos critérios de exclusão adotados foi rejeitado.

3.1.4 Critérios de Qualidade

Os Critérios de Qualidade (CQ) ajudam a filtrar os estudos pela qualidade e relevância para o tema depois de executada a primeira iteração da seleção dos estudos primários [32]. Os níveis dos CQ podem ser categóricos (Alto, médio e baixo) e ser divididos em várias dimensões para sistematizar sua aplicação [111]. Nesta revisão, um estudo que foi classificado como tendo nível baixo em pelo menos uma das dimensões, foi rejeitado. As três dimensões de qualidade que um estudo deve apresentar são:

- Qualidade da execução do estudo: relativo à capacidade do estudo aos padrões científicos exigidos por uma determinada área técnica científica;
- Adequação à questão da revisão: relativo à capacidade do estudo se adequar exatamente ao assunto alvo da revisão sistemática;

- Adequação ao foco da revisão: relativo à similaridade do contexto para o qual os resultados da revisão será útil;

Cabe salientar que, em termos contextuais, o resultado da pesquisa será útil para o desenvolvimento de uma proposta de ABP para desenvolvimento de software que se preocupe com a UX como atributo em um curso de tecnologia em análise e desenvolvimento de sistemas em uma cidade do interior do estado de Goiás.

3.1.5 Validação do Protocolo

Um segundo pesquisador aprovou o protocolo de pesquisa. Uma busca piloto foi feita na ACM Digital Library. Colocar as palavras relativas à autenticidade diminuía muito a quantidade de estudos. Este fenômeno acontece porque, muitas vezes, os autores não mencionam explicitamente a palavra autenticidade ou sinônimos em seus estudos mesmo que descrevam desenvolvimento de software que estejam ligados a contextos do mundo real. Portanto, essas palavras foram retiradas. A *string* de busca final adotada foi: “software development” AND (classroom OR learning OR teaching OR education) AND (“project-based learning” OR PBL OR “project based learning” OR “ProjBL” OR “PjBL”).

3.2 Condução

Com base no que foi definido e aprovado no protocolo, foi possível executar as tarefas da fase de condução. A fase de execução adotada é definida como seleção dos estudos.

3.2.1 Seleção dos Estudos Primários

Após a execução da Busca Automática utilizando a *string* de busca definida no protocolo, foram realizados os seguintes ajustes: Na base ACM Digital Library e IEEE Xplore, a *string* foi reescrita da seguinte forma: “**software development**” AND (“**classroom**” OR “**learning**” OR “**teaching**” OR **education***) AND (“**project-based learning**” OR PBL OR “**project based learning**” OR “**ProjBL**” OR “**PjBL**”) NOT (“**problem-based learning**” OR “**problem based learning**”). Depois disso, ela foi usada no campo de busca da base e o horizonte temporal da busca foi limitado para o intervalo de 2009 a 2019. A opção do campo de busca da IEEE Xplore foi definida como *Full Text & Metadata*. Tal busca resultou em **26 estudos** para ACM Digital Library e **21 estudos** para IEEE Xplore.

Na base Web of Science, a *string* foi usada da seguinte forma: “**software development**” AND (classroom OR “**learning**” OR “**teaching**” OR education) AND (“**project-based learning**” OR PBL OR “**project based learning**” OR “**ProjBL**”

OR “PjBL”) NOT (“problem-based learning” OR “problem based learning”), em vista da grande quantidade de resultados retornados sobre aprendizagem baseada em problemas. Tal busca resultou em **38 estudos**.

Na base <https://www.scopus.com/Scopus>, a *string* foi usada da seguinte forma: (ALL(“software development” AND (classroom OR “learning” OR “teaching” OR education) AND (“project-based learning” OR pbl OR “project based learning” OR “ProjBL” OR “PjBL”)) AND NOT ALL (“problem based learning” OR “problem-based learning”)) AND PUBYEAR > 2008, também em vista da grande quantidade de resultados retornados sobre aprendizagem baseada em problemas. Tal busca resultou em **353 estudos**.

A Busca Manual nas bases relacionadas na Tabela 3.2 resultou em **47 trabalhos**. O volume de artigos encontrados em cada fonte de busca é demonstrado na Figura 3.2.

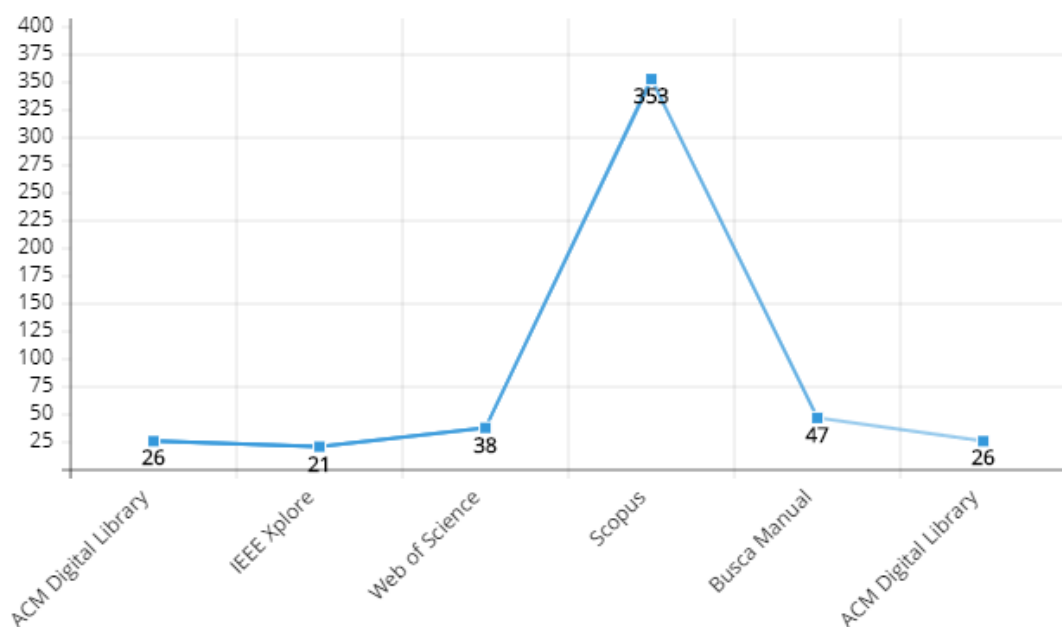


Figura 3.2: Quantidade de artigos selecionados por biblioteca digital.

Do montante geral dos artigos (**485 estudos primários**), 73% vieram da busca na Scopus, 8%, vieram da busca na Web of Science, 5% vieram da busca na ACM Digital Library, 4% vieram da busca na IEEE Xplore e 6% vieram da busca manual nos anais das conferências e periódicos. A Figura 3.3 apresenta a distribuição dos resultados obtidos de acordo com as fontes de busca.

A seleção dos estudos primários ocorreu através da aplicação dos critérios de inclusão e exclusão apresentados na Seção 3.1.3 e foram executados em três etapas (Tabela 3.4). Na primeira, os artigos foram aceitos e/ou rejeitados com base na leitura do resumo e

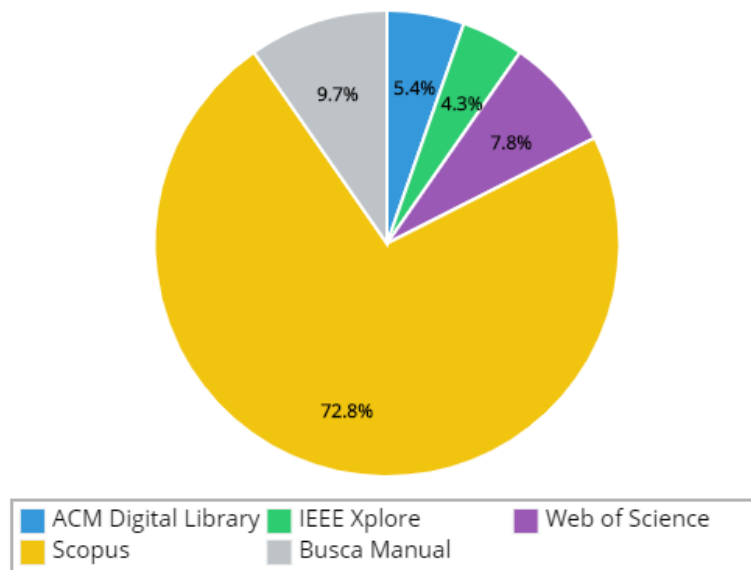


Figura 3.3: Proporção de artigos por fonte de busca

palavras chaves. Nesta etapa foram selecionados **179 artigos**. Na segunda etapa, os estudos foram selecionados e/ou rejeitados a partir da leitura completa do artigo. No final, foram selecionados **40 artigos**. Um segundo pesquisador efetuou a leitura completa dos estudos selecionados e aplicou os critérios de qualidade adotados. Após essa fase, **23 estudos primários** foram selecionados para responder (fase de extração de dados) as questões de pesquisa que foram propostas.

3.2.2 Extração de Dados

A etapa de extração de dados foi conduzida com auxílio da ferramenta StArt e também da ferramenta Mendeley, cujas funções permitem agregar, organizar e fazer o fichamento dos artigos selecionados.

3.3 Resultados

A extração das informações dos artigos de uma RSL deve ser feita de forma que responda às questões de pesquisa estabelecidas no protocolo. Para facilitar essa tarefa, uma das funções da ferramenta StArt permite criar campos de texto e itens de seleção onde é possível registrar os dados de forma facilitada. Com base nas questões descritas na Tabela 3.1, os campos a serem preenchidos na ferramenta StArt foram:

1. Qual o nível escolar da(s) turma(s) onde foi aplicada a abordagem de ABP?

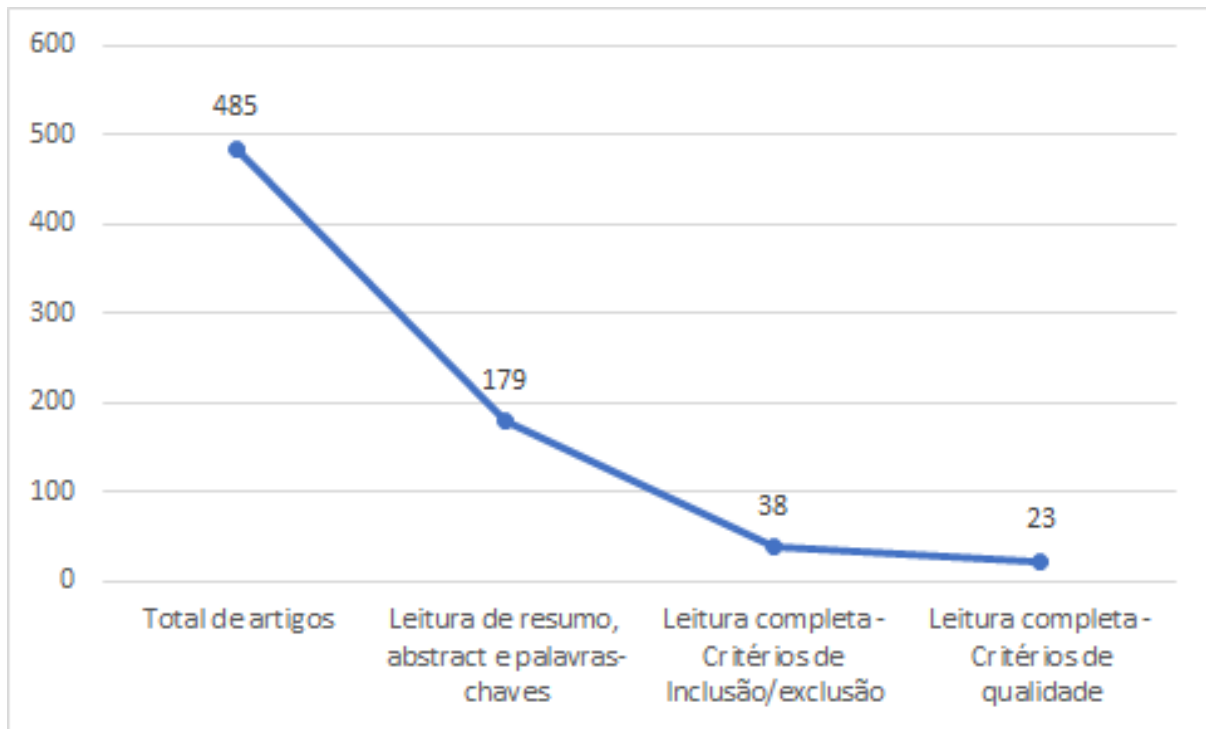


Figura 3.4: Quantidade de artigos por etapa da RSL

2. Qual o tempo disponível para a execução da ABP?
3. A ABP acontece de forma interdisciplinar?
4. A ABP acontece de forma multidisciplinar?
5. Qual a quantidade de alunos e como eles foram divididos em grupos?
6. Existe uma parceria entre indústria e academia?
7. Qual técnica ou método de Engenharia de Software foi usado?

A respostas das perguntas desses campos alimentaram a resposta final às questões secundárias (QS) e conseqüentemente à questão principal de pesquisa (QP). Todas as respostas foram extraídas dos artigos selecionados, os quais são apresentados na Tabela 3.3. A coluna ID fornece a identificação para os estudos selecionados, a coluna Ref indica a referência bibliográfica do estudo primário, a coluna Ano indica em que ano o estudo foi publicado, a coluna Tipo descreve se o artigo foi publicado em conferência ou anais de congressos (C) ou se foi publicado em periódico (P), a coluna Descrição fornece uma breve descrição do estudo e a coluna QS indica quais questões secundárias foram respondidas por intermédio do respectivo trabalho.

Tabela 3.3: Artigos selecionados na RSL

ID	Ref	Ano	Tipo	Descrição	QS
E1	[112]	2018	C	Comparação entre sala de aula invertida e disciplina tradicional em uma unidade de ABP.	QS.1, QS.2, QS.3, QS.4
E2	[113]	2018	C	Descrição de uma disciplina de Engenharia de Software que usa competição entre alunos, ABP e relação com clientes da indústria.	QS.1, QS.2, QS.3, QS.4
E3	[114]	2014	P	Aplicação da ABP para melhora das habilidades de desenvolvimento de software.	QS.2, QS.3, QS.4
E4	[115]	2017	C	Uso da abordagem ágil e de métodos formais em projeto de um contexto acadêmico.	QS.1, QS.2, QS.3, QS.4
E5	[116]	2014	C	Avaliação da abordagem de Mini-projetos implementada em uma universidade da Venezuela.	QS.1, QS.2, QS.3, QS.4
E6	[117]	2012	C	Apresentação de um modelo de reputação chamado <i>KommGame</i> para ambientes de ABP.	QS.1, QS.2, QS.3
E7	[12]	2015	C	Descrição de uma disciplina de projeto de curso que usa aprendizagem de serviço e colaboração entre departamentos de uma universidade.	QS.1, QS.2, QS.3, QS.4
E8	[118]	2018	P	Análise de um projeto multicultural que mantém os alunos como co-produtores.	QS.2, QS.4
E9	[119]	2013	C	Descrição de um curso que usa experiências didáticas para construir um produto que os clientes realmente precisam.	QS.1, QS.2, QS.3
E10	[120]	2017	C	Combinação de aprendizado mobile com <i>Just-in-time teaching</i> , projetos autênticos e <i>peer review</i> para ensino de desenvolvimento de software.	QS.1, QS.2, QS.3, QS.5

E11	[121]	2017	P	Descrição de experiências na condução de aulas de engenharia de software em cooperação com a indústria.	QS.1, QS.2, QS.3, QS.4
E12	[122]	2012	C	Relato de um estudo iterativo para criação de um modelo para trazer contextos do mundo real por meio do desenvolvimento de módulos educacionais.	QS.2, QS.3
E13	[123]	2018	P	Apontamento de padrões e anti-padrões para implementar um ambiente para ensino sobre <i>start-ups</i> por meio da ABP.	QS.1, QS.2, QS.3
E14	[124]	2018	C	<i>Workshop de Design Thinking</i> para ajudar os alunos com a geração de inovação para seus projetos de ABP.	QS.3, QS.4
E15	[125]	2014	C	Relato da aplicação do <i>Scrum</i> para resolver problemas de comunicação.	QS.1, QS.2, QS.3
E16	[126]	2011	C	Relato de como os alunos aprendem além do material disponibilizado em aula de aula trabalhando juntos em um <i>Vertically-Integrated Project (VIP)</i> chamado <i>eStadium</i> .	QS.1, QS.2, QS.3
E17	[127]	2018	P	Descrição da evolução do desenvolvimento de um curso em Engenharia de Requisitos para os primeiros anos de um curso de graduação	QS.3, QS.4
E18	[128]	2014	P	Descrição de uma unidade de ABP interdisciplinar	QS.1, QS.2, QS.3, QS.4, QS.5
E19	[129]	2018	P	Proposição de um método de monitoramento reflexivo semanal mediado por supervisores em unidades letivas que envolvem a construção de projetos de software.	QS.1, QS.2, QS.3
E20	[130]	2010	C	Discussão acerca de cursos pilotos que usaram projetos do mundo real no ensino de várias disciplinas de computação.	QS.1, QS.3, QS.4

E21	[131]	2010	P	Apresentação de um modelo de ABP no currículo para educar os profissionais de Tecnologia da Informação que se especializam em projeto e desenvolvimento de sistemas de informação para pequenas e médias empresas.	QS.2, QS.4
E22	[132]	2015	P	Experimento sobre níveis de cognição estudantil na utilização da aprendizagem baseada em projetos interdisciplinares.	QS.5
E23	[133]	2015	P	Apresentação de um desenho de um curso que usa ABP e <i>Scrum</i> e sua atualização para a inserção de abordagens <i>lean</i> para o desenvolvimento de software.	QS.1, QS.2, QS.3

3.3.1 QS.1 - Como o conteúdo teórico e técnico é ensinado aos alunos enquanto eles desenvolvem os projetos propostos nas disciplinas?

O modelo de ABP sempre deve levar em consideração o aprendizado dos alunos que dela fazem parte [70, 22]. Entregar o conteúdo teórico de uma unidade de ABP, sendo ele curricular ou não, não é uma tarefa fácil, afinal há sempre que se considerar a natureza dos projetos acadêmicos para os quais os alunos terão apenas uma disponibilidade de tempo reduzida que deverá ser dividida entre a execução dos projetos e o aprendizado dos conceitos teóricos.

É importante salientar que, em muitas aplicações de ABP, os alunos participantes estão em um nível mais avançado de conhecimento, depois de terem estudado os conceitos fundamentais da computação e engenharia de software em outras disciplinas [116, 120, 129]. Entretanto, mesmo nesses casos, é necessário algum tipo de ensino formal aos alunos envolvidos nas unidades de ABP e esse ensino nem sempre pode ser executado em sala de aula. Por causa disso, em algumas aplicações de ABP há a escalação de mais de um professor [116] e até de uma equipe de monitores não docentes para auxiliar na tarefa de ensino dos alunos [126, 125, 112] e acompanhá-los no desenvolvimento de seus projetos. Isso auxilia na manutenção da qualidade dos artefatos finais destes projetos, tornando-os mais autênticos [129]. O trabalho apresentado por Marques et al. [129] aborda especificamente essa questão ao analisar características de dois grupos em várias unidades

de ABP, tendo como principal diferença entre eles a presença constante de monitores. Os resultados desse estudo mostram evidências de que os alunos sob supervisão tiveram valores significativamente melhores do que os alunos não supervisionados em termos de efetividade, coordenação e senso de pertencimento a um time. A efetividade, foi analisada pela entrega dos requisitos essenciais para o funcionamento da aplicação, o qual é um aspecto muito importante para o alcance da autenticidade em um artefato de software. Entretanto, a produtividade, outro aspecto analisado por Marques et al. [129] que é importante para a autenticidade, não retornou melhores resultados nas equipes de alunos sob supervisão de monitores.

Aibin e Hunter [112] apresentam um estudo das diferenças entre equipes supervisionadas e equipes não supervisionadas durante um projeto com uma amostra menor na intenção de verificar a efetividade da abordagem de sala de aula invertida para um projeto em um contexto de ABP. O trabalho concluiu que não há diferenças significativas entre os dois grupos em relação ao *feedback* do cliente (o grupo não supervisionado é apenas um pouco melhor) e à satisfação dos alunos. A forma de distribuição dos conteúdos em uma ABP pode acontecer de inúmeras formas:

- **Instrução Direta:** Configura-se como uma estratégia que faz uso das aulas expositivas nos moldes tradicionais dadas geralmente nas primeiras semanas da unidade de ABP [116, 133, 121, 12, 115]. Quando o projeto é feito dentro de sala de aula, o tempo alocado para essa estratégia varia de 2 a 4 semanas [116, 133, 121], mas as instruções também podem ser dadas regularmente durante toda a unidade letiva seguindo a estratégia de módulos ou de execução do projeto fora da sala de aula [117, 112, 129];
- **Coaching:** Tutoria especializada para resolução de problemas frequentemente aliada à estratégia de “ensino puxado” (quando uma instrução é dada pontualmente para a resolução de um problema específico fora de uma estrutura linear de cronograma de ensino) [130, 128, 116, 115]. Sua realização quase sempre está ligada com a presença de tutores, sejam eles docentes [116] ou não docentes [129, 125];
- **Disponibilização de materiais de ensino:** Esta estratégia envolve a proatividade dos alunos, que deverão consumir os conteúdos fundamentais por eles mesmos [116, 130, 119, 126, 112]. Os exercícios de leitura (e também de escrita) possuem importante papel na retenção dos conteúdos destes materiais [119] e para esta estratégia, mesmo os dispositivos móveis podem ser de grande ajuda, dada sua natureza portátil [120];
- **Alunos ensinando alunos:** Também conhecida como *Peer Learning* ou *Peer Teaching*, esta é uma forma de ensino baseada em princípios construtivistas e de apren-

dizagem ativa que coloca o aluno como responsável pelo ensino dos colegas por meio da instrução direta ou suporte entre os pares [123, 117, 126];

- **Aulas Práticas:** Também podem ser chamadas de laboratórios. Elas podem aparecer no formato de mini-projetos [113, 119], ou pode ser ministradas por especialistas da área [113, 128];
- **Ensino sob demanda (*Just-in-time teaching*):** Trata-se de um método de ensino que, aliado à Sala de Aula Invertida, permite que os alunos consumam os conteúdos teóricos e resolvam exercícios de aquecimento fora de sala de aula e, em sala, tenham seus pontos de fraqueza no conteúdo reforçados por aulas pontuais com o professor [120].

3.3.2 QS.2 - Qual o tempo disponível utilizado nas aplicações de aprendizagem baseada em projetos para criação de softwares que serão ou podem ser usados em um contexto real?

Como discutido na Subseção 3.3.1, o tempo pode se tornar um fator muito limitante em uma unidade de ABP. Não apenas em relação à entrega dos conteúdos, mas também em relação à qualidade geral do artefato desenvolvido. As equipes de desenvolvimento de software de uma fábrica de software (indústria) dispõe de uma configuração de tempo muito diferente daquelas disponíveis em um ambiente acadêmico. Dentre as mais limitantes restrições, estão o encaixotamento dos projetos dentro de semestres letivos e a impossibilidade de contar com os alunos todos os dias, como é comumente praticado em um ambiente real de trabalho, embora haja projetos mais ambiciosos que proponham a criação de uma fábrica de software literal baseada em ABP no ambiente acadêmico [123].

Essas restrições depõem contra muitos princípios das práticas ágeis, por exemplo. Em um ambiente ágil, é esperado que as equipes trabalhem juntas a todo momento, exercitem a reflexão sobre o seu trabalho todos os dias e, se possível, mantenham o cliente sempre por perto [125]. Portanto, é importante que estratégias sejam postas em prática para uma correta execução de uma abordagem de ABP que visa entregar um software real para um cliente que aguarda para usá-lo. Algo que pode ser feito é aproveitar ao máximo o tempo dos alunos fora de sala de aula para que estes façam os seus projetos [117]. No entanto, essa estratégia pode ter suas complicações, dado que os alunos quase sempre estão matriculados em outras disciplinas e têm uma grande quantidade atividades/exercícios dessas disciplinas para fazer.

As abordagens de ABP são normalmente executadas dentro de unidades letivas de um semestre [112, 129, 113, 121, 133, 114, 122, 117, 120, 128], embora essa não seja

uma regra geral. Existem casos em que a unidade de ABP é executada em um tempo inferior a um semestre [116, 12], e outras em que ela dura mais de um semestre letivo [131, 125, 118, 119, 126]. Embora a maioria dos estudos encontrados na literatura estejam dentro de um semestre letivo, a quantidade de semanas e horas disponíveis varia bastante, conforme apresentado na Tabela 3.4.

Tabela 3.4: Tempo programado para a unidade de Aprendizagem Baseada em Projetos

Referência	Tempo em semanas	Tempo em horas
[112]	15 semanas	1 hora por semana
[129]	20 semanas	1 hora por semana
[133]	15 semanas	não informado
[121]	10 semanas	não informado
[123]	não informado	6 horas por dia, 5 dias por semana
[116]	12 semanas	20 horas por semana
[12]	10 semanas	2 horas por semana
[129]	12 semanas	não informado
[115]	não informado	240 horas de trabalho por estudante
[128]	16 semanas	4 horas por semana em POO e IHM, 3 horas por semana em TS1
[119]	2 semestres de 16 semanas cada	não informado
[120]	não informado	100 horas de trabalho por estudante
[117]	16 semanas	2 horas por semana

Intayoad [128] descreve uma distribuição de horas diferente para três disciplinas e uma unidade de ABP, uma vez que sua proposta é interdisciplinar. Nesta proposta participam os alunos de uma disciplina de Programação Orientada a Objetos (POO), Interação Humano Computador (IHC) e Tópicos Seleccionados I (TS1). As horas em sala podem ser usadas tanto para aulas teóricas, quanto para construção do projeto ou realização de atividades de laboratório.

Na abordagem apresentada por Kilamo et al. [117], as horas em sala de aula são bem definidas, sendo a primeira delas para discutir algum tópico específico do conteúdo teórico da disciplina e a última para discussão relacionada às contribuições feitas pelos alunos ao projeto durante a semana. Aibin e Hunter [112] também descrevem uma possível divisão de tempo que pode ser feita para uma abordagem ABP, embora nem todas as horas sejam dedicadas às reuniões de sala de aula. Neste trabalho, é mencionado duas aplicações de ABP que possuem tempo reservado para reunião do time de alunos, encontros de sala de aula e encontro com o cliente. O que marca a diferença entre as duas aplicações é um tempo reservado para encontros com o tutor.

Uma outra forma diferente de utilizar o tempo disponível é apresentada no trabalho de Ilkan et al. [131]. Os autores desenham todo o currículo do qual a unidade de ABP faz

parte para um período de dois semestres e um treinamento de 40 dias dedicados à ABP. No treinamento, os alunos passam um tempo na indústria em busca de um problema que deve ser definido/resolvido no tempo proposto. Já no primeiro semestre, eles marcam reuniões semanais com seu orientador, e no semestre seguinte desenvolvem a solução em uma disciplina específica.

3.3.3 QS.3 - Quais modelos e/ou técnicas de engenharia de software são usados no projeto?

Um modelo de ABP possui alguns princípios, mas em essência é genérico para que possa comportar unidades letivas de qualquer disciplina [28, 70, 26, 27]. Embora a produção de software dentro de uma ABP deva seguir os mesmos princípios de um ABP convencional, frequentemente ela é conduzida utilizando um processo de desenvolvimento de software e técnicas de engenharia de software. Isso se torna ainda mais importante quando a ABP é um meio para o ensino do próprio processo como no caso apresentado por Martinez-Arias e Gerardo [119].

Um modelo bastante comum para instanciações de ABP é o *Scrum* [127, 112, 113, 133, 121, 115, 12, 125, 123]. O *Scrum* é capaz de prover controle sobre o progresso do projeto e de garantir um ritmo constante de desenvolvimento. Sua configuração nem sempre é a mesma em todas as aplicações, havendo divergências no tamanho da *Sprint* (de 1 a 4 semanas) [115, 12, 133, 133, 121, 127, 123], e na forma como os eventos *Sprint Planning*, *Daily Scrum* (este sendo aplicado semanalmente e não diariamente), *Sprint Review* e *Sprint Retrospective* são conduzidos [115, 12, 123, 121, 125]. Embora Martinez-Arias e Gerardo [119] não usem o *Scrum* no maior projeto de sua unidade ABP, faz uso dele em conjunto com o *eXtremme Programming* (XP) para ensinar métodos ágeis aos alunos por meio de pequenos projetos que também são autênticos.

Algumas técnicas isoladas do XP também podem ser utilizadas dentro de uma unidade de ABP para viabilizar tanto o aprendizado quanto a construção de um software. Os autores dos estudos primários identificados nessa RSL fazem uso de integração contínua [127, 133, 117] e *test-driven development* [120]. Existem relatos do uso de *ticket driven development* [125], acompanhamento de *bugs* [117, 115, 125], rastreamento de trabalho executado [113, 130, 120, 126, 129], e ferramentas de “desenvolvimento puxado” como o *Kanban* [133, 123, 127].

Embora os métodos ágeis sejam populares em unidades de ABP para desenvolvimento de software, Marques et al. [129] argumenta que o uso de métodos prescritivos são mais recomendados para alunos inexperientes e faz a aplicação de um modelo chamado *Simple Software Process*. Em seu modelo de Fábrica de Software, Pariatra e Montañó [114]

propõem o uso do modelo Cascata (*Waterfall*) e sugere a divisão dos alunos em grupos especializados (Desenvolvimento, Documentação, Projeto e Teste). Além dele, Intayoad [128] também faz uso de um método prescritivo, providenciando a divisão de alunos em tarefas especializadas de acordo com a disciplina que cursam e Martinez-Arias e Gerardo [119] declaram o uso do RUP (*Rational Unified Process*) em sua abordagem.

Devido a especificidade dos projetos desenvolvidos pelos alunos em seu estudo, Rick et al. [122] desenvolveram um processo de desenvolvimento diferente dos usuais na indústria de software. Na aplicação de ABP, os alunos desenvolvem artefatos de “mini-mundo” com modelos de ambientes reais para serem usados para motivar alunos de escolas primárias e de ensino médio a gostar de informática. Para isso, é necessário que o processo de desenvolvimento se divida em duas fases, denominadas de estágio pré-educacional, onde os participantes da ABP fazem incursões etnográficas no ambiente real de onde se derivará o “mini-mundo” para coletar material e o estágio educacional, onde os participantes da ABP contextualizam estes materiais em um ambiente lúdico educacional desenvolvido por meio da plataforma *Greenfoot*.

Finalmente, em alguns casos, processos criativos podem ser usados para emergir ideias inovadoras dos participantes da ABP, entre os quais é possível citar os *hackathons* [120], *brainstorming* e *Design Thinking* [124]. Como também, há evidências do uso de estratégias de desenvolvimento distribuído [127, 125, 116, 115, 117], justificada pela dificuldade em se manter os alunos todos os dias trabalhando juntos [125].

3.3.4 QS.4 - Como são formados os grupos de alunos que participam do desenvolvimento de software em sala de aula?

A ABP é uma atividade substancialmente cooperativa que envolve a organização entre grupos de alunos para atingir objetivos complexos e fundamentados em um contexto do mundo real. Este é um aspecto importante das unidades de ABP: elas auxiliam o desenvolvimento de habilidades não técnicas importantes na indústria de software que só podem ser construídas por intermédio da aprendizagem social (Seção 2.1.5). Softwares de qualidade são produzidos através de equipes maduras em um esforço de cooperação mútua e isso não é diferente para as equipes de desenvolvimento situadas em ambientes acadêmicos. No entanto, os professores dessas unidades podem enfrentar problemas com turmas com uma grande quantidade de alunos, visto que recomenda-se adotar a formação de pequenos grupos [113], por causa dos problemas que isso pode causar à comunicação e à própria coordenação do trabalho.

A maioria dos trabalhos selecionados na RSL dividem os alunos em grupos de 2 a 5 pessoas [131, 116, 118, 112, 130, 128, 124, 114, 115, 121, 12], cada qual fazendo sua

própria versão do projeto. Em alguns casos, esses projetos vêm de uma mesma coleção de requisitos [113, 116, 118, 114, 127], podendo ter como estratégia a divisão dos grupos para tratar de um mesmo projeto, com cada aluno trabalhando em um conjunto de recursos do artefato [127] ou em aspecto transversal deste, como documentação, *design*, desenvolvimento e teste [128]. Em outros são projetos diferentes para clientes diferentes [131, 112, 130, 128, 124, 115, 12], para os quais há a possibilidade de adição de um elemento de competição como forma de motivação aos alunos e de obtenção de melhores resultados nos projetos [113].

Nem sempre esses grupos são formados por alunos de uma única disciplina. No trabalho apresentado por Intayoad [128], alunos de POO, IHC e TS1 formavam grupos interdisciplinares. Rupakheti et al. [127] apresentam uma aplicação específica da ABP de um único projeto multiplataforma unindo alunos das disciplinas de Engenharia de Requisitos, Projeto de Software e Manutenção de Software. Nesse projeto, as equipes foram distribuídas ao longo de diferentes times de desenvolvimento para diferentes plataformas (*Android*, *iOS*, *Web* e *Backend*), além de um grupo de Garantia de Qualidade.

Foster et al. [118] difere dos demais trabalhos apresentados no sentido de que vários grupos de alunos foram se revezando ao longo do tempo em um mesmo projeto, obrigando-os a terem que gerenciar o conhecimento de um grupo para o outro. No caso do trabalho de Cadenas et al. [116], os primeiros projetos alimentaram os projetos posteriores e também tiveram os desafios da gestão do conhecimento entre os grupos de projetos ao longo do tempo.

Rupakheti et al. [127] apresentam uma maneira não usual de se formar os times, permitindo que os grupos façam propostas de soluções de software para serem desenvolvidos e tentem convencer os demais colegas a trabalhar com eles em suas propostas. É uma abordagem com certa similaridade ao apresentado por Detmer et al. [130] na aplicação de ABP para uma disciplina de Linguagem de Programação em Java, onde os alunos selecionados como líderes indicaram com quem gostariam trabalhar e com quem não gostariam de trabalhar.

3.3.5 QS.5 - Houve mudança no rendimento acadêmico dos alunos após participarem das atividades práticas de aprendizagem propostas durante a unidade letiva?

Os benefícios da ABP para o desempenho acadêmico dos alunos são bem conhecidos [70, 22, 26, 28]. Embora seja importante existir um certo cuidado com o objetivo de transformar uma unidade de ABP em um processo meramente tecnicista de produção de software que negligencia o aprendizado dos alunos, uma ABP bem realizada pode resultar

em benefícios tanto para os alunos, quanto para os clientes que se disponibilizaram a estabelecer parcerias com a academia no intuito de auxiliar a execução de uma unidade letiva desta natureza. Em termos acadêmicos, algumas evidências demonstram benefícios de aprendizado e retenção de conteúdo. No trabalho de Schefer-Wenzl e Miladinovic [120] é relatado que a percepção dos alunos sobre seu próprio aprendizado em uma unidade de ABP para desenvolvimento *Android* aumentou em torno de 53% e teve um desvio padrão desta percepção reduzida em 37% com base em um estudo que usou aplicações de pré-teste e pós-teste. Nesta unidade de ABP, 70% dos alunos terminaram o curso com notas boas ou excelentes e o conhecimento em *Android* aumentou 5,8 vezes, o que combina com a auto avaliação dos alunos que inicialmente se classificaram com conhecimento básico em sua maioria, mas posteriormente obtiveram um expressivo número de 76% que se consideraram avançados no conhecimento de *Android*.

No trabalho apresentado Stozhko et al. [132] é realizada a integração da ABP com os conceitos de química analítica. Os resultados apresentam que 79,5% dos alunos que receberam instrução dessa maneira atingiram os níveis médio e superior da taxonomia de Bloom revisada [134]. Este número representa uma quantidade maior que a quantidade de alunos que receberam instrução tradicional, levando a crer que eles eram melhores em transpor o conhecimento para contextos diferentes, combinar elementos dentro de um novo padrão e além disso, fazer julgamentos de acordo com uma série de critérios.

Intayoad [128] também demonstra avanços na retenção do conhecimento por parte dos alunos, quando por meio de teste estatístico mostrou que a execução da abordagem de ABP integrada com as disciplinas de POO e IHC saiu de uma média de 5,03 para 7,58 com uma amostra de $n = 61$, na primeira disciplina e 7,79 para 11,09 em uma amostra de $n = 64$, na segunda disciplina. Estes resultados reforçam a ideia de que a ABP possui múltiplos benefícios que podem ser aproveitados por professores de áreas relacionadas ao desenvolvimento de software, mesmo que ela seja executada tendo como referência a construção de um software autêntico.

3.3.6 QP - Como aplicações de *Aprendizagem Baseada em Projetos* para *projetos de desenvolvimento de software durante os cursos de graduação* tornam um *software autêntico*?

Levando em consideração as respostas encontradas às QS, elucida-se que tornar um software autêntico em uma unidade de ABP envolve múltiplos fatores, dentro os quais é possível encontrar o suporte educacional aos alunos, a montagem adequada de um processo de engenharia de software e o tempo disponível para a execução desta unidade.

Foi possível identificar que o *Scrum* destaca-se como o método com mais aplicações, levando crer que sua adoção pode ajudar na construção da autenticidade do artefato [127, 112, 113, 133, 121, 115, 12, 125, 123]. A Tabela 3.5 apresenta a proporção de uso do Scrum em relação a outros processos de softwares relatados nos estudos primários identificados nesta RSL.

Tabela 3.5: Uso dos processos de desenvolvimento de software

	Scrum	Processo Prescritivos
Quantidade	10	5
Referências	[127], [112], [113], [133], [121], [115], [12], [125], [123], [119]	[119], [129], [114], [128], [122]

Embora Marques et al. [129] argumentem a favor de métodos prescritivos para alunos não experientes com a disciplina de engenharia de software, Yuen [12] descreve o mecanismo de Condição de Satisfação, relacionado com o modelo ágil de entrega por iterações, como sendo uma importante ferramenta para avaliação da autenticidade. No caso do trabalho de pesquisa relatado nesta dissertação, fazer uso da condição de satisfação faz sentido, uma vez que ela serve de métrica para a verificação de requisitos de UX explicitamente definidos. O estudo apresentado por Yuen [12] demonstra bons resultados da produção de softwares autênticos que foram instanciados em contextos reais e alimentaram outras pesquisas.

Marques et al. [129] fazem grande contribuição para esta dissertação ao demonstrar os resultados de seu estudo com uma amostra representativa indicando os benefícios de se manter o monitoramento das equipes de estudantes durante o desenvolvimento do software que é um artefato de uma unidade de ABP, bem como descrevendo um modelo de reflexão aplicado com sucesso em seu estudo, que possui objetivos similares à retrospectiva das *sprints* do Scrum.

Muitos trabalhos coletados nessa RSL apontaram a importância de se fazer uma divisão dos grupos de alunos durante as atividades de construção de software em uma unidade de ABP [131, 116, 118, 112, 130, 128, 124, 114, 115, 121, 12], além disso [113] argumenta que a competição entre os alunos traz benefícios úteis para o alcance de um artefato autêntico. Esses trabalhos dão *insights* de como estruturar a unidade de ABP proposta no Capítulo 5.

Capítulo 4

Estudos de Caso Exploratórios

Os estudos de casos exploratórios são uma variação dos estudos de caso descritivos. Eles podem ser aplicados quando se tem pouco conhecimento sobre o domínio do problema que se deseja resolver e não se sabe exatamente quais as principais hipóteses a serem confrontadas por um estudo mais formal [135]. A observação e os questionários disponibilizados aos alunos foram as principais fontes de coleta de dados. Para o cenário dessa pesquisa, nas revisões iniciais da literatura, poucos trabalhos que relatassem o uso da *Design Sprint* em sala de aula foram encontrados e nenhum deles propunha alinhá-la à ABP. Por isso foi necessário que, antes de tomar a decisão de estudar com mais profundidade a forma como a *Design Sprint* poderia contribuir para a ABP nos cursos de desenvolvimento de software, a *Design Sprint* fosse testada dentro de um cenário realista de aula, com todas as restrições que um cenário como esse enfrenta. Assim, antes de determinar a proposta, e em paralelo com a RSL e o estudo mais aprofundado das questões acerca da UX, foi iterada duas versões de uma *Design Sprint* adaptada para cenários educacionais no formato de estudos de casos exploratórios, de forma que a cada rodada, houvesse um aumento no conhecimento das questões particulares de uma *Design Sprint* adaptada para o cenário educacional.

4.1 Estudo de caso 1: ABP usando *Design Sprint* em uma disciplina de Engenharia de Requisitos

O modelo de *Design Sprint* proposto por Knapp et al. [2] foi adaptado para ser executado durante a disciplina regular de Engenharia de Requisitos do curso Superior em Análise e Desenvolvimento de Sistemas (TADS) do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Campus Formosa. Esta disciplina se situa no quinto semestre do curso que é composto, no total, por seis semestres, e possui como pré-requisito a

disciplina de Engenharia de Software. Duas aulas dessa disciplina foram usadas, uma para a aplicação das etapas denominadas Entender, Encontrar Soluções / Desenhar, Decidir e Prototipar e outra para a etapa de Validação com usuários finais. Participaram desse estudo de caso 13 alunos, sendo 10 deles do sexo masculino e 3 deles do sexo feminino, 1 servidor do setor de Coordenação de Integração Empresa-Escola (COSIE-E) do IFG - Campus Formosa e o pesquisador no papel de facilitador.

O objetivo geral desse estudo de caso foi *Explorar a Design Sprint e suas ferramentas em sala de aula no menor intervalo possível* e teve como questões de pesquisa:

1. Como a a *Design Sprint* se comporta como ferramenta de aprendizagem ativa na sala de aula?
2. Quais prováveis benefícios é possível extrair da *Design Sprint* como ferramenta de aprendizagem ativa?

Nenhuma instrução sobre o que é a *Design Sprint* e como ela funciona foi dada aos alunos na intenção de observar a capacidade de aprendabilidade da Design Sprint em seu próprio processo de execução, o que pode ser verificado nos resultados sobre o aprendizado sobre a Design Sprint reportados no questionário na Seção 4.1.6. Essa decisão foi tomada considerado que, na ABP, a reflexão sobre os problemas, a formulação de objetivos e a prototipagem ocorrem nos primeiros dias de aula, como proposto por Bender [28]. Nesse momento, os alunos de alguns cursos de análise e desenvolvimento de software provavelmente possuem pouca instrução em conceitos teóricos mais profundos sobre prototipagem, desenvolvimento centrado no usuário e elicitação de requisitos.

Para este estudo de caso, com base em uma revisão da literatura exploratória que incluiu artigos, os livros de Knapp et al. [2] e Banfield et al. [31], uma agenda foi construída com base nas ferramentas determinadas como mais representativas de cada passo definido pela *Design Sprint* [2]. Essa agenda foi:

- **Pré-Sprint**

- Escolha do desafio da *Sprint*
- Coleta e abstração do fluxo do negócio
- Convite aos especialistas do negócio

- **Entender**

- 10 min - Contextualização
- Palestras relâmpagos sobre os problemas e desafios
 - * 5 min - Palestra relâmpago feita pelo especialista

- * 5 min - Palestra relâmpago feita pelos estudantes
- 5 min - Explicação do fluxo do processo mapeado
- *How Might We (HMW)*
 - * 5 min - Explicação
 - * 10 min - Escrita do HMW
 - * 5 min - Votação silenciosa
 - * 5 min - Ordenação dos HMW

- **Encontrar soluções / Desenhar**

- *Crazy 8's*
 - * 5 min - Explicação
 - * 5 min - Desenho rápido de oito variações
 - * 10 min - Esboço final de uma solução

- **Intervalo**

- **Decidir**

- Votação silenciosa
 - * 5 min - Explicação
 - * 30 min - Processo de decisão

- **Prototipar**

- **Pós-*Sprint***

- **Validação com usuários finais**
- Resposta aos questionários

4.1.1 **Pré-*Sprint***

Com o objetivo de abordar um problema específico e de interesse dos alunos, um dos pilares da educação construcionista [20], foi lhes perguntado qual era o maior problema que eles enfrentavam durante o curso. A maioria dos alunos relataram ter alguma dificuldade de encontrar uma empresa para a realização de estágio. Com isso, foi decidido abordar esse problema neste primeiro estudo de caso. Foram coletadas informações sobre o fluxo atual do processo de estágio adotado pela Instituição junto ao órgão responsável pelo estágio (COSIE-E). O coordenador do COSIE-E foi convidado para participar da *Design Sprint* e executar o papel de especialista do negócio durante sua aplicação.

Sua participação foi essencial, dado que seria ele era a pessoa que mais estava a par dos problemas de estágio que os alunos disseram passar. O fluxo do processo de estágio foi coletado e desenhado seguindo as recomendações do fluxo de atividades a serem executadas no primeiro dia de uma *Sprint* [2]. Essas atividades correspondem às fases iniciais da segunda-feira (Entender) de uma *Design Sprint* convencional e foram conduzidas com o objetivo de obter as ideias iniciais para resolver o problema.

4.1.2 Entender

Na sala de aula, o objetivo da *Sprint* foi escrito no quadro branco e foi explicado aos alunos e ao especialista do negócio como ela iria funcionar. Esse foi o momento de contextualização. Assim, foram conduzidas as **palestras relâmpago**, das quais surgiram as seguintes considerações:

- **Especialista:** (1) O estágio não é obrigatório, portanto, a empresa deve pagar um salário ao estagiário; (2) Há falta de contato com o Centro de Integração Empresa-Escola (CIEE) e com o Instituto Euvaldo Lodi (IEL) pelo IFG e estudantes.
- **Alunos:** (1) Os estagiários são contratados para realizar trabalhos que não estão diretamente relacionados as atividades do curso; (2) Os empregadores exigem conhecimentos avançados que os alunos ainda não obtiveram no curso; (3) As empresas não conhecem as especificidades do curso e pensam que o curso de informática prepara os alunos para trabalhar exclusivamente com infraestrutura; (4) Empresas de desenvolvimento são escassas em Formosa; (5) Há uma falta de estrutura nas empresas existentes, que são pequenas e carecem de demanda para o desenvolvimento de softwares complexos; (6) Não há conexão do IFG com o Site Nacional de Emprego (SINE); (7) As empresas locais desconhecem a existência do curso TADS na cidade; (8) O IFG não conhece empresas que possam oferecer estágio aos seus estudantes.

Com base nessas considerações, foram definidas as seguintes questões de pesquisa para serem abordadas na *Sprint*: **É possível despertar o interesse das empresas locais pelos estagiários do TADS, mesmo que elas tenham que efetuar o pagamento de salário? É possível melhorar a comunicação entre o IFG e as empresas interessadas em ter os estudantes como estagiários?** Então, o fluxo resumido do processo de estágio no quadro branco foi desenhado e cada etapa foi explicada.

Durante o *How Might We (HMW)*, os participantes da *Sprint* escreveram o máximo de dúvidas que conseguiram obter das palestras em notas autoadesivas, sendo limitados por um cronômetro que marcava 10 minutos. Esse exercício produziu cinquenta e três notas HMW. Na **votação silenciosa** cada aluno recebeu dois adesivos de bolinha para escolher

quais as questões escritas eram as mais importantes para o problema em questão. A regra dizia que eles poderiam escolher qualquer nota HMW para votar e, inclusive, votar duas vezes em uma mesma nota, se quisessem. Trinta e nove notas não receberam votos, sete delas receberam um voto, quatro receberam dois votos, uma recebeu três votos, uma recebeu cinco votos e uma delas recebeu seis votos. Os cartões que não receberam votos ou receberam apenas um voto foram desconsiderados para as etapas seguintes. A lógica dizia que, como cada participante podia votar em seu próprio cartão, provavelmente esse tinha recebido voto do próprio autor ou era tão irrelevante para os outros que deveriam ser colocados de lado e recuperados em iterações subsequentes apenas se necessário.

Tabela 4.1: Notas do How Might We (HMW)

How Might We	Conteúdo das notas
Notas com 2 votos	Como poderemos criar uma comunicação entre empresas e estudantes interessados em estágios através de um sistema em que uma empresa procura estagiários e estagiários à procura de empresas?
	Como poderemos despertar o interesse das empresas em contratar estagiários?
	Como poderemos fazer COSIE-E conhecer as empresas que fornecem estágio para o curso?
	Como poderemos resolver o problema das vagas disponíveis?
Nota com 3 votos	Como poderemos atrair mais empresas para o IFG para que elas saibam como funcionam os cursos e quais as áreas de foco?
Nota com 5 votos	Como poderemos integrar o IFG e as empresas para ter uma melhor comunicação?
Nota com 6 votos	Como poderemos resolver o problema da falta de experiência dos alunos?

Encontrar soluções/Desenhar

Durante o *Crazy 8's* os participantes da *Sprint* deveriam desenhar oito variações de sua solução em um espaço de tempo de 5 minutos como forma de aquecimento mental para o desenho do esboço final. Nem todos os alunos desenharam as oito variações. Contudo, esses alunos se recuperaram na etapa do **esboço final** com desenhos melhores baseados naqueles produzidos durante o exercício do *Crazy 8's*.

4.1.3 Decisão

O processo de decisão começou após o intervalo. Devido às recomendações de intervalos nas seções da *Sprint* [2], foi decidido reservar um período de descanso com lanche. Ele foi dividido em duas etapas com aplicação de votos simples, representados pelos adesivos de

bolinha menores e supervotos representados pelos adesivos de bolinha maiores. Apenas os esboços que receberam supervotos e outro que, devido ao grande número de votos simples, foi considerado relevante pela equipe participante da *Sprint*, foram selecionados para a construção do protótipo.

4.1.4 Prototipar

Para a prototipagem, separamos os alunos em 4 grupos, onde cada um deveria desenhar protótipos em papel de uma etapa do fluxo que tinha sido contemplada com as soluções geradas no *crazy 8's*. Esses protótipos deveriam ser representações de telas que, conectadas, formariam um fluxo de navegação. Usamos o aplicativo POP 2.0 [136] para transformar as telas de papel em telas dinâmicas que seriam usadas na etapa de validação. As ideias geradas a partir da *Sprint*, culminaram em um protótipo de 14 telas que, se percorridas, levam o usuário a 5 fluxos diferentes.

A tela de login se divide entre o fluxo do aluno e o fluxo da empresa, que levam a telas iniciais diferentes. Fazendo login como aluno, a tela inicial apresenta botões onde se escolhe o curso no qual o aluno está matriculado. No protótipo, apenas o botão TADS funciona. Na tela inicial do aluno há um menu com link para uma seção contendo vídeo aulas e arquivos de texto sobre conhecimentos específicos que são cobrados nos ambientes de estágio. Esta tela surgiu do HMW mais votado que tratava do problema da falta de experiência dos alunos, este que originou a partir do apontamento realizado durante a palestra relâmpago dos alunos sobre a cobrança de trabalhos que exigiam habilidades e conhecimentos que os estagiários ainda não tinham.

4.1.5 Pós-Sprint

Tendo finalizado a aula regular, dois alunos ficaram para a fase de **validação** no dia seguinte. Esta atividade foi realizada com um prazo de trinta e seis minutos. O método de avaliação, inspirado na validação de Knapp et al. [2] foi definido da seguinte forma: (1) o protótipo era entregue para ser usado pelo usuários; (2) um dos alunos guiava a interação pedindo para que o usuário executasse tarefas específicas, enquanto o outro observava o comportamento do usuário e tomava notas em um caderninho; (3) O usuário dizia o que mais gostou, o que não gostou, o que ele diria se tivesse que recomendar o aplicativo para alguém e dava três sugestões de melhora do aplicativo. A validação foi feita com um professor da área da informática, um aluno do TADS e um técnico-administrativo que trabalha na COSIE-E, que não era o mesmo especialista do negócio da *Sprint*.

O aluno escolhido para validar foi um portador de necessidades especiais (cadeirante) e a conversa com ele trouxe bons *insights* sobre a usabilidade do aplicativo. O professor

fez algumas considerações sobre os nomes escolhidos para nomear os botões e a falta de intuitividade na interface do aplicativo. A aprendizagem sobre o problema foi bastante significativa nesta etapa. Após os alunos realizarem as três entrevistas, uma pequena conversa entre os membros da Instituição e os alunos revelou novas necessidades e, a partir delas, novas ideias foram surgindo. Entre elas:

1. Abrir a possibilidade da intermediação de serviços *freelancer* dos alunos pelo aplicativo, de forma a atender as necessidades das empresas da região, já que existem pouquíssimas delas cuja atividade-fim seja desenvolvimento de software;
2. Como um dos problemas relatados na validação foi o fato de as empresas, no geral, não conhecerem a finalidade do curso de análise e desenvolvimento de sistemas do IFG ou nem saberem que existe oferta de mão de obra de desenvolvedores e analistas de sistemas na cidade, o aplicativo poderia gerar postagens automáticas nas redes sociais com a oferta de serviços para que houvesse uma maior divulgação dos estagiários e de suas habilidades;
3. Sabendo que grande parte das empresas desejam estagiários do curso para trabalhos de infraestrutura, suporte, *help desk* e manutenção de sistemas, os quais não são especificamente pertencentes à área de conhecimento tratada no curso e que isso acontece por causa de uma falta de conhecimento dos nomes e jargões técnicos da área, o aplicativo poderia fornecer vídeos explicando o que é cada especialidade fornecida como habilidade de um aluno, levando em consideração que muitas vezes a empresa nem imagina que suas necessidades podem ser supridas por alguma ferramenta ou técnica específica aprendida em sala de aula;
4. O aplicativo proposto também poderia permitir adicionar *tags* com informações das habilidades dos alunos (como por exemplo: Programação em Java, Desenho de Interfaces, Implantação de banco de dados, Gestão de Projetos, dentre outras) e as empresas poderiam cadastrar suas necessidades (por exemplo: Desenvolver um *website*, realizar o mapeamento dos processos organizacionais, realizar a manutenção em um sistema legado, etc.). Além disso, o aplicativo poderia relacionar as habilidades dos alunos às necessidades correspondentes nas organizações/empresas locais interessadas no trabalho dos estagiários.

4.1.6 Questionário do Estudo de Caso 1

Foi fornecido aos alunos um questionário com algumas perguntas. O questionário ficou disponível por vinte e quatro horas. Para mitigar alguns problemas que a falta de confidencialidade poderia causar, os alunos responderam o questionário de forma anônima. Ao

final, foram coletadas 13 respostas. Tendo as respostas dos alunos, algumas interpretações puderam ser realizadas. O interesse por programação deste grupo de alunos é mais do que o dobro em comparação com a análise de sistemas e computação gráfica, conforme apresentado na Figura 4.1. Se essas duas áreas forem somadas, elas corresponderão a apenas 80% do índice total de interesse por programação. Isso pode indicar sinais de uma preocupação desequilibrada com a qualidade do código e qualidade dos aspectos estéticos, um dos fatores importantes em UX [88].

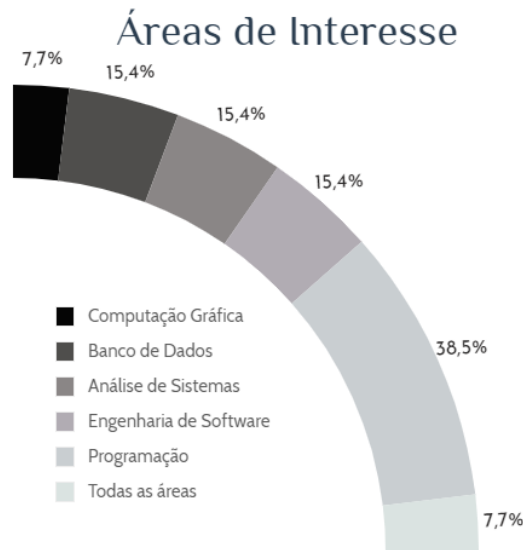
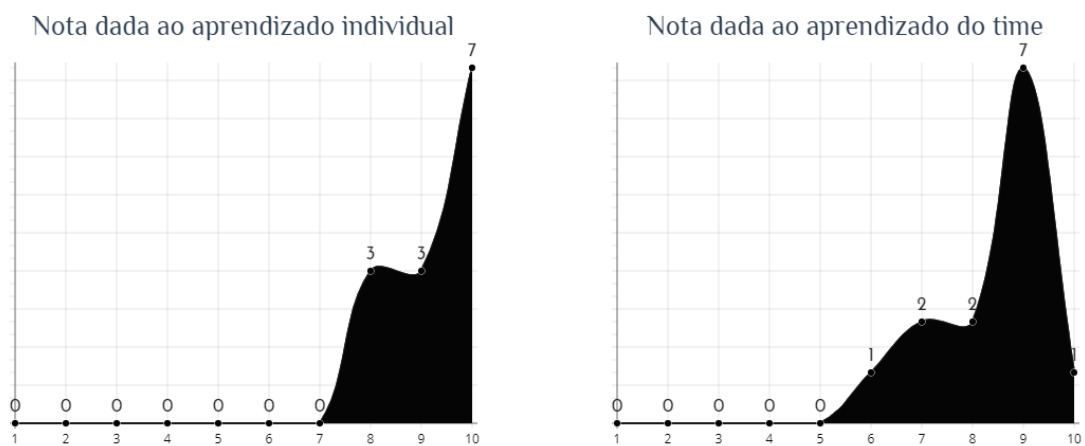


Figura 4.1: Áreas de interesse dos alunos

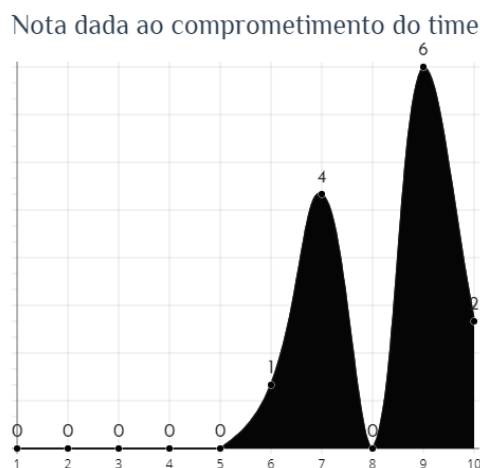
Os alunos relataram terem aprendido e terem percebido aprendizado nos colegas durante a execução da *Sprint*, embora não tenham sido tão veementes em relação ao comprometimento do time, conforme apresentado na Figura 4.2. Suas respostas em relação ao aprendizado não podem ser encaradas como comprovação final de que realmente aprenderam, pois este não era o objetivo final do estudo de caso. Ao contrário, elas devem indicar como os alunos se sentiram em relação ao próprio aprendizado. No máximo, era necessário avaliar empiricamente se os alunos compreendiam a mecânica da dinâmica enquanto as ações aconteciam. Não faria sentido cobrar deles conhecimentos teóricos sobre a *Design Sprint*, se a versão aplicada era um modelo diferente do usado na indústria.

Para a maioria dos alunos o tempo foi reconhecidamente um fator limitante. A sugestão geral foi a de que algo entre doze horas e três dias fosse mais adequado, o que é coerente, afinal, aliando as duas opções mais escolhidas poderíamos ter, por exemplo, quatro horas divididas em três dias, o que ainda é uma redução à versão mais nova da *Design Sprint*, chamada de *Design Sprint 2.0*, que sugere seis horas para cada dia de um total de quatro dias [137]. No geral, os alunos concordaram sobre a capacidade da



(a) Aprendizado Individual

(b) Aprendizado do time



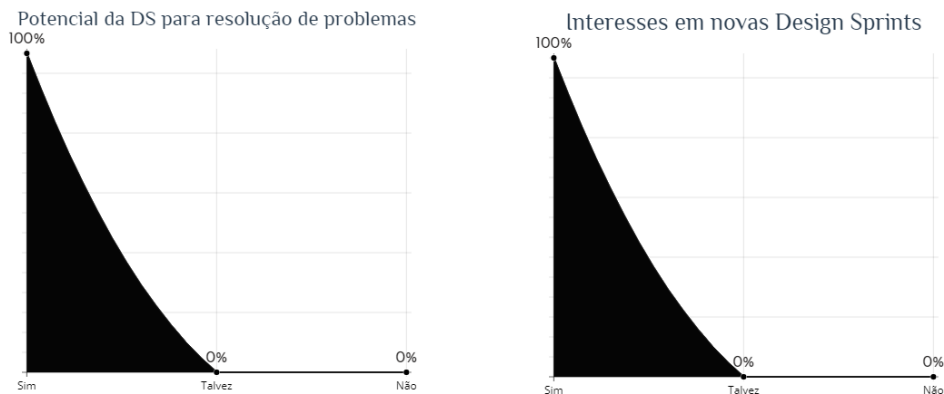
(c) Comprometimento do time

Figura 4.2: Aprendizado e comprometimento dos alunos

Design Sprint de resolver problemas de forma criativa e manifestaram unânime interesse em participar de novas aplicações de sua aplicação com propósitos educacionais, conforme apresentado na Figura 4.3. Isso representa um bom resultado, pois indica que participar dela foi uma experiência agradável aos alunos.

4.1.7 Considerações sobre primeiro estudo de caso

Tendo como referência as restrições de tempo de uma disciplina curricular, essa primeira tentativa de aplicar uma *Design Sprint* em sala de aula teve como meta fazê-la acontecer dentro da menor quantidade de tempo possível. Esse estudo de caso foi realizado sob a



(a) Potencial da *Design Sprint* para resolução de problemas

(b) Interesse em novas aplicações d *Design Sprint*

Figura 4.3: Avaliação geral da aplicação da *Design Sprint*

inspiração da versão original da *Design Sprint*, descrita em Knapp et al. [2]. Dessa forma, dado o formato prescritivo da *Design Sprint* de Knapp et al., o *mindset* que guiou essa adaptação tratou os exercícios como as unidades-chave da *Design Sprint*, e não as fases, como propõe Banfield et al. [31]. Uma revisão na literatura científica e na literatura cinzenta foi conduzida para que os exercícios mais representativos de cada fase fossem identificados, de modo que fosse possível cortar os outros. Com essa revisão, foi possível reduzir a *Design Sprint* de cinco dias para uma versão minimalista que continha o núcleo criativo (Contextualização do problema, busca por soluções e prototipação) em um dia, desde que o problema fosse identificado com antecedência e o protótipo fosse validado em outro dia.

A principal dificuldade percebida nesse formato foi o tempo reduzido. Desenvolver os exercícios da *Design Sprint* em uma versão minimalista, fez com que os limites de tempo para cada um deles fossem rigorosamente definidos. No entanto, em sua aplicação, esses limites não foram respeitados por causa da dificuldade dos alunos entenderem algumas ferramentas, da organização dos materiais necessários para a realização das atividades e, principalmente, por causa do atraso dos alunos no início da *Design Sprint* e na volta do intervalo. O efeito colateral do estouro de tempo foi uma certa inquietação dos alunos com a hora de ir embora, tendo, inclusive, alguns dos alunos realmente deixado a sala antes das atividades da *Design Sprint* acabar. Manter os alunos em silêncio nos exercícios de votação silenciosa também foi um desafio. Eles ficavam empolgados e a todo momento era necessário lembrá-los de que não deveriam conversar, para não enviesar seus pares.

Uma surpresa positiva foi o fato de que ter 13 alunos com *backgrounds* heterogêneos não tivesse impactado tão negativamente quanto foi imaginado que impactaria. No final

da atividade, notou-se que o principal benefício foi a solução não convencional que eles desenvolveram, embora ela tenha tentado resolver vários pontos problemáticos do processo do estágio e não ter mantido o foco em apenas um único ponto, conforme determinado pelo modelo do Diamante Duplo [1]. A fase mais determinante para este resultado foi o contato com pessoas reais com o protótipo em mãos durante a validação. Foi nela que a maioria dos *insights* criativos surgiram, levando a entender que o contato com as pessoas é, de fato, um dos fatores mais relevantes para um trabalho de *design* [97].

O que foi possível depreender dessa aplicação, levou a reformulações da adaptação que foi aplicado no segundo estudo de caso.

4.2 Estudo de caso 2: ABP usando *Design Sprint* em uma disciplina de Engenharia de Software

Baseando-se nos resultados obtidos com a aplicação do estudo de caso exploratório 1 (Seção 4.1), o modelo da *Design Sprint* foi adaptado para um segundo estudo de caso que foi executado durante as aulas de uma disciplina de Engenharia de Software do mesmo curso. Essa disciplina está no quarto semestre do curso e tem como pré-requisitos as disciplinas de Análise Orientada a Objetos e Programação Orientada a Objetos.

Foram usadas 4 aulas para a execução da *Design Sprint*, conforme recomendado pelos participantes do estudo de caso 1. 11 alunos participaram dessa iteração, sendo 8 deles do sexo masculino e 2 do sexo feminino, além do diretor do campus no papel do especialista na primeira aula e do pesquisador como facilitador. Nela, a *Design Sprint* aproximou-se mais ainda da proposta original de Knapp et al. [2], embora tivesse uma limitação de 3 horas por dia. Além disso, essa nova aplicação da *Design Sprint* foi distribuída ao longo de três semanas, tendo 1 dia de atividades na primeira e na terceira semana e 2 dias de atividades na segunda semana.

O objetivo geral desse estudo de caso foi o de *Explorar a Design Sprint e suas ferramentas em sala de aula em dias espalhados ao longo de mais de uma semana* e teve como questões de pesquisa:

1. Quais benefícios e desafios traz a *Design Sprint* dividida em aulas que estão em semanas diferentes?

A agenda criada para essa iteração foi:

- **Pré-Sprint**

- Escolha do desafio da *Sprint*

- Coleta e abstração do fluxo do negócio
- Convite ao especialista do negócio

- **Aula 1 - Questão âncora e imersão**

- 15 min - Palestra relâmpago sobre os desafios e problemas pelo especialista (Alunos escrevem o problema em notas autoadesivas)
- 25 min - Delimitação do escopo da *Sprint*
 - * 15 min - Explicação de notas autoadesivas e fixação delas no quadro
 - * 5 min - Agrupamento das notas autoadesivas
 - * 5 min - Votação e escolha da questão âncora
- 10 min - Escrita dos HMW
- Divisão dos grupos (*Voz do Usuário* e *Como as coisas funcionam*)

- **Aula 2 - Demonstrações relâmpago e Ideação**

- 20 min - Palestras relâmpago dos grupos (Alunos escrevem percepções em notas autoadesivas)
 - * Grupo *Voz do Usuário*
 - 7 min - Apresentação
 - 3 min - Perguntas
 - * Grupo *Como as coisas funcionam*
 - 7 min - Apresentação
 - 3 min - Perguntas
- 10 min - Votação nas notas HMW
 - * 3 min - Organização das notas HMW
 - * 5 min - Votação
 - * 2 min - Escolha do *Stakeholder* e da etapa do processo
- 15 min - Demonstrações relâmpago
 - * 12 min - Apresentação de soluções (limite máximo de 3 minutos por solução)
- 5 min - Divisão dos grupos por etapas do processo
- 30 min - Esboço em 3 etapas
 - * 10 min - Anotação livre
 - * 2 min - Explicação do *Crazy 8's*

- * 8 min - *Crazy 8's*
- * 10 min - Desenho do esboço final

- **Aula 3 - Decisão**

- 45 min - A decisão que cola
 - * 5 min - Fixação dos esboços nas paredes da sala
 - * 10 min - Processo de decisão
 - * 20 min - Críticas relâmpago
 - * 10 min - Pesquisa de intenção de votos
- 45 min - *Storyboard*

- **Aula 4 - Validação**

- 45 min - Revisão do protótipo
- 45 min - Validação final

- **Pós-*Sprint***

- Resposta aos questionários

4.2.1 Pré-*Sprint*

Neste estudo de caso, o problema foi coletado diretamente com um especialista de negócio, que para esta iteração foi o diretor da Instituição. Ele solicitou uma solução para os problemas de confiabilidade das assinatura dos documentos institucionais, que até então não eram feitos por meio de assinatura digital. Uma dimensão importante desse problema apontada pelo especialista foi o valor das soluções existentes no mercado para a adoção de assinatura digitais e dos *tokens* que precisam ser comprados para viabilizar esse processo. Nesta ocasião o fluxo do processo de assinatura foi esboçado pelo professor da disciplina e o diretor foi convidado para participar da *Sprint*. Assim, o objetivo da *Sprint* foi: “Como prover assinaturas confiáveis nos documentos institucionais sem grandes custos”.

4.2.2 Aula 1 - Questão âncora e imersão

O objetivo da *Sprint* e o mapa do processo foram desenhados no quadro branco antes de seu início. Cada aluno recebeu um bloco de notas autoadesivas e uma cartela de adesivos de bolinha. Então, tendo iniciada a aula e tendo passado as primeiras orientações sobre como a *Sprint* seria executada, o especialista do negócio realizou uma breve palestra de 15 minutos (**palestra relâmpago**) apresentando os malefícios de não ter um procedimento

padrão para a assinatura dos documentos institucionais na Instituição que não envolvessem um custo elevado e que fosse confiável. Durante a palestra, os alunos escreveram nas notas autoadesivas respostas à pergunta “Como a falta de um processo de assinatura confiável e barato afeta negativamente as pessoas?” a partir da apresentação do especialista e a partir de suas próprias percepções sobre o problema em questão.

Ao final da palestra relâmpago, os alunos colaram suas notas autoadesivas no quadro agrupando-as por categoria e relatando muito rapidamente o problema que eles identificaram como importante para a questão. Logo em seguida, os alunos usaram seus adesivos de bolinha para votarem nos problemas disponibilizados no quadro branco e o professor da disciplina formulou uma **questão âncora** seguindo a abordagem da ABP [28] de acordo com a categoria mais votada pelos participantes. Este exercício serviu para delimitar o escopo da *Sprint*.

Verificando as notas autoadesivas da categoria mais votada, os estudantes tiveram o limite de tempo de 10 minutos para formularem questões no formato HMW que estivessem estritamente relacionadas com a questão de pesquisa. As notas onde estavam escritas as questões no formato HMW foram afixadas nas paredes da sala de aula e agrupadas por similaridade temática.

Os alunos foram divididos em dois grupos com quatro alunos cada para que no interstício entre uma semana e outra pudessem entrevistar pessoas da comunidade sobre o problema e trazer informações a serem usadas na próxima aula. Estes grupos foram denominados de **Voz do Cliente** e **Como as coisas funcionam**. O grupo Voz do Cliente ficou responsável por entrevistar pessoas que envolvidas no processo de tramitação de documentos com assinatura do IFG (Grupo 1). Além disso, os grupos deviam entrevistar também pessoas consideradas especialistas em segurança da informação, para obter informações relativas à confiabilidade de uma assinatura digital (Grupo 2). Esta estratégia foi desenvolvida para mitigar o problema de falta de heterogeneidade dos alunos como equipe de uma *Sprint* sem o conhecimento necessário acerca de segurança da informação.

Como atividade adicional, os alunos deveriam realizar uma pesquisa sobre as soluções existentes na indústria e na academia que apresentassem funcionalidades úteis e/ou contribuísse para o endereçamento do problema identificado na *Sprint*. Conteúdos em vídeo também foram disponibilizados para que os alunos soubessem como conduzir entrevistas semiestruturadas para coleta de requisitos.

4.2.3 Aula 2 - Demonstrações relâmpago e ideação

Antes do início da segunda aula, o processo de assinatura digital dos documentos institucionais foi redesenhado, as notas HMW foram novamente fixadas e a questão de pesquisa foi reescrita no quadro branco da sala de aula. Desta aula em diante, o especialista do negó-

cio não participou da *Sprint*. No início das atividades da aula, os estudantes fizeram duas apresentações rápidas sobre o que aprenderam com suas entrevistas, um do grupo **Voz do Cliente** e outra do grupo **Como as coisas funcionam (Palestras relâmpago)**. Os estudantes tiveram tempo para escrever mais notas HMW durante as apresentações. Essas também foram afixadas às paredes, junto àquelas da primeira aula, sendo também agrupadas por similaridade temática.

Os alunos realizaram uma votação nas notas HMW consideradas mais importantes. As que receberam mais de um voto foram coladas em suas devidas posições no mapa do processo. A partir da identificação das regiões com mais notas, o professor escolheu o *stakeholder* e as etapas que deveriam ser seguidas na execução da *Sprint*. Na sequência, os estudantes tiveram um tempo definido para apresentarem as soluções encontradas em suas pesquisas na indústria e na academia (**Demonstrações relâmpago**). O tempo de apresentação foi limitado a 3 minutos para cada solução. Um dos estudantes foi eleito para documentar e projetar (através de desenhos) no quadro branco as funcionalidades descritas pelos participantes. Após esta etapa, os estudantes foram agrupados por etapa do mapa da jornada do usuário de acordo com suas preferências a fim de que o esboço das soluções propostas pudessem ser desenhados.

O desenho dos esboços foi realizado em 3 etapas. Na primeira etapa os estudantes observaram suas próprias anotações e as notas HMW afixadas no quadro branco e nas paredes da sala de aula e fizeram pequenos esboços experimentais em folhas A4. Na segunda etapa, os estudantes fizeram o exercício do *Crazy 8's* e por fim, na terceira etapa os estudantes desenharam os esboços finais que seriam sua proposta de solução.

4.2.4 Aula 3 - Decisão

Nos minutos iniciais da terceira aula, os estudantes afixaram seus esboços na parede da sala de aula em uma ordem que respeitava o fluxo do processo. Os estudantes de posse de sua cartela de adesivos de bolinha foram orientados a votar quantas vezes quisessem nas partes do esboço que mais gostaram e escrever em notas autoadesivas as preocupações e dúvidas existentes em relação aos esboços. Estas notas autoadesivas com preocupações e dúvidas foram afixadas embaixo do esboço correspondente.

Na próxima etapa, foram organizadas sessões de críticas aos esboços propostos. A sessão de crítica teve duração de 3 minutos para cada esboço (**críticas relâmpago**). Nessas sessões, o professor apresentou os esboços com foco nas partes marcadas pelos adesivos de bolinha. Um estudante foi escolhido para registrar as ideias em outras notas autoadesivas e afixá-las na parte de cima do esboço submetido à sessão de críticas. O professor e os estudantes revisaram as preocupações e dúvidas registradas nas notas afixadas abaixo do

esboço e, então, o criador do esboço se manifestou, comunicando o que não foi percebido em seu esboço e comentando as dúvidas e preocupações registradas pelos participantes.

Em seguida foi realizada a “**pesquisa de intenção de votos**” [2], onde os estudantes votaram com seus adesivos de bolinha para ajudar o professor no papel de Decisor a tomar uma decisão final sobre qual esboço se tornaria a base da solução proposta. Tendo decidido qual esboço iria se tornar a base do protótipo, os próximos 45 minutos foram dedicados ao desenho do **Storyboard** do esboço selecionado para a solução. O professor repassou aos estudantes as tarefas a serem executadas e estudadas para que pudessem realizar a validação da *Sprint*, conforme determina Knapp [2], ou seja, formularem o roteiro a ser seguido durante a aula para realizar a validação e desenvolverem o protótipo com uso de alguma ferramenta colaborativa online.

4.2.5 Aula 4 - Validação

A quarta aula foi dedicada à revisão e realização de ajustes no protótipo proposto e à realização da sua validação. Os estudantes foram divididos em grupos e organizaram a validação e coleta do *feedback* com diferentes perfis de usuários, entre eles, estudantes, técnicos-administrativos da instituição, professores e o próprio diretor (especialista do negócio). O processo de validação utilizado foi o mesmo do estudo de caso 1 (seção 4.1) com a diferença que cada dupla validou o protótipo com um perfil de usuário paralelamente. Ao final da entrevista de validação do protótipo, os estudantes retornaram para a sala de aula e fizeram uma reflexão em conjunto, avaliando os pontos positivos e negativos da solução, as sugestões de reformulação do protótipo e declarando o que tinham aprendido com a experiência.

A solução criada pelos alunos na execução da *Sprint* envolvia uma aplicação *Web* dividida em dois módulos: um de assinatura e validação digital que envolvia princípios de *blockchain* (*hash* composto pelo *timestamp* de cada novo documento e pelo conjunto de documentos da cadeia de blocos) e outro de validação de conteúdo de documentos físicos com base na leitura do conteúdo do documento e coleta randomizada de palavras para posterior conferência quanto à posição destas palavras no documento.

4.2.6 Questionário do Estudo de Caso 2

Neste estudo de caso, também foi aplicado um questionário online de avaliação da *Design Sprint* que ficou disponível por uma semana para ser respondido pelos participantes. Esse questionário também foi respondido de maneira anônima como forma de aumentar a confiabilidade das respostas. Nove alunos responderam ao questionário. Nenhum desses nove alunos participaram do primeiro estudo de caso (Seção 4.1). Assim como reportado

no questionário do estudo de caso 1 (Seção 4.1), o interesse dos alunos pela disciplina de linguagem de programação também permaneceu em destaque. Neste estudo de caso, 44,4% dos alunos reportaram maior interesse em programação. Com este grupo de alunos, o interesse pela disciplina de Banco de Dados também foi destaque, sendo 33,3% do total. Apenas um aluno reportou interesse na disciplina de Análise de Sistemas. O interesse por Computação Gráfica não foi sequer mencionado pelos participantes. A Figura 4.4 apresenta estes resultados.

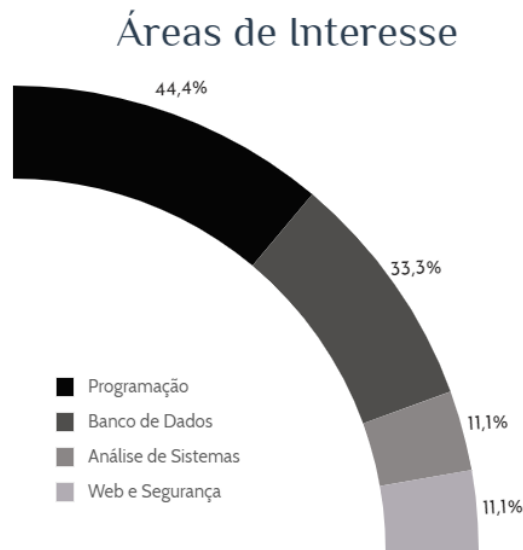
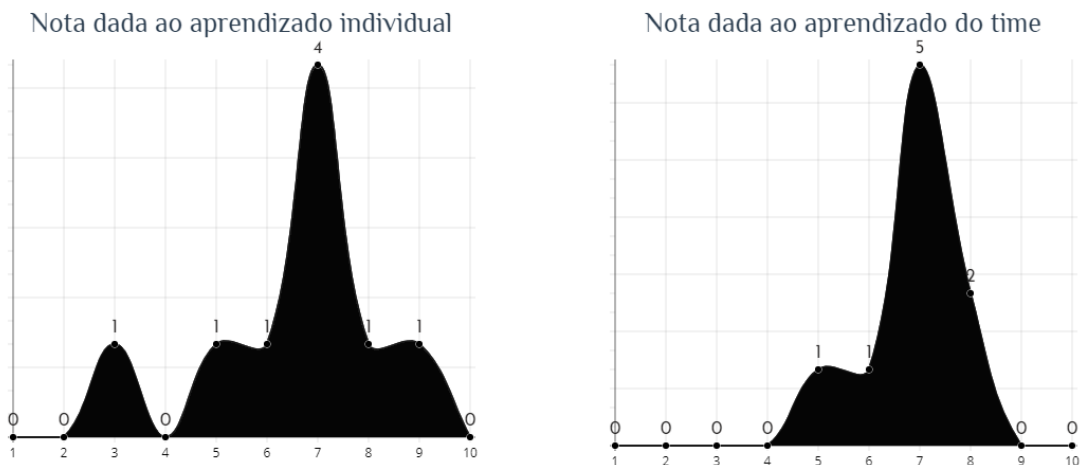


Figura 4.4: Áreas de interesse dos alunos

A distribuição das notas em relação ao próprio aprendizado dos participantes, do aprendizado dos colegas e do comprometimento dos colegas girou em torno da nota 7. Este é um resultado inferior ao apresentado no estudo de caso 1 (Seção 4.1). A Figura 4.5 apresenta a distribuição das atribuições de notas dadas pelos estudantes.

A avaliação positiva em relação à *Design Sprint* por parte dos estudantes também permaneceu no segundo estudo de caso, embora tenha sido bem menor que a avaliação do grupo do primeiro estudo. A Figura 4.6 apresenta este resultado. Diferente do estudo de caso 1 (Seção 4.1), alguns alunos desse estudo de caso apresentaram traços de introspecção. Estes alunos não tiraram muito proveito das atividades da *Design Sprint* que são colaborativas por natureza e é, portanto, razoável imaginar que as notas baixas possam ter origem da avaliação destes alunos. Uma razão para a timidez deles pode ter sido resultado da participação do Diretor da Instituição durante algumas atividades.

Mesmo com o aumento considerável do tempo disponível para a aplicação da *Design Sprint*, o tempo ainda continuou sendo considerado um fator limitador na opinião dos estudantes. Essa ponderação por parte dos estudantes pode estar relacionada com a



(a) Aprendizado individual

(b) Aprendizado do time



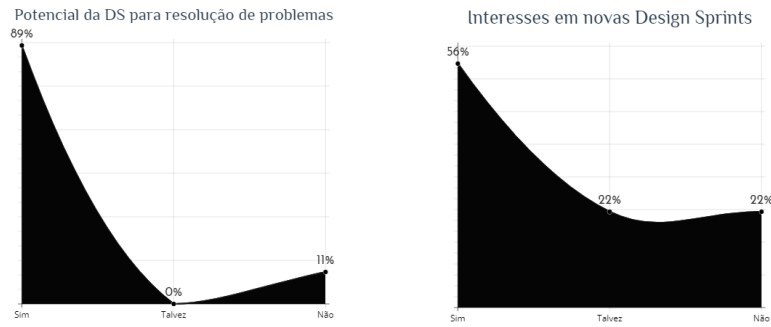
(c) Comprometimento do time

Figura 4.5: Aprendizado e comprometimento dos alunos

percepção dos alunos sobre as atividades naturalmente aceleradas de uma *Design Sprint*. Essa é uma questão que precisa ser melhor investigada em outros estudos de caso.

4.2.7 Considerações Sobre o Estudo de Caso 2

Essa nova iteração foi projetada para lidar com os problemas apontados no primeiro estudo de caso (Seção 4.1), principalmente àqueles referentes ao tempo e também inserir estratégias para mitigar efeitos da homogeneidade dos alunos, bem como potencializar os efeitos da inspiração e criatividade obtidos por meio da interação com pessoas externas à *Design Sprint*. Com mais tempo disponível, foi possível colocar mais exercícios da versão



(a) Potencial da *Design Sprint* para resolução de problemas

(b) Interesse em novas aplicações da *Design Sprint*

Figura 4.6: Avaliação geral da aplicação da *Design Sprint*

original da *Design Sprint* [2]. No entanto, foi necessário espalhar a *Design Sprint* durante 3 semanas, concentrando 1 aula de 3 horas na primeira semana e na terceira semana e 2 aulas de 3 horas na segunda semana. Ter um tempo maior disponível nas aulas para fazer as atividades foi muito importante, afinal, a exemplo do estudo de caso 1 (Seção 4.1), o tempo definido para uma boa parte dos exercícios da *Sprint* foi extrapolado. Nesse estudo de caso, o tempo total das atividades foi realizado para que ele fosse bem menor que o tempo disponível e isso ajudou a fazer correções, adaptações, dar novas aplicações e refazer exercícios que não foram bem executados.

Outro ponto positivo desse ajuste, foi ter a possibilidade de que os alunos entrassem em contato com pessoas de fora da *Design Sprint* entre a semana 1 e a semana 2 para fazer as entrevistas dos grupos Voz do Cliente e Como as Coisas Funcionam. Esse contato com as pessoas foi muito benéfico, afinal foi daí que surgiu a maior parte das ideias que ficaram na versão final do protótipo. Nota-se, portanto, como benefício recorrente o fato de que o contato com pessoas de fora da *Design Sprint* traz os melhores *insights*. Na primeira iteração, esses grandes *insights*, que são muito maiores do que aqueles que os alunos tiveram dentro das atividades de sala de aula, vieram na fase de validação. Esse é um ponto interessante, dado que Knapp et al. [2] e Banfield et al. [31] mencionam a atividade de validação como a parte mais importante da *Design Sprint*. É nessa fase de validação que o time da *Sprint* de Knapp et al. [2] entra em contato com o real usuário. Até então, nessa versão, o usuário é representado por algum especialista que integra o time da *Sprint* (geralmente, algum membro do setor de *marketing* ou relacionamento com o cliente) e todos os novos *insights* que terão, serão aplicados na solução em novas iterações da *Sprint*, o que pode acabar fazendo com que os melhores resultados raramente aconteçam com só uma semana (o tempo correspondente a uma única *Sprint*). Banfield et al. é mais diligente na inserção do usuário já dentro do processo de geração de ideias, propondo que se comece

a *Sprint* com uma série de dados de clientes coletados em pesquisas anteriores, além de sugerir que uma etapa da fase Entender seja usada para conversar com os usuários, embora também não tenha deixado claro a maneira como se deve fazer isso dentro de uma restrição de horário tão rigorosa quanto o de uma *Design Sprint*. Entrar em contato com outras ferramentas concorrentes da solução que eles estavam projetando também foi inspirador e ajudou a potencializar os exercícios de criatividade que viriam depois, como o *Crazy 8's*.

O ponto negativo dessa divisão da *Sprint* em mais de uma semana foi ter atrasos dos alunos no início das aulas e, em alguns casos, até mesmo falta. Dessa maneira, o grupo que começou a *Design Sprint*, não era o mesmo grupo que estava presente em todas etapas. Isso demonstra que não é muito fácil manter os alunos engajados a todo momento durante aplicações longas. Outro efeito negativo de ter a *Design Sprint* em mais de um dia foi não ter o especialista disponível para a *Sprint* durante toda ela, mas apenas no primeiro dia, onde o entendimento sobre o problema foi aprofundado. A falta de um foco claro no problema foi outro efeito indesejado dessa nova aplicação, conforme pode ser verificado na proposição de dois módulos, cada qual com uma solução diferente para o problema proposto.

As observações feitas nesse estudo de caso também demonstraram que alguns alunos que possuem personalidade introspectiva podem não se beneficiar das atividades da mesma forma que aqueles mais extrovertidos. Houve uma aluna com o perfil introspectivo que não participava ativamente das atividades e que, provavelmente, avaliou a *Design Sprint* com notas menores como demonstrado na seção 4.2.6.

Por fim, em ambos os estudos de casos, mostrou-se que é necessário que o professor facilitador tenha competências em gerenciamento de grupos, gerenciamento de tempo, capacidade de improviso e habilidades relacionadas à motivação. Realizar trabalhos como esse em substituição às aulas tradicionais elevam a carga de trabalho do professor tanto nos momentos dentro de sala de aula, quanto nos momentos fora dela e precisa de uma certa persistência. O fato de a coleta e estudo do problema e do fluxo das pessoas que sofrem com ele ser colocado fora da *Design Sprint* é extremamente necessário, pois traz uma certa dependência de organização e reorganização prévia por parte do professor que pode não encontrar o melhor problema a ser abordado ou o melhor especialista de negócio para participar das atividades da *Design Sprint*. Professores com alta carga de trabalho podem querer declinar da realização dessas atividades.

O próximo capítulo fará uso das considerações geradas nesses dois estudos de caso e da contribuição dada pela RSL (Capítulo 3 para construir uma proposta de ABP que contemple a UX como um dos atributos de autenticidade.

Capítulo 5

Proposta

A partir dos resultados obtidos com a Revisão Sistemática de Literatura (Capítulo 3) e das considerações feitas por intermédio da observação dos estudos de caso exploratórios (Capítulo 4) foi possível responder às questões de pesquisas que foram elaboradas para guiar esta pesquisa e também propor um modelo de Aprendizagem Baseada em Projetos (ABP) que considere requisitos de Experiência do Usuário (UX) para a construção de artefatos de software. A Figura 5.1 apresenta uma visão macro de uma unidade de ABP, cuja etapa de ideação e prototipação, comumente feita por *brainstorming*, foram substituídas pela *Design Sprint*.

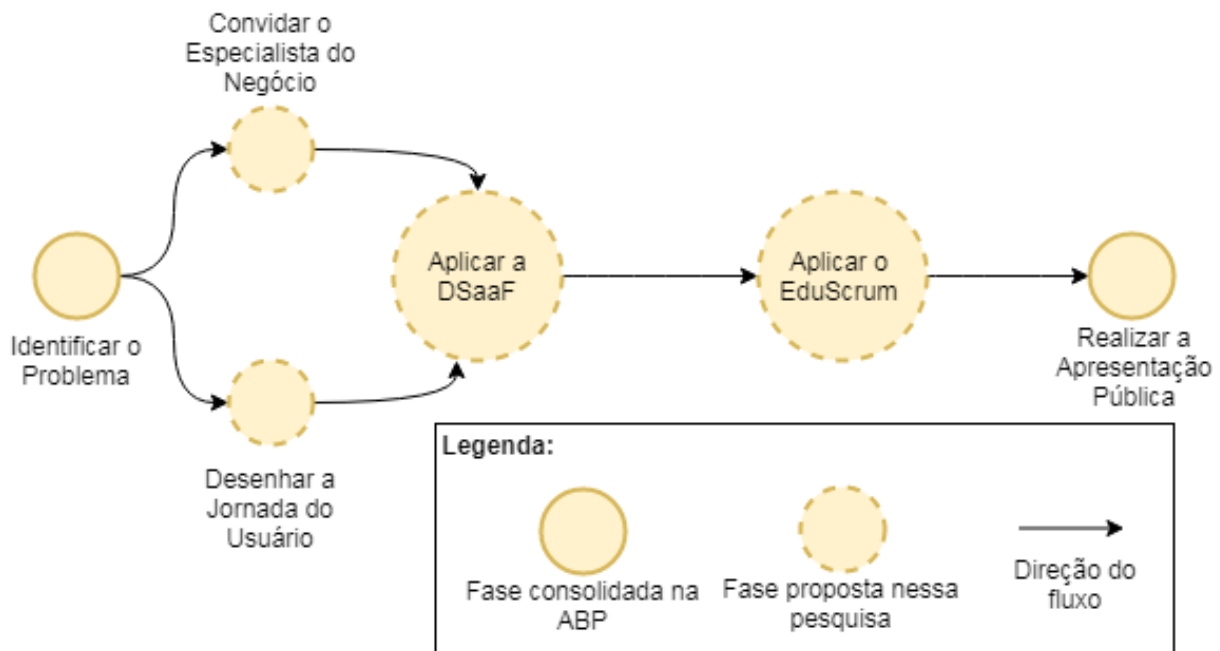


Figura 5.1: Modelo macro da proposta

O modelo proposto para a execução de uma ABP que considere os requisitos de UX faz uso da proposta de Bender [28] para a ABP, da *Design Sprint* descrita por Banfield et al.

[31] em que é considerada como um *framework* (DsaaF - *Design Sprint as a Framework*) e não como um modelo rígido e do *eduScrum* [138]. As fases do modelo proposto são descrição nas Seções 5.1, 5.2, 5.3, 5.4, 5.5 e 5.6.

Nesse modelo o uso da expressão *framework* denota uma estrutura conceitual recomendada, mas adaptável às necessidades particulares de contextos diferentes desde que se siga a macro estrutura determinada pelas fases que nomeia o título de cada uma das seções desse capítulo.

5.1 Identificar o Problema

A primeira fase do modelo acontece com a coleta do problema a ser resolvido. Embora Bender [28] não trate a ABP como uma série de fases a serem seguidas, ele deixa claro que é necessário partir de um problema que, posteriormente resultará em uma questão motriz [28]. Essa característica é compartilhada por outros autores da literatura sobre ABP conforme descrito na Seção 2.1.5.

A definição de autenticidade (Seção 2.1.6) deixa claro que o problema da ABP precisa ser fundamentado em um contexto de mundo real. Dessa maneira, propõe-se que ele seja captado com um *stakeholder* real e não criado artificialmente pelo professor da disciplina para fins didáticos. É descrito pela literatura que os agentes externos e o problema realista destes agentes possuem o poder de motivar os alunos em uma unidade de ABP, justificando a necessidade de que não seja o professor a definir esse problema [70]. Outro ponto que justifica como negativo o professor escolher o problema, remete ao fato de que o modelo proposto se preocupa com a UX como um dos atributos da autenticidade. Portanto, é importante que alguém de fato esteja enfrentando uma experiência ruim com o uso de algum software ou com a falta de um nas tarefas de seu dia-a-dia.

Devido a importância da valorização do tempo das aulas da unidade letiva de ABP, a captação do problema deve acontecer antes de ser dado início às atividades do projeto, de preferência, antes que o semestre/ano letivo tenha começado. Sem a pressão do tempo, o professor pode ter a liberdade de procurar por problemas relevantes e avaliar alternativas adequadas tanto no que se refere à autenticidade quanto ao que se refere à complexidade do problema. É importante que o problema seja ajustado à capacidade estimada dos alunos da turma que será participante da ABP, além de que é extremamente recomendável que o problema não envolva soluções para além do desenvolvimento de software. Por exemplo, escolher algum problema que envolva a segurança física de algum lugar levaria a um projeto considerado inadequado, pois remete à necessidade de inclusão de hardware, estruturas prediais e/ou mudança em políticas institucionais como parte de uma provável solução.

5.2 Convidar o Especialista do Negócio

Ainda antes das atividades da ABP, é necessário que o especialista do problema seja convidado. É também necessário que o convite seja feito com antecedência, pois se for verificada a indisponibilidade do especialista de estar presente pelo menos na aula em que se fará o entendimento do problema, na aula de validação e na apresentação pública, será necessário encontrar outro especialista com disponibilidade. Além disso, o convite deve ser feito de forma presencial, de uma maneira que o professor possa conversar sobre o problema e desenhar o mapa da jornada daquele usuário.

O especialista deverá participar do primeiro dia da DsaaF, onde o problema será contextualizado, sendo ele responsável pela primeira palestra relâmpago na fase de convergência, conforme apresentado na Figura 5.2 e na Seção 4.1. A fala do especialista contempla a estratégia de “âncora”, descrita no modelo de ABP de Bender [28]. A âncora é uma introdução com a entrega de informações básicas por meio de uma atividade que não seja uma aula expositiva. É usada para comunicar a questão motriz e, ao mesmo tempo, motivar os alunos participantes de uma unidade de ABP [28].

5.3 Desenhar a Jornada do Usuário

Durante o convite ao especialista, o professor deverá procurar entender a sequência de tarefas que ele executa que abrigam os problemas de experiência que ele está vivenciando. Essa sequência de tarefas é comumente chamada de Jornada de Usuário pela comunidade UX [88, 29]. Enquanto o especialista descreve sua jornada de usuário, o professor deve desenhá-la de forma que não fique tão detalhada a ponto de se torná-la complexa demais. Ele deverá ser curto, orientado ao objetivo final do usuário na realização de uma tarefa e considerar um fluxo com início, meio e fim de um caminho que ele percorre até ter seu objetivo concluído. Esse mapa será reproduzido em um quadro branco ou *flip-chart* na sala de aula, durante a execução da DSaaF.

Essa fase é também importante para que o professor consiga verificar se o problema a ser resolvido é complexo demais para os alunos ou extrapola o escopo de desenvolvimento de software. Caso isso seja verificado, o professor pode decidir escolher outro especialista ou escolher outro problema de UX que aquele mesmo especialista possui. É parte do processo estes ajustes, uma vez que é preciso contextualizar o problema com o cenário ao qual a unidade de ABP será aplicada.

5.4 Aplicar a *Design Sprint as a Framework* - DSaaF

Essa fase é baseada na formulação de Banfield et al. [31] para a *Design Sprint*. A principal motivação para escolher a *Design Sprint* de Banfield et al. [31] em detrimento da *Design Sprint* de Knapp et al. [31] como base teórica para esta proposta foi a liberdade dada pela *Design Sprint* de Banfield et al. [31] de incluir ou retirar exercícios de criatividade e de se distribuir ao longo do tempo, sem estar necessariamente vinculada a uma semana de cinco dias. Embora a *Design Sprint* seja flexível, os autores também valorizam sua aplicação em cinco dias de uma semana e a consideram como modelo ideal. Como forma de demonstrar que essa proposta considera mais adequada a proposição da *Design Sprint* flexível para aplicação no contexto educacional, foi escolhido a expressão *Design Sprint as a Framework* (DSaaF) para diferenciá-la da *Design Sprint* de cinco dias.

Essa fase acontece dentro de sala de aula como forma de imersão no problema, estudo de soluções, criação de protótipo e validação desse protótipo para extração dos requisitos de UX a serem endereçados na ABP. Alguns exemplos de exercícios possíveis de serem realizados na DSaaF estão descritos em Banfield et al. [31], nos estudos de caso descritos neste trabalho (Seção 4.1 e 4.2) e no livro *Gamestorming* de Gray et al. [139]. A Figura 5.2 apresenta o modelo proposta para a DSaaF.

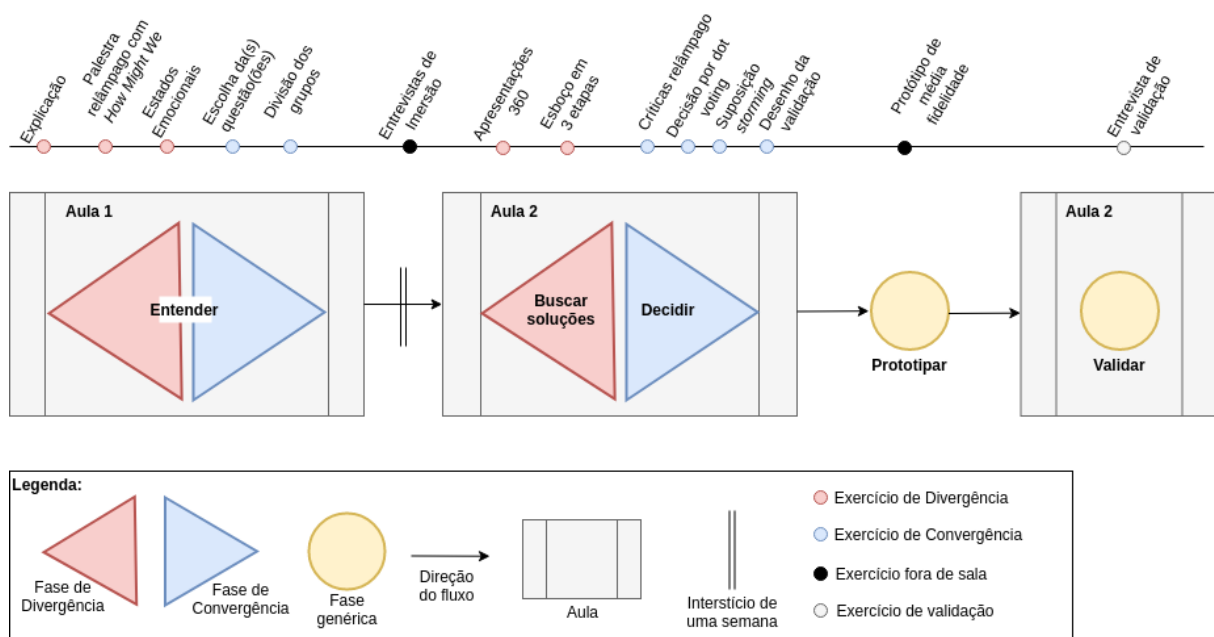


Figura 5.2: *Design Sprint as a Framework*

Antes de efetivamente começar a aula 1, o professor desenha o mapa da Jornada do Usuário coletado nas fases anteriores à DSaaF. Durante a aula, o problema é estudado de forma que haja um entendimento mais aprofundado dele. A DSaaF inicia com a palestra relâmpago do especialista, onde ele traz o máximo de informações possíveis sobre o

problema que ele e outros que compartilham do mesmo perfil passam. Tendo terminado a palestra relâmpago, o professor pergunta ao especialista se há algum ajuste a ser feito no mapa da Jornada de Usuário. Essa é a fase de divergência, dentro da aula do entendimento do problema. A fase de convergência é feita com a escolha de uma ou mais questões importantes a serem abordadas durante a DSaaF e também toda a ABP. Essa fase representa o princípio da questão motriz descrito na seção 2.1.5. O escopo do *stakeholder* a ser beneficiado pela solução deverá ser fechado no perfil de usuário representado pelo especialista, de forma a facilitar a execução das atividades da fase de validação.

Nesse dia é importante que, como relatado no estudo de caso 2 (Seção 4.2), os alunos sejam divididos nos grupos **Voz do Cliente** e **Como as coisas funcionam** para uma melhor investigação do problema a ser resolvido e mitigar o problema de viés trazido pelo relato de apenas um especialista de negócio. Além disso, essa investigação contempla o princípio da investigação na ABP descrito na seção 2.1.5. Outro importante grupo é aquele responsável por procurar por soluções concorrentes e/ou similares, de forma que essas soluções possam inspirar as novas propostas e que os alunos não "reinventem a roda" em sua proposta.

Antes da próxima aula, os alunos executarão as atividades dos grupos **Voz do Cliente** e **Como as coisas funcionam**, além de procurar por soluções concorrentes que possam inspirar a geração de soluções. Isso só é possível com o interstício de uma semana entre a aula 1 e a aula 2. Nesse dia, os alunos e o professor devem também concordar sobre um número importante a ser diminuído ou aumentado, caso a solução derivada da DSaaF seja implementada. Esse número é uma métrica que representa numericamente o estado futuro do usuário onde será desencadeada uma situação de satisfação. Ele pode ser, por exemplo, a quantidade de experiência profissional colocada no currículo de um aluno enquanto ele está na universidade (no caso de um número que deve ser aumentado) ou a quantidade de dias que um aluno espera para ser contratado depois de ter enviado um currículo (no caso de um número que deve ser diminuído). Ter esse número em mente durante a DSaaF evitará que a solução proposta tente resolver muitos problemas e seja difícil de mensurar sua adequabilidade à solicitação do especialista.

Na aula 2, os exercícios devem ser feitos na intenção de gerar o maior número de soluções possíveis (Fase Divergente) e, em seguida, convergi-la para uma única solução por meio de uma decisão feita com votação silenciosa como descrita no estudo de caso 2 4.2 (Fase Convergente). O protótipo deve ser gerado pelos próprios alunos entre a aula 2 e a aula 3, em casa ou em uma aula reservada para isso. Esse protótipo deve ser apenas um *wireframe* fácil de ser feito, devendo levar em consideração principalmente o fluxo das telas e a interação do usuário com o sistema. Recomenda-se usar *templates* de design já prontos, de forma a evitar que os alunos percam tempo discutindo aspectos de estética.

Um roteiro mais específico do que acontece na primeira e segunda aula está incluído na seção de anexos dessa dissertação.

A aula 3 é dedicada à validação, que será feita com o especialista. Embora Banfield et al. [31] sugiram que essa validação seja feita com pelo menos quatro usuários, isso tornaria a fase de validação impraticável devido a restrição de tempo que o ambiente educacional proporciona. Esse problema é mitigado pela reiteração do exercício de validação no final da segunda *Sprint* do *eduScrum*, antes do artefato ser apresentado publicamente. A validação é feita com base em uma ou mais suposições críticas que foram levantadas durante a DSaaF. Ela poderá ser realizada na própria sala de aula, com a tela do computador ou dispositivo móvel onde está sendo feito o teste compartilhada em um projetor (no caso de dispositivos móveis, é possível compartilhar a tela por meio de aplicativos como o *Screencast* [140]). O teste de validação consiste em que o especialista do negócio tente alcançar o objetivo de sua tarefa especificado no mapa da jornada do usuário sem que seja lhe dado nenhuma instrução específica do que se deve fazer. Enquanto isso, os alunos observam e anotam comportamentos e falas do especialista ligadas problemas de uso do protótipo da solução.

Tendo feito a validação, os alunos devem se reunir para fazer uma retrospectiva da DSaaF e deles extrair um mapa de empatia do perfil de usuário para quem eles querem destinar a solução e as *job stories* que endereçam as dores no mapa de empatia e compõem o *backlog* do artefato. Todos os artefatos gerados pela DSaaF serão importantes nessa atividade.

O mapa de empatia é criado com base nas atividades realizadas em sala de aula, sobretudo pelas entrevistas que acontecem entre o dia 1 e o dia 2. Ele é uma ferramenta útil para a definição de personas do público alvo, voltada para desenvolver um entendimento sobre o comportamento, as aspirações, as preocupações e o ambiente de um determinado público alvo [141]. A imagem 5.3 mostra um diagrama de mapa de empatia.

Já as *job stories* são similares às *user stories*, mas se diferem delas no sentido de que elas são vinculadas ao contexto do usuário e começam com a declaração da situação presente (Quando eu...) e não com a declaração do persona da *story* (Eu, como...) [142]. É importante que a declaração da situação presente esteja ligada aos estados emocionais negativo do especialista causado pela insatisfação com uma tarefa que ele já realiza ou a possíveis obstáculos para o alcance do objetivo por meio da execução dessa tarefa. É também importante que o resultado esperado esteja ligado a um estado emocional positivo que representa como o usuário ficará quando a experiência ruim com a tarefa atual for melhorada. Elas deverão focalizar os desejos do usuário que, se não cumpridos, transformarão a jornada do usuário com o software desenvolvido uma experiência negativa. Após a escrita das *job stories*, elas deverão ser validadas com o especialista. A Figura 5.4

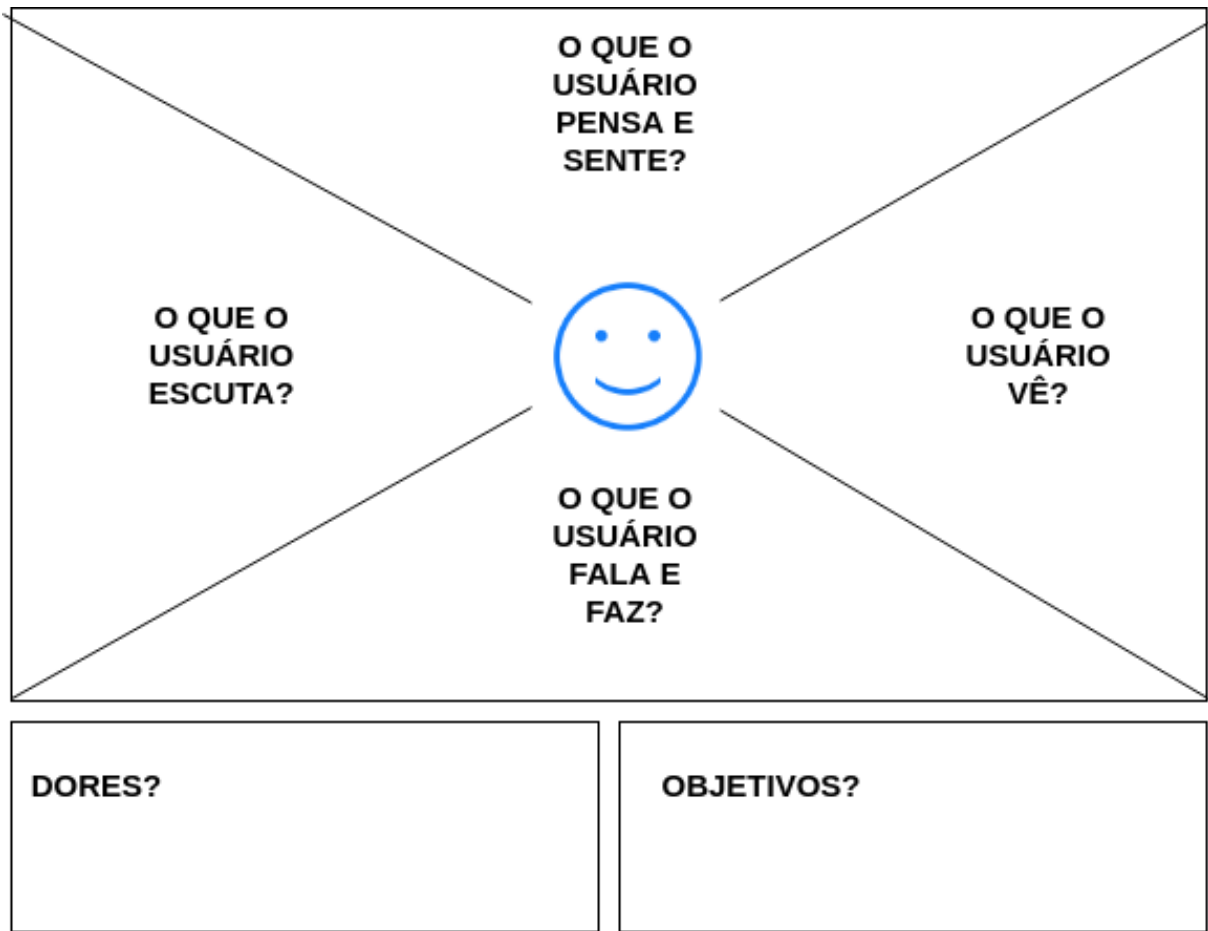


Figura 5.3: Mapa de Empatia

apresenta um exemplo de *job story* a ser usada nesse modelo de ABP.

The Job Story template is a yellow rectangular box with a gold border. It contains three lines of text, each followed by a horizontal line for input and an emoji:

- Quando _____ ☹️
- Eu quero _____
- Então eu posso _____ 😊

Figura 5.4: *Job Story* de Experiência do Usuário

5.5 Aplicar o *eduScrum*

O *eduScrum* é um método de educação baseado no *Scrum*. É uma estrutura para treinar estudantes onde a responsabilidade do processo de aprendizagem é delegada dos professores aos estudantes [138]. Sua escolha para contemplar a fase de desenvolvimento de software desse modelo se deu em virtude dos resultados obtidos na RSL (Seção 3). Além disso, o *eduScrum* é adequado à visão construcionista, pois preza pela autonomia do aluno e pela colaboração entre os pares, o que preserva os atributos da autenticidade presente nos modelos atuais de ABP [138].

Os eventos do *eduScrum* são [138]:

- ***Sprints***: unidade básica do *eduScrum*, é um conjunto coerente de material de aprendizagem que leva a equipe a atingir os objetivos previamente estabelecidos. Dentre delas estão incluídas a reunião de planejamento das *sprints*, os *stand-ups*, a revisão das *sprints* e a retrospectiva das *sprints*;
- **Reunião do planejamento das *sprints***: evento onde se forma a equipe, se definem as metas de aprendizado e se planeja o trabalho da *Sprint*. As metas de aprendizado são configurações fixas definidas pelo professor do que ele espera ser entregue ao final da *Sprint*. A definição do como será entregue e do quê será entregue fica a cargo dos estudantes;
- ***Stand-Up***: evento com o tempo fixo de 5 minutos que acontece no início de cada aula dentro de uma *Sprint*. Nele os alunos dizem o que já fizeram e o que farão para ajudar o time, além de elencar os obstáculos que os impedem de atingir as metas de aprendizado;
- **Revisão das *sprints***: evento que acontece ao final de cada *Sprint*, onde as equipes de alunos demonstram o que aprenderam durante a *Sprint*;
- **Retrospectiva das *sprints***: a exemplo da revisão das *sprints*, também é um evento que acontece no final de cada *Sprint*, mas que tem o foco na inspeção dos alunos sobre si mesmos e sobre a equipe. O resultado da retrospectiva é um plano de melhoria individual feito por cada aluno.

O *eduScrum* usado nesse modelo é dividido em duas *sprints*, sendo a primeira delas para configuração da infraestrutura necessária ao desenvolvimento do artefato e criação de modelos de arquitetura de software e a segunda para o desenvolvimento em si. Na primeira *Sprint*, os objetivos de aprendizagem são ligados ao projeto de software, ao planejamento das atividades do projeto e à tomada de decisões arquiteturais. Seu tempo deve ocupar apenas um quarto do tempo disponível para o *eduScrum* da ABP. Na segunda *Sprint*, os

objetivos de aprendizagem orbitam ao redor das técnicas de desenvolvimento e teste de software, da gestão da configuração (versionamento), da garantia de qualidade, do uso das ferramentas e do trabalho em equipe. Ela terminará com uma segunda validação feita com o especialista, dessa vez no software produzido. Seu tempo ocupará os outros três quartos do tempo disponível para o *eduScrum*.

No *eduScrum* desse modelo, os alunos são divididos em grupos de 4 ou 5 pessoas. Cada grupo tem seu próprio *eduScrum Master*, que possui a responsabilidade de escolher os membros do seu time *eduScrum*, cuidar do *Flip* (uma espécie de quadro Kanban da equipe por onde ela mede o próprio progresso) e garantir que as regras do *eduScrum* sejam cumpridas pela equipe que lidera. O professor age como o *Product Owner* da *eduScrum*, sendo responsável pelo *backlog* com as *job stories* e por definir os critérios de aceitação de cada *Sprint*. O critério de aceitação da segunda *Sprint* deverá ser vinculado à validação feita com o especialista no final da segunda *Sprint*. Além disso, o professor é responsável por determinar as necessidades de treinamento da equipe, provendo suporte educacional sempre que necessário por meio de qualquer das estratégias de ensino descritas no capítulo da RSL (Seção 3.3.1). Por sua vez, a equipe *eduScrum* é responsável por alcançar os objetivos de aprendizagem, desenvolver os artefatos da ABP, validar os artefatos com o professor e o especialista e entrarem em consenso sobre quando o trabalho terá terminado (definição de pronto) e quais as condições necessárias para a garantia de um trabalho agradável (definição de diversão) [138].

Para o modelo proposto neste trabalho é desejável ter um monitor (docente ou discente), dadas as considerações feitas no Capítulo 3 (Seção 3.3.1). Esse monitor deve ter habilidades em construção de software, de forma que auxilie o professor na segunda *Sprint*, que é a mais complexa e importante da ABP. Além disso, é importante que a maioria das atividades aconteçam dentro de sala de aula, a fim de que as estratégias de ensino como *coaching* e ensino sob demanda sejam possíveis.

5.6 Realizar a Apresentação Pública

Outro importante requisito para a autenticidade da ABP é fazer com que o artefato produzido pelos alunos em sala de aula seja apresentado publicamente. A apresentação pública é um dos pilares definidos pelo construcionismo, que afirma que os objetos de conhecimento construídos pelos alunos devem ser percebidos por outros (Seção 2.1.4). É também uma fase existente na proposta de ABP de Bender [28]. É dito na literatura que deixar claro que os alunos irão apresentar o resultado de seu trabalho para uma plateia externa representa um ganho na motivação e no senso de responsabilidade, afetando positivamente a qualidade do artefato que será construído (Seção 2.1.6).

Nessa fase os alunos apresentam publicamente seu artefato. O formato da apresentação não deve ser determinado pelo professor, mas escolhido pelos próprios estudantes. A plateia dessa apresentação pode ser variada em termos de tamanho e *background* técnico, mas é importante que o especialista faça parte dela. Assim, é recomendado que haja agentes educacionais que possam perceber e valorizar as contribuições pedagógicas da ABP presentes na plateia. Dessa forma, a cultura do aprendizado por meios construcionistas pode ser reforçada, fazendo com que outras iterações da ABP sejam mais bem aceitas na comunidade acadêmica e mais problemas sejam elencados para posterior tratamento dentro de futuras unidades de ABP.

Capítulo 6

Conclusão

Este trabalho apresentou uma Revisão Sistemática de Literatura (RSL) e dois estudos de caso exploratórios. A RSL permitiu fazer uma ampla busca na literatura a fim de encontrar um volume relevante de trabalhos sobre o assunto proposto neste trabalho. Os artigos identificados permitiram responder às questões de pesquisa secundárias e a questão principal envolvendo a autenticidade de unidades de Aprendizagem Baseada em Projetos que propõem o desenvolvimento de software como atividade principal.

As respostas às Questões Secundárias da RSL permitiram descobrir como o conteúdo teórico e técnico é ensinado nas unidades de ABP, quanto tempo duram essas unidades, quais os modelos e/ou técnicas de engenharia de software são usados em conjunto com elas, como são formados os grupos dos alunos e qual o impacto delas no rendimento acadêmico dos alunos. A partir dessas respostas, pôde-se inferir como as aplicações de ABP em projetos de desenvolvimento de software nos cursos de graduação tornam software autêntico. Notou-se que o *Scrum* é o modelo frequentemente usado para conduzir as unidades de ABP, que, em sua maioria, os alunos são divididos em pequenos grupos e que a presença de tutores durante as aulas pode impactar positivamente na autenticidade do software desenvolvido. Estas informações foram úteis na construção do modelo final proposto neste trabalho, pois ajudaram a balizar o modelo com base em aplicações que chegaram ao objetivo de ter o software funcionando e sendo usado por usuários reais, mitigando, assim, os vieses da aplicação de um baixo número de estudos de caso.

Além disso, este trabalho descreveu e analisou dois estudos empíricos, cuja finalidade foi verificar como a *Design Sprint* se comporta dentro de um contexto educacional. Por ser uma técnica extremamente dinâmica e multidisciplinar, a possibilidade de sucesso em aplicações no contexto das aulas de uma disciplina de engenharia de software é importante.

O estouro de tempo em um dos dois estudos (Seção 4.1) foi crucial para que verificássemos que não é viável fazer uma *Design Sprint* em apenas uma aula. Com base nisso, concluímos que o principal cuidado a ser tomado é em relação ao tempo. Mas

também há considerações a se fazer quanto à carga de trabalho do professor no papel de facilitador, principalmente nos dias anteriores à execução da *Design Sprint*, à necessidade de habilidades relacionadas ao gerenciamento do tempo, ao gerenciamento de grupos e à motivação de alunos, principalmente porque nem todos os alunos possuirão o perfil extrovertido e tentarão declinar da participação das atividades ou faltar em um ou mais dias da *Design Sprint*. Além disso, o professor facilitador precisa de algum conhecimento em como uma *Design Sprint* funciona e o que é importante na avaliação de requisitos de *User Experience*.

É importante ter também um especialista de negócio disponível, pelo menos para os momentos iniciais de coleta e abstração do Fluxo da Jornada do Usuário e palestra relâmpago e para o momento de validação. Encontrar esse especialista de negócio e o problema adequado ao nível dos alunos pode não ser rápido, por isso é importante que o planejamento da *Design Sprint* aconteça antes que o semestre letivo aconteça.

No entanto, foi possível notar que a *Design Sprint* possui um grande potencial de propor inovações na substituição de processos existentes que possuem algum problema de experiência de usuário (UX).

Os resultados dos estudos de caso permitiram identificar:

- O uso da *Design Sprint* é adequado para fornecer um direcionamento para os *insights* no desenvolvimento de produtos de software utilizando a abordagem de aprendizagem baseada em projetos.
- Os estudantes consideraram a experiência satisfatória na participação e aprendizado em sala de aula no formato de *Design Sprint*.
- A *Design Sprint* aplicada em um cenário educacional necessita de um período maior para o seu aprendizado e fixação.
- Houve a aceitação do uso da metodologia pelos estudantes, o que contribui para o aprendizado construcionista e a autenticidade de uma intervenção pedagógica.
- Pode haver problemas com absenteísmo dos alunos na *Design Sprint* espalhada ao longo de mais de uma semana, o que leva à reflexão da necessidade de incentivos.

Tais informações, aliadas a teorias e métodos já consolidados na literatura, permitiram construir um modelo de ABP que leve em conta requisitos de UX como atributos de autenticidade, principalmente no que tange a como os requisitos dos softwares a serem produzidos podem ser definidos.

Outros estudos empíricos serão realizados para validar o modelo proposto em duas etapas, verificando a viabilidade de sua execução. Como trabalhos futuros pretende-se avaliar se o conjunto de requisitos extraídos da DSaaS contém propriedades desejadas de

UX, de forma que guie a construção de software no modelo de ABP proposto. Outra validação necessária é a adequabilidade do *eduScrum* de suportar o desenvolvimento de um software que contemple os requisitos extraídos da DSaaF. Também espera-se nos trabalhos futuros ver que efeitos a *Design Sprint* traz para a motivação, já que este é um dos principais benefícios da educação autêntica reportados na literatura [70, 27], além de experimentar a *Design Sprint* em outros contextos específicos como no desenvolvimento de Trabalho de Conclusão de Curso (TCC) ou de Projetos de Iniciação Tecnológica (PIBIT) e observar se a necessidade de entrega de um projeto como requisito de avaliação nesses casos se configura como um incentivo capaz de mitigar o absenteísmo dos alunos.

Adequações podem ser feitas caso seja verificado alguma questão não percebida na construção desse modelo.

Referências

- [1] Council, British Design: *The design process: What is the double diamond?*, 2001. <https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>. xii, 29, 30, 31, 69
- [2] Knapp, Jake, John Zeratsky e Braden Kowitz: *Sprint: How to solve big problems and test new ideas in just five days*. Simon and Schuster, New York, NY, USA, 2016. xii, 2, 3, 4, 5, 31, 32, 33, 59, 60, 62, 63, 64, 68, 69, 74, 77
- [3] Brasil. Conselho Nacional de Educação - CNE/CES: *Parecer 436/2001 do Ministério da Educação: Trata de Cursos Superiores de Tecnologia – Formação de Tecnólogos*, 2001. <http://portal.mec.gov.br/setec/arquivos/pdf/legislacao/superior/legisla{ }superior{ }parecer4362001.pdf>. 1
- [4] Brasil. Ministério da Educação: *Catálogo Nacional de Cursos Superiores de Tecnologia*, 2016. <http://portal.mec.gov.br/index.php?option=com{ }docman{ }view=download{ }alias=98211-cncst-2016-a{ }category{ }slug=outubro-2018-pdf-1{ }Itemid=30192>. 1
- [5] Schön, Donald A: *The reflective practitioner: How professionals think in action*. Routledge, 2017. 1
- [6] Pohl, Klaus: *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010. 1
- [7] Bourque, Pierre e Richard E. Fairley (editores): *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, CA, version 3.0 edição, 2014, ISBN 978-0-7695-5166-1. <http://www.swebok.org/>. 1
- [8] Hassenzahl, Marc: *Experience design: Technology for all the right reasons*. Synthesis Lectures on Human-Centered Informatics, 3(1):1–95, janeiro 2010. <https://doi.org/10.2200/s00261ed1v01y201003hci008>. 2, 22, 23, 25
- [9] Brown, John Seely, Allan Collins e Paul Duguid: *Situated Cognition and the Culture of Learning*. Relatório Técnico, Institute for Research on Learning, 1968. 2, 16
- [10] Dewey, John: *Experience and education*. Kappa Delta Pi, 1998. 2, 18
- [11] Kilpatrick, William Heard: *The Project method: The use of the purposeful act in the educative process*. Teachers college, Columbia university, 1922. 2, 15

- [12] Yuen, Timothy T.: *Scrumming with educators: Cross-departmental collaboration for a summer software engineering capstone*. Proceedings - 2015 International Conference on Learning and Teaching in Computing and Engineering, LaTiCE 2015, páginas 124–127, 2015. 2, 48, 51, 53, 54, 55, 56, 58
- [13] Francese, Rita, Carmine Gravino, Michele Risi, Giuseppe Scanniello e Genoveffa Tortora: *Using Project-Based-Learning in a mobile application development course-An experience report*. Journal of Visual Languages and Computing, 31:196–205, 2015, ISSN 1045926X. <http://dx.doi.org/10.1016/j.jvlc.2015.10.019>. 2
- [14] Ferreira, Avelino e Gomes Filho: *Modelo de Ensino baseado nos Métodos Ágeis de Desenvolvimento de Software*. Tese de Doutorado, Universidade Federal do Rio de Janeiro, 2016. 2
- [15] Dubinsky, Yael e Orit Hazzan: *A framework for teaching software development methods*. Computer Science Education, 15(4):275–296, 2005, ISSN 0899-3408. <http://www.tandfonline.com/doi/abs/10.1080/08993400500298538>. 2
- [16] Maícias, José A.: *Enhancing project-based learning in software engineering lab teaching through an e-portfolio approach*. IEEE Transactions on Education, 55(4):502–507, 2012, ISSN 00189359. 2
- [17] Jaime, Arturo, José Miguel Blanco, César Domínguez, Ana Sánchez, Jónathan Heras e Imanol Usandizaga: *Spiral and Project-Based Learning with Peer Assessment in a Computer Science Project Management Course*. Journal of Science Education and Technology, 25(3):439–449, 2016, ISSN 15731839. 2
- [18] Jazayeri, Mehdi: *Combining Mastery Learning with Project-Based Learning in a First Programming Course: An Experience Report*. Proceedings - International Conference on Software Engineering, 2:315–318, 2015, ISSN 02705257. 2, 8
- [19] Noss, Richard: *Reconstructing Constructionism*. Constructionism, páginas 1–4, 2010. 2, 16
- [20] Papert, Seymour e Idit Harel: *Situating constructionism*. Constructionism, páginas 1–14, 1991. 2, 16, 61
- [21] Ackermann, Edith: *Piaget's Constructivism, Papert's Constructionism: What's the difference?* Future of learning group publication, 5(3), 2011. 2, 16, 22
- [22] Helle, Laura, Päivi Tynjälä e Erkki Olkinuora: *Project-Based Learning in Post-Secondary Education – Theory, Practice and Rubber Sling Shots*. Higher Education, 51(2):287–314, 2006, ISSN 0018-1560. <http://link.springer.com/10.1007/s10734-004-6386-5>. 2, 17, 18, 21, 50, 56
- [23] Splitter, Laurance J.: *Authenticity and constructivism in education*. Studies in Philosophy and Education, 28(2):135–151, 2009, ISSN 00393746. 2, 16, 19, 20, 21, 22

- [24] McDermott, Roger, Mark Zarb, Mats Daniels, Aletta Nylén, Arnold Pears, Ville Isomöttönen e Michael Caspersen: *The authenticity of 'Authentic' assessment some faculty perceptions*. Proceedings - Frontiers in Education Conference, FIE, 2017-Octob:1–9, 2017, ISSN 15394565. 2, 20, 21, 22
- [25] Liegel, Kimberly May: *Project-based learning and the future of project management*. Em *PMI global conference*, North America, Anaheim, CA, 2004. Project Management Institute. <https://www.pmi.org/learning/library/project-based-learning-future-project-management-8273>. 2
- [26] Thomas, John W: *A Review of Research on Project-Based Learning*. <http://www.bie.org/index.php/site/RE/pbl{ }research/29>, Working paper, 2000. 2, 16, 17, 18, 21, 22, 41, 54, 56
- [27] Condliffe, Barbara, Janet Quint, Mary G Visser, Michael R Bangser, Sonia Drohojowska, Larissa Saco e Elizabeth Nelson: *Project-Based Learning A Literature Review Working Paper Prepublication copy: Release*. Working paper, 2017. 2, 17, 18, 19, 54, 91
- [28] Bender, William N: *Project-based learning: Differentiating instruction for the 21st century*. Corwin Press, Thousand Oaks, California, USA, 2012. 2, 17, 18, 19, 21, 54, 56, 60, 72, 79, 80, 81, 87
- [29] Stull, Edward: *UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, Writers, Designers, and Developers*. Apress, Berkely, CA, USA, 1st edição, 2018, ISBN 1484238109, 9781484238103. 2, 5, 23, 24, 81
- [30] Gobble, MaryAnne M.: *The Persistence of Brainstorming*. Research-Technology Management, 57(1):64–67, 2014, ISSN 0895-6308. 3, 4
- [31] Banfield, Richard, C Todd Lombardo e Trace Wax: *Design Sprint: A Practical Guidebook for Building Great Digital Products*. " O'Reilly Media, Inc. ", "Sebastopol, CA", 2015. 3, 31, 32, 33, 60, 68, 77, 80, 82, 84
- [32] Kitchenham, Barbara: *Procedures for performing systematic reviews*. Keele, UK, Keele University, 33(TR/SE-0401):28, 2004, ISSN 13537776. [http://csnotes.upm.edu.my/kelasmaya/pgkm20910.nsf/0/715071a8011d4c2f482577a700386d3a/{\protect\T1\textdollar}FILE/10.1.1.122.3308\[1\].pdf{ }5Cnhttp://tests-zingarelli.googlecode.com/svn-history/r336/trunk/2-Disciplinas/MetodPesquisa/kitchenham{ }2004.pdf](http://csnotes.upm.edu.my/kelasmaya/pgkm20910.nsf/0/715071a8011d4c2f482577a700386d3a/{\protect\T1\textdollar}FILE/10.1.1.122.3308[1].pdf{ }5Cnhttp://tests-zingarelli.googlecode.com/svn-history/r336/trunk/2-Disciplinas/MetodPesquisa/kitchenham{ }2004.pdf). 6, 37, 38, 39, 41, 43
- [33] Gil, Antonio Carlos: *Métodos e técnicas de pesquisa social*. 6. ed. Editora Atlas SA, Sao Paulo, SP, BR, 2008. 6
- [34] Salvador, César Coll, Mariana Miras Mestres, Javier Onruvia Goñi e Isabel Solé Gallart: *Psicologia da educação*. Penso Editora, 2016. 8, 9
- [35] Moursund, David G: *Project-based learning using information technology*. International society for technology in education Eugene, OR, 1999. 8, 18

- [36] Diehl, W, T Grobe, H Lopez e C Cabral: *Project-based learning: A strategy for teaching and learning*. Boston: Center for youth development and education, 1999. 8, 18, 21
- [37] Grant, Michael M: *Getting a grip on project-based learning: Theory, cases and recommendations*. Meridian: A middle school computer technologies journal, 5(1):83, 2002. 8, 11
- [38] Morgan, Alistair: *Theoretical Aspects of Project- Based Learning in Higher Education*. British Journal of Educational Technology, 14(1):66–78, 1975. 8, 14, 17, 18
- [39] Thorndike, Edward L: *The contribution of psychology to education*. Journal of Educational Psychology, 1(1):5, 1910. 9
- [40] Thorndike, Edward L.: *Educational psychology, Vol 1: The original nature of man*. Teachers College, 1919. <https://doi.org/10.1037/13763-000>. 9
- [41] Thorndike, Edward L.: *Educational psychology, Vol 3: Mental work and fatigue and individual differences and their causes*. Teachers College, 1914. <https://doi.org/10.1037/13796-000>. 9
- [42] Thorndike, Edward L: *The principles of teaching: Based on psychology*. Routledge, 2013. 9
- [43] Judd, Charles Hubbard: *Genetic psychology for teachers*, volume 55. D. Appleton, 1903. 9
- [44] Judd, Charles Hubbard: *Psychology of secondary education*. Ginn & Company, 1927. <https://doi.org/10.1037/14634-000>. 9
- [45] Moreira, Marco Antonio: *Teorias de aprendizagem*, volume 2. Editora pedagógica e universitária São Paulo, 1999. 9, 10, 11, 12
- [46] outros, Kester Carrara e: *Introdução à psicologia da educação: seis abordagens*. Editora Avercamp, first edição, nov 2003. 9, 13
- [47] Bidarra, Maria da Graça e Maria Isabel Festas: *Construtivismo(s): Implicações e Interpretações Educativas*. Revista Portuguesa de Pedagogia, 39(2):177–195, 2005. <http://gaius.fpce.uc.pt/niips/gbidarra/Bidarra2005{ }Construtivismo.pdf>. 9, 11, 13
- [48] Pritchard, Duncan: *What is this thing called Knowledge?* Routledge, 3ª edição, 2013, ISBN 0415657210,9780415657211,9780415657068. <http://gen.lib.rus.ec/book/index.php?md5=5738974328335fc0c3711103bf3a9ec4>. 9
- [49] Nelson Piletti, Solange Marques Rossato, Geovanio Rossato: *Psicologia do Desenvolvimento*. Editora Contexto, São Paulo, SP, first edição, jun 2014. 9, 10, 11, 12, 13

- [50] Ostermann, Fernanda e CJ de H CAVALCANTI: *Teorias de aprendizagem*. Universidade Federal do Rio Grande do Sul–Instituto de Física, 2010. 9, 10, 11
- [51] Skinner, Burrhus Frederic: *Ciência e comportamento humano*, volume 10. Martins Fontes São Paulo, 2003. 10
- [52] Groome, David: *An Introduction to Cognitive Psychology*. Taylor & Francis, London and New York, 2014, ISBN 020397798X. 10
- [53] Tolman, E. C.: *A behavioristic theory of ideas*. *Psychological Review*, 33(5):352–369, 1926. 11
- [54] Raaijmakers, Jeroen G. e Richard M. Shiffrin: *Search of associative memory*. *Psychological Review*, 88(2):93–134, 1981. <https://doi.org/10.1037/0033-295x.88.2.93>. 11
- [55] Huitt, William e John Hummel: *Piaget’s theory of cognitive development*. *Educational psychology interactive*, 3(2):1–5, 2003. 11, 12
- [56] Vygotsky, Lev: *Interaction between learning and development*. *Readings on the development of children*, 23(3):34–41, 1978. 11, 18
- [57] *Vygotsky (1) - construtivismo x socioconstrutivismo*. <http://didatics.com.br/index.php/psicologia/socioconstrutivismo/vygotsky/parte-1-construtivismo-e-socioconstrutivismo>. 13
- [58] Vygotsky, Lev Semenovich: *Mind in society: The development of higher psychological processes*. Harvard university press, 1980. 13
- [59] Langford, Peter E: *Vygotsky’s developmental and educational psychology*. Psychology Press, 2004. 13
- [60] Souza, Gilcênio Vieira: *Estudando vygotky*, Feb 2011. <https://www.ufrgs.br/psicoeduc/gilvieira/2011/02/02/teoria-historico-cultural-e-aprendizagem-contextualizada/>. 13
- [61] Pereira, Eliana, Jackeline Ribeiro Martins, Vilmar dos Santos Alves e Evaldo Inácio Delgado: *A contribuição de John Dewey para a educação*. *Revista Eletrônica da Educação*, 3(1):154–161, 2009. <http://www.reveduc.ufscar.br/index.php/reveduc/article/view/38/37>. 14, 15
- [62] Bell, Stephanie: *Project-Based Learning for the 21st Century: Skills for the Future*. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 83(2):39–43, 2010, ISSN 0009-8655. <http://www.tandfonline.com/doi/full/10.1080/00098650903505415>. 14, 22
- [63] Kolmos, Anette e Erik de Graaff: *Problem-based and project-based learning in engineering education*. Em Johri, Aditya e Barbara M. Olds (editores): *Cambridge Handbook of Engineering Education Research*, páginas 141–160. Cambridge University Press, 2017. <https://doi.org/10.1017/cbo9781139013451.012>. 14, 21

- [64] Barron, Brigid J.S., Daniel L. Schwartz, Nancy J. Vye, Allison Moore, Anthony Petrosino, Linda Zech e John D. Bransford: *Doing With Understanding: Lessons From Research on Problem- and Project-Based Learning*. Journal of the Learning Sciences, 7(3-4):271–311, 1998, ISSN 1050-8406. <http://www.tandfonline.com/doi/abs/10.1080/10508406.1998.9672056>. 14, 19, 20, 22
- [65] Teitelbaum, Teitelbaum e Michael Apple: *John Dewey*. Curriculo sem Fronteiras, 1(2):194–201, 2001. 14
- [66] Cunha, Marcus Vinicius da: *John dewey: filosofia, política e educação*. Perspectiva, 19(2):371–388, 2001. 14, 15
- [67] Backe, Andrew: *John Dewey and Early Chicago Functionalism*. History of Psychology, 4(4):323–340, 2001, ISSN 10934510. 14
- [68] Knoll, Michael: *The Project Method: Its Origin and International Development*. Progressive Education Across the Continents., 27(September 1996):40–44, 1997. 15, 18
- [69] Marques, Liliana: *William Kilpatrick e o Método de Projeto*. Cadernos de Educação de Infância, 1(107):4–5, 2016. 15
- [70] Blumenfeld, Phyllis C, Elliot Soloway, Ronald W Marx, Joseph S Krajcik, Mark Guzdial e Annemarie Palincsar: *Motivating project-based learning: Sustaining the doing, supporting the learning*. Educational psychologist, 26(3-4):369–398, 1991. 17, 18, 19, 20, 21, 22, 50, 54, 56, 80, 91
- [71] Morgan, Alistair: *Theoretical aspects of project-based learning in higher education*. British Journal of Educational Technology, 14(1):66–78, 1983. 18
- [72] Jones, Beau Fly, Claudette M Rasmussen e Mary C Moffitt: *Real-life problem solving: A collaborative approach to interdisciplinary learning*. American Psychological Association, 1997. 18
- [73] Thomas, J, A Michaelson e JR Mergendoller: *Introduction to project based learning*. Buck Institute for Education Project Based Learning Handbook, 1999. 18
- [74] Adderley, Kenneth *et al.*: *Project methods in higher education*, volume 24. Society for research into higher education, 1975. 18
- [75] Krajcik, Joseph S, Phyllis C Blumenfeld, Ronald W Marx e Elliot Soloway: *A collaborative model for helping middle grade science teachers learn project-based instruction*. The elementary school journal, 94(5):483–497, 1994. 18
- [76] Aebli, Hans: *Psychologische Didaktik: Didaktische Auswertung der Psychologie von Jean Piaget*. Klett, 1963. 18
- [77] Marx, Ronald W., Phyllis C. Blumenfeld, Joseph S. Krajcik e Elliot Soloway: *Enacting Project-Based Science*. The Elementary School Journal, 97(4):341–358, 1997, ISSN 0013-5984. <https://www.journals.uchicago.edu/doi/10.1086/461870>. 19, 20

- [78] Archibald, Doug e Fred M Newmann: *Beyond Standardized Testing: Assessing Authentic Academic Achievement in the Secondary School*. Relatório Técnico, National Association of Secondary:School Principals, Madison, WI, 1988. <http://files.eric.ed.gov/fulltext/ED301587.pdf>. 21
- [79] Newmann, Fred M e Doug A Archibald: *The nature of authentic academic achievement*. Toward a new science of educational testing and assessment, páginas 71–83, 1992. 21
- [80] Gordon, Rick: *Balancing real-world problems with real-world results*. Phi Delta Kappan, 79(5):390, 1998. 22
- [81] Lepper, Mark R.: *Motivational Considerations in the Study of Instruction*. Cognition and Instruction, 5(4):289–309, 1988. 22
- [82] Allam, A H, Ab Razak e Che Hussin: *User Experience : Challenges and Opportunities*. Journal of Research and Innovation in Information Systems, páginas 28–36, 2009. 23, 26, 27, 28
- [83] Nielsen, Jakob e Thomas K. Landauer: *A Mathematical Model of the Finding of Usability Problems*. Em *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Amsterdam, NE, 1993. ACM Press, ISBN 0897915755. http://delivery.acm.org/10.1145/170000/169166/p206-nielsen.pdf?ip=152.11.88.1&id=169166&acc=ACTIVESERVICE&key=A79D83B43E50B5B8.2003C2B88ABD935A.4D4702B0C3E38B35.4D4702B0C3E38B35&{}_{}_acm{}_{}_=1540407948{}_bdce6a77cc527ebad722f2dd84e367da. 23
- [84] Vermeeren, Arnold P O S, Effie Lai chong Law e Virpi Roto: *User Experience Evaluation Methods: Current State and Development Needs Arnold*. Em *User experience evaluation methods. Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries*, páginas 521–530, Reykjavik, Iceland, 2010. ACM Press, ISBN 978-1-60558-934-3. 23, 27, 28
- [85] *Ergonomics of human system interaction - part 210: Human-centred design for interactive systems (formerly known as 13407)*, 2008. 23
- [86] Bargas-avila, Javier A e Kasper Hornbæk: *Old Wine in New Bottles or Novel Challenges ? A Critical Analysis of Empirical Studies of User Experience*. Em *Proceedings of Conference on Human Factors in Computing Systems*, páginas 2689–2698, Vancouver, BC, Canada, 2012. ACM, ISBN 9781450302678. 23, 26
- [87] Law, Effie Lai Chong, Virpi Roto, Marc Hassenzahl, Arnold P.O.S. Vermeeren e Joke Kort: *Understanding, scoping and defining user experience*. Em *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, páginas 719–728, 2009, ISBN 9781605582467. <http://dl.acm.org/citation.cfm?doid=1518701.1518813>. 23

- [88] Hassenzahl, Marc e Noam Tractinsky: *User experience - A research agenda*. Behaviour and Information Technology, 25(2):91–97, 2006, ISSN 0144929X. 23, 26, 66, 81
- [89] Forlizzi, Jodi e Katja Battarbee: *Understanding experience in interactive systems*. Proceedings of the 2004 conference on Designing interactive systems processes, practices, methods, and techniques - DIS '04, página 261, 2004. <http://portal.acm.org/citation.cfm?doid=1013115.1013152>. 23, 25, 26
- [90] Travis, David e Philip Hodgson: *Think Like a UX Researcher*. CRC Press, Boca Raton, FL, 1ª edição, 2019, ISBN 9781138365292. 23, 24
- [91] McCarthy, John e Peter Wright: *Technology as Experience*. The MIT Press, Cambridge, Massachusetts, 2004, ISBN 0262134470. 25
- [92] Russell, James A: *Core affect and the psychological construction of emotion*. Psychological review, 110(1):145–72, 2003, ISSN 0033-295X. <http://www.ncbi.nlm.nih.gov/pubmed/12529060>. 25
- [93] Bentley, Todd, Lorraine Johnston e Karola von Baggo: *Putting some emotion into requirements engineering*. Em *Proc. of the 7th Australian Workshop on Requirements engineering*, páginas 227–241, Melbourne, Victoria, AU, 2002. Deakin University. 26, 27
- [94] Callele, David, Eric Neufeld e Kevin Schneider: *An introduction to experience requirements*. Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010, páginas 395–396, 2010. 26, 27
- [95] Csikszentmihalyi, Mihaly: *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. Jossey-Bass, San Francisco, USA, 1975. 27, 34
- [96] Katja Tschimmel: *Design Thinking as an effective Toolkit for Innovation Katja Tschimmel*. XXIII ISPIM Conference: Action for Innovation: Innovating from Experience, página 20, 2012. 28, 29
- [97] Brown, Tim: *Change by design*. Collins Business., New York, NY, USA, 2009. 28, 29, 30, 69
- [98] Pressman, Andrew: *Design Thinking*. Routledge, New York, NY, USA, 2019. <http://www.icsid.org/smallbox4/file.php?sb4bdef72141c99>. 28, 29
- [99] Pressman, Roger S: *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005. 29
- [100] Blank, Steve e Bob Dorf: *Manual do Empreendedor*. Alta Books, Rio de Janeiro, RJ, BA, 2014. 30
- [101] MacDonald, Craig M. e Lillian Rozaklis: *Assessing the implementation of authentic, client-facing student projects in user experience (UX) education: Insights from multiple stakeholders*. Proceedings of the Association for Information Science and Technology, 54(1):268–278, 2017, ISSN 23739231. 34, 35

- [102] Getto, Guisepppe e Fred Beecher: *Toward a Model of UX Education : Training UX Designers*. IEEE Transactions on Professional Communication, 59(2):153–164, 2016. 35
- [103] Conradie, Peter D, Cesar Vandeveld, Jolien De Ville e Jelle Saldien: *Prototyping Tangible User Interfaces: Case Study of the Collaboration between Academia and Industry*. International Journal of Engineering Education, 32(2, A):726–737, 2016, ISSN 0949-149X. 35
- [104] Gerber, Elizabeth M., Jeanne Marie Olson e Rebecca L. D. Komarek: *Extracurricular Design-Based Learning: Preparing Students for Careers in Innovation**. International Journal of Engineering Education, 28(2):317–324, 2012, ISSN 0949149X (ISSN). 35
- [105] Biolchini, Jorge, Paula Gomes Mian, Ana Candida Cruz Natali e Guilherme Horta Travassos: *Systematic review in software engineering*. System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES, 679(05):45, 2005. 37
- [106] Uman, Lindsay S: *Systematic reviews and meta-analyses*. Journal of the Canadian Academy of Child and Adolescent Psychiatry, 20(1):57, 2011. 39, 41
- [107] Tacconelli, Evelina: *Systematic reviews: Crd’s guidance for undertaking reviews in health care*. The Lancet Infectious Diseases, 10(4):226, 2010. 39
- [108] Methley, Abigail M, Stephen Campbell, Carolyn Chew-Graham, Rosalind McNally e Sudeh Cheraghi-Sohi: *Pico, picos and spider: a comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews*. BMC health services research, 14(1):579, 2014. 39
- [109] Gomes Filho, AVELINO FERREIRA: *Modelo de ensino baseado nos métodos ágeis de desenvolvimento de software*. Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2016. 40
- [110] Kitchenham, Barbara, O Pearl Brereton, David Budgen, Mark Turner, John Bailey e Stephen Linkman: *Systematic literature reviews in software engineering—a systematic literature review*. Information and software technology, 51(1):7–15, 2009. 40
- [111] Dresch, Aline, Daniel Pacheco Lacerda e José Antonio Valle Antunes Júnior: *Design science research: método de pesquisa para avanço da ciência e tecnologia*. Bookman Editora, 2015. 43
- [112] Aibin, Michal e Aaron Hunter: *On Faculty Supervision in Industry Projects*. Western Canadian Conference on Computing Education, páginas 1–5, 2018. 48, 50, 51, 52, 53, 54, 55, 56, 58
- [113] Llopis, Fernando e Fernando G. Guerrero: *Introducing competitiveness and industry involvement as learning tools*. IEEE Global Engineering Education Conference, EDUCON, 2018-April:298–307, 2018, ISSN 21659567. 48, 52, 54, 55, 56, 58

- [114] Pariata, Mayra e Nora Montaña: *Software Factory, from professional environment to academic environment: Proposal to build competences through authentic activities in the context of software engineering*. Proceedings of the 2014 Latin American Computing Conference, CLEI 2014, páginas 1–10, 2014. 48, 52, 54, 55, 56, 58
- [115] Olszewska, Marta, Sergey Ostroumov e Mikolaj Olszewski: *To agile or not to agile students (with a twist): Experience report from a student project course*. Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, páginas 83–87, 2017, ISSN 00223654. 48, 51, 53, 54, 55, 56, 58
- [116] Cadenas, José Tomás, Rosseline Rodríguez e MacRinger Omaña: *Conceptualizing a didactics experience: Mini-Project Software Development*. Proceedings of the 2014 Latin American Computing Conference, CLEI 2014, 2014. 48, 50, 51, 53, 55, 56, 58
- [117] Kilamo, Terhi, Imed Hammouda e Mohamed Amine Chatti: *Teaching collaborative software development: A case study*. Proceedings - International Conference on Software Engineering, páginas 1165–1174, 2012, ISSN 02705257. 48, 51, 52, 53, 54, 55
- [118] Foster, David, Filippo Gilardi, Paul Martin, Wei Song, Dave Towey e Andrew White: *Students as co-producers in a multidisciplinary software engineering project: addressing cultural distance and cross-cohort handover*. Teachers and Teaching: Theory and Practice, 24(7):840–853, 2018, ISSN 13540602. 48, 53, 55, 56, 58
- [119] Martinez-Arias, Juan Carlos e Gerardo M. Sarria M.: *Didactic and interdisciplinary experiences in a software engineering course*. Proceedings - Frontiers in Education Conference, FIE, páginas 1800–1805, 2013, ISSN 15394565. 48, 51, 52, 53, 54, 55, 58
- [120] Schefer-Wenzl, Sigrid e Igor Miladinovic: *Game Changing Mobile Learning Based Method Mix for Teaching Software Development*. Proceedings of the 16th World Conference on Mobile and Contextual Learning - mLearn 2017, páginas 1–7, 2017. <http://dl.acm.org/citation.cfm?doid=3136907.3136912>. 48, 50, 51, 52, 53, 54, 55, 57
- [121] Mahnič, Viljan: *Student projects as a means of cooperation between academia and industry: Some experiences in the area of software engineering education*. World Transactions on Engineering and Technology Education, 15(3):239–244, 2017, ISSN 14462257. 49, 51, 52, 53, 54, 55, 58
- [122] Rick, Detlef, Marcel Morisse e Ingrid Schirmer: *Bringing contexts into the classroom*. Proceedings of the 7th Workshop in Primary and Secondary Computing Education on - WiPSCE '12, página 105, 2012, ISSN 0952-4649. <http://dl.acm.org/citation.cfm?doid=2481449.2481476>. 49, 52, 55, 58
- [123] Fagerholm, Fabian, Arto Hellas, Matti Luukkainen, Kati Kyllönen, Sezin Yaman e Hanna Mäenpää: *Designing and implementing an environment for software start-up education: Patterns and anti-patterns*. Journal of Systems and Software, 146:1–13, 2018, ISSN 01641212. <https://doi.org/10.1016/j.jss.2018.08.060>. 49, 52, 53, 54, 58

- [124] Pham, Yen Dieu, Davide Fucci e Walid Maalej: *A First Implementation of a Design Thinking Workshop During a Mobile App Development Project Course*. ACM/IEEE International Workshop on Software Engineering Education for Millennials, páginas 56–63, 2018, ISSN 02705257. <http://arxiv.org/abs/1803.10587>. 49, 55, 56, 58
- [125] Kizaki, Satoru, Yasuyuki Tahara e Akihiko Ohsuga: *Software development PBL focusing on communication using scrum*. Proceedings - 2014 IIAI 3rd International Conference on Advanced Applied Informatics, IIAI-AAI 2014, páginas 662–669, 2014. 49, 50, 51, 52, 53, 54, 55, 58
- [126] Abler, Randal, Ed Coyle, Agrita Kiopa e Julia Melkers: *Team-based software/system development in a vertically-integrated project-based course*. Proceedings - Frontiers in Education Conference, FIE, páginas T3F-1–T3F-7, 2011, ISSN 15394565. 49, 50, 51, 52, 53, 54
- [127] Rupakheti, Chandan R., Mark Hays, Sriram Mohan, Stephen Chenoweth e Amanda Stouder: *On a pursuit for perfecting an undergraduate requirements engineering course*. Journal of Systems and Software, 144:366–381, 2018, ISSN 01641212. <https://doi.org/10.1016/j.jss.2018.07.008>. 49, 54, 55, 56, 58
- [128] Intayoad, Wacharawan: *PBL framework for enhancing software development skills: An empirical study for information technology students*. Wireless Personal Communications, 76(3):419–433, 2014, ISSN 09296212. 49, 51, 52, 53, 55, 56, 57, 58
- [129] Marques, Maira, Sergio F. Ochoa, Maria Cecilia Bastarrica e Francisco J. Gutierrez: *Enhancing the Student Learning Experience in Software Engineering Project Courses*. IEEE Transactions on Education, 61(1):63–73, 2018, ISSN 00189359. 49, 50, 51, 52, 53, 54, 58
- [130] Detmer, Richard, Cen Li, Zhijiang Dong e Judy Hankins: *Incorporating Real-world Projects in Teaching Computer Science Courses*. Proceedings of the 48th Annual Southeast Regional Conference, páginas 24:1—24:6, 2010. <http://doi.acm.org/10.1145/1900008.1900042>. 49, 51, 54, 55, 56, 58
- [131] Ilkan, Mustafa, Hasan Amca e Ersun Iscioglu: *Grooming IT Students for Industry Through Industrial Training and Graduation Project Work*. Information-an International Interdisciplinary Journal, 13(4):1219–1242, 2010, ISSN 1343-4500. 50, 53, 55, 56, 58
- [132] Stozhko, Natalia, Boris Bortnik, Ludmila Mironova e Albina Tchernysheva: *Interdisciplinary project-based learning: technology for improving student cognition*. Research in Learning Technology, 23(1063519):1–13, 2015. 50, 57
- [133] Mahnič, Viljan: *The capstone course as a means for teaching agile software development through project-based learning*. World Transactions on Engineering and Technology Education, 13(3):225–230, 2015, ISSN 14462257. 50, 51, 52, 53, 54, 58
- [134] Bespalko, VP: *Standardization of education: basic ideas and concepts*. Pedagogy, 5:16–25, 1993. 57

- [135] Prodanov, Cleber Cristiano e Ernani Cesar de Freitas: *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. Editora Feevale, 2013. 59
- [136] WOOMOO: *Pop 2.0 - prototyping on paper*, Aug. 2018. <https://play.google.com/store/apps/details?id=in.woomoo.pop>. 64
- [137] Courtney, Jonathan: *The design sprint 2.0: What is it and what does it look like?*, Aug. 2018. <https://www.invisionapp.com/blog/design-sprint-2/>. 66
- [138] Arno Delhij, Rini Van Solingen, Willy Wijnands Jeff Sutherland: *O Guia EduScrum - As regras do jogo*, 2016. http://eduscrum.nl/en/file/CKFiles/0{}_guia{}_eduScrum.pdf. 80, 86, 87
- [139] Gray, Dave, Sunni Brown e James Macanuso: *Gamestorming: A playbook for innovators, rulebreakers, and changemakers*. " O'Reilly Media, Inc.", 2010. 82
- [140] *ScreenCast - apps on google play*. <https://play.google.com/store/apps/details?id=com.rosiapps.cast>. 84
- [141] Ferreira, Bruna, Williamson Silva, Edson Oliveira e Tayana Conte: *Designing Personas with Empathy Map*. Em *Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering*, páginas 501–505, 2015. 84
- [142] Klement, Alan: *Replacing the user story with the job story*, Nov 2013. <https://jtbd.info/replacing-the-user-story-with-the-job-story-af7cdee10c27>. 84

Apêndice A

Roteiro das Aulas da Design Sprint para Aprendizagem Baseada em Projetos

Dia 1

Tempo estimado: 1h e 05m

Materiais:

Item	Quantidade	Finalidade
Lápis	<i>n</i>	Anotações durante a palestra relâmpago do especialista
Borracha	<i>n</i>	Anotações durante a palestra relâmpago do especialista
Blocos de notas	<i>n</i>	Anotações durante a palestra relâmpago do especialista
Blocos de notas autoadesivas de 76mm x 102 mm	<i>n/2</i>	Escrita dos <i>How Might We</i> e das premissas
Bloco de notas autoadesivas de 47,5 mm x 47,5 mm	1	Desenho dos estados emocionais do <i>stakeholders</i> e das causas dos estados emocionais ruins
Cartolina	1	Terá o Mapa da Jornada do Usuário desenhado nela
Folhas impressas	<i>n</i>	Constará nelas o roteiro de perguntas da entrevista imersiva e serão distribuídas aos alunos
Canetas <i>Sharpie</i>	<i>n</i>	Escrita dos <i>How Might We</i> .

* *n* representa a quantidade de alunos participantes da Design Sprint

Antes da aula

- Desenhar Mapa da Jornada do Usuário no quadro

40 minutos - Divergência (Entender)

- 5 min - Explicação de como a DS funciona
- 30 min - Especialista faz uma palestra relâmpago respondendo às perguntas feitas pelo facilitador. Enquanto isso, os alunos fazem anotações no formato HMW para dúvidas que eles tiverem a respeito sobre como irão ajudar o especialista. Ao final do palestra, os alunos podem fazer suas próprias perguntas. O *mindset* para fazer boas questões HWM é: (1) Encontrar um obstáculo ou problema de experiência do *stakeholder* no caminho até o objetivo final; (2) Transformar o problema em uma pergunta HMW. Ex.: *Problema: O usuário se perde na quantidade imensa de*

notificações que chega; Questão HWM: Como poderíamos ajudar o usuário a não se perder com o grande volume de notificações?

- O roteiro de perguntas é:
 - o *Me conte um pouco da sua rotina onde o problema que você tem aparece?*
 - o *Me conte como foi a última vez que tentou (alguma tarefa que deixou o problema evidente)?*
 - o *Quais as piores partes disso tudo?*
 - o *Por que elas são ruins?*
 - o *Como você se sente quando passa por essas partes problemáticas?*
 - o *O que uma boa solução para esse problema deveria ter?*
 - o *Existe mais algo que queira contar sobre esse problema?*
- *5 min* - Os alunos identificam os estados emocionais do stakeholder segundo as etapas do mapa da Jornada do Usuário (O/A <causa> deixa o usuário <estado emocional>. Ex.: *O tempo perdido olhando as notificações* deixa o usuário *irritado*). O professor confere se aquilo que os alunos escreveram corresponde à realidade e faz eventuais correções, se necessário.

25 minutos - Convergência (Entender)

- *10 min* - Escolha do(s) HWM que seja(m) abordado(s) na Sprint com base em *dot voting* (quatro votos para cada pessoa) e de uma métrica que represente numericamente o estado futuro do usuário onde será encadeada uma situação de satisfação. Ex.: *aumentar quantidade de experiência profissional colocada no currículo de um aluno enquanto ele está na universidade ou diminuir a quantidade de dias que um aluno espera para ser contratado depois de ter enviado um currículo*. O(s) HWM escolhido(s) é(são) aquele(s) que visualmente tiver(em) recebido mais votos.
- *15 min* - Divisão e explicação da atividade e de como será a apresentação. As entrevistas são feitas com base no roteiro de perguntas abaixo e a apresentação do dia 2 é realizada com o professor refazendo essas perguntas para o grupo de alunos que obteve as respostas. Cada grupo deve documentar as respostas, pois elas serão úteis na criação do mapa de empatia e das *job stories* depois da Design Sprint.
 - o **Categorias dos grupos:**
 - **Voz do cliente**
 - Perguntas:
 - o Você sofre com problema X?
 - o Que outras pessoas você acha que sofre com o problema X?

- o Me conte sobre a última vez que sofreu com problema X
 - o O que te chateou mais?
 - o Porque isso te chateou?
 - o Como você lida com isso hoje?
 - o Por que não é satisfatório?
 - o *O que uma boa solução para esse problema deveria ter?*
- **Como as coisas funcionam;**
 - Perguntas:
 - o Quais tipos de pessoas sofre com o problema X?
 - o Por que elas sofrem com o problema X?
 - o Quais as formas possíveis de se resolver isso hoje em dia?
 - o Por que imagina que as pessoas não as resolvem dessa maneira?
- **Soluções concorrentes;**
 - Questões de pesquisa:
 - o O que já existe para resolver esse problema?
 - o Quais as funcionalidades específicas do produto/maneira de se fazer concorrente que lidam com o problema X?
 - o Quais os problemas das maneiras existentes de se resolver o problema X?
 - o Tem alguma funcionalidade de outros produtos que poderiam ser usadas para resolver o problema X?

Dia 2

Tempo estimado: 2h e 25m

Materiais:

Item	Quantidade	Finalidade
Lápis	n	Anotações durante as apresentações 360
Borracha	n	Anotações durante as apresentações 360
Bloco de notas	n	Anotações durante as apresentações 360 e desenho de esboços experimentais
Bloco de notas autoadesivas de 76mm x 102 mm	$n/2$	Escrita dos <i>How Might We</i> , das premissas e complementação dos painéis do esboço final da solução
Bloco de notas autoadesivas de 47,5 mm x 47,5 mm	1	Escrita das considerações das críticas relâmpago
Fita adesiva	1	Afixar os esboços finais na parede
Folha a4 em branco	$n * 3$	Exercício do <i>Crazy 8's</i> e desenho da proposta final de solução
Cartela adesivos de bolinha - cor 1	n	<i>Dot voting</i>
Cartela adesivos de bolinha - cor 2	1	Marcação das funcionalidades que possuem objetivos iguais
Cartolina	1	Terá o Mapa da Jornada do Usuário desenhado nela
Canetas <i>Sharpie</i>	n	Escrita dos <i>How Might We</i> .
Data Show	1	Apresentações 360

* n representa a quantidade de alunos participantes da Design Sprint

Antes da aula

- Redesenhe ou cole a cartolina com o mapa da Jornada do Usuário no quadro
-

1 hora e 05 minutos - Divergência (Buscar soluções)

- 5 min - Preparação das apresentações
- 10 min por grupo - Apresentações 360: Cada grupo dos três da aula passada responde à entrevista que o professor fez baseado no roteiro entregue aos alunos na aula 1. Enquanto isso, os grupos que não estão apresentando fazem anotações e desenhos das ideias promissoras.
- 30 min - Esboço em três etapas: A primeira etapa serve para revisar as informações geradas até o momento, a segunda etapa serve para aquecer o lado criativo do cérebro e a última etapa para fazer a proposta final através do esboço. As soluções dessa etapa devem endereçar as questões levantadas na rodada de *How Might We*, a mudança do estado emocional ruim do *stakeholder* e o benefício futuro representado pela métrica escolhida na aula 1.
 - o 10 min - 1ª etapa: Todos andam pela sala, observam as soluções do quadro, as anotações, o objetivo de longo prazo, as perguntas da *Sprint*, outras soluções na internet e fazem anotações e/ou pequenos esboços experimentais no bloco de notas.
 - o 2 min - 2ª etapa: Explicação do *Crazy 8's* e do MESCRAI (Modifique, Elimine, Substitua, Combine, Rearranje, Adapte, Inverta) para criar variações.
 - o 8 min ☺ - 2ª etapa: Por meio da técnica *Crazy 8's* os alunos desenham, em 8 minutos, 8 variações de uma ideia de solução.
 - o 10 min - Com base em tudo que fizeram até agora e também depois de terem aquecido o lado criativo do cérebro com o *Crazy 8's*, são entregues folhas A4 para desenho das propostas finais. A folha deve ser dividida em 3 partes. Uma parte tem o desenho do começo da interação, outra tem o desenho do meio da interação e outra tem o desenho do fim da interação (alcance do objetivo) do *stakeholder* com o novo sistema; Os alunos desenharão propostas que englobem todo o fluxo do mapa da jornada do usuário, endereçando as perguntas escolhidas no HMW, a mudança dos estados emocionais ruins encontrados na aula 1 e com foco na métrica de longo prazo que representa a satisfação do usuário.

1h20 minutos - Convergência (Decidir)

- Aprox. 24 min - Críticas relâmpago: Com os esboços dos alunos fixados na parede, haverá uma sessão de crítica estruturada que deve ter como teto o tempo de 3 minutos por esboço. As críticas funcionam assim: (1) Primeiro os alunos dizem o que gostaram e o que não gostaram, enquanto o professor escreve essas considerações em notas

autoadesivas e as fixa perto do esboço (2) O autor se revela e faz suas considerações. Há a oportunidade de retirar notas autoadesivas que tenham sido respondidas.

- *6 min* - Decisão por *dot voting*: Os alunos votam nas partes dos esboços que mais gostaram, quantos votos quiserem.
- *40 min* - É necessário juntar as partes com mais *dots* em soluções numa ótima e o desenho de um *storyboard* de uso da solução com começo, meio e fim considerando as funcionalidades mais bem votadas.
- *10 min* - Suposição *storming*: Os alunos escrevem em notas autoadesivas o que eles sabem ou acham que sabem sobre o uso da solução que estão propondo. Depois disso, o professor classifica com a ajuda dos alunos a posição destas notas autoadesivas em um gráfico de esforço e importância.
- *20 min* - Com base na(s) suposição(ões) escolhida(s) na atividade de suposição *storming* deve-se tentar responder à seguinte pergunta: Como testar isso? A partir de então um teste será projetado.

Dia 3

Tempo estimado: 2h e 25m

Materiais:

Item	Quantidade	Finalidade
Lápis	<i>n</i>	Anotações durante a validação
Borracha	<i>n</i>	Anotações durante a validação
Bloco de notas	<i>n</i>	Anotações durante a validação
Data Show	1	Apresentação da tela sendo testada
Computador	1	Execução do protótipo e auxílio na apresentação de tela sendo testada

* *n* representa a quantidade de alunos participantes da Design Sprint

Antes da aula

- Confira os equipamentos (Data Show, Computador, Dispositivo móvel, etc.)

40 minutos - Validação

- *Aprox. 5 min* – Cumprimente o usuário e apresente o protótipo. Diga ao usuário que algumas coisas não funcionarão e ele que se algo assim acontecer, ele não deve se preocupar.
- *Aprox. 20 min* – Dê ao usuário uma ou mais missões a serem executadas que sejam baseadas nas soluções classificadas por meio da atividade de suposição *storming*. Peça ao usuário pensar em voz alta enquanto ele navega na solução.
- *Aprox. 10 min* – Perguntas (*Debriefing*)
 - o O que acha desse produto em relação ao que já tem?
 - o Do que gostou desse produto? E do que não gostou?
 - o Como descreveria este produto para um amigo?
 - o Se tivesse três desejos para melhorar esse produto, quais seriam?