



UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Uma abordagem ágil para o escopo das linhas de produtos de software: um estudo de caso em uma microempresa

Trabalho de Conclusão de Curso

Amanda Israel Graeff Borges



Cascavel-PR

2021

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ - UNIOESTE

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Amanda Israel Graeff Borges

Uma abordagem ágil para o escopo das linhas de produtos de software: um estudo de caso em uma microempresa

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador(a): Ivonei Freitas da Silva

Cascavel-PR

2021

AMANDA ISRAEL GRAEFF BORGES

**UMA ABORDAGEM ÁGIL PARA O ESCOPO DAS LINHAS DE
PRODUTOS DE SOFTWARE: UM ESTUDO DE CASO EM UMA
MICROEMPRESA**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Ivonei Freitas da Silva (Orientador)
Colegiado de Ciência da Computação, UNIOESTE

Prof. Victor Francisco Araya Santander
Colegiado de Ciência da Computação, UNIOESTE

Prof. Luiz Antonio Rodrigues
Colegiado de Ciência da Computação, UNIOESTE

Cascavel, 28 de julho de 2022

Resumo

A combinação das Linhas de Produto de Software (LPS) e Metodologias ágeis é conhecida como uma estratégia aplicada em empresas com aplicações similares e que apresentam cenários de requisitos voláteis, pois possuem muitas variabilidades. Entretanto, uma das abordagens que combinam essas duas técnicas, RISE-DA (SILVA, 2013), não foi desenvolvida em micro e pequenas empresas, como as *very small companies*. Dessa forma, este trabalho propõe aplicar suas regras, atividades e artefatos na empresa BIT - Empresa Júnior de Ciência da Computação. Para avaliar essa proposta, a pesquisa será reportada por meio de um estudo de caso, em que serão avaliados o mercado que a empresa atua, suas aplicações e quais *features* principais precisarão ser detalhadas, além de aplicar reuniões de validação e um *feedback* final para conseguir uma análise melhor de quais deveriam ser as funcionalidades desenvolvidas para o reuso. A abordagem RISE-DA consegue suprir as necessidades de *very small entities* ao catalogar e detalhar as *features* mais importantes para o domínio de negócios de uma microempresa (enquanto esse domínio for utilizado) que possua a possibilidade de reuso em sua estrutura. Dessa forma, essa análise de domínio permite que os membros da empresa utilizem os dados catalogados para desenvolverem aplicações de qualidade mais rapidamente e com um alto retorno de investimento.

Palavras-chave: *feature*. *subfeature*. RISE-DA. reuso. similaridade. variabilidade.

Lista de figuras

Figura 1 – Ciclo de vida das LPSs	17
Figura 2 – Árvore de <i>features</i>	22
Figura 3 – Componentes do PuLSE	26
Figura 4 – Comparativo do ISO/IEC 29110	29
Figura 5 – Representação dos segmentos das empresas de software	31
Figura 6 – Censo 2018 do MEJ	32
Figura 7 – Processo do RISE-DA	35
Figura 8 – Parte 1 do processo RISE-DA	36
Figura 9 – Construção da Análise de Mercado	39
Figura 10 – Construção do mapa de produtos	40
Figura 11 – Priorização das <i>features</i> principais	42
Figura 12 – Parte 2 do processo RISE-DA	42
Figura 13 – Iterações do processo ágil	43
Figura 14 – Utilização do Scrum no processo RISE-DA	46
Figura 15 – Avaliação do processo RISE-DA	49
Figura 16 – Gráfico <i>burndown</i> - <i>Sprint</i> 1	72
Figura 17 – Árvore de <i>features</i> - CRUD	76
Figura 18 – Árvore de <i>features</i> - Controle de acesso dos usuários	82
Figura 19 – Gráfico <i>burndown</i> - <i>Sprint</i> 2	85
Figura 20 – Árvore de <i>features</i> - Criação de formulários	89
Figura 21 – Árvore de <i>features</i> - download de informações	90
Figura 22 – Árvore de <i>features</i> - Sistema de notificação	91
Figura 23 – Árvore de <i>features</i> - Visualização de relatórios	92
Figura 24 – Gráfico <i>burndown</i> - <i>Sprint</i> 3	95
Figura 25 – Árvore de <i>features</i> - Experiência do usuário	99
Figura 26 – Árvore de <i>features</i> - Páginas informativas	104
Figura 27 – Gráfico <i>burndown</i> - <i>Sprint</i> 4	107
Figura 28 – Árvore de <i>features</i> - Modelagem dos dados (parte 1)	115
Figura 29 – Árvore de <i>features</i> - Modelagem dos dados (parte 2)	116
Figura 30 – Árvore de <i>features</i> - Modelagem dos dados (parte 3)	117
Figura 31 – Árvore de <i>features</i> - Modelagem dos dados (parte 4)	118
Figura 32 – Gráfico <i>burndown</i> - <i>Sprint</i> 5	121
Figura 33 – Árvore de <i>features</i> - Controle de caixa	124
Figura 34 – Árvore de <i>features</i> - Gerência de compra e venda	125
Figura 35 – Árvore de <i>features</i> - Atendimento ao cliente	126
Figura 36 – Árvore de <i>features</i> - Envio de currículo	127

Lista de quadros

Quadro 1 – Análise de Mercado	38
Quadro 2 – Mapa de Produtos	39
Quadro 3 – Avaliação das <i>Features</i> Principais	41
Quadro 4 – Classificação do Potencial de Reuso e Riscos	41
Quadro 5 – Outros Critérios de Classificação	41
Quadro 6 – Backlog do Domínio	44
Quadro 7 – Backlog do <i>Sprint</i>	44
Quadro 8 – <i>Template</i> relacionado ao Backlog do <i>Sprint</i>	44
Quadro 9 – Análise de Mercado - Parte 1	55
Quadro 10 – Análise de Mercado - Parte 2	56
Quadro 11 – Lista de aplicações	58
Quadro 12 – Lista de <i>features</i>	60
Quadro 13 – Mapa de Produtos - parte 1	61
Quadro 14 – Mapa de Produtos - parte 2	63
Quadro 15 – Priorização das <i>features</i> principais - parte 1	64
Quadro 16 – Priorização das <i>features</i> principais - parte 2	65
Quadro 17 – Backlog do Domínio	70
Quadro 18 – Backlog da <i>Sprint</i> 1	71
Quadro 19 – CRUD - parte 1	73
Quadro 20 – CRUD - parte 2	74
Quadro 21 – Controle de acesso dos usuários - parte 1	78
Quadro 22 – Controle de acesso dos usuários - parte 2	79
Quadro 23 – Controle de acesso dos usuários - parte 3	80
Quadro 24 – Backlog da <i>Sprint</i> 2	84
Quadro 25 – Sistema de notificação, Visualização de relatórios, Criação de formulários e Download de informações - parte 1	86
Quadro 26 – Sistema de notificação, Visualização de relatórios, Criação de formulários e Download de informações - parte 2	87
Quadro 27 – Backlog da <i>Sprint</i> 3	94
Quadro 28 – Experiência do usuário - parte 1	96
Quadro 29 – Experiência do usuário - parte 2	97
Quadro 30 – Páginas informativas - parte 1	101
Quadro 31 – Páginas informativas - parte 2	102
Quadro 32 – Páginas informativas - parte 3	103
Quadro 33 – Backlog da <i>Sprint</i> 4	106
Quadro 34 – Modelagem de dados - parte 1	109

Quadro 35 – Modelagem de dados - parte 2	110
Quadro 36 – Modelagem de dados - parte 3	111
Quadro 37 – Modelagem de dados - parte 4	112
Quadro 38 – Modelagem de dados - parte 5	113
Quadro 39 – Modelagem de dados - parte 6	114
Quadro 40 – Backlog da <i>Sprint 5</i>	120
Quadro 41 – Controle de caixa - parte 1	122
Quadro 42 – Gerência de compra e venda, Atendimento ao cliente e Envio de currículo - parte 1	129
Quadro 43 – Controle de caixa - parte 2	130
Quadro 44 – Gerência de compra e venda, Atendimento ao cliente e Envio de currículo - parte 2	131
Quadro 45 – Controle de caixa - parte 3	132
Quadro 46 – Gerência de compra e venda, Atendimento ao cliente e Envio de currículo - parte 3	133

Lista de abreviaturas e siglas

APLE	<i>Agile Product Line Engineering</i>
CMMI	<i>Capability Maturity Model Integration</i>
DSDM	<i>Dynamic Software Development Method</i>
ED	Engenharia de Domínio
EA	Engenharia de Aplicação
FDD	<i>Feature Driven Development</i>
LPS	Linhas de Produto de Software
MARES	Método de Avaliação de Processo de Software
MEJ	Movimento Empresa Júnior
OORT	<i>Object-Oriented Reading Techniques</i>
PA	Proposed Approach
PR	Paraná
PuLSE	<i>Product Line System and Software Engineering</i>
RAPID	<i>Rapid Assessment for Process Improvement for Software Development</i>
RISE-DA	<i>Reuso in Software Engineering - Domain Analysis</i>
RISE-CBD	<i>Reuso in Software Engineering - Component-Based Development</i>
SPINI	<i>Software Process Improvement Initiation</i>
SPIRE	<i>Software Process Improvement in Regions of Europe</i>
SPLICE	<i>Software Product Line Integrated Construction Environment</i>
UnB	Universidade de Brasília
UNESP	Universidade Estadual Paulista
UNIOESTE	Universidade Estadual do Oeste do Paraná
UFMG	Universidade Federal de Minas Gerais
UFP	Universidade Federal de Pelotas

UFPE	Universidade Federal de Pernambuco
UFSC	Universidade Federal de Santa Catarina
UECE	Universidade Estadual do Ceará
USP	Universidade de São Paulo
VSE	<i>Very Small Entity</i>

Sumário

1	Introdução	11
2	Panorama do referencial	15
2.1	Reuso de software	15
2.2	Linhas de Produto de Software	16
2.2.1	<i>Features</i> em Linhas de Produto de Software	18
2.2.2	Escopo de Linhas de Produto de Software	19
2.2.3	Variabilidade e Similaridade em Linhas de Produto de Software	20
3	Trabalhos Relacionados	23
3.1	SPLICE	23
3.2	APLE	24
3.3	PuLSE	25
3.4	Metodologia Ágil utilizando Linhas de Produto de Software	26
3.5	Outras propostas	28
4	Cenário de Pesquisa	30
4.1	Micro e pequenas empresas	30
4.2	Empresas Juniores	31
4.3	BIT - Empresa Júnior de Ciência da Computação	32
5	RISE-DA	35
5.1	Parte 1	36
5.2	Parte 2	40
6	Metodologia	47
7	Limitações do Processo	50
7.1	Problemática BIT	50
7.2	Membros	51
7.2.1	Processo seletivo	51
7.3	Materiais	52
8	Resultados e Discussão	53
8.1	Treinamento	53
8.2	Pré-Análise	54
8.2.1	Análise de Mercado	54

8.3	Mapa de Produtos	57
8.3.1	Lista de Aplicações	57
8.3.2	Lista de <i>features</i>	57
8.3.3	Mapa de Produtos - Primeira versão	59
8.4	Priorização das <i>features</i> principais	62
8.5	Processo iterativo	67
8.5.1	Backlog do Domínio	68
8.5.2	Backlog do <i>Sprint</i>	68
8.5.2.1	<i>Sprint 1</i>	69
8.5.2.2	<i>Sprint 2</i>	83
8.5.2.3	<i>Sprint 3</i>	93
8.5.2.4	<i>Sprint 4</i>	105
8.5.2.5	<i>Sprint 5</i>	119
8.6	Questionário e Entrevista	128
9	Conclusão e trabalhos futuros	136
9.1	Lições Aprendidas	138
	 Referências	 140
	 Glossário	 144

1

Introdução

As empresas de software são diferentes em tamanho, setor de mercado, tempo de entrega, gerenciamento, cultura e localização geográfica, assim como em outros aspectos referentes ao seu escopo. Para quem desenvolve software, isso levanta a questão de qual seria a melhor abordagem para as empresas em relação a processos que diminuem o tempo de produção ao aumentar a produtividade. Independentemente do tamanho da organização, da quantidade de recursos disponíveis e da maturidade e estabilidade do segmento de mercado, é fundamental que haja a compreensão de quais são os produtos de software que precisam ser desenvolvidos, ou seja, compreender o domínio em que a empresa está inserida ou poderá estar ao construir sistemas de software (JOHN; EISENBARTH, 2009).

Para entender o domínio de uma organização, a primeira ação a ser realizada é definir o escopo deste domínio (JOHN; EISENBARTH, 2009). Esta atividade pode ser feita por meio de três tipos de escopo: escopo de produtos, escopo de domínio e escopo de ativos que serão explicados detalhadamente no capítulo 2. Entretanto, atividades de escopo tendem a acarretar em custos e esforços muito altos para as empresas, porque é necessário um time de especialistas que determinam quais dados são importantes para o mapeamento de escopo por meio de reuniões periódicas, uma vez que essa atividade leva muito tempo para ser mapeada e acaba trazendo um retorno a longo prazo para as empresas (JOHN; EISENBARTH, 2009).

Em times ou organizações de software muito pequenos, o domínio, os produtos, as *features* e os ativos costumam apresentar instabilidades e são suscetíveis a muitas mudanças. Por conta disso, para esses segmentos de negócios, uma atividade de escopo acaba se tornando inviável ao se utilizar métodos de mapeamento de escopo feitos para domínios tradicionais, como o método PuLSE (*Product Line System and Software Engineering*) (JOHN et al., 2006) que é usado em médias e grandes empresas. Isso acontece porque seria gerado ao final do processo documentos que não valeriam a longo prazo, por conta da instabilidade do domínio. Sendo assim, um método que possa moldar os artefatos, pela priorização das *features* principais do domínio

poderia ser mais apropriado para esse cenário, tornando esse processo mais leve, porque não haveria uma documentação exaustiva para a organização, uma vez que times pequenos costumam utilizar metodologias ágeis em seus processos.

Dentro da metodologia de Linhas de Produto de Software voltadas para o reuso, encontra-se a etapa de RISE-DA (*Reuso in Software Engineering - Domain Analysis*) (SILVA, 2013). Este método é capaz de estruturar um escopo de domínio que endereça cenários de pequenas empresas. O escopo de domínio é uma fase da análise de escopo que coleta informações de todos os domínios da empresa, de forma que consiga visualizar as similaridades entre eles. A primeira parte do processo do método RISE-DA tem como objetivo construir os artefatos de análise de mercado, mapa de produtos e priorização das *features* principais da empresa por meio da aquisição e avaliação dos dados de escopo da empresa com os *stakeholders* envolvidos. A segunda parte utiliza da metodologia Scrum com o objetivo de analisar as *subfeatures* e as similaridades e variabilidades entre essas *features*, atualizando os artefatos da primeira fase e construindo uma árvore de *features* que terão um detalhamento maior. Contudo, ainda não foi avaliada sua viabilidade em um cenário de microempresa, tal como, o exemplo de uma *Startup* de software ou empresa júnior.

Considerando as empresas brasileiras que atuam exclusivamente em desenvolvimento e produção de software, cerca de 95,3% podem ser classificadas como micro e pequenas empresas, segundo análise realizada pelo critério de número de funcionários (até 99 funcionários) (ABES, 2020), colocando esse nicho de mercado como uma importante fonte de pesquisas na área de Engenharia de Software. As instabilidades e outros desafios nessas pequenas organizações já foram identificados em outras estratégias de desenvolvimento de software como relatado nos artigos de SÁNCHEZ-GORDÓN e O’CONNOR (2016), MOLL (2013), O’CONNOR e COLEMAN (2009) e MAJCHROWSKI et al. (2016).

Nesse cenário de micro e pequenas empresas, temos as empresas juniores de tecnologia da informação que se enquadram na categoria de *very small entities*. Essas empresas estão restritas a diversas regulamentações que vem tanto da legislação do seu país, quanto da universidade que estão inseridas (JÚNIOR, 2021a). Portanto, seu cenário propõe uma escassez de recursos muito maior que a proposta por uma empresa que não esteja vinculada a essas entidades. As restrições impostas as empresas juniores, coloca o domínio da empresa muito longe de um cenário estável.

As empresas juniores são compostas por estudantes que estão envolvidos com empreendedorismo e querem uma experiência de trabalho dentro da faculdade, ou seja, a aquisição de conhecimento é o maior impulsionador dos funcionários, uma vez que estes não são remunerados, segundo a legislação brasileira (JÚNIOR, 2021a). Dessa forma, obter um domínio bem documentado seria de grande importância, pois geraria um legado organizacional muito rico a ser oferecido aos integrantes da organização.

O método RISE-DA seria apropriado para o ajuste e detalhamento do escopo de uma empresa júnior, porque a capacidade de customização das aplicações geraria uma maior facilidade

de aprendizado por meio das *features* dos produtos, onde é possível replicar o conhecimento passado entre [Desenvolvedores](#). Além disso a análise do escopo contribui para uma melhor visão do mercado que estão inseridos e, portanto, podem decidir se o caminho que a empresa está seguindo é do interesse dos estudantes.

Outras propostas, como APLE (*Agile Product Line Engineering*) ([KIANI et al., 2021](#)) e SPLICE (*Software Product Line Integrated Construction Environment*) ([VALE et al., 2014](#)), que possuem a mesma dinâmica de estruturar LPS com metodologias ágeis, não foram escolhidas uma vez que seria necessário uma equipe de análise de domínio e desenvolvimento operando juntos. Entretanto, nos anos de 2021 e 2022, na qual a abordagem foi aplicada, trouxe pouca mão de obra e um conhecimento não aprofundado dos especialistas envolvidos, fazendo com que apenas a etapa de escopo da Engenharia de Domínio de LPS fosse apropriada por meio das informações adquiridas de alguns especialistas e artefatos legados da empresa. Se os documentos gerados forem bem desenvolvidos antes dos *stakeholders* aplicarem esses conhecimentos nas aplicações, menos tempo e esforços serão perdidos ao se executar esse processo. Os capítulos 7 e 8 apresentam mais detalhes referentes as limitações do processo e como foram realizadas as etapas de treinamento e construção dos artefatos apesar das limitações apresentadas.

O objetivo deste estudo é avaliar a viabilidade do método RISE-DA, proposto por [Silva \(2013\)](#), em definir um escopo de domínio dentro de uma micro organização, particularmente em *very small entities*, que sejam voltadas ao ramo das Empresas Juniores envolvidas com tecnologia. O caso de estudo apresentado neste trabalho irá examinar o método RISE-DA na empresa BIT-Empresa Júnior de Ciência da Computação. Sua avaliação ocorrerá por meio de uma metodologia empírica de processos de software que inclui observações em campo para estudar cada parte do processo e estimar se ele gera os resultados adequados para o contexto da organização.

Os resultados apresentados pela pesquisa incluem a formação de uma árvore de *features* e um mapa de produtos que detalham cada uma das *features* e suas *subfeatures* presentes nas aplicações da empresa em seu domínio específico. Esses artefatos determinam se os produtos da organização possuem variabilidade e similaridade por meio da combinação das *features* analisadas e dessa forma, concluem se a empresa estudada possui um potencial de domínio para que suas aplicações possam ser implementadas para o reuso. O tempo para realizar as atividades propostas foi contabilizado entre as diferentes *Sprints* da metodologia ágil integrada ao método RISE-DA, e foi gerado um gráfico *burndown* de produtividade de cada *Sprint*. A possibilidade da viabilidade do processo para micro e pequenas empresas foi feita por meio de um questionário feito aos *stakeholders* que foram os especialistas inclusos em cada parte do processo, assim como uma entrevista para compreender e ter uma visão mais ampla das respostas fornecidas.

Este trabalho de conclusão de curso está estruturado em 8 capítulos principais. No capítulo 2 serão descritos os panoramas gerais do referencial necessárias para a contextualização da abordagem RISE-DA. Dessa forma, é necessário a introdução dos conceitos de Reuso de software ([SOMMERVILLE, 2011](#)) e Linhas de Produto de Software (LPS) ([SOMMERVILLE,](#)

2011). Para a estratégia de LPS, fala-se sobre *features* em LPS (ERIKSSON; BÖRSTLER; BORG, 2009), escopo de LPS (JOHN et al., 2006) e sobre variabilidade e similaridade em LPS (GURP; BOSCH; SVAHNBERG, 2001) que são os conceitos principais abordados neste trabalho.

No capítulo 3 são abordados os trabalhos relacionados. Aqui será descrita a análise de domínio PuLSE empregada para grandes empresas (JOHN et al., 2006), trabalhos que relacionam metodologias ágeis com a estratégia de LPS e outras propostas de abordagens engajadas em melhorar a produtividade de empresas em diversos contextos. Além disso, será falado da estratégia APPLE e SPLICE que possuem o mesmo objetivo da estratégia RISE-DA.

No capítulo 4 será contextualizado o cenário da pesquisa, especificando onde as micro e pequenas empresas se encontram no mercado e qual é a importância das empresas juniores no Brasil. Além disso, será feita a introdução da organização onde o caso de estudo deste trabalho será realizado.

No capítulo 5 será especificado a abordagem RISE-DA, apresentando os fluxogramas, os templates e os passos que precisarão ser realizados em cada parte do processo. Já no capítulo 6 será apresentado a abordagem utilizada para realização do caso de estudo, assim como o será o andamento do processo dentro da organização proposta para o caso de estudo. Além disso, no capítulo 7 serão apresentadas as limitações do processo e como isso afetou o processo de desenvolvimento da pesquisa na empresa selecionada.

No capítulo 8 serão apresentados os resultados e discussão da abordagem RISE-DA aplicada na empresa. Serão apresentados todos os documentos e artefatos criados e refinados durante todo o processo, assim como cada uma das fases executadas, dados catalogados e a entrevista feita por meio de um questionário com os membros da empresa. Por fim, o capítulo 9 terá as conclusões desse trabalho, assim como os possíveis trabalhos futuros, algumas recomendações e lições aprendidas.

2

Panorama do referencial

O trabalho apresentado é focado no mapeamento das similaridades e variabilidades dos produtos fabricados no contexto de pequenas empresas, utilizando a abordagem RISE-DA proposta por (SILVA, 2013). Essa proposta é fundamentada nas teorias de Linhas de Produtos de Software que é uma abordagem de reuso e, portanto, é utilizada para criar uma linha de produtos, em que esses produtos fazem parte de diferentes aplicações, mas que possuem uma arquitetura comum e componentes compartilhados (SOMMERVILLE, 2011). Para identificar um produto pertencente a uma linha de produtos, utiliza-se o conceito de *features*, pois elas são a principal forma de visualizar similaridades entre diferentes produtos, uma vez que essa funcionalidade especificada pela *feature* se aplica a mais de um produto em um mesmo domínio (ERIKSSON; BÖRSTLER; BORG, 2009).

O escopo é a primeira atividade a ser realizada na estratégia de Linhas de Produtos de Software, pois para que seja possível fazer uma análise de domínio em uma empresa, é necessário que existem ativos disponíveis, catalogados e estruturados para que ocorra uma análise bem-feita de cada um deles, com o objetivo de separá-los conforme sua similaridade e variabilidade (JOHN et al., 2006). Essa separação é necessária, pois ao fazer essa distinção é possível classificar um produto como potencialmente reusável (GURP; BOSCH; SVAHNBERG, 2001).

2.1 Reuso de software

A engenharia de software baseada no reuso é uma prática que orienta o processo de desenvolvimento dos produtos, levando em conta artefatos reusáveis definidos dentro do espaço do domínio da aplicação, que possui similaridades e variabilidades. O reuso de software foi um conceito primeiramente proposto há mais de 40 anos por Doug McIlroy, ao mostrar o intuito de fabricar componentes em massa (SOMMERVILLE, 2011).

Apenas nos anos 2000 foi considerado como uma estratégia de negócios para as empresas

de software. Começou-se a ver o software como um ativo valioso no mercado que alavancou a busca por menores custos de produção e manutenção, menor tempo de entrega e maior qualidade e produtividade no desenvolvimento de software. Dessa forma, passou-se a promover a estratégia de reuso para aumentar os investimentos nesses ativos (SOMMERVILLE, 2011).

Nos últimos 20 anos, foram desenvolvidas diversas técnicas para atender o reuso de software, explorando as semelhanças entre os domínios das aplicações ao verificar seu potencial de reuso. Cada uma das técnicas utilizadas depende dos requisitos do sistema em desenvolvimento, a tecnologia e os ativos reusáveis disponíveis durante o processo, além das habilidades da equipe de desenvolvimento (SOMMERVILLE, 2011).

A técnica utilizada no presente trabalho para atender o reuso de software, é baseada nas Linhas de Produtos de Software. Ao escolher a estratégia de reuso, procura-se atender alguns benefícios que o reuso oferece as empresas de software. Os softwares reusados proporcionam uma confiança aumentada, uma vez que as funcionalidades implementadas com reuso foram testadas exaustivamente e já estão em produção em outros produtos da empresa. Portanto, é mais fácil implementar essas funcionalidades em produtos que estão sendo produzidos, levando a um desenvolvimento acelerado. Além disso, é possível encapsular o conhecimento de especialistas em ferramentas, linguagens de programação e lógicas otimizadas, onde uma padronização pode ser criada para refletir boas práticas de software.

2.2 Linhas de Produto de Software

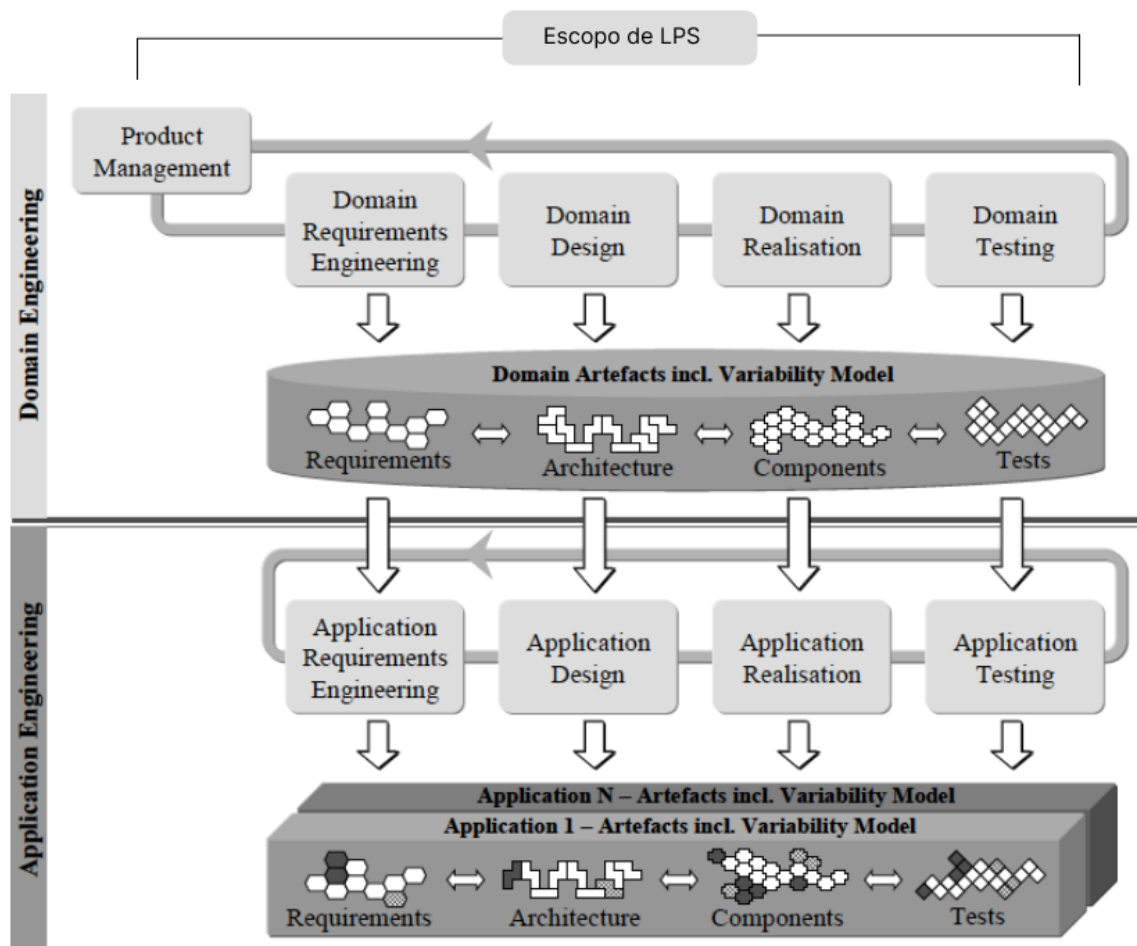
Linhas de Produto de Software (LPS) é uma estratégia de reuso que generaliza a aplicação avaliada em torno de uma arquitetura comum para que possa ser adaptada às necessidades de diferentes clientes, minimizando os custos de desenvolvimento. Desenvolver as aplicações utilizando-se de plataformas reusáveis implica em usar customizações em massa por meio do gerenciamento de variabilidade e similaridade para organizar os produtos de software comuns em uma família de produtos de software. Esse processo inclui avaliar as semelhanças e diferenças entre produtos de software em termos de requisitos, arquitetura, componentes, testes e artefatos de software. Dessa forma, a linha de produto pode ser modelada de um modo genérico e que sirva para todas as aplicações propostas considerando o domínio que estão inseridas (SOMMERVILLE, 2011).

As LPSs geralmente surgem de aplicações já existentes, chamadas de *ad hoc*. Essas aplicações utilizam ciclos completos de construção, projetadas para atender necessidades específicas de um usuário. Quando esses sistemas foram produzidos, não havia interesse em pensar em funcionalidades de alto nível, que seriam projetadas para atender mais de um mesmo usuário que pertenceria ao mesmo nicho de mercado. No entanto, cada vez que uma nova instância é desenvolvida, a mudança tende a corromper a estrutura da aplicação. Isso acontece porque torna-se difícil utilizar o código desenvolvido para o reuso, uma vez que ele acaba ficando com

mais dependências dentro do código. Assim, pode-se tomar a decisão de criar uma linha de produtos genéricos ao identificar funcionalidades comuns nas instâncias da aplicação para incluir em uma aplicação base para uso futuro (SOMMERVILLE, 2011).

Ao utilizar a estratégia de LPS, é necessário que ocorra mudanças na mentalidade da organização (ERIKSSON; BÖRSTLER; BORG, 2009). A transição de um sistema *ad hoc* para um sistema LPS, exige a revisão de todos os produtos desenvolvidos pela companhia para montar uma arquitetura que apresente uma família de produtos. Para isso, o processo de LPS apresenta dois ciclos de vida (POHL; BÖCKLE; LINDEN, 2005): Engenharia de Domínio e Engenharia de Aplicação, conforme mostrado na Figura 1.

Figura 1 – Ciclos de vida da estratégia de Linhas de Produto de Software: Engenharia de Domínio e Engenharia de Aplicação.



Fonte: Pohl, Böckle e Linden (2005)

A Engenharia de Domínio (ED) é responsável por desenvolver os ativos de reuso, com o objetivo geral de criar componentes em massa. Alguns artefatos muito utilizados na engenharia de domínio e que fazem parte de uma linha de produtos são obtidos por meio da aquisição e documentação de requisitos de cada produto e pela especificação do domínio que esse produto pertence, ou seja, as características do seu nicho de mercado e quais componentes, dentro das

funcionalidades que ele necessita, ele mais utiliza. Existem também outros tipos de artefatos que possibilitam uma maior facilidade na identificação dos produtos feitos para reuso, por exemplo, casos de testes, documentações e contratos para realização do projeto e partes de códigos (ERIKSSON; BÖRSTLER; BORG, 2009).

A Engenharia de Aplicação (EA) é responsável pelo desenvolvimento dos produtos utilizando os ativos desenvolvidos pela Engenharia de Domínio. Durante a engenharia de requisitos de um determinado projeto, os requisitos documentados serão avaliados e comparados com a linha de produtos produzida, resultando em projetos que incluirão requisitos da linha de produtos e os novos requisitos formulados para o produto em questão (ERIKSSON; BÖRSTLER; BORG, 2009).

A etapa da estratégia de Linhas de Produto de Software vinculada a este trabalho é a de Engenharia de Domínio. Neste caso de estudo, será identificado, documentado e minerado todos os ativos pertencentes e disponíveis da empresa para identificar todas as similaridades e variabilidades presentes e assim conseguir formar uma especificação de quais artefatos podem ser utilizados como componentes reusáveis na etapa de Engenharia de Aplicação. Como apresentado na Figura 1, a primeira etapa da engenharia de domínio é a análise de escopo, uma vez que todos os artefatos utilizados necessitam do conhecimento de para quais domínios a empresa vende seus produtos e quais as *features* e *subfeatures* de cada um desses domínios.

2.2.1 *Features* em Linhas de Produto de Software

Ter um conjunto bem definido de requisitos é um dos passos da especificação da Engenharia de Requisitos, sendo esta umas das etapas mais importantes ao se envolver com produção de software, pois nada pode ser mais ameaçador a um projeto de software do que ter requisitos ineficientes e incompletos. Portanto, é um item essencial para que não ocorra mudanças inesperadas que normalmente resultam em custos altos e atrasos no projeto (ERIKSSON; BÖRSTLER; BORG, 2009). Por conta disso, a etapa de Engenharia de Requisitos de uma empresa possui grande influência em como o processo de Linhas de Produto de Software opera, uma vez que ela é dependente do domínio que a empresa está inserida e, portanto, quais produtos ela implementa e já implementou.

Ao se olhar para diversos produtos em uma família de produtos a fim de identificar o potencial de reuso entre elas, os requisitos passam a ser referenciados como *features*, uma vez que cada um dos requisitos aplicados a produtos individuais, passam a ser reconhecidos como *features* quando esse se aplica a mais de um produto pertencente a um mesmo domínio.

Segundo Gulp, Bosch e Svahnberg (2001), *features* são uma forma de abstração de requisitos. Considera-se que *features* e requisitos estão em uma relação de n para n , em que um requisito pode se aplicar a diversas *features*, enquanto que uma *feature* pode se aplicar a mais de um requisito. Para Gulp, Bosch e Svahnberg (2001) as *features* se caracterizam da seguinte

forma:

- **Features externas:** não é diretamente relacionada ao sistema, porém o sistema usa e é dependente dessa funcionalidade. Pode ser tanto uma API necessária para obter dados que venham de uma determinada fonte, quanto um serviço implementado por uma entidade não relacionada a organização;
- **Features obrigatórias:** são capazes de identificar um produto. Para que um produto seja considerado como pertencente ao domínio automobilístico, por exemplo, ele precisa conter a *feature* de sistema de transmissão;
- **Features opcionais:** podem ou não serem habilitadas, mas podem adicionar componentes interessantes ao se relacionar com as outras *features* do sistema dependendo do contexto da aplicação;
- **Features variáveis:** é uma abstração de um conjunto de *features* obrigatórias ou opcionais, uma vez que essa *feature* tem a capacidade de ser opcional ou obrigatória dependendo do domínio e do produto que se relaciona.

Em LPS, as *features* aparecem como sendo a primeira classe de abstrações que moldam o raciocínio dos *stakeholders*, usadas para representar as variabilidades e similaridades entre as partes de uma linha de produtos. No artigo *Feature-oriented domain analysis*, [Kang et al. \(1990\)](#) define uma *feature* como “um aspecto, qualidade ou característica visível ao usuário de um sistema ou sistemas de software” e como “funcionalidades distintas e abstratas que devem ser implementadas, testadas, entregues e mantidas”. Por outro lado, [Chen et al. \(2005\)](#) define as *features* como “características do produto a partir da visão do usuário ou cliente, que consiste essencialmente em um conjunto coeso de requisitos individuais”.

Um sistema LPS pode conter centenas de *features*, cada uma com pontos de variação, ou seja, a variabilidade irá organizá-las em dois tipos: as que dependem umas das outras, e as que se distinguem entre si. Isso representa o ponto de variação no domínio do sistema de software, mostrando as *features* que estão presentes ou não nas aplicações ([SILVA, 2013](#)).

2.2.2 Escopo de Linhas de Produto de Software

O processo de análise do escopo de uma empresa considerada *ad hoc*, geralmente envolve conhecimento de um domínio que naturalmente foi concebido por experiências acumuladas durante o tempo que diversos sistemas foram implementados. Na construção da infraestrutura de uma linha de produtos, a análise de variabilidade e similaridade é feita para categorizar as *features* e sua distribuição entre esses produtos. Dessa forma, abstrações genéricas podem ser isoladas e reusadas ([SILVA, 2013](#)).

O objetivo principal de uma análise de escopo é capturar o contexto do domínio e assim identificar os recursos mais importantes e as restrições de projeto que irão pertencer a linha de produtos, ou seja, identificar os produtos e as suas *features* principais, os subdomínios e priorizar certos ativos para que se tenha critérios de avaliação ao se chegar ao produto final, colocando o foco em funcionalidades reusáveis que prometem um retorno de investimento ótimo. Cada uma dessas tarefas é executada conforme o contexto que o sistema está inserido (POHL; BÖCKLE; LINDEN, 2005). Existem três tipos de escopos de Linhas de Produto de Software que são uma forma de planejamento para o reuso (JOHN et al., 2006):

- **Escopo de domínio:** determina e delimita os domínios que tem potencial de reuso para as linhas de produto. Ao separar os domínios de uma empresa, a próxima etapa é identificar os subdomínios, suas *features* e *subfeatures*;
- **Escopo de produtos:** determina os produtos e suas *features* principais que estarão nas linhas de produto. Para cada um dos produtos dentro do domínio especificado, são relacionadas as *features* pertencentes a ele. Dessa forma, é possível visualizar quais *features* possuem uma maior importância dentro do domínio;
- **Escopo de ativos:** tem o objetivo de determinar os ativos que serão importantes para as linhas de produto. Essa etapa ocorre ao decorrer da mineração de artefatos durante o processo de detalhamento das *features* e *subfeatures* do domínio (Figura 7), uma vez que é possível identificar quais serão os ativos necessários para reformular as aplicações para serem utilizadas para reuso e com reuso durante a Engenharia de Aplicação.

As etapas de escopo de domínio e escopo de produtos estão presentes no *template* do Mapa de Produtos (Quadro 2) pertencente na abordagem RISE-DA. O escopo de ativos estará pouco presente durante o processo de Engenharia de Domínio deste trabalho. A investigação de código legado poderá levar a uma etapa de especificação de código, onde será descrito para cada trecho de código, onde está presente o potencial de reuso pela sua similaridade com outras aplicações e quais são específicas do contexto da organização, representando a variabilidade.

2.2.3 Variabilidade e Similaridade em Linhas de Produto de Software

Variabilidade é a capacidade de customização de um sistema de software (GURP; BOSCH; SVAHNBERG, 2001). Introduzir variabilidades em um sistema faz com que fique mais fácil fazer mudanças, porém é impossível antecipar quais variabilidades serão catalogadas conforme o tempo (GURP; BOSCH; SVAHNBERG, 2001). Segundo Pohl, Böckle e Linden (2005), variabilidade pode ocorrer conforme o tempo e o espaço em que os artefatos de software estão inseridos. Isso significa que podem existir diferentes versões válidas de um mesmo artefato em tempos diferentes e que os artefatos podem ter diferentes formatos no mesmo espaço de tempo. Porém, essas mudanças de versões podem fazer com que um produto não faça mais parte

de uma família de produtos criada e isso gere a necessidade de ocorrer uma nova análise de domínio.

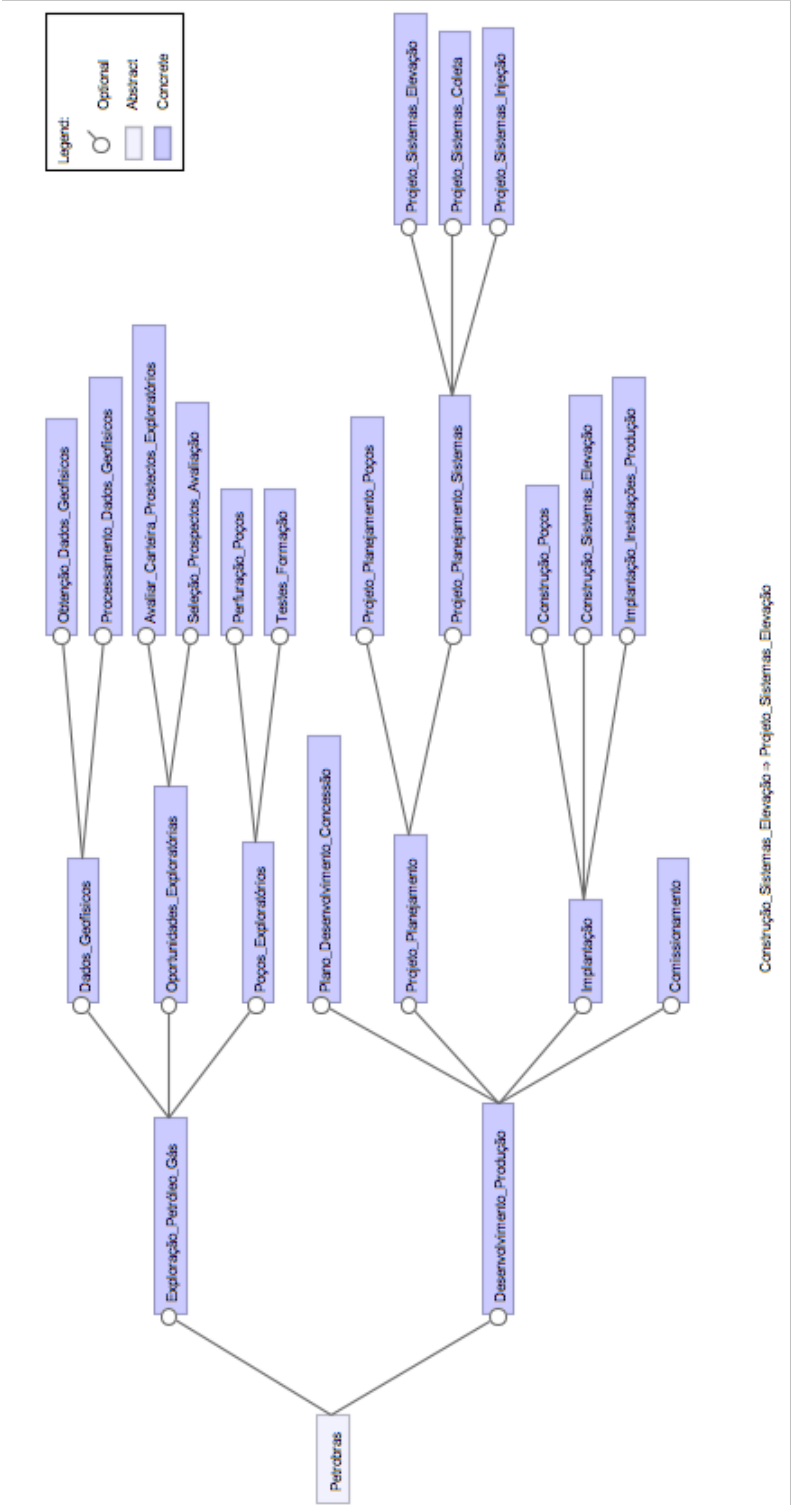
Variabilidades podem ser internas ou externas do ponto de vista do usuário (SILVA, 2013), ou seja, variabilidades externas fazem parte da documentação, sistemas em execução e outros artefatos com alta abstração e as variabilidades internas representam os sistemas legados como linhas de código e decisões técnicas, representando baixa abstração. As variabilidades externas estão presentes na primeira etapa do processo de análise de domínio da RISE-DA (Figura 8) e as variabilidades internas na etapa dois (Figura 12).

Em contrapartida, a similaridade é o que procuramos na fase de Engenharia de Domínio, sendo encontrada no contexto de diferentes produtos pertencerem ao mesmo domínio e suas *features* apresentarem as mesmas funcionalidades, fazendo com que isso passe a se tornar uma similaridade e, portanto, um potencial de reuso. Essas similaridades podem então ser estruturadas, especificadas, documentadas e catalogadas em cada um dos artefatos existentes na empresa para que na Engenharia de Aplicação.

A Figura 2 exemplifica a relação lógica entre as *features* dentro de um domínio de gestão de exploração de petróleo e gás em uma organização petrolífera. É possível configurar por meio da árvore de *features*, *features* abstratas (representada em lilás) e *features* concretas (representada em roxo). As *features* abstratas podem ser derivadas para *features* concretas que vão representar uma funcionalidade específica de uma aplicação. Essas *features* podem ser opcionais (conexões brancas) ou obrigatórias (conexões pretas), significando que, para diversas aplicações, essa funcionalidade é obrigatória para todas ou é opcional, uma vez que ao menos uma aplicação não possui essa *feature*. Podem haver também relações entre *features* e *subfeatures* que possuem a mesma *feature* raiz. Essas relações correspondem a operação lógica OR ou XOR, ou seja, para o OR, é necessário ter pelo menos 1 das *features* ou as duas, e para o XOR, é obrigatório ter pelo menos 1, mas as duas não podem funcionar ao mesmo tempo.

Com a árvore de *features* é possível visualizar mais facilmente as dependências entre as *features* e *subfeatures* e dessa forma conseguir analisar as similaridades e variabilidades entre os produtos do domínio mais facilmente, uma vez que essas dependências ajudam a construir uma família de produtos. Um exemplar desta família seria representada por uma ramificação da árvore.

Figura 2 – Template de uma árvore de *features* ou modelo de *features* presente na abordagem RISE-DA (SILVA, 2013).



3

Trabalhos Relacionados

Neste capítulo, será apresentado a abordagem SPLICE e APLE que possuem o mesmo objetivo que o RISE-DA, mostrando suas especificidades e objetivos. Além disso, será apresentado o PuLSE que é uma etapa de análise de escopo, assim como acontece no RISE-DA, mas com uma estratégia voltada para grandes organizações. É apresentado também, abordagens que tentaram utilizar a metodologia ágil com as linhas de produtos de softwares em contextos de grandes empresas. Existem outras abordagens que também podem ser aplicadas nas empresas que não estão atreladas a uma análise de escopo como o RISE-DA, mas também se preocupam com o aumento da produtividade ao otimizar o tempo de desenvolvimento, como mostrado pelas “Outras Propostas” apresentadas nesse capítulo.

3.1 SPLICE

No artigo de [Vale et al. \(2014\)](#), é proposto a abordagem SPLICE (*Software Product Line Integrated Construction Environment*). Essa estratégia tem por objetivo promover a combinação das Linhas de Produto de Software com uma estratégia ágil e adota uma abordagem de classificação dos requisitos imediatos de um sistema ou sistemas de software pela maior prioridade para poder construir componentes semelhantes de software. Esse artigo também aborda a aplicação do SPLICE em cenários de pequenos times de desenvolvimento de software e adotam essas práticas em aplicações mobile que promovem assistências emergenciais aos usuários para validar a proposta. Seu objetivo é apresentar uma estratégia mais leve de desenvolvimento combinando essas duas abordagens para propor menos custos e um retorno de investimento rápido.

Essa proposta propõe um detalhamento dos processos de software por meio de uma linha de produtos em mercados dinâmicos ou empresas mais estabilizadas. Segundo [Vale et al. \(2014\)](#), uma consequência da abordagem faz com que os *stakeholders* possam implementar essa estratégia em seu ambiente de trabalho, ou seja, a proposta é especificamente para guiar os

stakeholders na adoção da estratégia, sem necessitar de um [Especialista em análise do domínio](#). Por conta disso, junto com a parte iterativa do processo, existem os artefatos de implementação e testes das *subfeatures* encontradas. Consequentemente, também existem as reuniões iniciais, de resolução de problemas e de finalização para definir se existe alguma nova funcionalidade a ser implementada ou ajustes a serem feitos, além de verificar se são necessários mais testes em diferentes níveis ou se a *Sprint* está finalizada.

A abordagem descrita neste trabalho de conclusão de curso, por outro lado, propõe a aplicação apenas da etapa de análise de domínio (escopo da LPS) em uma organização através da análise de similaridades e variabilidades das *features* dos produtos de um domínio. O objetivo é obter conhecimento de um domínio já existente para construir uma documentação e dessa forma obter os ativos que podem estar em uma linha de produtos para implementá-los em etapas posteriores, tais como a etapa de RISE-CBD (*Reuso in Software Engineering - Component-Based Development*). Essa etapa propõe as unidades e os produtos de trabalho e os papéis necessários para implementar os artefatos reusáveis ([SILVA, 2013](#)). Dessa forma, os papéis de desenvolvedor e pessoal de teste apenas trabalham de forma ativa na segunda parte do processo com uma documentação já bem estruturada anteriormente.

3.2 APLE

Segundo [Kiani et al. \(2021\)](#), a abordagem APLE (*Agile Product Line Engineering*) integra a abordagem ágil com as Linhas de Produtos de Software para reduzir as tarefas de design e aumentar a reusabilidade e customização enquanto o desenvolvimento de *software* passa a ser mais flexível e adaptável às mudanças. Neste artigo é proposto uma nova maneira de incrementar a abordagem APLE, de forma que exista um gerenciamento mais dinâmico das variabilidades encontradas durante o reuso de *features*. A nova proposta chama-se PA (*Proposed Approach*) e garante que as regras das LPSs, das metodologias ágeis e características positivas da APLE sejam mantidas como fases separadas da abordagem PA, por exemplo, etapas e papéis do Scrum, atividades, artefatos e reuniões. O objetivo do estudo é fazer com que a fase de Engenharia de Domínio e a fase de Engenharia de Aplicação sejam aplicadas de forma iterativa e incremental durante todo o processo aplicado, além de envolver linhas de produto com características ágeis para formar uma estratégia reusável que faça sentido para a organização em questão.

A integração de metodologias ágeis nas duas fases da Linhas de Produto de Software (ED e EA) é mais desafiadora e é a proposta tanto da abordagem SPLICE, quanto da abordagem APLE. Entretanto, devido ao contexto da empresa (estudo de caso deste trabalho de conclusão de curso) em que a primeira etapa da abordagem RISE-DA está sendo aplicada, ficaria inviável promover as duas etapas na etapa iterativa do processo, pois apesar da sincronização das duas etapas serem importantes para manter os documentos criados relevantes e de manter obstáculos de desenvolvimento mais claros durante as etapas de reuso, é importante considerar que é necessário

uma rápida construção e evolução dos produtos criados por meio da linha de produtos. Isso envolve um refatoramento interno ocorrendo rapidamente, por meio da alocação de times que provém estabilidade e conhecimento relevantes para implementar as atividades de escopo e também os ativos necessários na etapa de Engenharia de Aplicação.

Portanto, o contexto abordado neste trabalho de conclusão de curso não se integra bem a essas duas abordagens por se tratar de funcionários que são estudantes não remunerados e de uma cultura organizacional que possui uma relação restrita com a instituição de ensino a que pertence, possuindo limitações de tempo, conhecimento e estrutura. Todas as limitações do processo estarão citados mais detalhadamente no capítulo 7.

3.3 PuLSE

A abordagem de escopo PuLSE (*Product Line System and Software Engineering*) permite a concepção de Linhas de Produto de Software em diversos contextos de grandes empresas. Isso acontece, pois o processo de escopo do PuLSE é genérico, sendo base para diversas aplicações (JOHN et al., 2006). A Figura 3 demonstra as fases do processo, que consistem em:

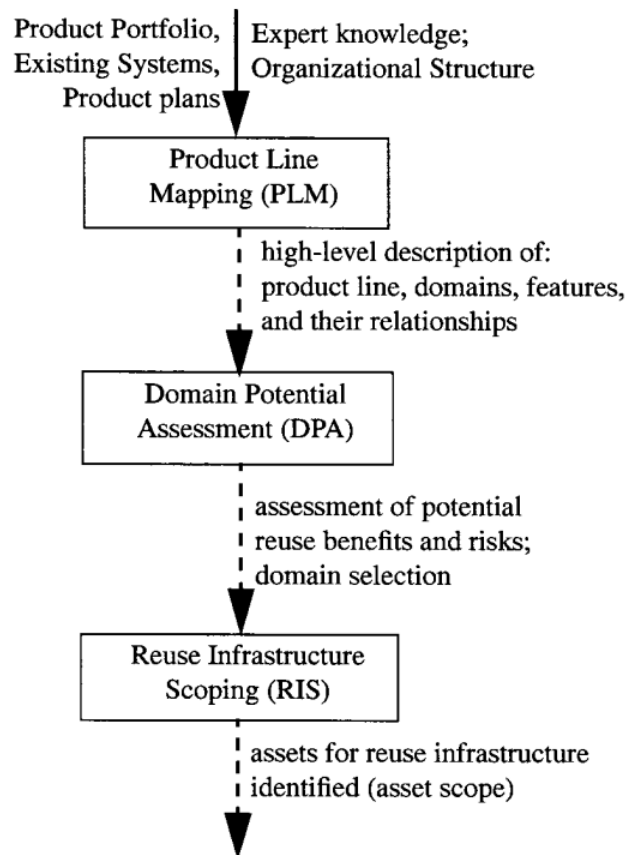
- **Mapeamento das linhas de produtos:** o objetivo dessa fase é ter uma visão geral da linha de produto e observar a distribuição das *features* nos produtos;
- **Avaliação do potencial de domínio:** o objetivo dessa fase é avaliar os domínios com potencial de reuso;
- **Infraestrutura de escopo reusável:** o objetivo dessa fase é identificar ativos, como componentes e documentações que tem um potencial de reuso.

O mapeamento das linhas de produtos é feito pela identificação dos produtos apropriados para a linha de produtos, onde será descrito seu potencial de mercado, suas *features* principais e outras informações que são relevantes para a linha de produto. As *features* e *subfeatures* são utilizadas nessa etapa para descrever os produtos, ou seja, quaisquer funcionalidades do produto. Essa fase é importante para identificar domínios internos no sistema como base para uma análise qualitativa ao cobrir todo o escopo (JOHN et al., 2006).

A avaliação do potencial do domínio consegue estruturar uma estratégia de reuso para os diferentes domínios da empresa por meio das linhas de produtos. Para tal prática, o processo precisa de um determinado período para catalogar e avaliar o potencial de reuso das *features* no domínio da aplicação, havendo etapas de planejamento, preparação dos dados e identificação dos recursos necessários. Dependendo de quantos subdomínios forem enumerados, pode haver uma variação no tempo de análise (JOHN et al., 2006).

A etapa de infraestrutura de linhas de produto reusáveis consegue catalogar quais ativos são feitos para reuso dos componentes e quais são específicos das aplicações. Para a arquitetura

Figura 3 – Componentes do PuLSE: Mapeamento das linhas de produtos, Avaliação do potencial de domínio e Infraestrutura de linhas de produto reusáveis.



Fonte: Schmid (2002)

trabalhada, o foco desta etapa são as implementações para reuso, estabelecendo requisitos reusáveis que são relevantes para o desenvolvimento (JOHN et al., 2006).

Como pode ser visualizado na Figura 3, a abordagem utilizada leva em consideração uma ordem de execução de cada umas das etapas necessárias para construção das linhas de produtos da empresa estudada. O PuLSE, portanto, utiliza fases sequenciais, longo planejamento, projetos com custos, escopo e cronograma fixos, o que não enquadra bem ao contexto de uma *very small entity*, uma vez que seus artefatos são instáveis e precisariam ser revistos com periodicidade. A abordagem RISE-DA ao contrário, possui a estratégia do Scrum atrelada ao seu funcionamento, que garante mais flexibilidade no mapeamento, avaliação e detalhamento de cada uma das suas linhas de produtos.

3.4 Metodologia Ágil utilizando Linhas de Produto de Software

Uma das pesquisas que tentam juntar as técnicas de LPS com a metodologia ágil foi no artigo de Klünder, Hohl e Schneider (2018). Essa proposta é aplicada no desenvolvimento de

linhas de produto para um sistema de software automotivo, diminuindo os custos e melhorando o tempo de entrega dos produtos. Em uma empresa de grande porte podem existir muitas variabilidades no produto, causando uma maior complexidade da evolução rápida em sistemas automotivos e por isso, não sendo tão adaptáveis aos modelos ágeis. Por conta disso, tentou-se aplicar a metodologia ágil mantendo os benefícios da LPS, mas especificando um modelo adaptado para o contexto da organização e que não será necessariamente funcional para todas as organizações. Segundo ele, é necessário adotar a análise de escopo para que seja possível identificar as melhores práticas para cada organização.

Outro estudo feito por [Hayashi, Aoyama e Kobata \(2017\)](#), elabora um caso de estudo para descobrir o porquê a empresa estudada conseguiu integrar o processo LPS com uma metodologia ágil. A empresa selecionada para o caso de estudo corresponde a uma grande empresa, onde sua sede fica na Finlândia, mas produz diversos serviços para mais de 90 países. Para este trabalho de conclusão de curso, assume-se que os métodos ágeis requerem que as *features* sejam pequenas para que possam ser analisadas em *Sprints* curtas, normalmente uma a duas semanas. Por conta disso, a metodologia utilizada foi aplicada dentro da própria sede da organização, com atividades mais localizadas.

Na tese de [Snorre \(2009\)](#), a questão da pesquisa é entender como um método de desenvolvimento ágil consegue ser combinado dentro das práticas de LPS em duas grandes companhias de software. Segundo [Snorre \(2009\)](#), os métodos ágeis não possuem a característica de adquirir o conhecimento do domínio da empresa como acontece nos processos de escopo de LPS, pois são priorizados os códigos ao invés da documentação, além disso os métodos de desenvolvimento ágil geralmente constroem os requisitos enquanto desenvolvem, enquanto os processos de escopo de LPS colocam esforço em priorizar quais *features* são válidas e corretas ao se pensar em reuso no domínio da aplicação ao longo de uma sequência de fases de análise e, portanto, essa combinação não é bem-sucedida para grandes empresas

Entretanto, segundo [Snorre \(2009\)](#) ainda é possível utilizar as duas técnicas para formar os documentos necessários e conseguir compreender a direção e o futuro dos produtos que estão sendo implementados na linha de produtos. Isso acontece, porque um método ágil é bastante útil integrado com os processos de escopo de LPS quando é analisado os requisitos de cada produto baseado no que o consumidor precisa, gerando muitos benefícios para mercados dinâmicos e que mudam rapidamente. [Snorre \(2009\)](#) também recomenda que a combinação dessas duas técnicas pode acontecer se a falta de documentação do processo ágil for suprida com o *feedback* dos usuários e com a participação do [Especialista do Domínio](#) em ciclos iterativos para ajudar a priorizar os recursos que precisam ser colocados no desenvolvimento e assim conseguir construir uma linha de produtos. Isso faz com que a LPS seja mais voltada as necessidades imediatas do domínio da aplicação e que as linhas de produtos sejam mais capazes de sofrer mudanças, o que acontece na abordagem RISE-DA.

A maioria dos trabalhos relacionados pesquisam formas de trabalhar em grandes empresas

que estão dentro da área de aplicação da LPS, porém tentam adotar metodologias ágeis com o intuito de aumentar a produtividade das organizações para diminuir o tempo de entrega de produtos no mercado que estão inseridas. Entretanto, na tese de [Silva \(2013\)](#), é mostrado a teoria da junção da metodologia ágil com o processo de LPS em contexto voláteis que possuem variabilidade e similaridade entre os produtos do domínio para reuso. Nesse contexto é necessário também que essas empresas se adaptem a tarefas pequenas que possam ser executadas em pouco tempo. Portanto, essa metodologia está relacionada também ao contexto de micro e pequenas empresas que possuem um portfólio de produtos, tal qual é a tentativa deste estudo em aplicar em uma empresa júnior de tecnologia.

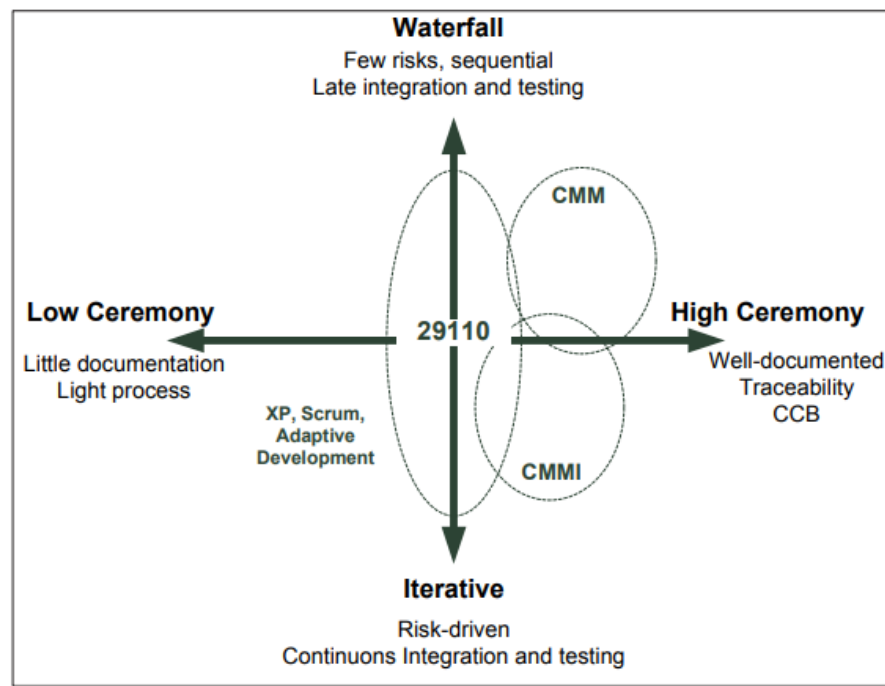
3.5 Outras propostas

Uma das propostas de práticas de Engenharia de Software feitas à comunidade de *very small entities* (VSE) é a ISO/IEC 29110 “Software Engineering – Lifecycle profiles for Very Small Entities (VSEs)” que promove dois tipos de processos: gerenciamento de projeto e implementação de software ([SÁNCHEZ-GORDÓN; O’CONNOR, 2015](#)). A definição de *Very Small Entity* (VSE) foi definida pela ISO/IEC JTC1/SC7 e depois adotada pelo processo ISO/IEC 29110. O objetivo geral desse novo processo é auxiliar as pequenas empresas a aumentar a performance do processo de desenvolvimento de software ([SÁNCHEZ-GORDÓN; O’CONNOR, 2015](#)). É possível representar um domínio de abordagens que podem ser aplicadas ao desenvolvimento de software em comparação com o processo ISO/IEC 29110. Na Figura 4 é possível observar que no eixo x é representado o nível de formalidade presente nas abordagens, indo desde pouca documentação presente em uma metodologia ágil por exemplo, até uma metodologia que precisa de uma documentação mais rigorosa, como o processo CMMI (*Capability Maturity Model Integration*). No eixo y observa-se o nível do risco de cada abordagem, que variam entre as estratégias *Waterfall* e estratégias mais iterativas, como *Risk-driven*. O processo ISO/IEC 29110 está localizado no centro dos dois eixos ([LAPORTE; O’CONNOR; PAUCAR, 2015](#)).

Existem outras técnicas utilizadas em *very small entities* além da ISO/IEC 29110, como por exemplo o processo ISO/IEC/IEE 12207 e ISO/IEC/IEEE 15289 ([LAPORTE; O’CONNOR; PAUCAR, 2015](#)). Além disso, há o processo Crystal Clear ([NUNEZ, 2014](#)) e ESA PSS-05-0 ([EUROPEAN SPACE AGENCY, 1996](#)). Alguns outros métodos adaptados ao contexto das *very small entities* foram desenvolvidas em conformidade com o ISO/IEC 15504: RAPID (*Rapid Assessment for Process Improvement for Software Development*), SPINI (*Software Process Improvement Initiation*), MARES (Método de Avaliação de Processo de Software), e SPIRE (*Software Process Improvement in Regions of Europe*) ([SÁNCHEZ-GORDÓN; O’CONNOR, 2015](#)).

Entretanto, apesar das técnicas utilizadas pelas *very small entities* nos trabalhos analisados, nenhuma delas fornece práticas associadas as LPSs que possuem um processo de análise de

Figura 4 – Abordagens aplicadas ao desenvolvimento de software, baseando-se no processo ISO/IEC 29110.



Fonte: Laporte, O'Connor e Paucar (2015)

domínio. O processo de análise de escopo realizado em uma empresa é essencial, pois é um trabalho contínuo de análise e monitoramento de desenvolvimento das aplicações, tecnologias e requisitos do usuário nos nichos de mercado relevantes para a empresa. Esse processo possui o propósito de tirar as dúvidas sobre quais produtos serão parte da linha de produtos de software e se eles são capazes de serem utilizados para o reuso, que é uma fonte de recursos economicamente viável às empresas (JOHN; EISENBARTH, 2009).

4

Cenário de Pesquisa

Neste capítulo, é apresentado o contexto da empresa onde o caso de uso dessa pesquisa será aplicado. A empresa BIT é uma empresa júnior de Ciência da Computação da UNIOESTE, com o objetivo de proporcionar aos estudantes uma aproximação da academia com o mercado de trabalho e aumentar as habilidades sociais e técnicas dos membros efetivados. Esse contexto possui algumas especificações que serão apresentadas a seguir que irão impactar no andamento da pesquisa, como falta de capacitação, alta rotatividade de membros e restrições organizacionais.

4.1 Micro e pequenas empresas

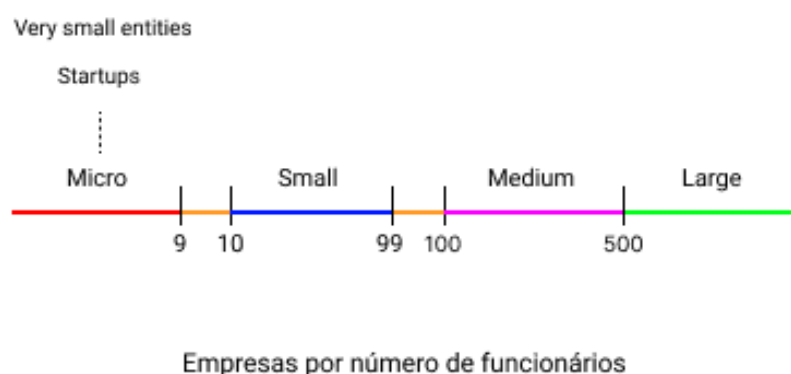
A engenharia de software fornece ferramentas, processos, técnicas e métodos para auxiliar as empresas de software no gerenciamento e no desenvolvimento para que mantenham os produtos com foco nos interesses do mercado e gerem qualidade, custo e prazos controlados. Empresas conhecidas como *micro companies* ou *small companies* apresentam dificuldades em aplicar essas práticas, pois funcionam em cenários distintos, apresentando número limitado de funcionários, uma infraestrutura limitada e uma flexibilidade que pode não estar em concordância com as práticas de engenharia de software, uma vez que essas práticas podem não ser adequadas as suas características (MÁRCIO; WESLEY; IVONEI, 2020).

A associação brasileira das empresas de software (NETO, 2021) define que para uma empresa ser definida como *large company* é necessário conter mais de 500 funcionários, já uma empresa definida como *medium company* necessita entre 100 a 500 funcionários. Uma *small company* teria uma estimativa de 10 a 99 funcionários, enquanto uma *micro company* é necessário ter menos de 10 funcionários e menos de 2 milhões de *turnover* anual para ser considerada uma microempresa, representando 77% das empresas no Brasil (Figura 5).

As empresas juniores podem se enquadrar na categoria de *micro company* como *Very Small Entity* (Figura 5), pois a rotatividade de alunos é uma constante, não gerando um crescimento

e acúmulo de funcionários muito grande com o decorrer do tempo. As *very small entities* são organizações com objetivos de negócios voláteis, constantemente experimentando novas formas de otimizações e nichos de mercado, mesmo tendo muitas vezes grandes crescimentos e retornos, trabalham com recursos limitados, incluindo financiamento, recurso intelectual e de pessoal (SANCHEZ-GORDON; SÁNCHEZ-GORDÓN; LUJÁN-MORA, 2016).

Figura 5 – Representação dos segmentos das empresas de software com base no número de funcionários. Mais à esquerda estão as empresas com o menor número de funcionários e à direita com o maior número.



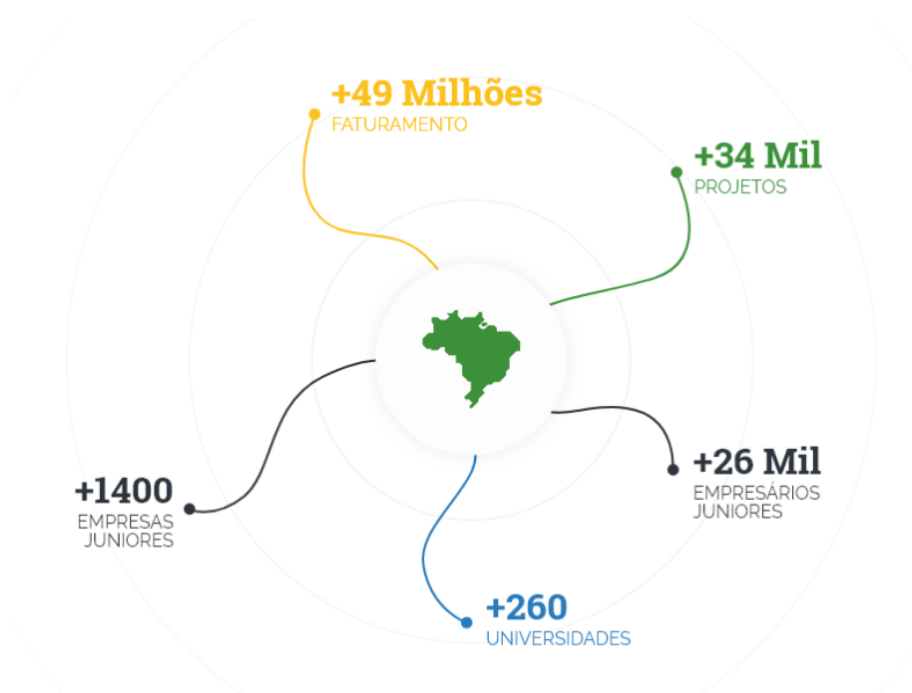
4.2 Empresas Júniores

O Movimento Empresa Júnior (MEJ) é um movimento de empreendedorismo universitário que foi elaborado e aprovado em 2010 e posteriormente reformulado em 2015 pela legislação brasileira (JÚNIOR, 2021a). A publicação da lei nº 13.267/16 regulamenta as empresas juniores, reconhecendo o nível legal dessas organizações como instituições que promovem a educação empreendedora no ensino superior brasileiro (JÚNIOR, 2021a). Segundo o Art. 2º da lei nº 13.267/16 “Considera-se empresa júnior a entidade organizada nos termos desta Lei, sob a forma de associação civil gerida por estudantes matriculados em cursos de graduação de instituições de ensino superior, com o propósito de realizar projetos e serviços que contribuam para o desenvolvimento acadêmico e profissional dos associados, capacitando-os para o mercado de trabalho” (ROUSSEFF, 2016).

Os funcionários das empresas juniores são estudantes regularmente matriculados na instituição de ensino superior e no curso de graduação que a entidade esteja vinculada, exercendo obrigatoriamente trabalho voluntário na empresa, sendo a renda obtida revertida exclusivamente para atividades que venham contribuir e promover o melhor desenvolvimento dos profissionais no mercado de trabalho (JÚNIOR, 2021a).

Em 2021, foram catalogadas mais de 1400 empresas juniores no mercado brasileiro (figura 6), segundo a Júnior (2022), gerando um impacto real sobretudo para as micro e pequenas empresas, que fazem parte de 17,6% e 31,1% dos principais clientes das empresas juniores (JÚNIOR, 2021b), sendo compostos também por grandes empresas, órgãos públicos, médias empresas, empreendedor individual e pessoas físicas (JÚNIOR, 2018). Desde 2010, o MEJ já impactou mais de R\$ 70.000.000,00 na economia brasileira, que são integralmente reinvestidos na educação empreendedora dos estudantes, realizando mais de 34 mil projetos e formando mais de 26 mil jovens que passaram pelo MEJ (JÚNIOR, 2021b).

Figura 6 – Resultados do censo de 2018 do Movimento Empresa Júnior.



Fonte: Júnior (2022)

4.3 BIT - Empresa Júnior de Ciência da Computação

O cenário escolhido para este trabalho foi uma Empresa Júnior de Ciência da Computação (BIT). Segundo o Estatuto Social da Empresa Júnior de Ciência da computação (SCHAEFER; DALSSASSO; MATOS, 2016), artigo 1º: "A Empresa Júnior de Ciência da Computação, também denominada BIT, fica constituída uma associação civil, sem fins econômicos, formada e gerida por alunos do curso de graduação de Ciência da Computação, da Universidade Estadual do Oeste do Paraná - Campus Cascavel (UNIOESTE - Cascavel) e orientada por um professor do Curso de Ciência da Computação, que se regerá pelo presente Estatuto Social, pelas disposições legais aplicáveis, e por atos que forem baixados pelos órgãos competentes". Todos os membros efetivos, estudantes do curso de Ciência da Computação, precisam ser aprovados no processo seletivo da BIT, estabelecido pelo regulamento interno.

Segundo o artigo 5º (SCHAEFER; DALSSASSO; MATOS, 2016), a BIT terá como finalidade:

- Proporcionar aos seus membros as condições de aplicação prática dos seus conhecimentos relativos à área de formação profissional;
- Promover palestras, cursos, seminários e eventos, por meio de seus próprios recursos, ou em parceria(s) com organizações idôneas, objetivando desenvolver e aprimorar conhecimentos da área de Ciência da Computação, Tecnologia da Informação e Empreendedorismo;
- Valorizar e apoiar o Movimento Empresa Júnior, seguindo o seu Código de Ética, participando de seus órgãos reguladores e respeitando as suas deliberações;
- Desenvolver em seus membros habilidades profissionais no âmbito da vivência gerencial, trabalho em equipe, relações interpessoais e análise lógico-analítica de problemas cotidianos empresariais;
- Realizar estudos e projetos sobre assuntos referentes à área de graduação de seus associados, visando o seu desenvolvimento profissional;
- Desenvolver atividades em caráter prático e treinamento, visando a futura inserção no mercado de trabalho, sempre com subordinação profissional competente;
- Fomentar o espírito empreendedor de seus membros;
- Valorizar alunos e professores da UNIOESTE no mercado de trabalho e no âmbito acadêmico, bem como a referida instituição;
- Desenvolver ações que contribuam para a realização das finalidades, princípios e objetivos da UNIOESTE, sempre em observância da indissociabilidade entre ensino, pesquisa e extensão.

Todo o patrimônio social da BIT será constituído das contribuições dos próprios membros devido aos trabalhos realizados em nome da empresa, além de subvenções, legados, recursos recebidos, treinamentos, cursos e seminários (SCHAEFER; DALSSASSO; MATOS, 2016). Os membros da BIT são supervisionados por um orientador do curso que zela pela qualidade, imagem e pelo bom nome da empresa, além de cumprir com o regimento interno do MEJ. A Diretoria Executiva da BIT, eleitos dentre os membros efetivos, é composta por 7 membros, sendo eles: 1 Presidente, 1 Tesoureiro, 1 Secretário, 1 Diretor de Marketing, 1 Diretor de Vendas e 2 Diretores de Projetos (SCHAEFER; DALSSASSO; MATOS, 2016).

Pela rotatividade grande de funcionários dentro da empresa e pelos requisitos serem muito instáveis, acaba existindo um conhecimento muito imaturo sobre o domínio por parte dos estudantes, sem existir a preocupação de um aprendizado vindo dos trabalhos anteriores, apenas

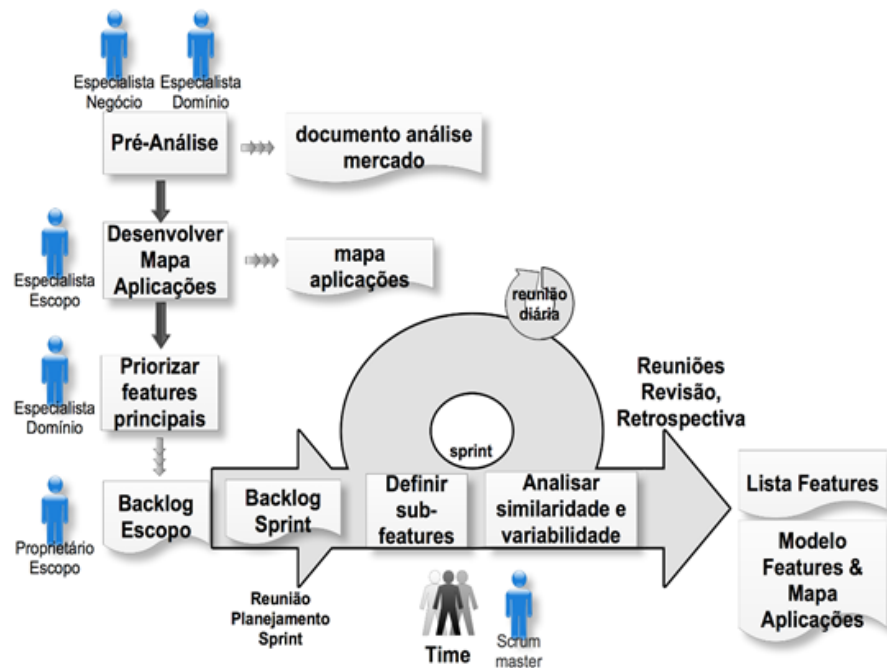
no desenvolvimento que será executado ao se adquirir um projeto. Como os membros da BIT não são remunerados e estão exclusivamente dependentes de todo o regimento proposto pela universidade, isso faz com que ela seja uma empresa com poucos recursos.

5

RISE-DA

A abordagem escolhida para este estudo de caso foi o *Reuso in Software Engineering - Domain Analysis* (RISE-DA) (SILVA, 2013). A análise de variabilidade e similaridade entre as aplicações do domínio precisa de unidades e produtos de trabalho (atividades, tarefas e passos). O processo utilizado pelo RISE-DA está descrito na Figura 7. O processo mostrado na Figura 7 será separado em duas partes.

Figura 7 – Processo do RISE-DA que detalha as atividades realizadas de maneira *top-down*.



Fonte: Silva (2013)

A primeira parte concentra a criação e especificação de cada um dos artefatos utilizados: documento de análise de mercado, mapa de produtos e priorização de *features* principais por meio

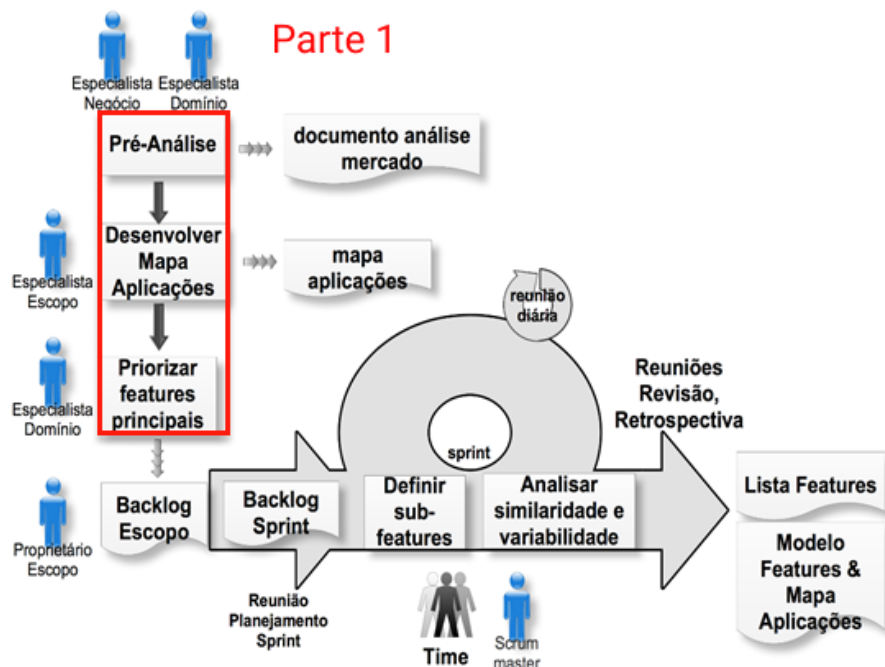
dos especialistas necessários para essa etapa. As responsabilidades dos especialistas utilizados na abordagem estão descritos no glossário.

A segunda parte utilizará os artefatos criados anteriormente e por meio da priorização das *features* principais, haverá a criação de um Backlog do Domínio. Por meio de reuniões periódicas, será definido o Backlog do *Sprint* para que cada *feature* seja detalhada em *subfeatures* e dessa forma obter uma visualização melhor de cada uma das similaridades e variabilidades do domínio que ela se encontra.

5.1 Parte 1

A primeira parte da abordagem RISE-DA (Figura 8) é responsável por realizar o processo que está inserido na etapa inicial da Engenharia de Domínio, com o objetivo de extrair uma análise de domínio da empresa, documentando uma análise de mercado, um mapa de produtos e uma priorização das *features* principais entre as aplicações da empresa. A saída esperada por essa etapa serão artefatos de alto nível que possuem as similaridades e as variabilidades entre as aplicações de um mesmo domínio (Figura 8).

Figura 8 – Parte 1 do processo de análise de domínio da RISE-DA, buscando as similaridades e variabilidades entre as *features* principais.



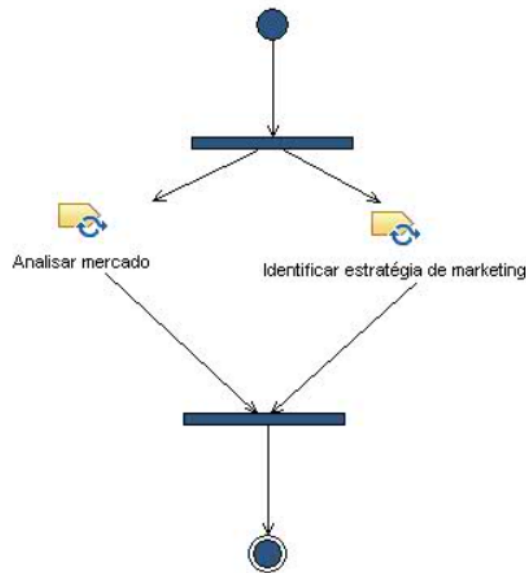
Fonte: Silva (2013)

O primeiro artefato a ser documentado na parte 1 do processo é a análise de mercado (Quadro 1 e Quadro 9). Esse documento obtém as informações sobre o mercado e a estratégia de marketing do contexto analisado, avaliando seus impactos no desenvolvimento dos artefatos

reusáveis. Os papéis desempenhados durante o processo de análise de mercado e estratégia de marketing serão definidos aos cargos de [Especialista do negócio](#) e o [Especialista em análise do domínio](#). Abaixo estão citados alguns dos principais itens analisados.

- **Estratégia de marketing:** analisa como a empresa está presente no mercado, considerando as características e especificações da prestação de serviços;
- **Necessidade do mercado:** analisa quais são as necessidades do mercado que a empresa supre;
- **Concorrência:** cataloga os concorrentes que podem ou não impactar no negócio da empresa e quais são as *features* dessas empresas que são interessantes para aplicar no domínio analisado;
- **Características dos usuários:** analisa qual é o ambiente computacional, o perfil e os níveis de habilidades dos usuários dos produtos desenvolvidos;
- **Tecnologias de desenvolvimento:** cataloga as tecnologias atuais usadas no processo de desenvolvimento;
- **Restrições culturais/legais:** analisa as restrições que a empresa precisa seguir e os regimentos que estão interligados a ela, assim como as barreiras culturais tanto dos funcionários quanto dos usuários dos sistemas;
- **Tempo de entrega:** analisa como funciona a periodicidade das entregas e como isso afeta na produção do sistema;
- **Objetivos de negócio:** analisa as forças, fraquezas, oportunidades e ameaças relacionadas no planejamento estratégico da empresa e como ela escolhe planejar seus projetos e estruturar sua cultura organizacional;
- **Objetivos de reuso:** analisa os principais objetivos da empresa para a realização de um desenvolvimento voltado para o reuso, na etapa de engenharia de aplicação.

Figura 9 – A análise de mercado é um documento fundamental para os *stakeholders* compreenderem qual é o contexto das aplicações, as necessidades dos clientes e as restrições da sua empresa.



Fonte: Silva (2013)

O segundo artefato presente na parte 1 fornece as informações em alto-nível de como as aplicações estão relacionadas dentro do domínio. As similaridades são identificadas em função das *features* presentes nas aplicações (Quadro 2 e Figura 10). Os papéis desempenhados para construir o mapa de produtos/aplicações são dirigidos para o **Especialista do Domínio** e o **Especialista em análise do domínio**.

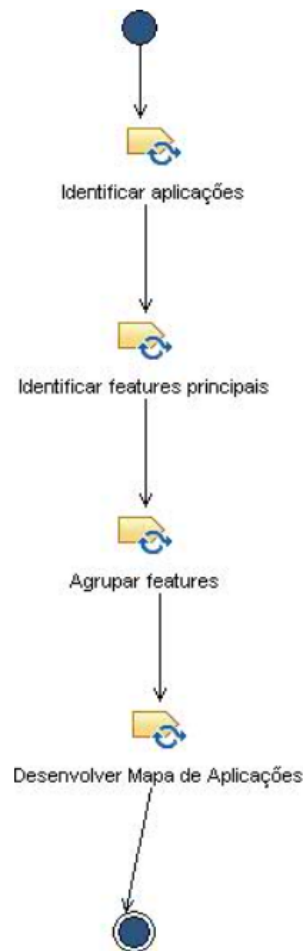
Quadro 2 – *Template* do mapa de aplicações ou mapa de produtos da primeira parte da abordagem RISE-DA.

Mapa de Produtos/Aplicações				
Domínios	Features Principais	subfeatures	Produtos	
			Produto 1	Produto 2

Fonte: Silva (2013)

O terceiro artefato presente na parte 1 cumpre a tarefa de priorizar as *features* principais identificadas no mapa de produtos (Quadro 3 e Figura 11). Isso acontece, pois elas serão detalhadamente analisadas durante o processo incremental do Scrum. Os papéis desempenhados para catalogar as *features* principais são dirigidos para o **Especialista do negócio**, o **Especialista em análise do domínio** e o **Especialista do Domínio**. Assim que forem obtidos os documentos com as informações necessárias, é preciso que ocorra sua avaliação e aprovação pelos especialistas presentes em cada processo. Para classificar o potencial de reuso e risco, será utilizada as *tags* do quadro 4 e a viabilidade e os benefícios por meio das *tags* do quadro 5.

Figura 10 – O mapa de produtos fornece em alto-nível a cobertura que as *features* oferecem às aplicações.



Fonte: [Silva \(2013\)](#)

5.2 Parte 2

A parte 2 do processo é baseado na metodologia Scrum e tem o objetivo de definir as *subfeatures* e analisar a similaridade e variabilidade de modo incremental (Figura 12). Para cada *Sprint* presente no processo, uma reunião de planejamento é realizada com a ajuda de um Backlog do Domínio para gerar um Backlog da *Sprint* a fim de conseguir um detalhamento maior das *features* principais identificadas na parte 1 do processo.

O framework Scrum é um método ágil proposto por Jeffrey Sutherland e Ken Schwaber em 1995 ([SCHWABER; SUTHERLAND, 1995](#)). Dentre os métodos ágeis, ele é o mais conhecido pela sua fácil adoção pelas empresas. O Scrum é definido pelos seus papéis, eventos e artefatos ([VALENTE, 2020](#)). Além disso, ele tem a característica principal de adotar ciclos curtos e iterativos de desenvolvimento, sendo esse implementado de forma gradativa (Figura 13). Segundo [Qumer e Henderson-Sellers \(2007\)](#), o Scrum é um método flexível, adaptável e iterativo que funciona muito bem com as ideias de um processo de controle organizacional,

Quadro 3 – *Template* da avaliação das features principais da segunda parte da abordagem RISE-DA.

Avaliação das <i>Features</i> Principais		
Critério de Avaliação/ <i>Feature</i> Principal		<i>Feature</i> 1
Potencial de Reuso		<i>Feature</i> 2
Risco		
Viabilidade	Maturidade	
	Estabilidade	
	Restrições de recursos/organizacionais	
Benefícios	Potencial de mercado	
	Similaridade e Variabilidade	
	Acoplamento e coesão	
	Potencial de artefatos legados existentes (em termos de código)	

Fonte: [Silva \(2013\)](#)

Quadro 4 – *Tags* utilizadas para classificar o potencial de reuso e o risco das *features* principais.

Potencial de Reuso/Risco
Muito Alto
Alto
Médio
Baixo
Muito Baixo

Fonte: [Silva \(2013\)](#)

Quadro 5 – *Tags* utilizadas para classificar a viabilidade e os benefícios das *features* principais.

Outros Critérios
Completa
Predominante
Parcial
Nenhuma

Fonte: [Silva \(2013\)](#)

mais especificamente para micro, pequenas e médias empresas de software, onde o time de desenvolvimento é menor do que 10 pessoas.

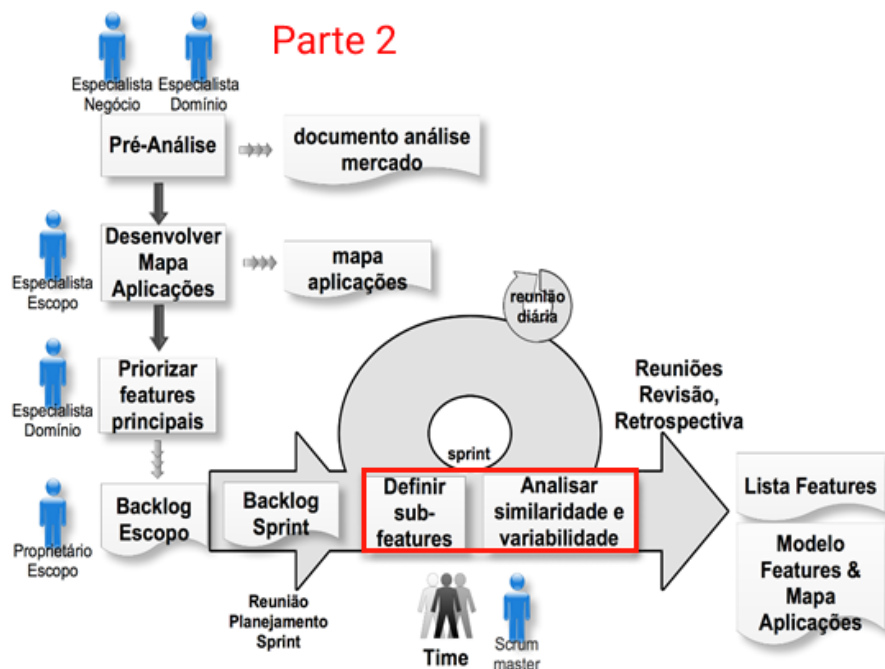
O Scrum é especializado em gerenciamento de projetos, que não precisam estar envolvidos em desenvolvimento de software, mas pode ser usado em um processo de análise de *escopo ágil* como a abordagem RISE-DA. Essa análise utiliza de iterações para providenciar aos *stakeholders* mecanismos que constroem adaptações no escopo, como elicitação, especificação, verificação, priorização, estimação e validação conforme o tempo de cada tarefa ([VALENTE, 2020](#)). Durante

Figura 11 – A priorização das *features* principais é importante para a análise iterativa da próxima etapa.



Fonte: [Silva \(2013\)](#)

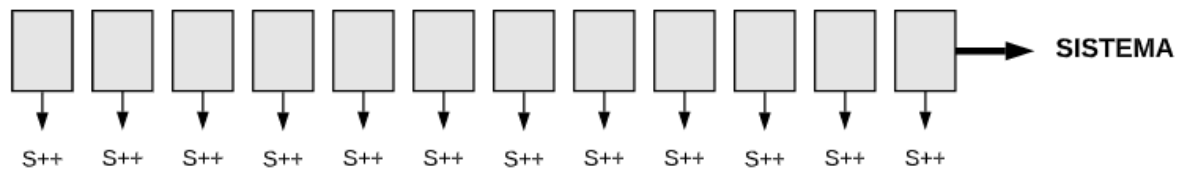
Figura 12 – Parte 2 do processo de análise de domínio da RISE-DA, buscando o detalhamento das *features* principais e do mapa de produtos.



Fonte: [Silva \(2013\)](#)

os ciclos iterativos da parte 2 da abordagem, os *features* são avaliadas em termos de tamanho do projeto que participam, tamanho do time, estilo de desenvolvimento e codificação, tecnologias,

Figura 13 – Processo ágil feito por meio de implementações gradativas, sendo cada iteração um incremento no sistema que é validado e testado pelos usuários.



Fonte: Valente (2020)

ambiente, cultura organizacional e mecanismos de abstração para que dessa forma elas sejam priorizadas de acordo com a flexibilidade, velocidade, qualidade, aprendizado e responsividade de cada produto do domínio (Quadro 3).

Os times de Scrum no RISE-DA são formados pelo **Especialista do Domínio**, o **Scrum Master** e de três a nove **Desenvolvedores**. Os eventos principais da metodologia Scrum são as *Sprints*, o Planejamento da *Sprint*, as Reuniões Diárias, a Revisão da *Sprint* e a Retrospectiva. Uma *Sprint* é uma iteração de tarefas. A conclusão de várias *Sprints* ao longo do tempo acaba por estruturar um potencial artefato até este se tornar tangível ao final do projeto. Já o Planejamento da *Sprint* é uma reunião em que o **Especialista do Domínio** propõe as histórias que devem ser mapeadas, documentadas e analisadas e os **Desenvolvedores** decidem se as histórias estão coerentes com o tempo total para a entrega do artefato e o tempo de duração para cada tarefa. As Reuniões propõem a comunicação entre o time de desenvolvimento para que todos saibam sobre o andamento do projeto. A Revisão da *Sprint* avalia os resultados de uma *Sprint*, podendo ocorrer ajustes ou ser aprovadas pelo **Especialista do Domínio**. Por fim, a Retrospectiva é encarregada de analisar os pontos de melhoria do processo, nas pessoas, nas ferramentas e nos relacionamentos ao final de uma *Sprint*. Cada um dos processos citados precisam ter uma duração definida, chamada de *time-box* de uma atividade (VALENTE, 2020).

Os principais artefatos que compõem o Scrum são o Backlog do Domínio e o Backlog do *Sprint*. O Backlog do Domínio representa um conjunto de histórias que resumem as tarefas que devem ser analisadas, e como um artefato dinâmico, pode ser constantemente atualizado pelo **Especialista do Domínio**, para que atenda todas as mudanças que surgirem das *features* analisadas. Na abordagem RISE-DA, o Backlog do Domínio/Escopo é representado pelo um quadro 6. Já o Backlog do *Sprint* é um artefato dinâmico que terá a lista de *Sprints* com suas respectivas durações, podendo ser adaptável somente no conjunto de horas de cada *Sprint*, mas trabalhando com objetivos fechados que foram definidas pelo **Especialista do Domínio** (VALENTE, 2020) (Quadro 7).

Cada *Sprint* de 1 a n deverá conter um objetivo, duração, data de início, data de encerramento, data de revisão, definições prontas, local e hora das reuniões, **Scrum Master**,

Especialista do Domínio e os integrantes da equipe. Para as *features* principais selecionadas, define-se as tarefas, o proprietário dessa *Sprint*, qual é o seu status e se possui alguma observação (Quadro 8).

Quadro 6 – *Template* do Backlog do Domínio/Escopo da segunda parte da abordagem RISE-DA.

Backlog do Domínio/Escopo			
Domínio	<i>Features</i> Principais	Prioridade	Estimativa
Domínio 1	<i>Feature</i> 1	Muito Alta	8

Fonte: [Silva \(2013\)](#)

Quadro 7 – *Template* do Backlog do *Sprint* da segunda parte da abordagem RISE-DA.

Sprint n	
Objetivo	Analisar a <i>feature</i> principal X
Duração	X horas
Data de Início	XX/YY/ZZ
Data de Encerramento	XX/YY/ZZ
Data de Revisão	XX/YY/ZZ
Definições Prontas	Modelo de features, mapa do produto e lista de features validados
Local/hora das reuniões diárias	Sala X as Yh
Scrum Master	Especialista 1
Especialista do Domínio/Dono do Produto	Especialista 2, Especialista 3
Integrantes da Equipe	Especialista 4

Fonte: [Silva \(2013\)](#)

Quadro 8 – *Template* relacionado ao Backlog do *Sprint*.

<i>Features</i> Principais Selecionadas	Tarefas	Proprietário	Status	Observação
<i>Feature</i> 1	Tarefa 1	Especialista 1	Feito	Nenhuma
<i>Feature</i> 2	Tarefa 2	Especialista 2	Parcial	
<i>Feature</i> 3	Tarefa 3	Especialista 3	À fazer	

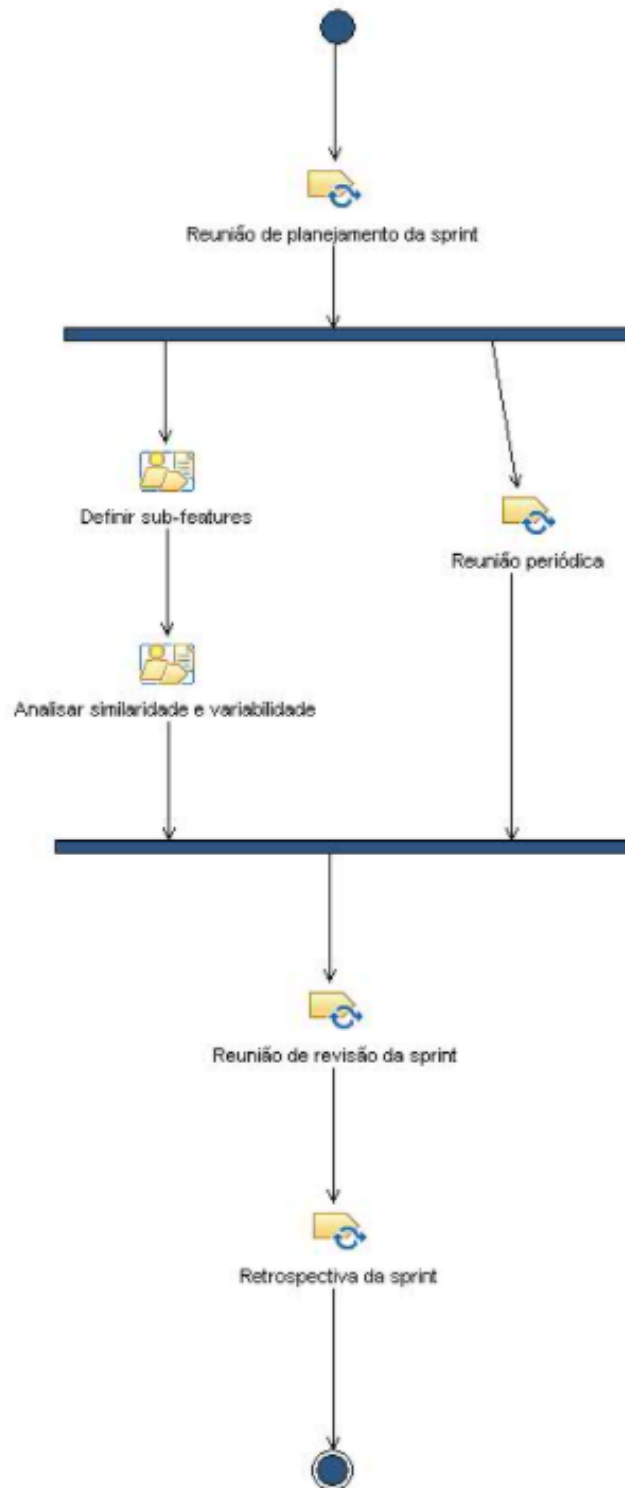
Fonte: [Silva \(2013\)](#)

Nessas iterações definem-se as tarefas que serão realizadas para especificar as *features* principais e suas prioridades no domínio das aplicações. Com isso, é possível analisar as

similaridades e variabilidades entre todas as aplicações da empresa por meio dessas *features*. Dessa forma, essa análise documenta o potencial de reusabilidade do domínio para oferecer os subsídios necessários para o desenvolvimento voltado para o reuso. Além disso, é possível observar o potencial de mercado, a maturidade, a estabilidade, quais são as restrições dos recursos, a similaridade e variabilidade, o acoplamento e a coesão de cada um dos produtos. Portanto, essa análise fornece um panorama para a real possibilidade de reuso do sistema e fornece quais são as customizações possíveis que as aplicações futuras poderão utilizar. Durante todo o processo, serão inspecionados e atualizados os documentos criados na parte 1, durante as reuniões de revisão e retrospectiva da *Sprint* (Figura 14).

Os principais artefatos gerados pela parte 2 do processo RISE-DA são a árvore de *features* e o mapa de produtos em um alto nível de detalhamento. Toda a documentação gerada será destinada para o desenvolvimento dos serviços reusáveis dos projetos futuros da empresa BIT. A empresa então ficará encarregada de utilizar o processo de RISE-CBD para o desenvolvimento baseado na especificação do domínio que foi criado, o que guiará todo o processo de construção dos artefatos reusáveis que estarão presente no repositório da empresa (POHL; BÖCKLE; LINDEN, 2005).

Figura 14 – Processo incremental que a partir da reunião de planejamento da *Sprint*, há o processo de definição das *subfeatures* e análise das similaridades e variabilidades durante reuniões periódicas até que todas as tarefas sejam finalizadas e assim ocorrer a revisão e a retrospectiva da *Sprint*.



Fonte: [Silva \(2013\)](#)

6

Metodologia

A metodologia empírica utilizada neste trabalho de conclusão de curso é baseada em um estudo de caso da aplicação da etapa RISE-DA na empresa BIT - Empresa Júnior de Ciência da Computação. Essa pesquisa aplicada será feita a partir da lógica indutiva, com o propósito descritivo-exploratório, sob o viés da abordagem interpretativista. Será feita uma coleta de dados qualitativos por meio da observação em campo dos *stakeholders* e análise dos documentos produzidos durante o estudo. A partir disso, os dados serão observados para uma maior compreensão dos efeitos da abordagem na empresa e quais os benefícios gerados (WOHLIN; AURUM, 2014).

Durante todo o processo de análise de dados, serão feitos processos de identificação, validação, especificação e documentação, organização de diagramas e quadros, inspeção de código e análise dos artefatos escolhidos. Todos os dados coletados serão priorizados pela sua capacidade de reusabilidade por meio dos níveis de variabilidade e similaridade das *features* analisadas entre os produtos do domínio, ou seja, serão escolhidos os melhores tipos de dados necessários para suportar as próximas decisões para que os problemas iniciais sejam detectados com o intuito de evitar esforço desnecessário mais tarde.

O processo de análise começará capturando informações sobre as circunstâncias em que a empresa se encontra, por exemplo, seu lugar no mercado, concorrentes, propósitos, metas, estrutura, cultura organizacional e a disposição dos papéis dos *stakeholders* dentro da organização. Dessa forma será possível descobrir se a empresa possui interesse e capacidade de investir em um processo de escopo voltada para o reuso de software. A próxima etapa é administrar treinamentos dos *stakeholders* nas etapas da abordagem RISE-DA e descobrir se os membros possuem alguma dificuldade em entender o processo. Para cada etapa serão feitos ajustes de todos os artefatos criados, caso for necessário conforme a pesquisa avança, para que o detalhamento seja o mais fiel possível aos dados obtidos. Essa etapa é importante, uma vez que é preciso localizar e remover os possíveis erros durante a captura dos artefatos de alto nível, para que não seja necessário

reestruturar todo o cenário criado ao realizar o detalhamento dos artefatos de baixo nível.

A observação em campo da abordagem RISE-DA levará em conta as experiências das pessoas envolvidas no estudo de caso, além do contexto da empresa envolvida no processo de análise. Além disso, o [Especialista em análise do domínio](#) e o [Scrum Master](#) são membros da empresa e, portanto, contribuirão ativamente com as respostas empregadas durante todos os processos necessários para a coleta de dados. Além disso, todas as reuniões serão gravadas para capturar informações referentes a como a empresa se organiza, como é sua cultura e como são os processos envolvidos no desenvolvimento de um projeto, assim como as reuniões de *Sprint* que ocorrem com a presença dos [Especialista do Domínio](#) que desenvolveram as aplicações analisadas.

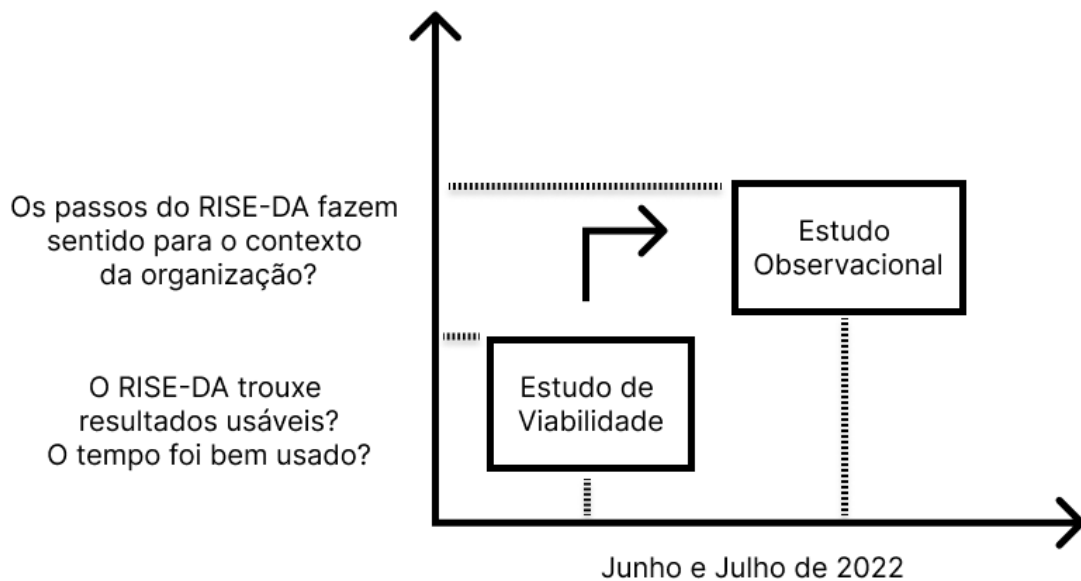
Todo o repositório privado da empresa será analisado para criação dos artefatos analisados, incluindo código legado, contratos com as empresas, atas de reuniões, planilhas de fluxo de caixa, memorandos, minicursos realizados, processos seletivos, documentos das eleições dos membros da diretoria executiva, documentos de análise de requisitos dos projetos antigos, estatutos, regimentos internos, além de entrar em contato com os membros antigos para obter conhecimento do domínio. Ademais, será feita uma observação em campo de todo o processo de realização de um projeto de desenvolvimento para que se obtenha o conhecimento das tecnologias em uso, técnicas, métodos e processos que venham a contribuir para uma melhor análise de escopo.

Por fim, será aplicado um questionário aos *stakeholders* para coletar os dados qualitativos sobre quais foram os efeitos da abordagem RISE-DA na Empresa Júnior de Ciência da Computação. Entretanto, apenas essa métrica de avaliação não será o suficiente para gerar o nível desejado de detalhamento nas respostas. Por conta disso, o questionário será aplicado durante uma entrevista com os *stakeholders*. Dessa forma, a coleta de dados será mais eficiente por proporcionar que os *stakeholders* tenham mais liberdade nas respostas.

A avaliação do processo de análise do RISE-DA na empresa BIT será feita por meio da figura 15, em que é possível observar as perguntas que fazem parte desse processo de avaliação. Quando todos os artefatos forem criados, será possível aplicar um estudo de avaliação da abordagem. O questionário utilizado durante a entrevista com os *stakeholders*, será feito com o propósito de responder a primeira questão mostrada na Figura 15, em que a questão mais ampla é o estudo de viabilidade do processo. Entretanto, a segunda questão apresentada da Figura 15 (estudo observacional) também pode ser abordada durante a entrevista ao descobrir se a abordagem faz sentido ao ser aplicada no contexto da empresa BIT, quando forem analisados os resultados gerados. Portanto, o estudo de viabilidade e o estudo observacional serão utilizados para avaliar a abordagem RISE-DA na empresa BIT.

As perguntas a serem respondidas incluem: “O RISE-DA trouxe resultados usáveis?”, “O tempo foi bem usado?” e “Os passos do RISE-DA fazem sentido para o contexto da organização?”. O tempo será contabilizado por meio das gravações realizadas em cada uma das etapas e por meio de gráficos *burndown* de cada *Sprint*.

Figura 15 – Avaliação do processo de análise da abordagem RISE-DA no contexto da empresa BIT. Para a pesquisa em questão, serão considerados o estudo de viabilidade e o estudo observacional.



Fonte: adaptado de [Shull, Carver e Travassos \(2001\)](#)

Sendo assim, o objetivo da realização da abordagem RISE-DA na Empresa Júnior de Ciência da Computação é analisar todos os ativos da empresa BIT e mapear os possíveis artefatos que permitem a customização em massa. Dessa forma, é possível avaliar se a proposta do RISE-DA é eficiente em situações tão extremas como o cenário de pesquisa proposto neste trabalho.

7

Limitações do Processo

Para realização do estudo de caso, foi fundamental considerar algumas limitações presentes no contexto do cenário de pesquisa em que a abordagem RISE-DA seria aplicada. Dessa forma, foi essencial entender o que precisaria ser realizado em nível de etapas do processo, para que não houvesse problemas ao aplicar as técnicas desejadas e dessa forma poder corrigi-las. Para melhor contribuição das técnicas utilizadas, foi preciso considerar também como os *stakeholders* se organizavam para conseguir catalogar os dados de forma que sua aquisição estivesse em sincronia com o ambiente e estilo de trabalho que eles estavam acostumados.

7.1 Problemática BIT

O mapeamento inicial da empresa foi realizado durante uma das reuniões semanais para que houvesse a introdução da ideia de aplicação da abordagem RISE-DA e se os membros tinham interesse em aplicá-la. Alguns pontos foram levantados pelos membros da diretoria:

- A empresa não possuía nenhuma metodologia de desenvolvimento aplicada formalmente na empresa, uma vez que não possuíam nenhum especialista como parte do time. Apenas havia rascunhos de aquisição de requisitos, gravações e outros artefatos, às vezes incompletos, para organização das etapas de desenvolvimento;
- Ocorria problemas na organização dos membros durante o processo de desenvolvimento, uma vez que não havia separação de papéis de forma efetiva. Portanto, havia dificuldade em atribuição de tarefas, execução das tarefas e pouca documentação de projetos passados;
- Não utilizavam nenhuma ferramenta de organização de tarefas, porém utilizavam ferramentas para versionamento de código e ambientes de desenvolvimento;
- As reuniões não estavam acontecendo de forma recorrente no período em que a empresa teve que adotar reuniões remotas e isso teve um grande impacto na organização da empresa;

- Não possuem documentação de código e, portanto, não utilizam código legado como fonte de conhecimento.

Por conta dos itens anteriores, ocorriam as seguintes consequências: o prazo de entrega dos projetos era adiado, não havia uma comunicação efetiva entre os membros da equipe e ocorria falhas na aquisição de requisitos, uma vez que algumas funcionalidades do projeto não eram atendidas.

Durante as reuniões de contextualização do RISE-DA, os membros da empresa concordaram em aplicar a abordagem pelos seguintes motivos: conseguir obter um repositório com códigos documentados, aumentar a escala de eficiência, qualidade, produtividade, performance e diminuir tempo de entrega, difundir conhecimento entre equipes de desenvolvimento, aumentar a previsibilidade de custos, cronogramas e qualidade do serviço, obter a capacidade de customização das aplicações, ter uma melhor visualização do domínio do negócio e ajudar os membros da equipe na aquisição de conhecimento por meio da documentação dos códigos. Para isso, é necessário fazer a primeira etapa do processo, que é o RISE-DA e a partir disso, buscar a segunda etapa do processo que é o RISE-CBD.

7.2 Membros

Os membros efetivos da empresa são compostos por estudantes de graduação de Ciência da Computação da UNIOESTE. No começo da pesquisa, participavam o presidente, o secretário e 6 [Desenvolvedores](#). Dentre os 8 estudantes, apenas 3 fizeram a disciplina de estágio obrigatório e, por conta disso, demonstravam alguma experiência de mercado. Ao longo do período da pesquisa ocorreram 5 afastamentos das obrigações e atividades da BIT, incluindo o presidente e 4 [Desenvolvedores](#), pois não possuíam mais tempo para os compromissos da empresa.

7.2.1 Processo seletivo

Devido ao grande afastamento de membros durante o período da pesquisa, foi necessário realizar um processo seletivo na empresa para que fosse possível aplicar as etapas iterativas e o questionário com o intuito de possuir um número significativo de respostas. Pelo processo seletivo, 8 *trainees* foram efetivados e passaram pelo período de teste até se tornarem [Desenvolvedores](#) da empresa. Entre os 8 *trainees*, 3 estão cursando o terceiro ano e 5 estão no primeiro ano da universidade, tendo ainda um conhecimento raso sobre como funcionam os procedimentos dentro de uma empresa de software, assim como em desenvolvimento e conhecimento de ferramentas e linguagens. Entretanto, o processo de treinamento dos novos membros se tornou prioridade entre as atividades administradas dentro da organização.

7.3 Materiais

Os materiais utilizados durante a execução da abordagem RISE-DA, incluem todos os materiais disponíveis para aplicação da abordagem. Foram utilizadas as referências bibliográficas de [Silva \(2012\)](#) e [Silva \(2013\)](#), além de documentos HTML que apresentavam todos os passos a serem seguidos, os *templates* que foram apresentados no capítulo 5 e os papéis dos especialistas necessários em cada etapa. Para o treinamento, foram utilizados slides introdutórios de “Introdução ao reuso”, “Adoção do reuso”, “Programa de reuso”, “Conceitos de análise de domínio” e “RISE-DA parte 1”. Todos os materiais utilizados fazem parte dos documentos internos fornecidos pelo autor da abordagem RISE-DA e RISE-CBD que é orientador deste trabalho de conclusão de curso.

Para a criação da árvore de *features* ou modelo de *features*, foi utilizado o FeatureIDE que é uma ferramenta de código livre voltada para desenvolvimento de software baseado em *features* integrada ao ambiente do Eclipse ([FEATUREIDE, 2020](#)).

Para análise de código legado, foi necessário entrar em contato com os antigos membros da BIT para recuperar alguns dos códigos desenvolvidos desde 2016, pois o repositório privado que a empresa utiliza para guardar suas versões estava incompleto. Foi possível resgatar 4 projetos dos 11 catalogados. Além disso, todas as documentações encontradas para análise da empresa estavam armazenados na nuvem ou estavam arquivados juntos com as pastas da empresa. Algumas documentações também foram recuperadas com os membros afastados.

8

Resultados e Discussão

8.1 Treinamento

O primeiro treinamento realizado trouxe a introdução e o levantamento de conceitos do processo ágil Scrum, mostrando como funcionam ciclos curtos e iterativos de desenvolvimento, seus principais artefatos e eventos (Backlog do Produto, *Sprint* e Backlog do *Sprint*), além de apresentar os possíveis papéis que cada um poderia desempenhar durante a pesquisa.

Os slides introdutórios de “Introdução ao reuso”, “Adoção do reuso”, “Programa de reuso”, “Conceitos de análise de domínio” e “RISE-DA parte 1” mostravam todo o processo da abordagem RISE-DA como apresentado no capítulo 5. Foram apresentados a Parte 1 da abordagem (Pré-análise, Mapa de Produtos, Priorização de *features* principais) e a Parte 2 em que são definidos as *subfeatures* e como ocorre o processo de análise de similaridade e variabilidade. Em cada parte do treinamento foram trazidas observações do que seria necessário obter durante a pesquisa, como foi apresentado no capítulo 7. Para esse treinamento foi avaliado a ficha técnica de produtos, as aplicações documentadas e o repositório privado da empresa, além das planilhas de preço de serviço. Por conta disso, houve uma reunião para pesquisa de concorrência e preços de *software* ao tentar situar a empresa no mercado.

No total foram utilizadas duas reuniões para os treinamentos de introdução à metodologia ágil Scrum e explicação da abordagem RISE-DA. A reunião para o primeiro treinamento levou 30 minutos e para o segundo treinamento levou 1 hora e 10 minutos. Todas essas etapas foram gravadas e documentadas com a permissão dos *stakeholders*. Como os membros da empresa já possuíam facilidade com o Scrum, durante o período da pesquisa, foi incentivado que realizassem a disciplina de reuso de software que estava sendo ofertada pela UNIOESTE para os alunos de Ciência da Computação.

8.2 Pré-Análise

Todos os resultados da pré-análise estão presentes nos quadros de 9 e 10. As respostas catalogadas para construção desses artefatos foram obtidas por meio de reuniões consecutivas de discussão e validação dos pontos analisados.

8.2.1 Análise de Mercado

A partir da análise de mercado (Quadros 9 e 10) foi possível identificar que a BIT tem por objetivo aumentar a proximidade do mercado de trabalho com o meio acadêmico e proporcionar aos estudantes um desenvolvimento de capacidades sociais e técnicas ao trabalhar com nichos de mercado variados. A estratégia de marketing da empresa é levar ao cliente (pessoas físicas ou jurídicas) um custo abaixo do custo de mercado, mas trazendo qualidade e performance aos seus sistemas de software, além de prestação de serviços especializados e treinamentos com os responsáveis técnicos da aplicação. A empresa supre qualquer necessidade de mercado que venha a ser de interesse dos estudantes. Entretanto, as manutenções corretivas acontecem apenas durante os 3 meses consecutivos depois da entrega da aplicação, uma vez que os direitos autorais e o código fonte passam a ser de responsabilidade do cliente. No momento da pesquisa estavam sendo desenvolvidas principalmente aplicações web, mobile e híbridas, utilizando HTML, CSS, JavaScript, NodeJS, React, Wordpress ([WORDPRESS, 2022](#)), MongoDB e VueJS como as principais tecnologias de desenvolvimento.

A listagem de concorrentes inclui tanto empresas da cidade de Cascavel-PR, onde a BIT reside, quanto empresas juniores de várias partes do Brasil. As empresas juniores catalogadas são a Citi da UFPE, Acens da UECE, icmc júnior da USP, eccjr da UNESP, Hut8 da UFP, iJunior da UFMG, ECOMP da UFP, CJR da UnB e Ejec que pertence ao curso de Engenharia da Computação da UFSC. As empresas juniores foram contatadas por *email* para que respondessem algumas perguntas referentes a sua estratégia de negócio e se possuíam interesse em reutilização de software. Apesar de 4 das 9 empresas terem respondido o *email*, apenas 1 empresa respondeu o questionário. Como não houve respostas significativas, ele não foi considerado. Entretanto, é possível contactá-las em um estudo futuro para aplicar a abordagem RISE-DA em outros contextos de empresas juniores.

Esse artefato foi construído pelo [Especialista em análise do domínio](#) com ajuda e validação de 1 especialista do domínio, 2 especialistas do negócio e 2 [Desenvolvedores](#). O período necessário para construção e validação desse artefato foi durante os meses de Fevereiro, Março e Abril de 2022. Para o questionário realizado, foi revisado e validado novamente o artefato de análise de mercado.

Quadro 9 – Artefato da análise de mercado - parte 1, em que é possível observar informações do mercado onde a BIT sem encontra, características dos seus usuários e suas tecnologias.

Análise de Mercado - parte 1					
Estratégia de marketing	Necessidade de mercado	Concorrência	Características dos usuários		
			Perfil dos clientes	Níveis de habilidade	Tecnologias de desenvolvimento
Custo do sistema é abaixo do custo de mercado	Sistemas sob demanda	SetaDigital	Pessoas físicas ou jurídicas da região de Cascavel	Equipe realiza treinamento para as funcionalidades entregues	HTML
Prestação de serviços especializados	Desenvolvimento e consultoria	Wealth Systems	Possível expansão do negócio	Usuários já treinados irão atuar no treinamento de outros usuários	CSS
Instalação do software nos dispositivos do cliente	Aplicações desktop	Linx		Treinamento ocorre entre o especialista do negócio e os responsáveis técnicos da aplicação	JavaScript
Entregas totais dos sistemas ao cliente	Aplicações web com acesso via navegador	Citi (UFPE)		Dependendo da aplicação são escritos manuais sobre o treinamento	NodeJS
Existem avaliações e feedbacks periódicos dos clientes sobre o sistema	Aplicações híbridas	Acens (UECE)		Comunicação entre desenvolvedores e clientes permanece para o recebimento de feedback sobre o funcionamento (3 meses)	React
		ICMC Júnior (USP)		Não são aplicados questionários sobre a qualidade do treinamento	Wordpress (WORDPRESS, 2022)
		ECCJR (UNESP)			MongoDB
		Hut8 (UFF)			VueJS
		iJúnior (UFMG)			
		Ejec (UFSC)			
		ECOMP (UFF)			
		CJR (UnB)			
		Intelitech			
		Smart BR Soluções			
		Bordoms			

Quadro 10 – Artefato da análise de mercado - parte 2, em que é possível observar as regras de negócio presentes na BIT, assim como seus objetivos de negócio e de reuso de software.

Análise de Mercado - parte 2			
Restrições culturais/legais	Tempo de entrega	Objetivos de negócio	Objetivo de reuso
As manutenções corretivas acontecem somente no período de 3 meses após a entrega do sistema. Depois desse prazo o código fonte e os direitos autorais da aplicação são de responsabilidade do cliente	Tempo de entrega estipulado por contrato (pode haver mais de um contrato no período de desenvolvimento)	Possuir um repositório com códigos bem documentados	Aumentar a escala de eficiência e produtividade
Internamente ligada ao regimento, estatuto, normas internas da instituição UNIOESTE	Entrega de softwares completos a partir da especificação do cliente (tempo de entrega depende do projeto)	Desenvolvimento de capacidades sociais e técnicas	Melhorar qualidade do serviço
Ligada ao código de ética do Movimento Empresa Júnior e as deliberações superiores da Diretoria Executiva		Oferecer soluções a baixo custo	Difundir conhecimento entre equipes de desenvolvimento
Alta rotatividade de funcionários		Aumentar a proximidade do mercado com o meio acadêmico	Diminuir o tempo de entrega dos serviços
Alta maleabilidade na escolha do nicho de mercado		Aumentar a habilidade de trabalhar com nichos de mercado variados	Documentação de boas práticas de software
Integrantes são declarados como voluntários na empresa, sem direito a remuneração, compensação ou indenização de qualquer tipo		Aumentar a habilidade de comunicação com os clientes	Aumentar a satisfação do cliente
<u>Desenvolvedores</u> participam de parte do processo da criação do sistema, sendo alocados para processos separados (front-end, back-end)		Aumentar qualidade, produtividade, performance e diminuir tempo de entrega	Aumentar a previsibilidade de custos, cronograma e qualidade do serviço
Os acadêmicos têm permissão da Diretoria Executiva para pedir afastamento no desenvolvimento em período de provas/trabalhos		Manter a segurança e consistência dos dados do sistema	Capacidade de customização das aplicações
Adequação das aplicações e direitos legais sob os regulamentos do MIT		Investimento aplicado na empresa é voltado para o treinamento dos desenvolvedores e melhoria da estrutura	Padronização do desenvolvimento
Avaliações periódicas de funcionários (faltas com advertência)		Realizar controle e avaliação do projeto (feedback semanal)	Melhor visualização do domínio de negócio
Sites podem ser produzidos por apenas um desenvolvedor		Abertura de negociações com os clientes	Ajudar os membros da equipe na aquisição de conhecimento
O feedback do andamento do projeto é semanal		Processo seletivo para adesão de novos membros	

8.3 Mapa de Produtos

Depois de encontrar os objetivos de negócio da empresa e onde ela se encontra no mercado, no mapa de produtos será possível observar cada uma das aplicações já desenvolvidas. A BIT trabalha desde 2017 e possui 14 aplicações desenvolvidas, divididas entre web, mobile e híbridas, que foram catalogadas por meio dos contratos realizados com os clientes. Os quadros referentes a essa etapa são: 11, 12, 13 e 14.

8.3.1 Lista de Aplicações

A lista de aplicações é desenvolvida para descrever cada uma das aplicações e suas funcionalidades gerais. Como observado no quadro 11, são desenvolvidas aplicações com módulos web, mobile e híbridas. Todos os módulos das aplicações, exceto “Apthus”, “Hyeperion” e “G4”, propõem apresentações de sites institucionais responsivos que contêm informações da aplicação e da entidade representativa da mesma, além de algumas solicitações de serviços (formulários, históricos, downloads, envio de currículos e gerenciamento de dados). A principal funcionalidade presente em todas as aplicações é a gestão de informações.

8.3.2 Lista de *features*

A lista de *features* é responsável por catalogar o domínio, as *features* principais, suas *subfeatures* e a respectiva descrição de cada uma conforme as informações obtidas anteriormente na lista de aplicações (Quadro 12). As *features* principais foram separadas em “Controle de acesso dos usuários”, “Modelagem dos dados”, “CRUD”, “Visualização de relatórios”, “Criação de formulários”, “Download de informações”, “Controle de caixa”, “Gerência de compra e venda”, “Páginas informativas”, “Atendimento ao cliente”, “Envio de currículo” e “Sistema de notificações”. Como dito anteriormente, os dados foram obtidos por meio dos contratos feitos com os clientes e algumas especificações existentes dentro do repositório privado da empresa. Portanto, as *subfeatures* especificadas nessa etapa são palavras-chaves obtidas para cada uma das aplicações.

O domínio que a BIT engloba é o de gestão de informações, uma vez que esse é o principal objetivo por trás do desenvolvimento das aplicações. O domínio apresentado nesse trabalho se mostrou muito abrangente, pois a BIT é uma empresa que possui muita variedade de nichos de mercado em suas aplicações, sendo sua principal correspondência derivada das funcionalidades descritas no quadro 11. Algumas das aplicações apresenta uma página pública de apresentação da empresa ou uma página privada para os funcionários que fazem o gerenciamento dos dados referentes ao propósito da aplicação.

Quadro 11 – Lista de descrições das aplicações já desenvolvidas pela BIT desde 2017.

Lista de aplicações	
Aplicação	Descrição
Siqueira Transportes	Plataforma WEB, composta por uma página de apresentação da empresa (site institucional) e uma parte interna decomposta em outras funcionalidades (CRUD das informações).
PrimorLab	Site institucional responsivo contendo as informações da empresa. Também há uma área restrita contendo formulários para solicitação de serviços, área de histórico e seção de downloads e cadastro de usuários.
Positiva Engetech	Site institucional responsivo contendo as informações da empresa. Também há uma <i>loading page</i> com destaques sobre a empresa, uma página contendo informações gerais da empresa, uma página com portfólio de produtos da empresa e um link para cada página individual dos produtos.
Hyperion	Um módulo WEB e dois módulos MOBILE. O sistema WEB é composto pela página de administradores que podem gerenciar os dados e página de funcionários que podem ter algumas tarefas habilitadas. O módulo MOBILE contém as mesmas autorizações de usuário para os funcionários, porém o modo de administrador tem algumas funcionalidades a mais.
Projeto Engenharia	Um módulo WEB desenvolvido contendo descrição da empresa, histórico e portfólio.
G4	Dois módulos do sistema WEB para o gerenciamento financeiro da empresa. Um módulo é o módulo de caixa e o segundo módulo é a de gerência de compra, vendas e orçamento.
CQD	Dois módulos MOBILE e um módulo WEB. Um dos módulos atende serviços para utilização de médicos, outro módulo para utilização dos pacientes e uma plataforma WEB para o uso de ambos.
CONNECT4SITE	Plataforma WEB, contendo um site de visitas responsivo contendo informações das lojas, quem somos, atendimento por meio de formulários, envio de currículos e garantias.
Apthus	Um sistema WEB composto pela página de administradores que podem gerenciar os dados e página de funcionários que podem ter algumas tarefas habilitadas.
GDIS	Um site híbrido (MOBILE e WEB) responsivo para visualização e edição de informações.
Univita	Gestão de funções administrativas para médicos, englobando funcionalidades de gestão de venda, acompanhamento de pacientes, agendar consultas e ver histórico de atendimento.

8.3.3 Mapa de Produtos - Primeira versão

A primeira versão do mapa de produtos está representada nos quadros 13 e 14. Cada um desses quadros mostra o domínio, as *features* principais e as *subfeatures*, assim como apresentado no quadro 12, porém, agora analisando como as aplicações apresentadas no quadro 11 se relacionam com as *features* e *subfeatures* catalogadas. As aplicações web e mobile de uma mesma aplicação foram consideradas como produtos diferentes.

Quadro 12 – Apresentação do domínio, das *features* principais, suas *subfeatures* e descrições.

Lista de <i>Features</i>			
Domínios	<i>Features</i> Principais	<i>Subfeatures</i>	Descrição
Sistema de gestão de informações	Controle de acesso dos usuários	Acesso à páginas distintas	Permissões de tela conforme o usuário logado. Dependendo do usuário, será possível utilizar diferentes partes do sistema.
		login e senha	Tipo de controle de acesso.
	Modelagem dos dados		Técnicas e princípios de coleta de informações referentes a aplicação e quais dados são relevantes guardar no banco de dados. Estruturação dos dados de um banco de dados relacional.
	CRUD		Criar, ler, atualizar e deletar dados
	Visualização de relatórios		Possibilidade de gerar relatórios
	Criação de formulários		Possibilidade de criar e preencher formulários
	Download de informações		Possibilidade de obter informações da aplicação
	Controle de caixa		Responsável por gerenciar um fluxo de entrada e saída de dinheiro dentro da aplicação
	Gerência de compra e venda		Responsável por expor produtos e vendê-los.
	Páginas informativas	Portfólio da empresa	Responsável por expor os produtos da empresa
		Imagens de destaque da empresa	Responsável pela exposição de imagens
		Produtos em destaque	Responsável pela exposição de produtos
		Informações da empresa	Responsável pela divulgação da empresa
		Responsividade	Ser adaptável a todos os dispositivos
	Atendimento ao cliente		Canal de comunicação entre a aplicação e os clientes
	Envio de currículo		Envio de informações por meio da aplicação
	Sistema de notificação	WhatsApp (INC, 2009)	Notificações de quaisquer ações dentro da aplicação
		SMS	
		email	

Quadro 13 – Apresentação do relacionamento entre cada uma das *features* principais e *subfeatures* do domínio de gestão de informações para as 10 aplicações web.

Mapa de Produtos - parte 1													
Domínio	Features Principais	Subfeatures	Produtos										
			Versão WEB										
			Apthus	Siqueira Transportes	PrimorLab	Positiva Engetech	Hyperion	Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita	
Sistema de gestão de informações	Controle de acesso dos usuários	Requisição de telas	X	X	X	X	X		X	X		X	
	Modelagem dos dados	login e senha	X	X	X	X	X		X	X		X	
			X	X	X	X	X	X	X	X	X	X	
	CRUD		X	X	X	X	X		X	X		X	
	Experiência do usuário		X	X	X	X	X	X	X	X	X	X	
	Visualização de relatórios		X	X	X		X		X			X	
	Criação de formulários		X	X	X		X			X		X	
	Download de informações			X				X					
	Controle de caixa								X				
	Gerência de compra e venda								X				
	Páginas informativas	Portfólio				X			X		X	X	X
		Imagens de destaque		X	X	X			X		X	X	X
		Produtos em destaque				X					X	X	X
		Informações da empresa		X	X	X			X		X	X	X
	Atendimento ao cliente	Responsividade	X	X	X	X				X	X	X	X
	Envio de currículo											X	
	Sistema de notificação	Email	X	X								X	
		WhatsApp (INC, 2009)		X									
SMS			X										

Ao analisar a primeira versão do mapa de produtos, já é possível observar que as *features* principais de “Modelagem dos dados” e “Experiência do usuário” possuem similaridade, uma vez que estão presentes em todas as aplicações da empresa. Portanto, é possível criar uma família de produtos para essas duas *features* principais.

Para cada uma das *features*, existe pelo menos uma correspondência com uma das aplicações, fazendo com que exista uma variabilidade entre elas. É possível, por exemplo, criar uma família de produtos para a *feature* principal “CRUD”, uma vez que existem produtos (Apthus, Siqueira Transportes, PrimorLab, Positiva Engetech, Hyperion, Hyperion mobile, G4, CQD, CQD mobile, Univita e Univita mobile) que possuem similaridade para essa *feature*. Existem pontos de variação para essa *feature* principal nos produtos “Projeto Engenharia”, “CONNECT4SITE” (Quadro 13) e “GDIS” (Quadro 14), uma vez que esses são direcionados respectivamente à aplicações web e híbrida voltadas à páginas informativas.

É possível observar também que existem algumas *features* relacionadas apenas com 1 aplicação. Isso acontece para as *features* principais de “Controle de caixa”, “Envio de currículo” e nas *subfeatures* “WhatsApp” e “SMS” da *feature* principal de “Sistema de notificação”. Dessa forma, essas *features* automaticamente terão uma prioridade inferior ao pensar em reusabilidade de código, comparado as outras *features* que apresentam mais de uma correspondência entre as aplicações.

Os artefatos foram construídos pelo especialista em análise de domínio com ajuda e validação de 3 especialistas do domínio e 2 desenvolvedores. Todos os especialistas envolvidos na pesquisa estavam no último ano do curso, apresentando mais de 2 anos de experiência na empresa. O domínio construído para essa pesquisa possuiu apenas o conhecimento desses especialistas com ajuda das documentações da empresa.

O período necessário para construção e validação desse artefato foi durante os meses de Fevereiro, Março e Abril de 2022. Para o questionário realizado (Subcapítulo 8.6), o artefato do mapa de produtos não foi revisado, uma vez que ocorreu muitas atualizações nessa tabela no período de Maio, Junho e Julho de 2022 e, portanto, no questionário foram validados os artefatos atualizados.

8.4 Priorização das *features* principais

A priorização das *features* principais está presente nos quadros 15 e 16. Essa etapa do processo foi avaliada por 2 especialistas do domínio e 1 especialista do negócio durante o mês de Maio de 2022, uma vez que é necessário nessa etapa, configurar para cada uma das *features* analisadas anteriormente, o seu potencial de reuso, riscos, viabilidade e benefícios de cada uma.

Quadro 14 – Apresentação do relacionamento entre cada uma das *features* principais e *subfeatures* do domínio de gestão de informações para 1 aplicação híbrida e 3 aplicações mobile.

Mapa de Produtos - parte 2						
Domínio	Features Principais	<i>subfeatures</i>	Produtos			
			Híbrido	Versão MOBILE		
			GDIS	Hyperion	CQD	Univita
Sistema de gestão de informações	Controle de acesso dos usuários	Requisição de telas		X	X	X
		login e senha		X	X	X
	Modelagem dos dados		X	X	X	X
	CRUD			X	X	X
	Experiência do usuário		X	X	X	X
	Visualização de relatórios			X		X
	Criação de formulários			X		X
	Download de informações					
	Controle de caixa					
	Gerência de compra e venda					
	Páginas informativas	Portfólio			X	X
		Imagens de destaque	X		X	X
		Produtos em destaque				X
		Informações da empresa	X		X	X
		Responsividade	X		X	X
	Atendimento ao cliente				X	X
	Envio de currículo					
	Sistema de notificação	Email				
		WhatsApp (INC, 2009)				
		SMS				

Quadro 15 – Avaliação das *features* principais de “Controle de acesso dos usuários”, “Modelagem dos dados”, “CRUD”, “Experiência do usuário”, “Visualização de relatórios” e “Criação de formulários” em relação ao potencial de reuso, risco, viabilidades e benefícios.

Avaliação das features principais							
Critério avaliação/Feature Principal		Controle de acesso dos usuários	Modelagem dos dados	CRUD	Experiência do usuário	Visualização de relatórios	Criação de formulários
Viabilidade	Potencial de reuso	Muito alto	Muito alto	Muito alto	Muito alto	Médio	Médio
	Risco	Médio	Médio	Muito baixo	Médio	Muito baixo	Muito baixo
	Maturidade	Completa	Parcial	Completa	Completa	Parcial	Parcial
	Estabilidade	Predominante	Completa	Completa	Parcial	Parcial	Parcial
	Restrições de recursos/organizacionais	Nenhuma	Nenhuma	Nenhuma	Nenhuma	Parcial	Parcial
Benefícios	Potencial de mercado	Completa	Completa	Completa	Completa	Parcial	Parcial
	Similaridade e variabilidade	Predominante	Predominante	Predominante	Completa	Parcial	Parcial
	Acoplamento e coesão	Predominante	Nenhuma	Predominante	Completa	Parcial	Parcial
	Potencial de artefatos legados existentes (em termos de códigos)	Completa	Parcial	Completa	Completa	Parcial	Parcial

Quadro 16 – Avaliação das *features* principais de “Download de informações”, “Controle de caixa”, “Gerência de compra e venda”, “Atendimento ao cliente”, “Envio de currículo”, “Páginas informativas” e “Sistema de notificação” em relação ao potencial de reuso, risco, viabilidades e benefícios.

Avaliação das <i>features</i> principais - parte 2							
Critério avaliação/Feature Principal	Download de informações	Controle de caixa	Gerência de compra e venda	Atendimento ao cliente	Envio de currículo	Páginas informativas	Sistema de notificação
Viabilidade	Potencial de reuso	Muito baixo	Muito baixo	Baixo	Muito baixo	Muito Alto	Muito baixo
	Risco	Muito baixo	Alto	Muito baixo	Muito baixo	Baixo	Médio
	Maturidade	Completa	Parcial	Parcial	Parcial	Completa	Parcial
	Estabilidade	Completa	Parcial	Parcial	Parcial	Completa	Parcial
Benefícios	Restrições de recursos/ organizacionais	Nenhuma	Predominante	Nenhuma	Nenhuma	Nenhuma	Nenhuma
	Potencial de mercado	Parcial	Parcial	Parcial	Parcial	Completa	Predominante
	Similaridade e variabilidade	Parcial	Parcial	Parcial	Parcial	Completa	Parcial
	Acoplamento e coesão	Nenhuma	Nenhuma	Nenhuma	Nenhuma	Completa	Parcial
	Potencial de artefatos legados existentes (em termos de códigos)	Nenhuma	Nenhuma	Nenhuma	Nenhuma	Parcial	Parcial

Ao se falar em nicho de mercado na priorização das *features* principais, infere-se que diversos nichos de mercado podem fazer parte do mesmo domínio com a condição de apresentarem algumas características em comum que referenciam ao domínio escolhido. No exemplo das *features* principais analisadas o domínio sistema de gestão de informações possui as aplicações de que gerenciam as informações de caminhoneiros (Siqueira Transportes), loja de vendas (Positiva Engetech e CONET4S1TE), contabilidade (Apthus), administração médica (CQD e Univita), glossário (GDIS) e administração de laboratório (PrimorLab) e administração de empresas (Hyperion, Projeto Engenharia e G4). Para a última, não havia informações suficientes sobre o nicho de mercado que elas pertencem, apenas as funcionalidades empregadas.

O potencial de reuso foi avaliado conforme os resultados da primeira versão do mapa de produtos (Quadros 13 e 14), uma vez que já era possível observar as similaridades e variabilidades dentro do domínio das aplicações. Portanto, “Controle de acesso dos usuários”, “Modelagem dos dados”, “CRUD” e “Experiência do usuário” ficaram com um potencial de reuso “Muito Alto”. O potencial de reuso para a modelagem de dados é caracterizada pela possibilidade de reutilizar informações já obtidas de entidades do mesmo nicho de mercado, uma vez que as etapas de aquisição de requisitos seriam parecidas. A experiência do usuário é caracterizada pelo uso da responsividade nas aplicações, um requisito muito importante que foi catalogado durante as reuniões que foram documentadas para o planejamento das aplicações. As “Páginas informativas” foram categorizadas como “Muito Alto”, uma vez que existem 10 das 14 aplicações que utilizam essa *feature* e 5 delas utilizam todas as suas *subfeatures*. Isso acontece também com a *feature* de “Controle de acesso dos usuários”, relacionada com 11 das 14 aplicações.

Como mencionado anteriormente, o potencial de reuso para as *features* de “Controle de caixa”, “Envio de currículo” e “Sistema de notificação” ficaram com o potencial de reuso “Muito baixo”, assim como “Download de informações” e “Gerência de compra e venda”, pois possuem relação com apenas 2 das 14 aplicações. Isso determina no Backlog do Domínio que essas *features* sejam as últimas a serem detalhadas no processo iterativo pelo seu nível de prioridade.

A validação dos riscos de cada uma das *features* para o negócio foram avaliadas segundo algumas observações adquiridas durante o processo de construção desse artefato. A validação de riscos aqui referencia aos riscos que as *features* principais representam para a organização levando em consideração a estrutura da empresa, cultura organizacional, habilidade dos membros ou tempo de desenvolvimento e possíveis manutenções.

O risco para a *feature* “Controle de acesso dos usuários” foi categorizado como “Médio” e o risco para “Controle de caixa” e “Gerência de compra e venda” foi categorizado como “Alto”. Isso acontece para essas *features*, pois elas precisam de boas práticas de programação e experiências vindas do desenvolvedor, uma vez que o requisito de segurança é essencial para que a aplicação tenha qualidade, já que a criptografia e a consistência na transferência de informações é essencial nesse tipo de contexto.

As avaliações de viabilidade e benefícios foram analisadas pela disponibilidade dos

códigos legados de cada uma das *features*. Onde foi obtido “Nenhuma” em “Potencial de artefatos legados e existentes (em termos de códigos)”, significa que não havia código para ser analisado, uma vez que uma limitação do processo era a lacuna de linhas de código para certas *features* das aplicações. O “Potencial de mercado” e a “Similaridade e variabilidade” foram definidas conforme o potencial de reuso demonstrado anteriormente. A “Maturidade”, “Estabilidade” e “Acoplamento e coesão” dependeram se o código analisado possuía um padrão arquitetural e utilização de boas práticas.

Se não havia código legado para categorizar o “Acoplamento e coesão”, a *feature* foi categorizada como “Nenhuma”. Já a “Maturidade” e a “Estabilidade”, foi analisado o conhecimento dos membros da empresa sobre a implementação de tais *features* e, dessa forma, indicar para cada uma delas as suas respectivas categorias.

As “Restrições de recursos/organizacionais” consideraram as *features* de “Controle de caixa” e “Gerência de compra e venda” como “Predominantes”. Isso acontece, porque como a empresa faz manutenções durante apenas 3 meses, qualquer problema que ocorrer no *software* depois desse período não seria de responsabilidade da empresa. Como os membros da BIT - Empresa Júnior de Ciência da Computação são estudantes, o contrato com uma aplicação que apresenta uma maior complexidade dependeria das habilidades dos [Desenvolvedores](#) envolvidos.

8.5 Processo iterativo

O processo iterativo ocorreu no mês de Junho de 2022 e compreendeu 5 *Sprints* que foram realizadas a partir das *features* dispostas no Backlog do Domínio (Quadro 17). As *sprints* 2 e 3 aconteceram simultaneamente, porque as tarefas de documentar, organizar a árvore de *features*, atualizar o mapa de produtos e separar os artefatos legados foram feitas apenas pelo especialista em análise do domínio, fazendo com que as reuniões com os outros especialistas se limitasse a 2 ou 3 reuniões durante o tempo da *sprint*. Os gráficos *burndown* apresentados para cada uma das *sprints* apresenta a quantidade de tarefas pendentes a cada dia daquela *sprint*. É importante considerar que mais de uma atividade foi feita em um único dia.

Dependendo da similaridade das *features*, elas foram realizadas em uma mesma *Sprint*. Os nomes dos especialistas que participaram de cada uma das iterações foram omitidas. A aplicação “Aphtus” junto com a “Siqueira transportes”, “GDIS” e “Univita” possuem a maior fidelidade as *subfeatures* analisadas, pois essas possuem códigos legados. Por conta disso, as outras aplicações foram analisadas apenas por seus documentos de especificação de requisitos. Esses documentos apresentavam histórias de usuários e alguns mapas mentais que auxiliavam os desenvolvedores a mapear o sistema a ser criado. A documentação encontrada era modelada de maneira informal, sem apresentar nenhuma regra ou padronização de aquisição desses requisitos.

8.5.1 Backlog do Domínio

A construção do Backlog do Domínio (Quadro 17) é utilizada para determinar a ordem em que as *features* principais serão executadas em cada iteração do Scrum. A determinação dessa prioridade levou em consideração os quadros 15 e 16, assim como a ajuda do [Especialista do Domínio](#) e [Especialista do negócio](#). A prioridade alocada para cada uma *features* corresponde ao potencial de reuso observadas nas etapas anteriores. A estimativa de horas de trabalho foi definida pelo [Especialista do Domínio](#) e ela foi ajustada para a *feature* de “Modelagem dos dados”, uma vez que foram necessárias duas reuniões com os especialistas para validação dessa *Sprint*, enquanto as outras necessitaram apenas de uma reunião. No total, foram utilizadas 75 horas para realizar todas as tarefas de cada *Sprint*.

8.5.2 Backlog do Sprint

A decisão de realizar as tarefas de mais de uma *feature* em uma mesma *Sprint* foram do [Especialista do Domínio](#) e do [Especialista em análise do domínio](#), uma vez que ao realizar as reuniões de identificação, documentação e detalhamento das *subfeatures* de uma *feature* principal, acabava-se por passar por informações que compreendiam aspectos de outra *feature* principal presente no Backlog do Domínio. O método RISE-DA não contém nenhuma regra que especifica a separação de *features* principais pela sua similaridade, para que sejam executadas em uma mesma *Sprint*. Isso é interessante, pois é possível diminuir o tempo de trabalho ao conseguir catalogar *subfeatures* de *features* diferentes em uma mesma reunião de identificação.

Para as *Sprints* 1, 2, 4 e 5 foram realizadas 8 atividades principais. Antes de realizar a reunião de mapeamento das *subfeatures*, foi necessário separar os artefatos legados que possuíam as *features* da *Sprint* a serem realizadas. A primeira etapa realizada foi identificar as *subfeatures* a partir das *features* principais pelo [Especialista do Domínio](#), em que seriam analisados documentações, código legado e feito perguntas sobre as aplicações que ele desenvolveu. Dessa forma era possível realizar a segunda atividade proposta, ou seja, especificar e documentar as *subfeatures* catalogadas. Quando a documentação estava pronta, era possível organizar um diagrama hierárquico (árvore de *features* ou modelo de *features*), agrupando e decompondo essas *features* e *subfeatures*, podendo observar seus relacionamentos e dependências.

Para modelar essa árvore de *features*, foi utilizado a ferramenta FeatureIDE ([FEATU-REIDE, 2020](#)) e, portanto, uma das atividades era estudá-la para conseguir fazer a modelagem. Além disso, foi atualizado o mapa de produtos realizado anteriormente com as *subfeatures* catalogadas. Os artefatos criados nessa etapa da abordagem para a *Sprint* 1 estão presentes nos Quadros 18, 19, 20, 21, 22 e 23 e nas Figuras 16, 17 e 18. Os artefatos criados na *Sprint* 2 estão presentes nos Quadros 24, 25 e 26 e nas Figuras 19, 20, 21, 22 e 23. Na *Sprint* 3, os artefatos relacionados são os Quadros 27, 28, 29, 30, 31 e 32 e as Figuras 24, 25 e 26. Já a *Sprint* 4 possui os Quadros 33, 34, 35, 36, 37, 38 e 39 e as Figuras 27, 28, 29, 30 e 31. Por fim, a *Sprint* 5 possui os Quadros 40, 41, 42, 43, 44, 45 e 46 e as Figuras 32, 33, 34, 35 e 36.

Durante cada *Sprint*, algumas observações foram feitas, uma vez que os artefatos da árvore de *features*, mapa de produtos e o estudo da ferramenta FeatureIDE estariam em andamento até a realização da *Sprint 5*. Todas as reuniões utilizadas para realizar as atividades foram gravadas para melhor análise dos dados catalogados ao criar os artefatos necessários. Todos os artefatos criados foram estruturados e organizados pelo [Especialista em análise do domínio](#).

A *Sprint 3* não possui a atividade de especificar e documentar as *subfeatures*, uma vez que os papéis de [Especialista do Domínio](#) e [Especialista em análise do domínio](#) foram realizadas pelo presidente da empresa e, portanto, a documentação não foi necessária. Entretanto, foi realizada uma reunião de validação dos artefatos criados com os outros membros da empresa.

No *link* apresentado no referencial de [Amanda \(2022\)](#) desse trabalho, será possível observar as relações de dependência entre todas as *features* e *subfeatures* criadas para cada uma das 14 aplicações (separadas em Web, Mobile e Híbrida), com base nos quadros criados e na árvore de *features*.

Cada um dos *pdf's* contém uma hierarquia de *features*, em que aparecem seus relacionamentos ao lado de cada uma. A obrigatoriedade de uma *feature* sobre outra depende do relacionamento entre elas na árvore de *features*. A bolinha mais escura representa *features* ou *subfeatures* obrigatórias. Bolinhas claras *features* ou *subfeatures* opcionais. O símbolo $\wedge$ fechado e pintado significa o relacionamento lógico OR entre as *features* ou *subfeatures* de mesmo nível. Já o símbolo $\wedge$ fechado e em branco, significa o relacionamento XOR entre as *features* ou *subfeatures* de mesmo nível.

Além disso, as marcações em cruz na cor verde clara, significam que aquela *feature* ou *subfeature* está diretamente relacionada com a aplicação em questão. Por conta disso, as marcações em cruz na cor verde escura representam as dependências daquela *feature* ou *subfeature* selecionada, ou seja, ao tentar implementá-las, será obrigatório considerar o restante da hierarquia. Já as *features* ou *subfeatures* que estão em branco não pertencem a aplicação.

8.5.2.1 *Sprint 1*

A *Sprint 1* foi executada entre os dias 3 de junho de 2022 à 10 de junho de 2022. No (Quadro 18) é apresentado o Backlog da *Sprint 1*, em que são definidos os objetivos, a duração, as tarefas, os especialistas envolvidos e quais são as *features* principais selecionadas. As *features* principais realizadas foram as de “CRUD” e “Controle de acesso dos usuários”, totalizando 15 horas de trabalho nas atividades propostas. A reunião de identificação das *subfeatures* levou 1 hora e 14 minutos, já a reunião de validação dos artefatos criados levou 40 minutos. O gráfico *burndown* apresentado na Figura 32 representa os dias de trabalho pelas tarefas realizadas.

As atividades de Sexta, Segunda e Quarta ficaram incompletas, entretanto, isso já era esperado, uma vez que todas as tarefas que precisavam ser realizadas durante o dia se repetiram durante vários dias no período proposto.

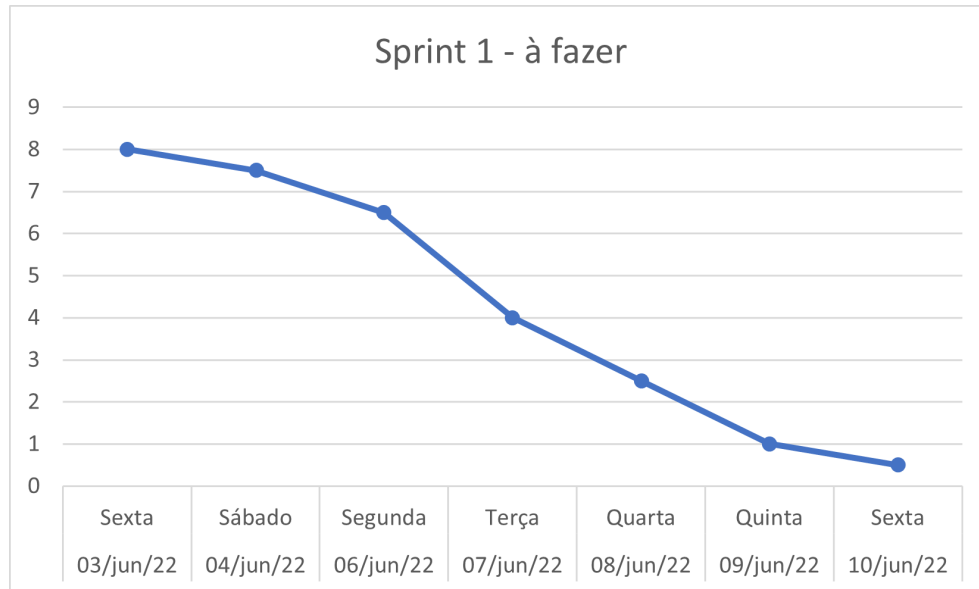
Quadro 17 – Utilização do Backlog do Domínio para separar as *features* de maior prioridade dentro do domínio de sistema de gestão de informações.

Backlog do domínio			
Domínio	<i>Features</i> principais	Prioridade	Estimativa (horas)
Sistema de gestão de informações	CRUD	Muito alto	8
	Controle de acesso dos usuários	Muito alto	7
	Modelagem dos dados	Muito alto	16
	Experiência do usuário	Muito alto	8
	Páginas informativas	Muito alto	8
	Criação de formulários	Médio	3
	Visualização de relatórios	Médio	3
	Sistema de notificação	Muito baixo	3
	Download de informações	Muito baixo	3
	Atendimento ao cliente	Baixo	3
	Envio de currículo	Muito baixo	3
	Gerência de compra e venda	Muito baixo	5
	Controle de caixa	Muito baixo	5

Quadro 18 – Backlog da *Sprint 1* para as *features* de “Controle de acesso do usuário” e “CRUD”.

Sprint 1		Features principais selecionadas	Tarefas	Proprietário	Status	Observação
Objetivo	Analisar as <i>features</i> principais de controle de acesso dos usuários e CRUD	Controle de acesso dos usuários e CRUD	Identificar as <i>subfeatures</i> a partir do especialista de domínio	Todos	Feito	
Duração	15 horas		Especificar e documentar as <i>subfeatures</i>		Feito	
Data de início	03/jun/22		Organizar diagrama hierárquico, agrupando e decompondo <i>features</i> (modelo de <i>features</i>)		Feito	Poderá ser continuado nas próximas Sprints
Data de encerramento	10/jun/22		Organizar as <i>subfeatures</i> no mapa de produtos (atualizar mapa de produtos)		Feito	Poderá ser continuado nas próximas Sprints
Data de revisão	10/jun/22		Inspecionar artefatos legados		Feito	
Artefatos prontos	Mapa de aplicações, Avaliação das <i>features</i> principais, Lista de aplicações e Lista do domínio de <i>features</i>		Separar artefatos legados reusáveis		Feito	
Local/hora das reuniões diárias	UNIOESTE, bloco F, sala 7 às 17h		Validar <i>features</i> do backlog de domínio e a <i>Sprint</i> atual		Feito	
Scrum master, Especialista em análise do domínio	Amanda Israel Graeff Borges		Estudar ferramenta para modelar <i>features</i> (FeatureIDE)		A fazer	Poderá ser continuado nas próximas Sprints
Especialista do domínio	1 especialista					
Integrantes da equipe	1 integrante					

Figura 16 – Gráfico *burndown* da *Sprint 1* representando os dias trabalhados pela quantidade de tarefas realizadas.



Cada um dos artefatos criados na *Sprint 1* estão presentes nos quadros 19, 20, 21 e 22 e nas figuras 17 e 18.

Considerando a *feature* principal “CRUD”, é possível observar nos quadros 19 e 20 que as aplicações possuem muitas correspondências entre as *subfeatures* detalhadas durante o processo incremental. As aplicações “Apthus”, “Siqueira transportes”, “PrimorLab”, “Positiva Engetech”, “Hyperion”, “G4”, “CQD”, “Univita”, “Hyperion”, “CQD mobile” e “Univita mobile” possuem relação com todas as *subfeatures*, exceto as *subfeature* “Drive de conexão entre SGBD e código” pertencente a *feature* “Script de conexão com o banco de dados”. Isso acontece, porque algumas aplicações utilizam bibliotecas que disponibilizam as implementações de banco de dados e, portanto, a conexão de um SGBD com a aplicação não se faz necessária. Por outro lado, como algumas aplicações não possuem código legado, não foi inferido que elas possuíam ou não essa característica. Como não foram todas as aplicações que tiveram correspondência com esta *feature* principal, não houve similaridades entre os produtos do domínio, mas pela quantidade significativa de correspondências, uma família de produtos de variabilidade pode ser criada.

Quadro 20 – Segunda parte do mapa de produtos atualizado para a *feature* principal “CRUD” e sua relação com o produto Híbrido e os produtos Mobile da empresa.

Domínio	Feature Principal	Mapa de Produtos					Produtos					
		Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Híbrido GDIS	Hyperion	Versão MOBILE			
Sistema de gestão de informações	CRUD	Criação da tabelas	Criação de atributos						X	X	X	
			Armazenar dados	Script de dados						X	X	X
			Atribuição de restrições	Chave primária						X	X	X
				Atributos únicos						X	X	X
				Relacionamento entre tabelas						X	X	X
				Tipos de atributos						X	X	X
		Tamanho de atributos						X	X	X		
		Funções de acesso	Permissão de usuário							X	X	X
			Especificar atributo(s)							X	X	X
			Especificar função SQL							X	X	X
			Especificar máscaras (data, string, quantidade de casas decimais) necessárias para inserção							X	X	X
			Tratamento de erro	status code 406	Tentativa de inserção chaves primárias/ únicos iguais					X	X	X
					Tentativa de update atributos chaves primárias/únicos iguais					X	X	X
				Status code 409	Tentativa de deleção de elementos vazios					X	X	X
	Status code 500			Erro ao encontrar o(s) dado(s)					X	X	X	
		Status code 400	Erro ao deletar					X	X	X		
			Erro imprevisto					X	X	X		
			Erro ao criar				Tentativa de update com atributos inválidos		X	X	X	
			Erro ao editar						X	X	X	
		Status code 403	Erro de permissão de usuário					X	X	X		
		Status code 201	Sucesso ao criar					X	X	X		
		Status code 204	Sucesso ao editar					X	X	X		
		Status code 200	Sucesso ao deletar					X	X	X		
		FindAll FindOne Create Delete Update		Sucesso ao encontrar o(s) dado(s)					X	X	X	
									X	X	X	
								X	X	X		
								X	X	X		
								X	X	X		
								X	X	X		
	Script de conexão com o banco de dados	Drive de conexão entre SGBD e código										
		Verificar conexão	Erro na conexão com o banco de dados						X	X	X	
					Sucesso na conexão com o banco de dados				X	X	X	

A construção da árvore de *features* para essa *feature* principal (Figura 17), levou em consideração uma *feature* abstrata chamada de “banco_de_dados” que incorpora em sua lógica a relação que a *feature* principal “CRUD” possui com outras *features* principais, uma vez que sua implementação é obrigatória ao tentar implementar outras *features* que necessitam de gestão de dados. É possível observar algumas *subfeatures* obrigatórias e opcionais presentes na estrutura. A *subfeature* obrigatória de “criacao_tabelas” possui a *subfeature* de “armazenas_dados” como opcional, pois algumas aplicações criam a lógica para os dados serem armazenados, mas o *script* de dados a serem inseridos serão feitos pelo próprio cliente. As outras *features* opcionais presentes, apenas referenciam diferentes possibilidades de implementações, uma vez que uma aplicação pode ou não precisar de uma máscara para os dados inseridos no banco de dados, por exemplo.

No *link* apresentado em [Amanda \(2022\)](#), será possível observar a relação entre as aplicações e a *feature* “CRUD”. Esse artefato possui a junção das informações presentes nos quadros e a árvore de *features* apresentados para essa *feature* principal.

Para a *feature* principal “Controle de acesso dos usuários”, é possível observar que nos quadros 21, 22 e 23 existem mais variações comparado a *feature* principal “CRUD”, mas não possuem também nenhuma similaridade entre os produtos do domínio.

No quadro 21 a aplicação “Aptus” possui correspondência com todas as *subfeatures* detalhadas, diferentemente das aplicações “Siqueira transportes”, “PrimorLap”, “Positiva Engetech” e “Hyperion”. As aplicações “Siqueira transportes” e “Positiva Engetech” possuem variação em 3 *subfeatures* e as aplicações “PrimorLap” e “Hyperion” possuem variação em 5 *subfeatures*.

No quadro 22, as aplicações “Projeto Engenharia” e “CONNECT4S1ITE” não possuem nenhuma correspondência com essa *feature* principal, enquanto as aplicações “G4” e “CQD” possuem 5 variações. A aplicação “Univita” possui 3 variações.

No quadro 23, a aplicação “GDIS” não possui nenhuma correspondência com essa *feature* principal, enquanto as aplicações “Hyperion” e “CQD” na versão mobile possuem 5 variações. A aplicação “Univita” na versão mobile possui 3 variações.

Ao não levar em consideração as aplicações que não possuem nenhuma correspondência com a *feature* de “Controle de acesso” (“Projeto Engenharia”, “CONNECT4S1ITE” e “GDIS”), é possível notar que existem correspondências entre todas as outras aplicações para 9 *subfeatures*, podendo gerar uma família de produtos significativa pra essas variabilidades. Essa *feature*, portanto, foi implementada em uma quantidade significativa de produtos e dessa forma é possível considerá-la para o reuso.

Quadro 21 – Primeira parte do mapa de produtos atualizado para a *feature* principal “Controle de acesso dos usuários” e sua relação com os produtos Web da empresa.

Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Mapa de Produtos				Produtos					
					Subfeatures 4	Subfeatures 5	Subfeatures 6	Subfeatures 7	Subfeatures 8	Versão WEB				
										Aptus	Siqueira transportes	PrimorLab	Positiva Engatech	Hyperion
Sistema de gestão de informações	Login	Autenticar se o usuário existe no banco de dados								X	X		X	
		Requisição do Cookie do browser do usuário para guardar o token com a autenticação do usuário logado	Verificar se o token criado é válido	Criação de um objeto criptografado (token) com a informação do ID do usuário ao logar						X	X			
			Deletar a senha informada para permanecer somente o ID do usuário no objeto criado ao logar no sistema							X				
			Utilizar variáveis de ambiente para não deixar o dado exposto (token)							X	X			
		Autenticar senha							X		X		X	
	Atualizar senha	Verificação da senha por meio de uma comparação entre as senhas criptografadas do banco e a fornecida pelo usuário									X	X		X
		Necessário confirmar duas vezes a nova senha ao atualizar a senha do usuário									X	X		X
		Validar usuário no banco de dados para atualizar a senha	Um token é mandado para o email do cliente por meio do protocolo SMTP							X				
		Requisição para criptografia de senha	Token aceito/ Token expirado	Status code 401						X	X	X	X	X
			Utilizar função hash								X	X	X	X
Restrição acesso dos usuários	Other token	Desloga o usuário e volta para a página de login	Token não existe	Excluir cookies da página				X						
			Token obtido é inválido						X					
			Permanece com o usuário logado	token é válido	Para todas as ações do sistema, é checado se o usuário possui permissão para realizá-las a partir de seu ID	Procurar os dados de permissão do usuário no banco de dados por meio das informações do usuário logado	permitir acesso do usuário a tela requisitada	X	X	X		X	X	

Quadro 22 – Segunda parte do mapa de produtos atualizado para a *feature* principal “Controle de acesso dos usuários” e sua relação com os produtos Web da empresa.

Mapa de Produtos																
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Subfeatures 6	Subfeatures 7	Subfeatures 8	Produtos						
										Projeto Engenharia	G4	CQD	CONECT4\$ITE	Univita		
Sistema de gestão de informações	Login	Autenticar se o usuário existe no banco de dados									X	X		X		
		Requisição do Cookie do browser do usuário para guardar o token com a autenticação do usuário logado	Verificar se o token criado é válido	Criação de um objeto criptografado (token) com a informação do ID do usuário ao logar											X	
			Deletar a senha informada para permanecer somente o ID do usuário no objeto criado ao logar no sistema													
			Utilizar variáveis de ambiente para não deixar o dado exposto (token)													
		Autenticar senha									X	X			X	
	Atualizar senha	Verificação da senha por meio de uma comparação entre as senhas criptografadas do banco e a fornecida pelo usuário										X	X		X	
		Necessário confirmar duas vezes a nova senha ao atualizar a senha do usuário														
		Validar usuário no banco de dados para atualizar a senha	Um token é mandado para o email do cliente por meio do protocolo SMTP													
		Requisição para criptografia de senha	Token aceito/Token expirado	Status code 401								X	X			X
			Geração do token por meio do ID do usuário										X	X		X
Restrição acesso dos usuários	Obter token			Utilizar função hash								X	X		X	
				Checa token obtido	Desloga o usuário e volta para a página de login	Token não existe	Excluir cookies da página									
						Token obtido é inválido							X	X		X
					Permanece com o usuário logado	token é válido	Para todas as ações do sistema, é checado se o usuário possui permissão para realizá-las a partir de seu ID	Procurar os dados de permissão do usuário no banco de dados por meio das informações do usuário logado	permitir acesso do usuário a tela requisitada		X	X		X		

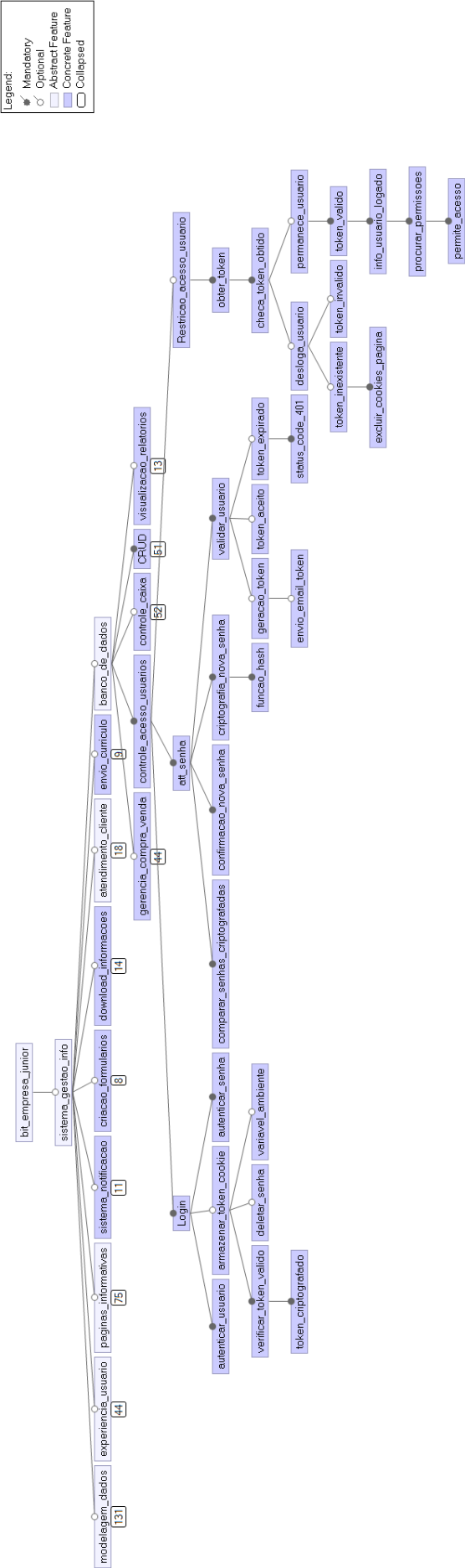
Quadro 23 – Terceira parte do mapa de produtos atualizado para a *feature* principal “Controle de acesso dos usuários” e sua relação com o produto Híbrido e os produtos Mobile da empresa.

Mapa de Produtos															
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Subfeatures 6	Subfeatures 7	Subfeatures 8	Produtos					
										Híbrido GDJS	Hyperion	Versão MOBILE Univita			
Sistema de gestão de informações	Login	Autenticar se o usuário existe no banco de dados									X	X	X		
		Requisição do Cookie do browser do usuário para guardar o token com a autenticação do usuário logado	Verificar se o token criado é válido	Criação de um objeto criptografado (token) com a informação do ID do usuário ao logar											
			Deletar a senha informada para permanecer somente o ID do usuário no objeto criado ao logar no sistema												
			Utilizar variáveis de ambiente para não deixar o dado exposto (token)										X		
	Atualizar senha	Autenticar senha										X	X	X	
		Verificação da senha por meio de uma comparação entre as senhas criptografadas do banco e a fornecida pelo usuário											X	X	X
		Necessário confirmar duas vezes a nova senha ao atualizar a senha do usuário											X	X	X
		Validar usuário no banco de dados para atualizar a senha	Um token é mandado para o email do cliente por meio do protocolo SMTP												
			Token aceito/Token expirado	Status code 401									X	X	X
	Restrição acesso dos usuários	Requisição para criptografia de senha	Utilizar função hash										X	X	X
		Other token	Checa token obtido	Desloga o usuário e volta para a página de login	Token não existe	Excluir cookies da página									
						Token obtido é inválido									
							Permanece com o usuário logado	token é válido	Para todas as ações do sistema, é checado se o usuário possui permissão para realizá-las a partir de seu ID	Procurar os dados de permissão do usuário no banco de dados por meio das informações do usuário logado	permitir acesso do usuário a tela requisitada		X	X	X

Na sua árvore de *features* (Figura 18), a *feature* principal de “Controle de acesso dos usuários” também é obrigatória, uma vez que as aplicações que possuem um banco de dados, automaticamente precisam de um controle de acesso, já que um sistema de gestão de informações de uma empresa necessita do controle de acesso às suas informações. Dessa forma, se faz necessário a obrigatoriedade de um processo de *login* e atualizações de senhas para os usuários.

A restrição de acesso dos usuários não se tornou obrigatória, pois dentre as possíveis aplicações da empresa, todos os usuários podem ter permissão de acessar qualquer parte do sistema. Essa *subfeature* faz essa checagem por meio de um *token* armazenado para esse usuário no momento do login. Dessa forma, é possível checar se esse *token* referencia as permissões de usuário corretas para a tela que está sendo acessada. Para uma aplicação que necessita dessa restrição, as *subfeatures* obrigatórias abaixo dessa serão acionadas para essa aplicação. Esses relacionamentos podem ser observados no *link* apresentado no referencial de [Amanda \(2022\)](#), pela *feature* de “Controle de acesso dos usuários” com todas as aplicações da empresa.

Figura 18 – Subárvore da árvore de *features* apresentando a *feature* principal “Controle de acesso dos usuários” e suas *subfeatures*

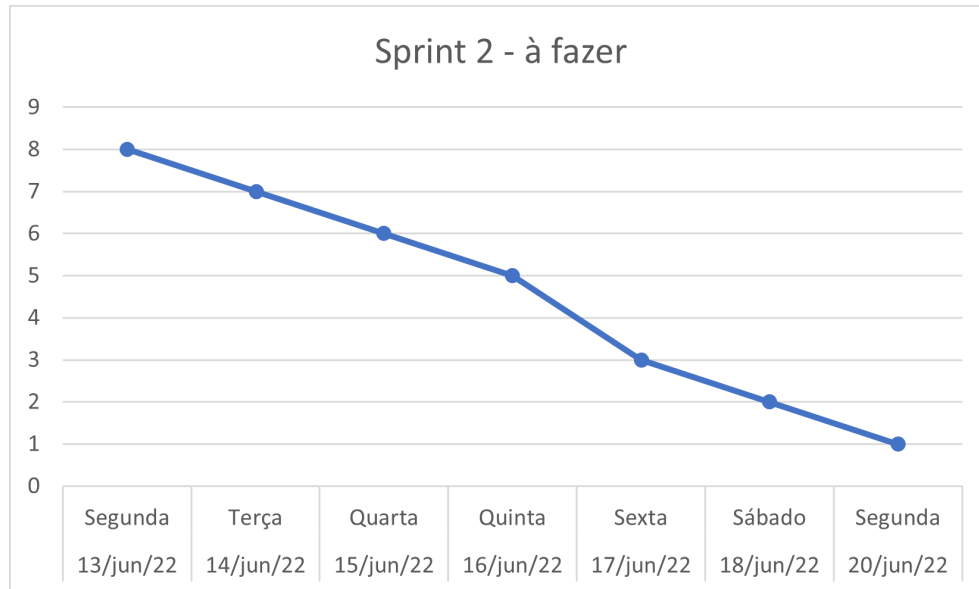


8.5.2.2 *Sprint 2*

A *Sprint 2* foi executada entre os dias 13 de junho de 2022 à 20 de junho de 2022. No (Quadro 24) é apresentado o Backlog da *Sprint 2*, em que são definidos os objetivos, a duração, as tarefas, os especialistas envolvidos e quais são as *features* principais selecionadas. As *features* principais realizadas foram as de “Visualização de relatórios”, “Sistema de notificação”, “Download de informações” e “Criação de formulários”, totalizando 12 horas de trabalho nas atividades propostas. A reunião de identificação das *subfeatures* levou 40 minutos, já a reunião de validação dos artefatos criados levou 50 minutos. O gráfico *burndown* apresentado na Figura 19 representa os dias de trabalho pelas tarefas realizadas.

As atividades que não foram realizadas de forma satisfatória durante os dias 13, 14 e 15 foram compensadas mais tarde, depois da validação de alguns artefatos. Todas as tarefas que precisavam ser realizadas durante o dia se repetiram durante vários dias no período proposto.

Figura 19 – Gráfico *burndown* da *Sprint 2* representando os dias trabalhados pela quantidade de tarefas realizadas.



Cada um dos artefatos criados na *Sprint 2* estão presentes nos quadros 25 e 26 e nas figuras 20, 21, 22 e 23.

No quadro 25 é possível observar que existem 5 entre as 14 aplicações (“Positiva Engetech”, “Projeto Engenharia”, “CQD”, “GDIS” e “CQD” mobile) que não possuem nenhuma correspondência entre as *features* analisadas. Não considerando essas aplicações, a *feature* principal que possui mais correspondências entre as aplicações é a de “Criação de formulários”, uma vez que existem mais entradas dentro dos sistemas do que saídas, como acontece em “Visualização de relatórios”, sendo essa *feature* a segunda com mais correspondências. Dessa forma, é possível criar uma família de produtos de variabilidade para essas duas *features* principais. Como observado nos quadros 13 e 14 a *feature* de “Sistema de notificações” e “Download de informações” continuam sem apresentar correspondências significativas entre as aplicações, apresentando uma grande variabilidade.

Quadro 26 – Segunda parte do mapa de produtos atualizado para as *features* principais “Sistema de notificação”, “Visualização de relatórios”, “Criação de formulários” e “Download de informações” e sua relação com o produto Híbrido e os produtos Mobile.

Domínio	Feature Principal	Mapa de Produtos					Produtos		
		Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Híbrido	Versão MOBILE	
							GDIS	Hyperion	CQD Univita
Sistema de gestão de informações	Sistema de notificação	Disparador de notificação	Corpo da mensagem	Input da tela					
			Remetente						
			Destinatário						
		Contratar servidor SMTP	Assunto						
			Padronização de informações						
	Visualização de relatórios	Tela de visualização de relatórios	Integridade						
			Sigilo						
			Segurança						
			Lista de relatórios	Data da última edição					
				Dados do relatório referente as informações desejadas	atributos			X	X
			pesquisa relatórios	find_all				X	X
				Barra de pesquisa					
				Botão pesquisa					
			selecionar relatório	Requisição HTTP como corpo da mensagem	find_by_corpo_msg			X	X
				Botão de visualizar	Inputs preenchidas com as informações			X	X
	Criação de formulários	Variáveis alocadas no front-end	Inputs		find_one			X	X
Download de informações	Recebimento das informações	Passagem das informações em um corpo de mensagem	Listener de mudança do input	Requisição HTTP	Chamada da API	Função create relatório		X	X
					PDF				
					Doc				
					Tabela				
					Arquivo de texto				
	Escolher as informações a ser baixada	Botão de download da informação	Escolha do formato do arquivo	Script de geração do relatório	Outro				
					Montar arquivo HTML	Documento padrão selecionado			
					Imagem				
					PDF com seleção	Pacotes de edição			

Ao observar a árvore de *features* para a *feature* de “Criação de formulários” (Figura 20), verifica-se que todas as *subfeatures* presentes são obrigatórias, apesar da *feature* principal em si ser apenas opcional. Isso acontece, pois para que seja possível a implementação dessa funcionalidade, todas as informações presentes na árvore precisam estar presentes. Portanto, para todas as aplicações que possuem a *feature* de “Criação de formulários”, ou seja, entradas no sistema, precisam apresentar essas *features*. Dessa forma, é possível compreender melhor como ocorreu as marcações nos quadros 25 e 26 pelas dependências de uma *feature*.

Para a árvore de *feature* da *feature* de “Download de informações” é possível notar que a criação dessa funcionalidade certamente depende de outra, mas aqui nós olhamos especificamente na funcionalidade a ser implementada. É obrigatório existir a escolha da informação a ser baixada, assim como a ação de baixar o arquivo, que nesse caso é por meio de um botão. Ao escolher o formato de um arquivo, é necessário um relacionamento baseado em XOR, uma vez que é preciso escolher um formato, mas não é possível escolher mais de um dos disponíveis. O *script* de geração de relatórios é obrigatório, mas a forma como essa *feature* é realizada, é de escolha do programador.

Já a árvore de *features* para a *feature* de “Sistema de notificação”, torna-se obrigatório um disparador de notificação que precisa de uma estrutura bem especificada, mas não a contratação de um serviço SMTP, esse sendo voltado especificamente a aplicações que precisam mandar notificações por email. Os outros modos de enviar notificações (SMS e por WhatsApp (INC, 2009)) não foram consideradas aqui, pois o conhecimento dos desenvolvedores não abrangia tais *subfeatures* e não havia código legado para ser analisado.

A árvore de *features* para a *feature* de “Visualização de relatórios” pertence a *feature* abstrata de “banco_de_dados”, pois ela possui algumas chamadas ao banco de dados da aplicação. Uma dessas chamadas ocorre ao tentar acionar uma pesquisa de informações do sistema, necessitando, portanto, da função “find_by_corpo_mensagem”. Essa pesquisa, entretanto, é opcional. Isso acontece também ao tentar buscar um relatório específico ou acionar a tela de visualização de relatórios, fazendo com que tenha que ser implementada as funções de “find_all” ou “find_one”. Para poder observar essas dependências nas aplicações da empresa, consulte o *link* apresentado no referencial de Amanda (2022).

Figura 20 – Subárvore da árvore de *features* apresentando a *feature* principal “Criação de formulários” e suas *subfeatures*

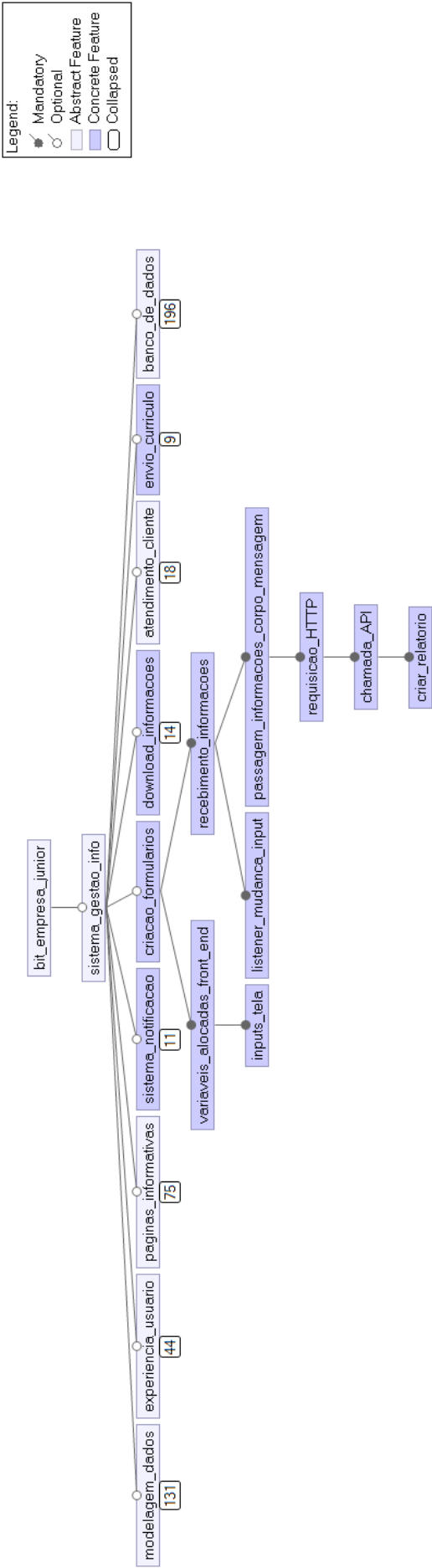


Figura 21 – Subárvore da árvore de *features* apresentando a *feature* principal “Download de informações” e suas *subfeatures*

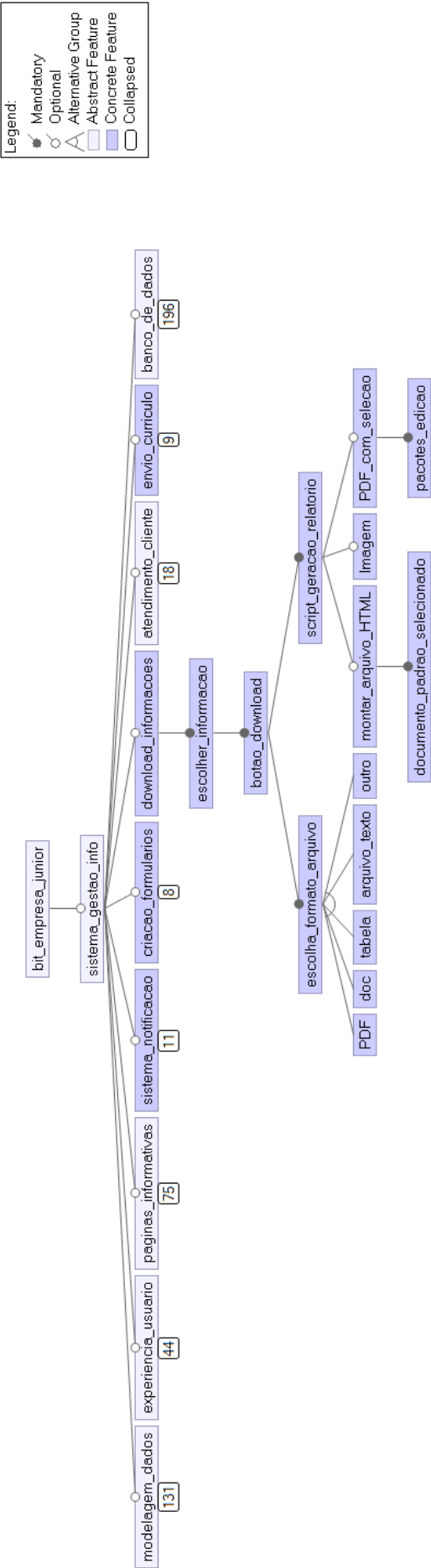
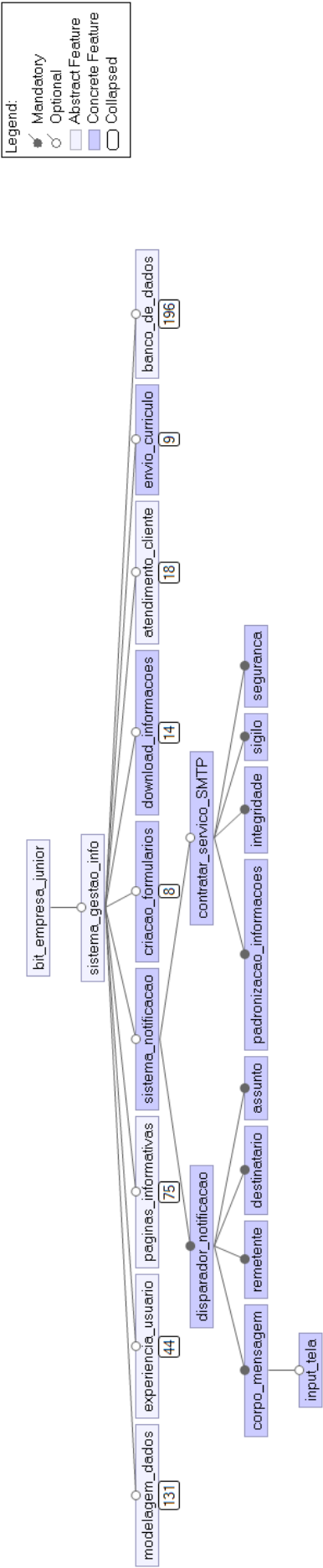


Figura 22 – Subárvore da árvore de *features* apresentando a *feature* principal “Sistema de notificação” e suas *subfeatures*



8.5.2.3 *Sprint 3*

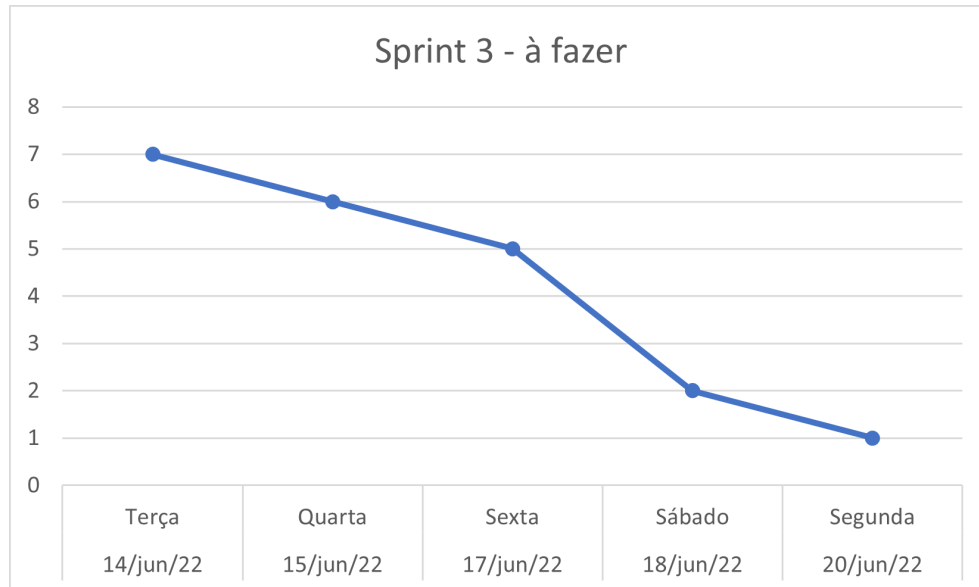
A *Sprint 3* foi executada entre os dias 13 de junho de 2022 à 20 de junho de 2022. No (Quadro 27) é apresentado o Backlog da *Sprint 3*, em que são definidos os objetivos, a duração, as tarefas, os especialistas envolvidos e quais são as *features* principais selecionadas. As *features* principais realizadas foram as de “Páginas Informativas” e “Experiência do usuário”, totalizando 16 horas de trabalho nas atividades propostas. A reunião de identificação das *subfeatures* levou 1 hora e 40 min, já a reunião de validação dos artefatos criados levou 30 minutos. O gráfico *burndown* apresentado na Figura 24 representa os dias de trabalho pelas tarefas realizadas.

As atividades de documentação e separação dos artefatos legados não foram realizadas de forma satisfatória durante os dias 14 e 15, pois levaram mais tempo que o esperado. Entretanto, ao validar com os membros da empresa, foi possível construir a árvore de *features* e o mapa de produtos mais rapidamente. Todas as tarefas que precisavam ser realizadas durante o dia se repetiram durante vários dias no período proposto.

Quadro 27 – Backlog da *Sprint 3* para as *features* de “Páginas informativas” e “Experiência do usuário”.

Sprint 3		Features principais selecionadas	Tarefas	Proprietário	Status	Observação
Objetivo	Analisar as features principais de páginas informativas e experiência do usuário	Páginas informativas e experiência do usuário	Identificar as <i>subfeatures</i> a partir do especialista de domínio	Todos	Feito	
Duração	16 horas		Organizar diagrama hierárquico, agrupando e decompondo features (modelo de features)		Feito	Poderá ser continuado nas próximas <i>Sprints</i>
Data de início	13/jun/22		Organizar as <i>subfeatures</i> no mapa de produtos (atualizar mapa de produtos)		Feito	Poderá ser continuado nas próximas <i>Sprints</i>
Data de encerramento	20/jun/22		Inspecionar artefatos legados		Feito	
Data de revisão	20/jun/22		Separar artefatos legados reusáveis		Feito	
Artefatos prontos	Mapa de aplicações, Avaliação das features principais, Lista de aplicações e Lista do domínio de features		Estudar ferramenta para modelar features (FeatureIDE)		À fazer	Poderá ser continuado nas próximas <i>Sprints</i>
	UNIOESTE, bloco F, sala 7 às 17h		Validar as <i>Sprints</i>		Feito	
Scrum master, Especialista em análise do domínio	Amanda Israel Graeff Borges					
Especialista do domínio	Amanda Israel Graeff Borges					
Integrantes da equipe	2 integrantes					

Figura 24 – Gráfico *burndown* da *Sprint* 3 representando os dias trabalhados pela quantidade de tarefas realizadas.



Cada um dos artefatos criados na *Sprint* 3 estão presentes nos quadros 28, 29, 30, 31, e 32 e nas figuras 25 e 26.

Nos quadros 28 e 29 já é possível observar que existem similaridades entre uma das *subfeatures* com todas as aplicações. Essa *subfeature* é chamada de “Estrutura”, e essa possui as *subfeatures* “Definição do posicionamento dos objetos” e “Conteúdo”. Isso acontece, porque todas as aplicações precisam ter alguma estrutura, seja ela referente a uma página institucional ou a uma página interna para os funcionários que realizam a gestão de informações. Portanto, estruturar as páginas de um sistema de forma padronizada pode ser vantajoso e por isso, criar uma família de produtos de similaridade para essa *subfeature* pode ser de interesse da empresa.

Considerando agora as aplicações que possuem variabilidade, é necessário apontar que a “Responsividade” não está presente em todas as aplicações pela falta de certeza de sua utilização, uma vez que não haviam códigos legados ou documentações descrevendo a presença dessa *subfeature* em todas as aplicações. Entretanto, ao levar em consideração a quantidade de aplicações que possuem essa característica (“Apthus”, “Siqueira transportes”, “PrimorLab”, “Positiva Engetech”, “Projeto Engenharia”, “CONNECT4SITE”, “Univita”, “GDIS” e “Univita mobile”), demonstra que essa *subfeature* é fundamental para a experiência dos usuários da empresa BIT e portanto, a criação de uma família de produtos ao considerar também a “Estrutura” e a “Consistência e padrão” demonstra o potencial de reuso dessas *subfeatures*.

Quadro 28 – Primeira parte do mapa de produtos atualizado para a *feature* principal “Experiência do usuário” e sua relação com os produtos Web.

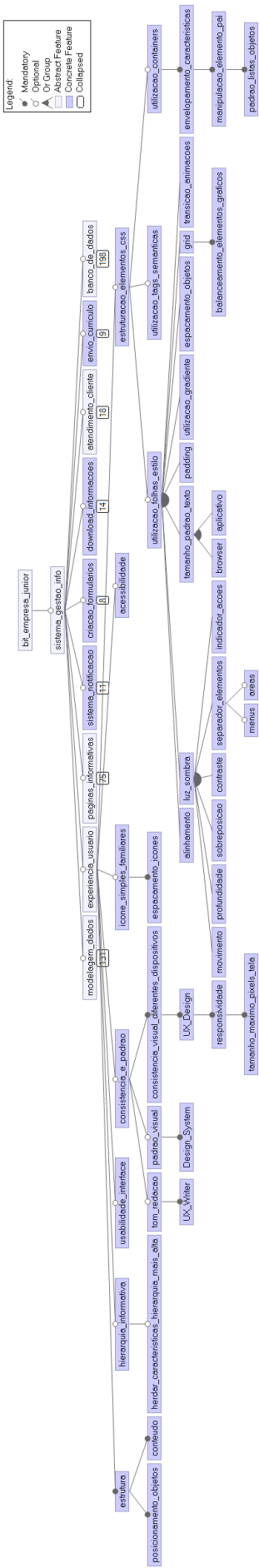
Domínio	Feature Principal	Mapa de Produtos					Produtos										
		Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Versão WEB										
							Aplhus	Siqueira Transportes	PrimorLab	Positiva Enggetech	Hyperion	Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita	
Sistema de gestão de informações	Experiência do usuário	Estrutura	Definição do posicionamento dos objetos				X	X	X	X	X	X	X	X	X	X	
			Conteúdo														
			Herdar características da hierarquia mais alta														
			Usabilidade da interface														
		Consistência e padrão	Tom de redação	UX Writer													
			Padrão visual	Design System					X								
			Consistência visual em diferentes dispositivos	UX Design	Responsividade	Tamanho máximo de pixels na tela	X	X	X	X		X			X	X	
			Espaçamento de ícones				X	X		X		X				X	
		Acessibilidade	Utilização de tags semânticas														
			Utilização de containers	Envolvimento de características	Manipulação diretamente do elemento pai	Padrão de listas de objetos	X	X		X		X				X	
			Utilização de folhas de estilo	Grid	Balanceamento dos elementos gráficos		X	X		X		X				X	
				Alinhamento				X	X		X		X				X
				Espaçamento de objetos				X	X		X		X			X	
					Movimento												X
Estruturação de elementos css	Luz e sombra	Profundidade															
		Sobreposição															
		Contraste															
		Separador de elementos	Menus importantes	X	X												
		Indicador de ações															
		Browser															
		Aplicativo															
		Tamanho padrão de texto															
	Utilização de gradiente																
	Padding																
Transição de animações																	

Quadro 29 – Segunda parte do mapa de produtos atualizado para a *feature* principal “Experiência do usuário” e sua relação com o produto Híbrido e os produtos Mobile.

Domínio	Feature Principal	Mapa de Produtos					Produtos			
		Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Híbrido	Versão MOBILE		
								GDJS	Hyperion	CQD
Sistema de gestão de informações	Experiência do usuário	Estrutura	Definição do posicionamento dos objetos					X	X	X
								X	X	X
			Herdar características da hierarquia mais alta					X		X
								X		X
								X		X
		Consistência e padrão	Tom de redação	UX Writer	Responsividade	Tamanho máximo de pixels na tela		X		X
			Padrão visual	Design System				X		X
			Consistência visual em diferentes dispositivos	UX Design				X		X
		Ícones simples e familiares	Espaçamento de ícones					X		X
		Estruturação de elementos css	Acessibilidade							
			Utilização de tags semânticas					X		X
			Utilização de containers	Envolvimento de características	Manipulação diretamente do elemento pai	Padrão de listas de objetos		X		X
								X		X
			Grid	Alinhamento	Balanceamento dos elementos gráficos			X		X
								X		X
			Espaçamento de objetos					X		X
			Luz e sombra		Movimento					
			Sobreposição	Contraste	Separador de elementos	Menus Áreas importantes		X	X	
								X		
			Indicador de ações	Browser	Aplicativo			X		X
								X		X
			Tamanho padrão de texto	Utilização de gradiente						
			Padding							
			Transição de animações					X		X

Na figura 25, já é possível observar que a *subfeature* “estrutura”, assim como suas *subfeatures*, são obrigatórias, uma vez que elas estão presentes em todas as aplicações. As outras *subfeatures*, são opcionais, mas ao ser selecionada por uma aplicação, é necessário que algumas *subfeatures* mais abaixo na hierarquia sejam obrigatórias (para ver mais detalhes dessas dependências, visualize o *link* apresentado no referencial de [Amanda \(2022\)](#)). A estruturação dos elementos em CSS traz algumas características interessantes presentes nas aplicações, por exemplo, a utilização de folhas de estilo. Entretanto nem todas são necessárias, e por isso, a relação OR é utilizada para representar a liberdade de escolha pelo programador ao querer utilizar um gradiente, *grids* ou apenas um *padding*.

Figura 25 – Subárvore da árvore de *features* apresentando a *feature* principal “Experiência do usuário” e suas *subfeatures*



As páginas informativas (Quadros 30, 31 e 32), por outro lado, não possuem similaridades entre os produtos do domínio. Essa *feature* principal considera principalmente aplicações que são voltadas para páginas informativas. Entretanto, as variabilidades entre essas aplicações com as outras *features*, conseguem trazer um número interessante de correspondências entre si. Considerando as *subfeatures* de primeiro nível, as que possuem mais correspondências entre todas as aplicações são as de “Design”, “Estilização”, “Criação de rotas” e “Script de dados”.

Ao observar as aplicações que possuem páginas informativas (“GDIS”, “Siqueira transportes”, “PrimorLab”, “Positiva Engetech”, “Projeto Engenharia”, “CQD”, “CONNECT4SITE” e “CQD mobile”), diferentemente das outras aplicações, elas têm correspondência com a *subfeature* “Layout”, uma vez que possuem listas de itens em destaque ou descrições importantes sobre os ativos oferecidos ou sobre as entidades representativas do software. Dentre as aplicações voltadas as páginas informativas, a aplicação “GDIS” é a que possui mais correspondências entre as *subfeatures*.

Na árvore de *features* para a *feature* de “Páginas informativas” (Figura 26) é possível observar que a *subfeature* “estilização” é obrigatória, entretanto suas *subfeatures* são opcionais, com exceção da “definicao_titulo”. Todas as outras *subfeatures* são opcionais, porque para formar uma página informativa pode existir muitas configurações diferentes para diferentes aplicações e, portanto, qualquer *subfeature* caracterizada como obrigatória seria uma restrição que não caberia em todas. Um dos exemplos disso, é a *feature* de criação de rotas, uma vez que existem páginas informativas que não necessitam de mais de uma página de apresentação. As otimizações e boas práticas apresentadas, também são opcionais, uma vez que podem ser utilizados *plugins* que já fornecem esse tipo de *feature* e, portanto, o programador não precisaria se preocupar com *backup* e *restore*, por exemplo.

Quadro 30 – Primeira parte do mapa de produtos atualizado para a *feature* principal “Páginas informativas” e sua relação com a primeira parte dos produtos Web.

Mapa de Produtos															
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos								
							Versão WEB								
							Apthus	Siqueira transportes	PrimorLab	Positiva Engetech	Hyperion				
Sistema de gestão de informações	Páginas informativas	Design	Ferramentas de Design	Criação de imagens	Diferentes tamanhos	Logo mobile									
						Logo web		X		X					
						Banner mobile									
						Banner web									
				Outras imagens de destaque			X	X							
		Estilização	Definição de cores	Edição de imagens	Mudança de escala			X	X	X					
						Mudança de resolução		X	X	X					
				Definição de fontes	Definição de cores	Fundo web	Fundo mobile	Links							
		Definição subtítulo													
		Layout	Barra superior	Menu	Mobile	drop down									
							Web	Horizontal	X						
								Vertical							
								Fixo	X						
								Ancora flutuante							
				Banner	Logo			X	X	X					
			Barra inferior	Direitos autorais	Copyright da entidade responsável pelo desenvolvimento	Definição da entidade representativa do site		X	X	X					
							Informações adicionais	Sobre o site	Contribuintes		X	X	X		
			Página de apresentação	Informações dos responsáveis pelos dados presentes do site	Especificação sobre os dados informativos presentes no site			X	X	X					
								X		X					
								X	X	X					
								X							
							X		X						
		Formulação das páginas internas	Criação da estrutura	Janelas informativas			X	X	X						
				Listas de quadros informativos			X								
				Portfólio	Lista de itens de interesse em destaque		X		X						
				Submenus			X								
		Saiba mais	Lista de páginas associadas												
		Criação de rotas	Botões				X	X	X	X					
			URLs					X	X	X	X				
		Script de dados	Fornecido pelo cliente					X	X	X	X				
			Adicionar em caixas de texto						X	X	X	X			
		Registrar um domínio							X						
		Provedor de hospedagem							X						
		Otimizações	Motor de busca (_SEO)	Google Analytics											
			Caching												
			Proteção contra Spam												
		Manual do usuário													
		Atualização de plugins													
		Backup e restore													
		Certificado SSL													

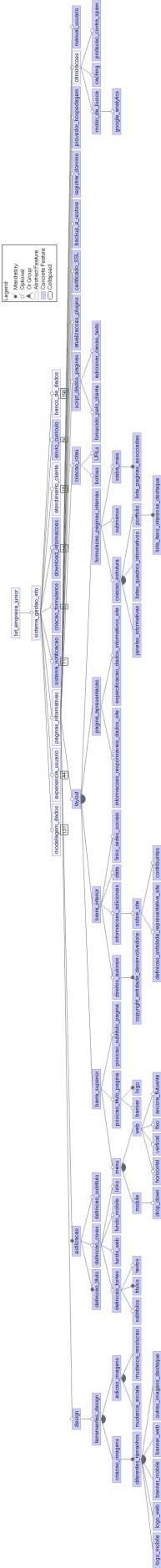
Quadro 31 – Segunda parte do mapa de produtos atualizado para a *feature* principal “Páginas informativas” e sua relação com a segunda parte dos produtos Web.

Mapa de Produtos													
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos						
							Versão WEB						
							Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita		
Sistema de gestão de informações	Páginas informativas	Design	Ferramentas de Design	Criação de imagens	Diferentes tamanhos	Logo mobile							
						Logo web	X		X	X	X		
						Banner mobile							
						Banner web							
				Outras imagens de destaque	X		X	X	X				
				Edição de imagens	Mudança de escala		X		X	X	X		
		Mudança de resolução	X				X	X	X				
		Estilização	Definição do título						X		X	X	X
				Definição de fontes	Cor dos Subtítulos								
						Títulos	X		X	X	X	X	
						Textos	X		X	X	X	X	
					Fundo web	X		X	X	X	X		
					Fundo mobile								
					Links								
				Definição subtítulo									
		Layout	Barra superior	Menu	Mobile	drop down							
						Horizontal							
					Web	Vertical							
						Fixo							
						Ancora flutuante							
						Posição Título da página	Banner						
			Posição Subtítulo da página	Logo		X		X	X	X	X		
			Barra inferior	Direitos autorais	Copyright da entidade responsável pelo desenvolvimento		X		X	X			
						Informações adicionais	Sobre o site	Definição da entidade representativa do site	X		X	X	
				Contribuintes				X	X				
				Data									
				Links para redes sociais									
				Página de apresentação	Informações dos responsáveis pelos dados presentes do site		X		X	X			
			Especificação sobre os dados informativos presentes no site					X	X				
			Formulação das páginas internas		Criação da estrutura	Janelas informativas		X		X	X		
						Listas de quadros informativos							
				Portfólio	Lista de itens de interesse em destaque	X		X	X				
				Submenus									
			Saiba mais	Lista de páginas associadas									
			Criação de rotas	Botões					X		X	X	
		URLs						X		X	X		
		Script de dados	Fornecido pelo cliente					X		X	X		
			Adicionar em caixas de texto					X		X	X		
		Registrar um domínio											
		Provedor de hospedagem											
		Otimizações	Motor de busca (SEO)	Google Analytics									
			Caching										
			Proteção contra Spam										
		Manual do usuário											
		Atualização de plugins											
		Backup e restore											
		Certificado SSL											

Quadro 32 – Terceira parte do mapa de produtos atualizado para a *feature* principal “Páginas informativas” e sua relação com o produto Híbrido e os produtos Mobile.

Mapa de Produtos											
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos				
							Híbrido	Versão MOBILE			
							GDIS	Hyperion	CQD	Univita	
Sistema de gestão de informações	Páginas informativas	Design	Ferramentas de Design	Criação de imagens	Diferentes tamanhos	Logo mobile	X		X	X	
						Logo web	X		X		
						Banner mobile	X				
						Banner web	X				
				Outras imagens de destaque	X		X	X			
					X		X	X			
		Estilização	Definição do título					X		X	X
				Definição de cores	Definição de fontes	Cor dos Subtítulos		X			
			Títulos			X		X			
			Textos			X		X			
			Fundo web			X					
			Fundo mobile		X		X	X			
			Links		X						
			Definição subtítulo				X				
		Layout	Barra superior	Menu	Mobile	drop down	X				X
						Horizontal	X				
					Web	Vertical					
						Fixo					
						Âncora flutuante					
						Posição Título da página	Banner	X			
				Posição Subtítulo da página	Logo	X		X	X		
						X					
			Barra inferior	Direitos autorais	Copyright da entidade responsável pelo desenvolvimento		X		X		
				Informações adicionais	Sobre o site	Definição da entidade representativa do site	X		X		
						Contribuintes	X		X		
				Data			X				
				Links para redes sociais							
			Página de apresentação	Informações dos responsáveis pelos dados presentes do site			X		X		
					Especificação sobre os dados informativos presentes no site		X		X		
				Formulação das páginas internas	Criação da estrutura	Janelas informativas		X		X	
						Listas de quadros informativos		X			
			Portfólio		Lista de itens de interesse em destaque	X					
			Submenus								
		Saiba mais	Lista de páginas associadas		X						
		Criação de rotas	Botões				X		X		
			URLs				X		X		
		Script de dados	Fornecido pelo cliente					X		X	
			Adicionar em caixas de texto					X		X	
		Registrar um domínio						X			
		Provedor de hospedagem						X			
		Otimizações	Motor de busca (_SEO)	Google Analytics				X			
			Caching					X			
			Proteção contra Spam					X			
		Manual do usuário						X			
		Atualização de plugins						X			
Backup e restore						X					
Certificado SSL						X					

Figura 26 – Subárvore da árvore de *features* apresentando a *feature* principal “Páginas informativas” e suas *subfeatures*



8.5.2.4 *Sprint 4*

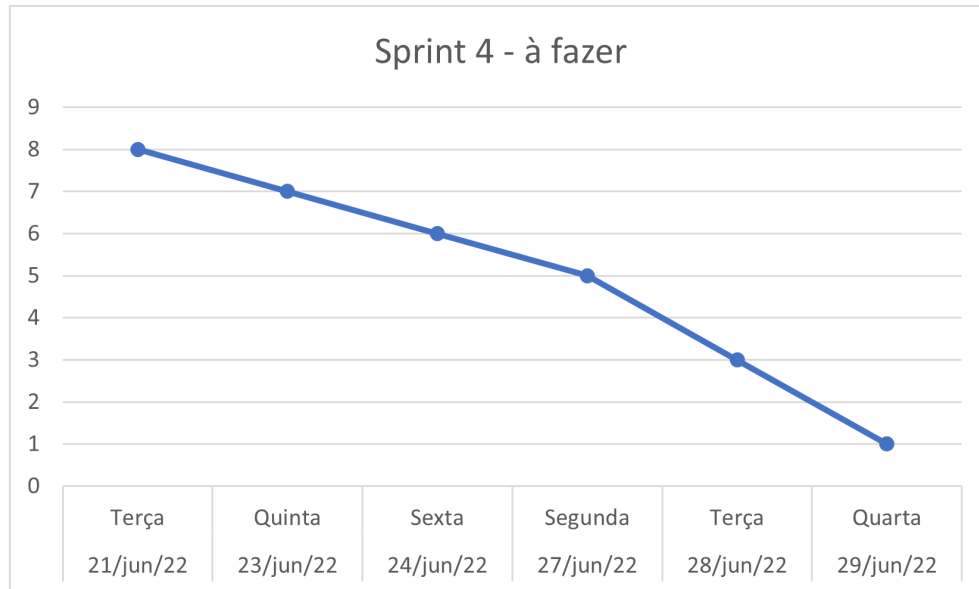
A *Sprint 4* foi executada entre os dias 21 de junho de 2022 à 28 de junho de 2022. No (Quadro 33) é apresentado o Backlog da *Sprint 4*, em que são definidos os objetivos, a duração, as tarefas, os especialistas envolvidos e quais são as *features* principais selecionadas. A *feature* principal realizada foi a de “Modelagem de dados”, totalizando 16 horas de trabalho nas atividades propostas. A reunião de identificação das *subfeatures* levou 43 min, já a reunião de validação dos artefatos criados levou 30 minutos. O gráfico *burndown* apresentado na Figura 27 representa os dias de trabalho pelas tarefas realizadas.

As atividades de identificação e documentação das *subfeatures* levaram mais tempo que o esperado, uma vez que não havia disponibilidade dos especialistas do domínio no começo da *Sprint*. Entretanto, ao executar essas tarefas com os especialistas, foi possível construir a árvore de *features* e o mapa de produtos mais rapidamente. Todas as tarefas que precisavam ser realizadas durante o dia se repetiram durante vários dias no período proposto.

Quadro 33 – Backlog da *Sprint 4* para a *feature* de “Modelagem de dados”.

Sprint 4		Features principais selecionadas	Tarefas	Proprietário	Status	Observação
Objetivo	Analisar as features principais de modelagem de dados	Modelagem de dados	Identificar as <i>subfeatures</i> a partir do especialista de domínio	Todos	Feito	
Duração	16 horas		Especificar e documentar as <i>subfeatures</i>		Feito	
Data de início	21/jun/22		Organizar diagrama hierárquico, agrupando e decompondo features (modelo de features)		Feito	Poderá ser continuado nas próximas <i>Sprints</i>
Data de encerramento	28/jun/22		Organizar as <i>subfeatures</i> no mapa de produtos (atualizar mapa de produtos)		Feito	Poderá ser continuado nas próximas <i>Sprints</i>
Data de revisão	28/jun/22		Inspecionar artefatos legados		Feito	
Artefatos prontos	Mapa de aplicações, Avaliação das features principais, Lista de aplicações e Lista do domínio de features		Separar artefatos legados reusáveis		Feito	
Local/hora das reuniões diárias	UNIOESTE, bloco F, sala 7 às 17h		Estudar ferramenta para modelar features (FeatureIDE)		À fazer	Poderá ser continuado nas próximas <i>Sprints</i>
Scrum master, Especialista em análise do domínio	Amanda Israel Graeff Borges		Validar as <i>Sprints</i>		Feito	
Especialista do domínio	2 especialistas					
Integrantes da equipe	-					

Figura 27 – Gráfico *burndown* da *Sprint* 4 representando os dias trabalhados pela quantidade de tarefas realizadas.



Cada um dos artefatos criados na *Sprint* 4 estão presentes nos quadros [34](#), [35](#), [36](#), [37](#), [38](#) e [39](#) e nas figuras [28](#), [29](#), [30](#) e [31](#).

Nos quadros [34](#), [36](#) e [38](#) é possível observar que existe similaridade entre os produtos do domínio para a *subfeature* “Papéis do projeto”, exceto para a *subfeature* “Back-end”, pois existem projetos que não possuem essa categoria de membros. Além disso, todos os quadros com suas *features* e *subfeatures* possuem variabilidades. O maior número de correspondências entre as *features* e as aplicações que possuem variabilidades são as *subfeatures* de “Entrevistas”, em que a maior parte das entrevistas documentadas pela empresa foram feitas informalmente e com perguntas abertas, a *subfeature* de “Levantamento de requisitos utilizando o Scrum”, pois a maioria dos projetos utilizam a metodologia Scrum, a *subfeature* de “Mapa mental” em “Princípios e técnicas” e as *subfeatures* de “regras de negócio” e “lista de requisitos” utilizadas dentro da empresa para “Documentar e detalhar requisitos”, além das *subfeatures* de “desenvolvedor” e “gerente” em “Papéis para detalhamento de requisitos”.

Todas essas correspondências demonstram como a empresa trabalha na modelagem de dados levando em consideração as *subfeatures* que foram detalhadas. Essa *Sprint* foi considerada importante, pois segundo os membros da empresa, existe uma carência de técnicas realmente implementadas e utilizadas. Dessa forma é possível visualizar a maneira que a BIT - Empresa Júnior de Ciência da Computação opera, além de demonstrar que existem muitas outras técnicas que podem ser utilizadas. Os artefatos presentes no *link* apresentado no referencial de [Amanda \(2022\)](#) desse trabalho, além de demonstrar a correspondência de cada *subfeature* com as aplicações, também pode ser configurada para aplicações a serem desenvolvidas. Por conta disso, a escolha do detalhamento de *subfeatures* fora do escopo da empresa foi feita pelos membros

da organização, ao ser útil para ajudar a estruturar uma modelagem de dados mais eficiente nas próximas aplicações.

Ao observar as árvores das figuras 28, 29, 30 e 31 é possível observar que as *subfeatures* de primeiro nível são todas opcionais.

Entretanto, ao olhar para as *subfeatures* de “ferramentas_gestao”, “boas_praticas” e “mudancas_requisitos” na figura 28, existem *subfeatures* obrigatórias nos níveis superiores. A *subfeature* “ferramentas_gestao” foi estruturada para citar alguns exemplos de ferramentas disponíveis no mercado. A ferramenta “Mermaid” é a utilizada pelos membros no momento que foi feita a pesquisa. Já a *subfeature* de “boas_praticas” foi estruturada na árvore de *features*, pois algumas das aplicações dentro do domínio da empresa não foram finalizadas pela falta de alguns entendimentos por parte dos desenvolvedores, uma vez que a modelagem de dados não tinha sido bem-feita. Por consequência, existe a *subfeature* de “mudancas_requisitos”, que pontuam os motivos dessas implementações não terem sido entregues ou finalizadas. Essas *subfeatures* também demonstram situações que podem ocorrer e o que deve ser feito para garantir boas práticas e quais as possíveis condições de evolução que podem ocorrer em um sistema de software.

Já nas figuras 29, 30 e 31, observa-se que dependendo da técnica escolhida, as *subfeatures* de nível superior passam a ser opcionais ou ter a relação XOR, uma vez que pelo menos 1 pode ser utilizada. Na figura 30 é possível perceber as *subfeatures* obrigatórias em “responsavel” que é uma *subfeature* de “papeis_projeto”, ou seja, as aplicações que possuíam papéis em seus projetos e um responsável, obrigatoriamente teriam de ter as *subfeatures* de níveis superiores ligadas a essas e, portanto, possuem uma similaridade entre os produtos do domínio que acaba acontecendo entre as aplicações.

Quadro 34 – Primeira parte do mapa de produtos atualizado para a *feature* principal “Modelagem de dados” e sua relação com a primeira parte dos produtos Web para a primeira parte de *subfeatures*.

Mapa de Produtos												
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos					
							Versão WEB					
							Apthus	Siqueira transportes	PrimorLab	Positiva Engetech	Hyperion	
Sistema de gestão de informações	Modelagem dos dados	Papéis do projeto	Responsável	Descrição técnica	Objetivos		X	X	X	X	X	
					Justificativa		X	X	X	X	X	
					Documentação		X	X	X	X	X	
				Metodologia		X	X	X	X	X		
				Prazos		X	X	X	X	X		
				Orçamento		X	X	X	X	X		
				Formas de pagamento		X	X	X	X	X		
			Membros	Front-end		X	X	X	X	X		
				Back-end		X	X	X	X	X		
		Técnicas para levantamento e elicitação de requisitos	Orientado a pontos de vista dos usuários				X					
			Etnografia	Observação do dia-a-dia								
			Workshops	período intensivo								
			Prototipagem	formas de interface			X					
				comunicação com outros sistemas								
			viabilidade de atendimento dos requisitos			X						
			Entrevistas	Formal								
				Informal			X	X	X	X	X	
				Fechadas								
			Abertas			X	X	X	X	X		
			Questionários	Separação geográfica dos grupos de usuários								
			Brainstorming	Várias reuniões	Geração	Coleta de ideias						
					Avaliação	Discussão de ideias						
			JAD	Planejamento e projeto	Levantar e especificar requisitos							
					Adaptação							
					Sessão							
					Finalização							
			Leitura de documentos									
			Análise de protocolos	Ordem de execução								
				Prioridade de tarefas								
			Participação dos usuários									
			Reuso de requisitos									
			Ferramentas de gestão	Kanban								
				(KANBANIZE, 2022)								
				BABOK								
				(IIBA, 2022)								
				OSRMT								
				(GITHUB, 2020)								
				IFPUG SNAP								
				(DEVMEDIA, 2014)								
			RE-Tools									
			(DEVMEDIA, 2015)									
		SOA com SoaML										
		(DEVMEDIA, 2012)										
		Mermaid	Criação de diagramas e visualização	Texto								
		(MERMAID, 2022)		Código								
		Boas práticas	Entendimento do domínio									
			Entendimento do problema									
			Entendimento do domínio do negócio									
			Entendimento das necessidades e restrições dos usuários									

Quadro 35 – Segunda parte do mapa de produtos atualizado para a *feature* principal “Modelagem de dados” e sua relação com a primeira parte dos produtos Web para a segunda parte de *subfeatures*.

Mapa de Produtos														
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos							
							Versão WEB							
							Apthus	Siqueira transportes	PrimorLab	Positiva Engetech	Hyperion			
Sistema de gestão de informações	Modelagem dos dados	Levantamento de requisitos utilizando o Scrum	Lista de prioridades	Backlog do produto	Priorização das atividades a serem executadas									
			Sprints	Reuniões semanais			X	X	X	X	X			
				Definição de duração			X	X	X	X	X			
				Porção dos requisitos a ser executada			X	X	X	X	X			
				Overview sobre o andamento das atividades			X	X	X	X	X			
			Sprint review	Avaliar objetivos			X	X	X	X	X			
				Adaptar Backlog do produto										
			Sprint retrospective	pontos positivos			X							
				pontos negativos			X							
			Métricas de software	Gestão de riscos	Tamanho do produto	Número de linhas de código								
		Número de pessoas no projeto					X							
		Número de defeitos no documento de requisitos					X							
		Tempo total de desenvolvimento			Tempo de desenvolvimento de uma funcionalidade	PDR (Project Delivery Rate)								
		Custo do produto												
		Grau de satisfação do cliente					X							
		Complexidade do produto												
		Eficiência												
		Confiabilidade												
		Facilidade de manutenção												
		Número de funções desenvolvidas por funções entregues em determinado período			SOD (Speed of Delivery)									
		Recursos					X							
		Requisitos					X							
		Ambiente												
		Tecnologia					X							
		Ferramentas					X							
		Documentar e detalhar requisitos			Princípios e técnicas	UML								
						Diagramas	i estrela							
			Caso de uso											
			Classe											
			Sequência											
			Histórias de usuário				X							
			Casos de teste											
			Mapa mental				X	X	X	X	X			
			Papéis para detalhamento de requisitos	Analista de métricas				X	X	X	X	X		
				Analista de negócios										
				Analista de testes										
				Desenvolvedor				X	X	X	X	X		
				Gerente				X	X	X	X	X		
			Lista de regras de negócio					X	X	X	X	X		
			Lista de requisitos					X	X	X	X	X		
		Mudanças a partir de requisitos	Condição de evolução	Sem conformidade com as solicitações										
				Ocorrência de erros										
				Ausência de detalhamento										
				Alterações no contexto	Datas		X	X						
					Abrangência		X	X						
					Alteração de legislação									

Quadro 36 – Terceira parte do mapa de produtos atualizado para a *feature* principal “Modelagem de dados” e sua relação com a segunda parte dos produtos Web para a primeira parte de *subfeatures*.

Mapa de Produtos												
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos					
							Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita	
Sistema de gestão de informações	Modelagem dos dados	Papéis do projeto	Responsável	Descrição técnica	Objetivos		X	X	X	X	X	
					Justificativa		X	X	X	X	X	
					Documentação		X	X	X	X	X	
				Metodologia		X	X	X	X	X		
				Prazos		X	X	X	X	X		
				Orçamento		X	X	X	X	X		
				Formas de pagamento		X	X	X	X	X		
			Membros	Front-end		X	X	X	X	X		
				Back-end			X	X		X		
		Técnicas para levantamento e elicitação de requisitos	Orientado a pontos de vista dos usuários				X			X		
			Etnografia	Observação do dia-a-dia								
			Workshops	período intensivo								
			Prototipagem	formas de interface			X			X		
				comunicação com outros sistemas								
				viabilidade de atendimento dos requisitos								
			Entrevistas	Formal								
				Informal			X	X	X	X		
				Fechadas								
			Questionários	Abertas			X	X	X	X		
				Separação geográfica dos grupos de usuários								
			Brainstorming	Várias reuniões	Geração	Coleta de ideias						
					Avaliação	Discussão de ideias						
			JAD	Planejamento e projeto	Levantar e especificar requisitos							
					Adaptação							
					Sessão							
					Finalização							
			Leitura de documentos									
			Análise de protocolos	Ordem de execução								
				Prioridade de tarefas								
			Participação dos usuários									
			Reuso de requisitos									
			Ferramentas de gestão	Kanban								
				(KANBANIZE, 2022)								
				BABOK								
				(IIBA, 2022)								
				OSRMT								
				(GITHUB, 2020)								
				IFPUG SNAP								
				(DEVMEDIA, 2014)								
			RE-Tools									
			(DEVMEDIA, 2015)									
			SOA com SoaML									
			(DEVMEDIA, 2012)									
			Mermaid	Criação de diagramas e visualização	Texto							
			(MERMAID, 2022)		Código							
			Boas práticas	Entendimento do domínio								
		Entendimento do problema										
		Entendimento do domínio do negócio										
Entendimento das necessidades e restrições dos usuários												

Quadro 37 – Quarta parte do mapa de produtos atualizado para a *feature* principal “Modelagem de dados” e sua relação com a segunda parte dos produtos Web para a segunda parte de *subfeatures*.

Mapa de Produtos														
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos							
							Versão WEB							
							Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita			
Sistema de gestão de informações	Modelagem dos dados	Levantamento de requisitos utilizando o Scrum	Lista de prioridades	Backlog do produto	Priorização das atividades a serem executadas		X			X				
			Sprints	Reuniões semanais				X	X					
				Definição de duração				X	X	X	X			
				Porção dos requisitos a ser executada				X	X	X	X			
				Overview sobre o andamento das atividades					X	X				
			Sprint review	Avaliar objetivos					X	X				
				Adaptar Backlog do produto										
			Sprint retrospective	pontos positivos										
				pontos negativos										
		Métricas de software	Gestão de riscos	Tamanho do produto	Número de linhas de código									
				Número de pessoas no projeto										
				Número de defeitos no documento de requisitos										
				Tempo total de desenvolvimento	Tempo de desenvolvimento de uma funcionalidade	PDR (Project Delivery Rate)								
				Custo do produto										
				Grau de satisfação do cliente										
				Complexidade do produto										
				Eficiência										
				Confiabilidade										
				Facilidade de manutenção										
				Número de funções desenvolvidas por funções entregues em determinado período	SOD (Speed of Delivery)									
				Recursos										
				Requisitos			X					X		
				Ambiente			X					X		
				Tecnologia			X					X		
				Ferramentas										
		Documentar e detalhar requisitos		Princípios e técnicas	UML									
					Diagramas	i estrela								
						Caso de uso								
						Classe								
						Sequência								
					Histórias de usuário									
				Casos de teste										
				Mapa mental				X	X					
				Papéis para detalhamento de requisitos	Analista de métricas									
					Analista de negócios					X	X			
					Analista de testes									
					Desenvolvedor			X	X	X	X	X		
				Gerente			X	X	X	X	X			
		Lista de regras de negócio						X	X					
		Lista de requisitos						X	X					
		Mudanças a partir de requisitos	Condição de evolução	Sem conformidade com as solicitações										
				Ocorrência de erros										
				Ausência de detalhamento										
				Alterações no contexto	Datas									
					Abrangência									
					Alteração de legislação									

Quadro 38 – Quinta parte do mapa de produtos atualizado para a *feature* principal “Modelagem de dados” e sua relação com o produto Híbrido e os produtos Mobile para a primeira parte de *subfeatures*.

Mapa de Produtos										
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos			
							Híbrido	Versão MOBILE		
							GDIS	Hyperion	CQD	Univita
Sistema de gestão de informações	Modelagem dos dados	Papéis do projeto	Responsável	Descrição técnica	Objetivos		X	X	X	X
					Justificativa		X	X	X	X
					Documentação		X	X	X	X
				Metodologia			X	X	X	X
				Prazos			X	X	X	X
				Orçamento			X	X	X	X
				Formas de pagamento			X	X	X	X
			Membros	Front-end			X	X	X	X
				Back-end				X	X	X
		Técnicas para levantamento e elicitação de requisitos	Orientado a pontos de vista dos usuários				X			
			Etnografia	Observação do dia-a-dia						
			Workshops	período intensivo						
			Prototipagem	formas de interface			X			
				comunicação com outros sistemas						
				viabilidade de atendimento dos requisitos						
			Entrevistas	Formal						
				Informal			X	X	X	
				Fechadas						
				Abertas			X	X	X	
			Questionários	Separação geográfica dos grupos de usuários						
			Brainstorming	Várias reuniões	Geração	Coleta de ideias				
					Avaliação	Discussão de ideias				
			JAD	Planejamento e projeto	Levantar e especificar requisitos					
					Adaptação					
					Sessão					
					Finalização					
			Leitura de documentos							
			Análise de protocolos	Ordem de execução						
				Prioridade de tarefas						
			Participação dos usuários							
			Reuso de requisitos							
	Ferramentas de gestão	Kanban (KANBANIZE, 2022)								
		BABOK (IIBA, 2022)								
	Boas práticas	OSRMT (GITHUB, 2020)								
		IFPUG SNAP (DEV MEDIA, 2014)								
		RE-Tools (DEV MEDIA, 2015)								
		SOA com SoaML (DEV MEDIA, 2012)								
		Mermaid (MERMAID, 2022)								
		Criação de diagramas e visualização								

Quadro 39 – Quinta parte do mapa de produtos atualizado para a *feature* principal “Modelagem de dados” e sua relação com o produto Híbrido e os produtos Mobile para a segunda parte de *subfeatures*.

Mapa de Produtos											
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos				
							Híbrido	Versão MOBILE			
							GDIS	Hyperion	CQD	Univita	
Sistema de gestão de informações	Modelagem dos dados	Levantamento de requisitos utilizando o Scrum	Lista de prioridades	Backlog do produto	Priorização das atividades a serem executadas		X				
			Sprints	Reuniões semanais					X	X	
				Definição de duração				X	X	X	
				Porção dos requisitos a ser executada				X	X	X	
				Overview sobre o andamento das atividades					X	X	
			Sprint review	Avaliar objetivos					X	X	
				Adaptar Backlog do produto							
			Sprint retrospective	pontos positivos				X			
				pontos negativos				X			
			Métricas de software	Gestão de riscos	Tamanho do produto	Número de linhas de código					
		Número de pessoas no projeto									
		Número de defeitos no documento de requisitos						X			
		Tempo total de desenvolvimento			Tempo de desenvolvimento de uma funcionalidade	PDR (Project Delivery Rate)					
		Custo do produto									
		Grau de satisfação do cliente					X				
		Complexidade do produto									
		Eficiência									
		Confiabilidade									
		Facilidade de manutenção					X				
		Número de funções desenvolvidas por funções entregues em determinado período			SOD (Speed of Delivery)						
		Recursos									
		Requisitos					X				
		Ambiente					X				
		Tecnologia					X				
		Ferramentas									
		Documentar e detalhar requisitos	Princípios e técnicas	UML							
				Diagramas	i estrela						
					Caso de uso						
					Classe						
				Sequência							
				Histórias de usuário							
				Casos de teste							
				Mapa mental				X	X		
				Papéis para detalhamento de requisitos	Analista de métricas						
					Analista de negócios				X	X	
			Analista de testes								
			Desenvolvedor				X	X	X		
			Gerente				X	X	X		
			Lista de regras de negócio					X	X		
			Lista de requisitos					X	X	X	
		Mudanças a partir de requisitos	Condição de evolução	Sem conformidade com as solicitações							
				Ocorrência de erros							
				Ausência de detalhamento							
				Alterações no contexto	Datas						
					Abrangência						
					Alteração de legislação						

Figura 28 – Subárvores da árvore de *features* apresentado para a *feature* principal “Modelagem de dados” e suas *subfeatures* de “ferramentas_gestao”, “boas_praticas” e “mudancas_requisitos”.

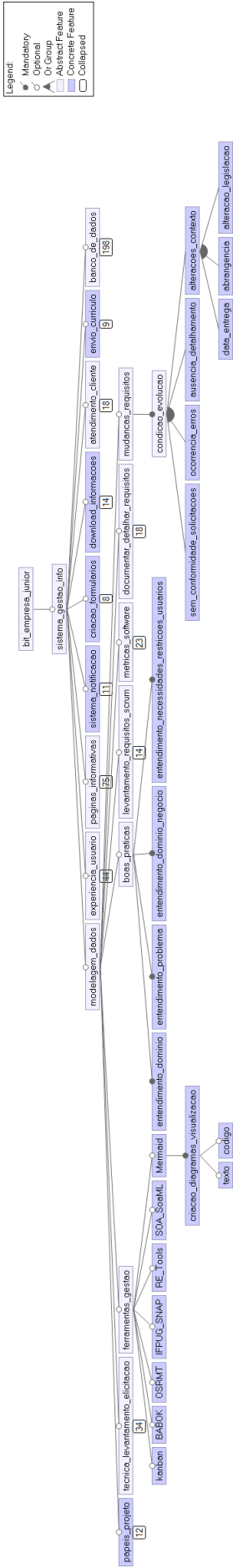
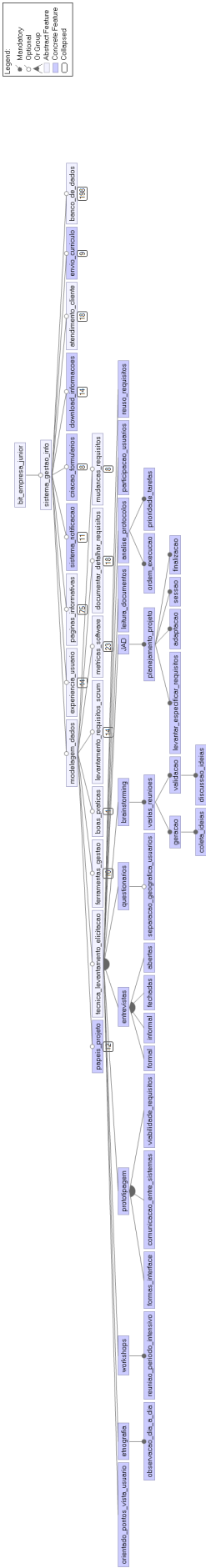


Figura 31 – Subárvore da árvore de *features* apresentado para a *feature* principal “Modelagem de dados” e sua *subfeature* de “*tecnica_levantamento_elicitacao*”.



8.5.2.5 *Sprint 5*

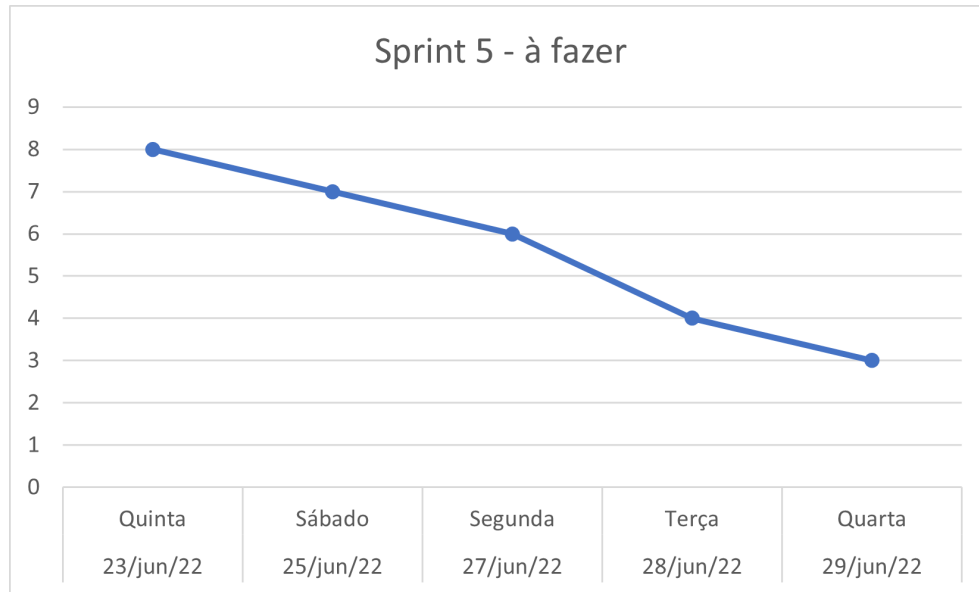
A *Sprint 5* foi executada entre os dias 23 de junho de 2022 à 29 de junho de 2022. No (Quadro 40) é apresentado o Backlog da *Sprint 5*, em que são definidos os objetivos, a duração, as tarefas, os especialistas envolvidos e quais são as *features* principais selecionadas. As *features* principais realizadas foram as de “Atendimento ao cliente”, “Envio de currículo”, “Gerência de compra e venda” e “Controle de caixa”, totalizando 16 horas de trabalho nas atividades propostas. A reunião de identificação das *subfeatures* levou 20 min, já a reunião de validação dos artefatos criados levou 10 minutos. O gráfico *burndown* apresentado na Figura 32 representa os dias de trabalho pelas tarefas realizadas.

As atividades de identificação e documentação das *subfeatures* começaram apenas no dia 23 de junho de 2022, uma vez que essa reunião foi feita com todos os membros da BIT no momento da pesquisa durante a reunião semanal da empresa. Os 8 integrantes adicionais durante a reunião de identificação das *subfeatures* foi essencial, uma vez que não existia nenhum tipo de documentação ou código para essas *feature* e, portanto, foi discutido com os membros quais seriam as possibilidades de implementação. Todas as tarefas que precisavam ser realizadas durante o dia se repetiram durante vários dias no período proposto.

Quadro 40 – Backlog da *Sprint 5* para as *features* de “Atendimento ao cliente”, “Envio de currículo”, “Gerência de compra e venda” e “Controle de caixa”.

Sprint 5		Features principais selecionadas	Tarefas	Proprietário	Status	Observação
Objetivo	Analisar as features principais de atendimento ao cliente, envio de currículo, gerência de compra e venda e controle de caixa	Atendimento ao cliente, envio de currículo, gerência de compra e venda e controle de caixa	Identificar as <i>subfeatures</i> a partir do especialista de domínio	Todos	Feito	Nenhuma
Duração	16 horas		Especificar e documentar as <i>subfeatures</i>		Feito	
Data de início	23/jun/22		Organizar diagrama hierárquico, agrupando e decompondo features (modelo de features)		Feito	
Data de encerramento	29/jun/22		Organizar as <i>subfeatures</i> no mapa de produtos (atualizar mapa de produtos)		Feito	
Data de revisão	29/jun/22		Inspecionar artefatos legados		Feito	
Artefatos prontos	Mapa de aplicações, Avaliação das features principais, Lista de aplicações e Lista do domínio de features	UNIOESTE, bloco F, sala 7 às 17h	Separar artefatos legados reusáveis		Feito	
Local/hora das reuniões diárias			Estudar ferramenta para modelar features (FeatureIDE)		Feito	
Scrum master, Especialista em análise do domínio	Amanda Israel Graeff Borges		Validar as <i>Sprints</i>		Feito	
Especialista do domínio	2 especialistas					
Integrantes da equipe	8 integrantes					

Figura 32 – Gráfico *burndown* da *Sprint 5* representando os dias trabalhados pela quantidade de tarefas realizadas.



Cada um dos artefatos criados na *Sprint 5* estão presentes nos quadros 41, 42, 43, 44, 45 e 46 e nas figuras 28, 29, 30 e 31.

No quadro 42, a aplicação “Positiva Engetech” possui a *feature* de “Gerência de compra e venda” e, portanto, possui correspondência entre 20 das 30 *subfeatures* catalogadas. Nesse mesmo quadro, a “PrimorLab” possui 2 das 11 *subfeatures* de “Atendimento ao cliente”.

No quadro 43, a única aplicação que possui correspondências com a *feature* de “Controle de caixa” é a “G4”. Ela possui 26 das 33 *subfeatures* catalogadas. Já no quadro 44, apenas a aplicação “CONNECT4SITE” possui correspondência com as *features* de “Atendimento ao cliente” e “Envio de currículo”, apresentando 6 das 11 correspondências para a primeira *feature* e todas as correspondências para a segunda *feature*.

Como é possível observar, nos quadros 41, 45 e 46 não existem correspondências entre as *features* e *subfeatures* com as aplicações da empresa. Entretanto, existem algumas *sufeatures* que satisfazem aplicações em específico, como foi mencionado anteriormente, mas variabilidades que correspondem apenas a 1 aplicação não formam uma linha de produtos que gere um potencial de reuso. Portanto, as *features* principais de “Atendimento ao cliente”, “Envio de currículo”, “Gerência de compra e venda” e “Controle de caixa” apenas se tornarão relevantes a uma família de produtos, caso existam novos produtos que também possuam essas *features*.

Como é possível observar na figura 33, todas as *subfeatures* da *feature* principal “Controle de caixa” são obrigatórias, uma vez que ao implementar a *feature* controle de caixa, é necessário que se crie dados atômicos, consistentes, isolados e duráveis, além de ter um controle de acesso dos usuários para que as operações realizadas sejam autorizadas. Além disso, deve

Quadro 41 – Primeira parte do mapa de produtos atualizado para a *feature* principal “Controle de caixa” e sua relação com a primeira parte dos produtos Web.

Mapa de Produtos												
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos					
							Versão WEB					
							Apthus	Siqueira transportes	PrimorLab	Positiva Engetech	Hyperion	
Sistema de gestão de informações	Controle de caixa	Atomicidade										
		Consistência										
		Isolamento										
		Durabilidade										
		Único usuário logado por vez	Acesso restrito	Comprovação de senha	Realizar operações de transação							
		Operações	Escrita									
			Leitura									
			Início transação									
			Fim transação									
		Tratamento de erro das transações	Sucesso	commit_transaction								
			Sem sucesso	abort_transaction	Rollback							
		Controle de entradas	Depósitos									
			Transferências									
			Recebimentos									
		Controle de saídas	Impostos									
			Equipamentos									
			Materiais									
		Tabela de transações	Atributos	Data								
				Valor	Entrada							
					Saída							
				Nota fiscal								
		Valor total	Saldo operacional	Descrição								
				Entradas menos as saídas								
				Saldo final	Saldo operacional mais saldo inicial							
		Relatório de controle de movimentação	Período da movimentação	Dia								
				Mês								
				Ano								
		Falhas de transação	Divisão por zero									
			Leitura de dado inexistente									
			Protocolo de deadlock									
			Parâmetros incorretos									
			Interrupção de transação									
Backups periódicos												

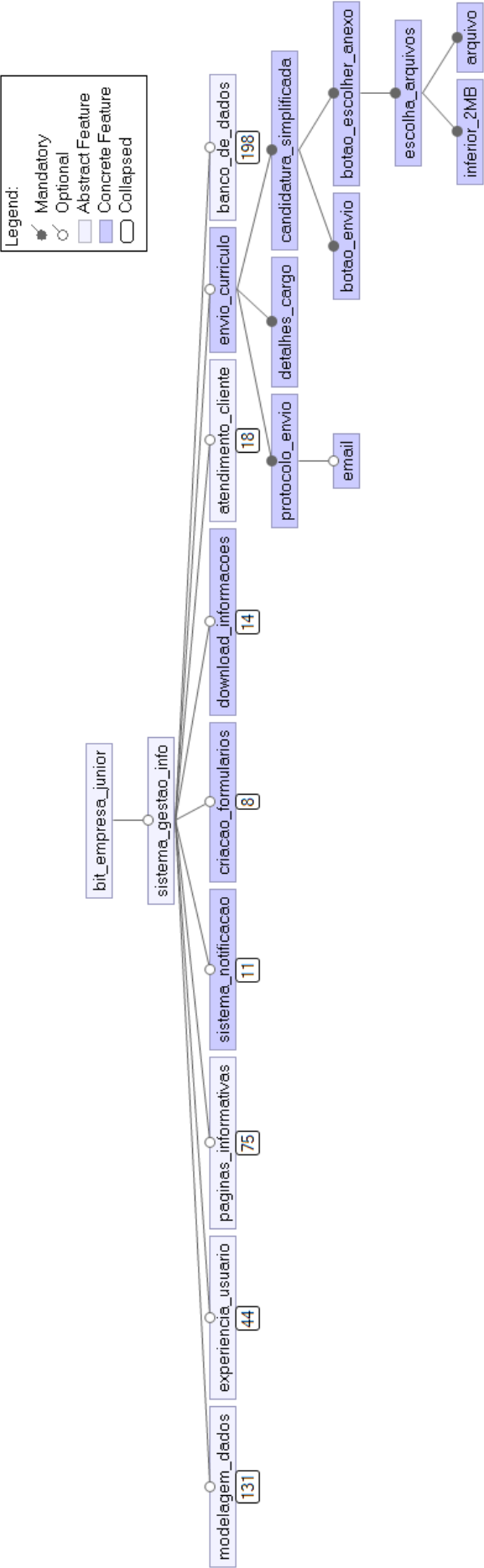
haver tratamentos de erros, controle de transações, relatórios de movimentos, além de *backups* periódicos. Todas essas *subfeatures* foram catalogadas pelos especialistas e pelos desenvolvedores durante a reunião de detalhamento. Como essas *features* principais da *Sprint 5* não possuem código e nem documentações bem estruturadas, optou-se por utilizar de vários membros para conseguir catalogar boas práticas ao detalhar essas *features*.

Da mesma forma, para a *feature* “Gerência de compra e venda” foi criada a árvore apresentada na figura 34, em que a maioria das *subfeatures* são obrigatórias, exceto as de “historico_compras_usuarios” e “notificacao”. Na reunião de detalhamento das *subfeatures* foi partido do princípio que essa *feature* era derivada de uma loja de vendas online e, portanto, os especialistas e integrantes detalharam as *subfeatures* dessa forma.

A figura 35 representa o detalhamento das *subfeatures* de “Atendimento ao cliente”. Foi concluído pelos especialistas, que essa *feature* principal não poderia ser derivada detalhadamente, pois nenhum dos desenvolvedores presentes na reunião possuíam conhecimento aprofundado sobre tal técnica. Caso algum cliente pedisse tal funcionalidade, seria proposto alguma aplicação terceirizada para integrar com o sistema a ser desenvolvido. Dessa forma, as *subfeatures* apresentadas nessa árvore são opcionais ou pode haver mais de uma opção dependendo da especificação do cliente.

A árvore 36 apresenta a *feature* principal de “Envio de currículo”, em que todas as *subfeatures* apresentadas são obrigatórias, exceto a opção de protocolo de envio. O modo de implementar essa *feature* foi decidida por todos os desenvolvedores, por se tratar de uma funcionalidade simples.

Figura 36 – Subárvore da árvore de *features* apresentado para a *feature* principal “Envio de currículo” e suas *subfeatures*.



8.6 Questionário e Entrevista

Como apresentado no capítulo 6, o questionário foi baseado em um estudo de viabilidade, em que foi buscado responder se o processo trouxe resultados usáveis, ou seja, se ele gerou artefatos úteis que podem ser utilizados posteriormente para o reuso, e se o tempo foi bem usado na geração desses artefatos. Além disso, foi proposto também fazer um estudo observacional, respondendo se os passos do processo fazem sentido para o contexto da organização. O objetivo geral dessa pesquisa é constatar se a abordagem RISE-DA faz sentido no contexto de uma empresa júnior. A seguir, são apresentadas as perguntas utilizadas no questionário e as respectivas respostas dos três especialistas de domínio presentes ativamente em todas as etapas da pesquisa. Esses especialistas tinham os cargos de presidente, tesoureiro e secretário da empresa e possuíam conhecimento aprofundado da mesma, pois executaram a maioria dos processos administrativos e de desenvolvimento.

- **Os conceitos utilizados no processo ficaram claros?** *Resposta:* Dois dos especialistas do domínio e membros da diretoria realizaram a disciplina de reuso e conseguiram entender completamente todas as etapas e conceitos utilizados durante a pesquisa. O outro especialista do domínio e membro da diretoria entendeu parcialmente os processos utilizados, entretanto o envolvimento do membro na avaliação e refatoramento das tabelas e da árvore de *features* fizeram com que as etapas e os objetivos do processo ficassem mais claros.
- **Qual sua opinião sobre a efetividade da abordagem adotada levando em consideração os dados encontrados? (Considerando as limitações do processo)** *Resposta:* Como foram encontradas possíveis famílias de produtos nas *sprints* 1, 2, 3 e 4 que contêm as *features* e *subfeatures* que obtiveram similaridade ou uma variabilidade pequena entre os produtos e as aplicações mapeadas da empresa, é possível utilizar a análise de escopo para agilizar o processo de desenvolvimento, uma vez que as tomadas de decisão seriam automatizadas pelas boas práticas e pela definição de quais funcionalidades são mais trabalhadas dentro do domínio da empresa. A parte de encontrar padrões se mostrou excelente e demonstra ser um possível facilitador para os alunos durante a criação de um projeto. Entretanto, como existe grande rotatividade de membros dentro da empresa, não é garantido que, mesmo automatizando boas práticas de programação, ainda assim pode ocorrer a não utilização dos possíveis módulos reusáveis que seriam desenvolvidos futuramente.
- **Utilizando essa abordagem, foi possível detectar dificuldades ao implementá-la na empresa?** *Resposta:* Sim, pois a empresa estava passando por um período pós-pandemia, mas que também herdou uma desorganização de documentos e repositórios de anos passados que foram um dificultador para todas as etapas do processo. Além disso, houve muita entrada e saída de funcionários no período da pesquisa e, por conta disso, as

Quadro 42 – Primeira parte do mapa de produtos atualizado para as *features* principais “Gerência de compra e venda”, “Atendimento ao cliente” e “Envio de currículo” e sua relação com a primeira parte dos produtos Web.

Mapa de Produtos															
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos								
							Versão WEB								
							Apthus	Siqueira transportes	PrimorLab	Positiva Engetech	Hyperion				
Sistema de gestão de informações	Gerência de compra e venda	Termos e condições								X					
		Registro de entrada e saída de produtos													
		Apresentação dos produtos	Lista de produtos	Nome							X				
				Preço							X				
				Imagem							X				
			Pesquisa	Barra de pesquisa de produtos							X				
				Botão de pesquisa de produtos											
				Procura por nome	Nome de produto						X				
		Histórico de compra do usuário	Front page do usuário	Algoritmo de recomendação	Categoria de produtos										
				Usuário logado	Controle de entrada e saída de usuários										
		Tela item		Preço do item								X			
				Descrição item unitário								X			
				Nome do item								X			
				Especificações do item								X			
		Pagamentos	Carrinho	Finalizar compra	Controle de produtos							X			
					Valor total							X			
						Especificação de cada produto									
						Valor unitário							X		
						Tipo de pagamento							X		
					Entrega	Tempo de chegada							X		
							Lugar							X	
					Frete								X		
					Subtotal									X	
					Notal fiscal									X	
				Descrição do produto											
				Quantidade									X		
	Notificação		Recomendações de produtos												
			Promoções												
			Entrega de produtos												
			Confirmação de compra												
	Atendimento ao cliente	Chat em tempo real	Contratar um serviço												
			Controle de dados	Avaliação de desempenho											
		Equipe de atendimento disponível								X					
		Página de perguntas frequentes	Priorização de perguntas	Quantidade de acesso											
			Pesquisa por associação de palavras												
		Taxa de conversão	Compra de produtos/ serviços												
		Integração de canais	omni chat	Aplicativos de celular											
				Site							X				
				Loja Online											
		Blog													
		Chatbots													
	Envio de currículo	Protocolo de envio	email												
		Detalhes do cargo													
		Candidatura simplificada	Botão de envio												
	Botão de escolher anexo		Escolha de arquivos	Inferior 2MB											
Arquivo															

Quadro 43 – Segunda parte do mapa de produtos atualizado para a *feature* principal “Controle de caixa” e sua relação com a segunda parte dos produtos Web.

Mapa de Produtos											
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos				
							Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita
Sistema de gestão de informações	Controle de caixa	Atomicidade						X			
		Consistência						X			
		Isolamento						X			
		Durabilidade						X			
		Único usuário logado por vez	Acesso restrito	Comprovação de senha	Realizar operações de transação			X			
		Operações	Escrita					X			
			Leitura					X			
			Início transação					X			
			Fim transação					X			
		Tratamento de erro das transações	Sucesso	commit_transaction				X			
			Sem sucesso	abort_transaction	Rollback			X			
		Controle de entradas	Depósitos					X			
			Transferências					X			
			Recebimentos					X			
			Impostos					X			
		Controle de saídas	Equipamentos					X			
			Materiais					X			
		Tabela de transações	Atributos	Data				X			
				Valor	Entrada			X			
				Saída				X			
				Nota fiscal				X			
		Valor total	Saldo operacional	Descrição				X			
				Entradas menos as saídas							
		Saldo final	Saldo operacional mais saldo inicial								
		Relatório de controle de movimentação	Período da movimentação	Dia				X			
				Mês				X			
				Ano				X			
		Falhas de transação	Divisão por zero								
			Leitura de dado inexistente					X			
			Protocolo de deadlock								
			Parâmetros incorretos								
Interrupção de transação											
Backups periódicos											

comunicações sobre a pesquisa com antigos membros eram inviáveis. Os novos membros se mostraram proativos em ajudar durante todo o período em que participaram, mas por não terem um conhecimento aprofundado sobre programação e processos de engenharia de software, não conseguiram ajudar em todas as partes da abordagem, principalmente na avaliação dos artefatos. Por se tratar de estudantes de uma universidade, o tempo para construir e validar os artefatos eram feitos durante reuniões em que alguns membros estavam disponíveis. Por conta da escassez de tempo, consequentemente foi necessário uma objetividade entre as reuniões, mas isso fez com que elas fossem melhor aproveitadas.

- **Considerando a abordagem RISE-DA, será possível utilizá-la para implementar a segunda parte do processo (RISE-CBD)? Que mudanças devem ocorrer para que isso aconteça?** *Resposta:* Mesmo a abordagem sendo um facilitador para realizar a segunda parte do processo, é recomendado algumas aulas de treinamento para os novos membros da BIT e tornar isso algo recorrente, uma vez que para utilizar todo o processo, é necessário que se mantenha os conhecimentos adquiridos durante a aquisição dos dados da pesquisa. Portanto, é recomendado que algum dos especialistas do domínio que participaram da abordagem RISE-DA continuem na abordagem RISE-CBD. É necessário também achar

Quadro 44 – Segunda parte do mapa de produtos atualizado para as *features* principais “Gerência de compra e venda”, “Atendimento ao cliente” e “Envio de currículo” e sua relação com a segunda parte dos produtos Web.

Mapa de Produtos													
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos						
							Versão WEB						
							Projeto Engenharia	G4	CQD	CONNECT4SITE	Univita		
Sistema de gestão de informações	Gerência de compra e venda	Termos e condições											
		Registro de entrada e saída de produtos											
		Apresentação dos produtos	Lista de produtos	Nome									
				Preço									
			Pesquisa	Imagem									
				Barra de pesquisa de produtos									
				Botão de pesquisa de produtos									
		Histórico de compra do usuário	Front page do usuário	Procura por nome	Nome de produto								
				Algoritmo de recomendação	Categoria de produtos								
				Usuário logado	Controle de entrada e saída de usuários								
		Tela item		Preço do item									
				Descrição item unitário									
				Nome do item									
				Especificações do item									
		Pagamentos	Carrinho		Controle de produtos								
					Finalizar compra	Valor total							
				Especificação de cada produto									
				Valor unitário									
				Tipo de pagamento									
				Entrega		Tempo de chegada							
						Lugar							
				Frete									
				Subtotal									
				Notal fiscal									
				Descrição do produto									
		Quantidade											
		Notificação		Recomendações de produtos									
				Promoções									
				Entrega de produtos									
	Confirmação de compra												
	Atendimento ao cliente	Chat em tempo real	Contratar um serviço										
			Controle de dados	Avaliação de desempenho							X		
		Equipe de atendimento disponível									X		
		Página de perguntas frequentes	Priorização de perguntas	Quantidade de acesso									
			Pesquisa por associação de palavras								X		
		Taxa de conversão	Compra de produtos/ serviços								X		
		Integração de canais	omni chat	Aplicativos de celular								X	
				Site								X	
				Loja Online									
				Blog									
	Chatbots												
	Envio de currículo	Protocolo de envio	email								X		
											X		
		Detalhes do cargo									X		
Candidatura simplificada			Botão de envio								X		
	Botão de escolher anexo	Escolha de arquivos	Inferior 2MB							X			
Arquivo									X				

Quadro 45 – Terceira parte do mapa de produtos atualizado para a *feature* principal “Controle de caixa” e sua relação com o produto Híbrido e os produtos Mobile.

Mapa de Produtos											
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos				
							Híbrido	Versão MOBILE			
							GDIS	Hyperion	CQD	Univita	
Sistema de gestão de informações	Controle de caixa	Atomicidade									
		Consistência									
		Isolamento									
		Durabilidade									
		Único usuário logado por vez	Acesso restrito	Comprovação de senha	Realizar operações de transação						
		Operações	Escrita								
			Leitura								
			Início transação								
			Fim transação								
		Tratamento de erro das transações	Sucesso	commit_transaction							
			Sem sucesso	abort_transaction	Rollback						
		Controle de entradas	Depósitos								
			Transferências								
			Recebimentos								
		Controle de saídas	Impostos								
			Equipamentos								
			Materiais								
		Tabela de transações	Atributos	Data							
				Valor	Entrada						
					Saída						
				Nota fiscal							
				Descrição							
		Valor total	Saldo operacional	Entradas menos as saídas							
			Saldo final	Saldo operacional mais saldo inicial							
		Relatório de controle de movimentação	Período da movimentação	Dia							
				Mês							
				Ano							
		Falhas de transação	Divisão por zero								
			Leitura de dado inexistente								
			Protocolo de deadlock								
			Parâmetros incorretos								
Interrupção de transação											
Backups periódicos											

uma melhor forma de visualizar os dados, uma vez que as tabelas não se mostram muito convidativas, uma vez que existem muitos dados a serem analisados e se não houver uma certa familiaridade, é muito difícil passar por todos os artefatos criados. A ferramenta Eclipse com a extensão FeatureIDE se mostrou muito difícil e não intuitiva de se utilizar e isso pode fazer com que esse artefato não seja utilizado ou não utilizado da maneira correta.

- **A parte 1 do processo permitiu que a empresa conseguisse mais conhecimento sobre seus objetivos, metas e seu lugar no mercado?** *Resposta:* Sim, pois com a análise de mercado, foi possível concluir que a empresa BIT é uma empresa de software de gerenciamento de informações e trabalha com pessoas físicas e jurídicas. Seu diferencial perante seus concorrentes mais próximos é oferecer um valor abaixo do mercado. A BIT trabalha com entregas completas do sistema, apesar de utilizarem uma metodologia ágil no desenvolvimento e oferecem apenas 3 meses de manutenção, não chegando a amadurecer nenhum projeto. O nível de complexidade dos sistemas de software desenvolvidos pela empresa é médio ou baixo, uma vez que existe muita rotatividade de alunos e, portanto, acabam existindo níveis de conhecimentos variados em programação e tecnologias e isso

Quadro 46 – Terceira parte do mapa de produtos atualizado para as *features* principais “Gerência de compra e venda”, “Atendimento ao cliente” e “Envio de currículo” e sua relação com o produto Híbrido e os produtos Mobile.

Mapa de Produtos										
Domínio	Feature Principal	Subfeatures 1	Subfeatures 2	Subfeatures 3	Subfeatures 4	Subfeatures 5	Produtos			
							Híbrido	Versão MOBILE		
							GDIS	Hyperion	CQD	Univita
Sistema de gestão de informações	Gerência de compra e venda	Termos e condições								
		Registro de entrada e saída de produtos								
		Apresentação dos produtos	Lista de produtos	Nome						
				Preço						
			Pesquisa	Imagem						
				Barra de pesquisa de produtos						
				Botão de pesquisa de produtos						
				Procura por nome	Nome de produto					
		Histórico de compra do usuário	Front page do usuário	Algoritmo de recomendação	Categoria de produtos					
				Usuário logado	Controle de entrada e saída de usuários					
		Tela item		Preço do item						
				Descrição item unitário						
				Nome do item						
				Especificações do item						
		Pagamentos	Carrinho	Controle de produtos						
					Valor total					
				Finalizar compra	Especificação de cada produto					
					Valor unitário					
					Tipo de pagamento					
					Entrega	Tempo de chegada				
						Lugar				
					Frete					
					Subtotal					
					Notal fiscal					
					Descrição do produto					
					Quantidade					
		Notificação	Recomendações de produtos							
				Promoções						
				Entrega de produtos						
				Confirmação de compra						
	Atendimento ao cliente	Chat em tempo real	Contratar um serviço							
			Controle de dados	Avaliação de desempenho						
		Equipe de atendimento disponível								
		Página de perguntas frequentes	Priorização de perguntas	Quantidade de acesso						
			Pesquisa por associação de palavras							
		Taxa de conversão	Compra de produtos/ serviços							
		Integração de canais	omni chat	Aplicativos de celular						
				Site						
				Loja Online						
				Blog						
		Chatbots								
	Envio de currículo	Protocolo de envio	email							
		Detalhes do cargo								
		Candidatura simplificada	Botão de envio							
			Botão de escolher anexo	Escolha de arquivos	Inferior 2MB	Arquivo				

influência no nível do projeto. Dessa forma, foi possível visualizar nesse processo que não existe nenhum produto em específico que a empresa oferece ao mercado, muito menos das tecnologias em uso. Portanto, a busca de conhecimento sobre boas práticas irá acontecer apenas nas funcionalidades mais genéricas da empresa.

- **A parte 2 do processo permitiu detalhar melhor as *features* utilizadas com base nas aplicações já realizadas?** *Resposta:* Sim, especialmente as *features* de maior prioridade da empresa (CRUD, Controle de acesso dos usuários, modelagem dos dados, experiência do usuário e páginas informativas). O maior detalhamento das *subfeatures* ocorreu na *feature* principal de Controle de acesso dos usuários, pois foram desenvolvidos 8 níveis. As *features* principais que possuíam maior prioridade tinham código legado e documentação para analisar e minerar durante o processo. Já as outras *features*, não possuíam essas informações, mas seu nível de prioridade era baixo ou muito baixo e, portanto, seu nível de detalhamento foi suficiente, uma vez que se um desenvolvedor precisar programar essas funcionalidades, já vai conseguir ter uma noção do que a maioria dos membros da BIT discutiram e analisaram sobre quais seriam as *subfeatures* mais interessantes a serem implementadas.
- **Foi possível identificar quais princípios e técnicas fazem mais sentido para a organização?** *Resposta:* Sim, por conta das prioridades que as *features* principais possuem na parte 2 da abordagem e pelo detalhamento das *subfeatures* dessas *features* principais, foi possível observar os padrões que foram utilizados nas implementações e nas documentações analisadas. Além disso, foi analisado que na *feature* principal “Modelagem de dados” podem existir muitas práticas e técnicas a serem utilizadas na empresa em Engenharia de Software, que é um dos processos que a empresa possui mais dificuldade. Segundo um dos membros da diretoria e especialista do domínio, é interessante buscar para projetos futuros uma ferramenta que consiga ser prática e intuitiva para fazer a modelagem de dados, para possuir uma visualização melhor dos requisitos de um projeto.
- **Foi possível identificar quais *features* se encaixam no domínio da organização pela sua reusabilidade? Quais são as *features* mais improváveis de serem implementadas para o reuso?** *Resposta:* Sim, com base no Backlog do domínio é possível observar que as *features* “CRUD”, “Controle de acesso dos usuários”, “Modelagem dos dados”, “Experiência dos usuários” e “Páginas informativas” são as que mais se encaixam no domínio da organização. Foi possível encontrar similaridade entre os produtos do domínio nas *features* de “Experiência do usuário” e “Modelagem dos dados” e um grande número de variabilidades nas outras *features* citadas. Por outro lado, as que menos se encaixam no domínio da organização, como “Sistema de notificação”, “Download de informações”, “Atendimento ao cliente”, “Envio de currículo”, “Gerência de compra e venda” e “Controle de caixa” se mostraram com pouco número de variabilidade (1 à 3 correspondências com as aplicações) e nenhuma similaridade entre os produtos do domínio. Dessa forma, fica

claro que as *features* que apresentam reusabilidade são as que possuem maior prioridade no Backlog do domínio.

9

Conclusão e trabalhos futuros

O objetivo desse estudo foi avaliar a viabilidade da abordagem RISE-DA, proposta por (SILVA, 2013) em definir um escopo de domínio dentro de uma micro organização, particularmente em uma *very small entity*. Um questionário e entrevistas foram utilizados nesse trabalho para coletar dados qualitativos sobre todos os artefatos realizados, ao buscar se a abordagem RISE-DA foi viável para a empresa BIT - Empresa júnior de Ciência da Computação. Foram analisados todos os artefatos durante os processos de detalhamento, documentação, criação e validação para controlar a qualidade dos dados em diferentes momentos da pesquisa, uma vez que foi utilizada a metodologia Ágil Scrum durante o processo iterativo. Todos os dados foram avaliados sobre diferentes perspectivas, tanto por diferentes integrantes ao longo das *sprints*, quanto diferentes especialistas do domínio e especialistas do negócio.

A análise foi iniciada pela aquisição dos dados da pré-análise e primeira versão do mapa de produtos. A partir desses artefatos, foi possível definir suas prioridades, e tentar criar uma análise mais detalhada. Foram apresentadas algumas dificuldades durante a pesquisa, principalmente pela falta de experiência dos membros novos e antigos da empresa, tanto em relação as aplicações realizadas anteriormente, quanto das documentações da empresa. Isso aconteceu por se tratar de uma empresa pequena e com muita rotatividade de clientes e, para que fosse possível realizar a pesquisa, os dados a serem minerados precisaram ser organizados e classificados de maneira correta. Além disso, conforme os artefatos aumentavam sua complexidade ao longo do processo, diminuía a quantidade de especialistas envolvidos para avaliá-los.

Os especialistas do domínio que participaram ativamente da pesquisa concluíram que os resultados conseguiram ajudar a descobrir como a empresa funciona dentro do seu domínio. Ao avaliar o mapa de produtos finalizado, é possível perceber que a maioria das *features* e *subfeatures* detalhadas apresentaram muita variabilidade entre as aplicações. Mesmo com esses resultados, ainda é possível categorizar o reuso em diferentes níveis. O primeiro nível enquadraria os projetos que precisam de uma área interna, onde aconteceria o gerenciamento de informações pelos

funcionários, ou seja, o primeiro nível corresponde às aplicações que precisam de um back-end. O segundo nível encontrado seria reusar estruturas do front-end genéricas que serviriam para todas as aplicações.

A construção da árvore de *features* proporcionou uma visão melhor das dependências entre as *features* e *subfeatures* catalogadas. Ao trabalhar com back-end, necessariamente vai existir a *feature* de “CRUD” e “Controle de acesso dos usuários”. Ao ter essas duas estruturas, pode-se adicionar “Visualização de relatórios”, “Criação de formulários”, “Controle de caixa” e “Gerência de compra e venda”.

Foi possível determinar também os motivos que levaram algumas *features* a terem prioridades tão baixas no Backlog do Domínio. A *feature* de “Sistema de notificação” ficava dependente de servidores terceirizados (SMTP, WhatsApp (INC, 2009) e SMS). Isso acontece também para *feature* de “Atendimento ao cliente”, em que um chat em tempo real é um tipo de funcionalidade muito complexa e imatura para ser implementada e por conta disso, uma empresa terceirizada deveria ser contratada. Outras *features* como “Gerência de compra e venda” e “Controle de caixa” obtiveram um alto risco de implementação, já que a BIT não consegue implementar projetos com muita maturidade, uma vez que seu processo de manutenção ocorre apenas durante 3 meses. Todas essas *features* são possíveis de serem desenvolvidas, mas isso depende do conhecimento técnico dos desenvolvedores.

O *feedback* que aconteceu entre Especialista em análise do domínio e os Especialista do Domínio, Especialista do negócio e Desenvolvedores aconteceu durante as reuniões marcadas e durante as reuniões semanais da empresa. A abordagem propõe esse *feedback* constante e rápido entre esses especialistas em todas as etapas, entretanto, apenas a definição e validação conseguiram cumprir esse papel completamente, uma vez que os times da empresa não se encontravam sempre disponíveis para realizar as atividades da pesquisa.

Com o estudo de caso, foi possível concluir que uma *very small entity* conseguiu trabalhar efetivamente com a proposta da abordagem RISE-DA, uma vez que as fases chave que acontecem durante todo o período da pesquisa conseguem se adaptar a um cenário que muda rapidamente, uma vez que foi possível observar os padrões de implementação que podem ser utilizados dentro do seu domínio. É recomendado reutilizar essa abordagem caso o domínio da empresa tenha mudanças, pois será preciso descobrir quais são suas novas limitações e o que ela consegue oferecer aos clientes de forma satisfatória.

No questionário, junto com as entrevistas, foi possível obter *feedbacks* dos especialistas envolvidos na pesquisa. Algumas das respostas obtidas no Subcapítulo 8.6 foram positivas em relação a abordagem, pois, segundo eles, os dados desse trabalho poderão ser utilizados para implementar *templates* que ficarão disponíveis para todos futuros desenvolvedores da BIT - Empresa Júnior de Ciência da Computação. Entretanto, também foram pontuados alguns problemas. Os problemas apontados possuem relação com a introdução desses conhecimentos adquiridos aos novos membros e sua continuidade durante os anos. Futuramente, é preciso

considerar os problemas mencionados pelos especialistas na pesquisa, para conseguir fazer um caso de estudo detalhado ao aplicar a abordagem RISE-CBD na empresa. Dessa forma, será preciso refatorar os dados catalogados pelos artefatos em ferramentas e estruturas mais intuitivas ou que funcionem para os membros da BIT. Para outros trabalhos futuros, também é possível realizar a abordagem RISE-DA em outros contextos de empresas júnior voltadas para tecnologia, como as empresas citadas no capítulo 7.

9.1 Lições Aprendidas

Ao aplicar a abordagem RISE-DA em uma empresa júnior, existem algumas especificidades que foram notadas. A necessidade de reuniões é uma parte essencial da abordagem e a experiência da pesquisa concluiu que comunicações a distância não funcionam bem para conseguir catalogar os dados necessários. Além disso, os estudantes não tem um vínculo empregatício forte com a empresa, uma vez possuem como prioridade as ocupações da faculdade e, por conta disso, o tempo dos estudantes é escasso. Comparando com as empresas no mercado de trabalho que não fazem parte desse cenário, a criação de times que trabalham apenas na parte de análise de domínio da empresa não é algo viável para uma empresa júnior ou uma *very small entity*, já que não existem setores de trabalho e todos fazem mais de uma atividade.

Durante a pesquisa ocorreu muita rotatividade de alunos (como já era esperado), mas o processo seletivo trouxe alunos de diversos períodos da faculdade que possuíam níveis de conhecimentos diferentes. Essa alta rotatividade foi o fator responsável por não garantir que todos os desenvolvedores presentes pudessem participar de algumas etapas da pesquisa, uma vez que não teriam nenhuma contribuição já que não possuíam conhecimento dos projetos que estavam sendo analisados, da cultura da empresa ou mesmo da tecnologia utilizada pelas aplicações.

Além disso, a pesquisa funciona apenas se existir algum especialista do domínio ativo na empresa (alguém que desenvolveu projetos) e uma boa documentação, o que não aconteceu com 9 de 14 dos projetos analisados nesse trabalho. A participação dos especialistas do domínio são essenciais, pois auxiliam ativamente em todas as etapas presentes na abordagem e ainda conseguem apontar onde seria interessante incrementá-la no contexto das *features* principais do domínio. Isso aconteceu ao separar *features* principais em uma mesma *sprint* por conta das suas similaridades já observadas em alto nível, o que facilitou o processo de coleta de *subfeatures*. As análises para outros produtos que não possuíam código ou documentação estruturada foi baseado em como os desenvolvedores programariam as *features* e *subfeatures* caso recebessem os requisitos daquele produto.

Dessa forma, para aplicar a abordagem RISE-DA em uma empresa júnior é necessário que exista conhecimento vindo de pessoas que estão há muito tempo na empresa ou de uma boa documentação que foi sendo gerada desde a introdução da empresa no mercado. Pela alta rotatividade de alunos, existem dois paradigmas a serem pensados para os trabalhos futuros:

como manter o conhecimento adquirido entre estudantes e como manter boas práticas, tanto das tecnologias utilizadas, padrões de documentação, etapas do RISE-DA e utilização dos códigos com reuso depois que o processo RISE-CBD fosse implementado na empresa. Dessa forma, caso fosse necessário aplicar a abordagem RISE-DA na empresa novamente, o processo seria facilitado por conta dessas práticas.

Referências

- ABES. *Mercado Brasileiro de Software: Panorama e Tendências*. 2020. Disponível em: <https://abessoftware.com.br/wp-content/uploads/2020/10/ABES-EstudoMercadoBrasileirodeSoftware2020.pdf>. Acesso em: 2022-03-12. Citado na página 12.
- AMANDA, I. G. B. *Artefatos criados a partir da árvore de features*. 2022. Disponível em: https://drive.google.com/drive/folders/1_UgErnraS5SXsLMM92GbsNiKZQbZNdeB?usp=sharing. Acesso em: 15 jul 2022. Citado 6 vezes nas páginas 69, 75, 81, 88, 98 e 107.
- CHEN, K. et al. An approach to constructing feature models based on requirements clustering. *13th IEEE International Conference on Requirements Engineering*, p. 31–40, 2005. Citado na página 19.
- DEVMEDIA. *SOA com SoaML: Análise e modelagem de serviços*. 2012. Disponível em: <https://www.devmedia.com.br/soa-com-soaml-analise-e-modelagem-de-servicos/24101>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.
- DEVMEDIA. *IFPUG SNAP: medindo requisitos não funcionais*. 2014. Disponível em: <https://www.devmedia.com.br/ifpug-snap-medindo-requisitos-nao-funcionais/30204>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.
- DEVMEDIA. *RE-Tools: Ferramenta para elicitação de requisitos*. 2015. Disponível em: <https://www.devmedia.com.br/re-tools-ferramenta-para-elicitaao-de-requisitos/31990>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.
- ERIKSSON, M.; BÖRSTLER, J.; BORG, K. Managing requirements specifications for product lines - an approach and industry case study. *J. Syst. Softw.*, Elsevier Science Inc., USA, v. 82, n. 3, p. 435–447, mar 2009. ISSN 0164-1212. Disponível em: <https://doi.org/10.1016/j.jss.2008.07.046>. Citado 4 vezes nas páginas 14, 15, 17 e 18.
- EUROPEAN SPACE AGENCY. *Guide to applying the ESA software engineering standards to small software projects*. EUA, 1996. Citado na página 28.
- FEATUREIDE. *FeatureIDE: the only eclipse plugin for feature oriented software development*. 2020. Disponível em: <https://www.featureide.de/>. Acesso em: 26 jun 2022. Citado 2 vezes nas páginas 52 e 68.
- GITHUB, U. do. *A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide)*. 2020. Disponível em: <https://github.com/osrmt/osrmt>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.
- GURP, J. van; BOSCH, J.; SVAHNBERG, M. On the notion of variability in software product lines. In: *Proceedings Working IEEE/IFIP Conference on Software Architecture*. [S.l.: s.n.], 2001. p. 45–54. Citado 4 vezes nas páginas 14, 15, 18 e 20.
- HAYASHI, K.; AOYAMA, M.; KOBATA, K. Agile tames product line variability: An agile development method for multiple product lines of automotive software systems. *Association for Computing Machinery*, p. 25–29, 2017. Citado na página 27.

IIBA. *A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide)*. 2022. Disponível em: <https://www.iiba.org/career-resources/a-business-analysis-professionals-foundation-for-success/babok/>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.

INC, W. *Simples. Seguro. Troque mensagens com confiança*. 2009. Disponível em: https://www.whatsapp.com/?lang=pt_br. Acesso em: 12 jul 2022. Citado 5 vezes nas páginas 60, 61, 63, 88 e 137.

JOHN, I.; EISENBARTH, M. A decade of scoping: A survey. In: *Proceedings of the 13th International Software Product Line Conference*. USA: Carnegie Mellon University, 2009. (SPLC '09), p. 31–40. Citado 2 vezes nas páginas 11 e 29.

JOHN, I. et al. A practical guide to product line scoping. In: *10th International Software Product Line Conference (SPLC'06)*. [S.l.: s.n.], 2006. p. 3–12. Citado 6 vezes nas páginas 11, 14, 15, 20, 25 e 26.

JÚNIOR, B. *Censo e Identidade*. 2018. Disponível em: <https://drive.google.com/file/d/1VH7guSPRyYCLiFzPEMftx1CzNVcBZCyx/view>. Acesso em: 06 mar 2022. Citado na página 32.

JÚNIOR, B. *Código de ética do movimento empresa júnior*. 2021. Disponível em: <https://drive.google.com/file/d/1SQD97cEDJjVnxPzmWAebYp8Xlb8dn3el/view>. Acesso em: 06 mar 2022. Citado 2 vezes nas páginas 12 e 31.

JÚNIOR, B. *jeitoBJ*. 2021. Disponível em: https://drive.google.com/file/d/1wrSedNQRCajc41WrZOo_wKL7JxhQsnf-/view. Acesso em: 06 mar 2022. Citado na página 32.

JÚNIOR, B. *Conheça o MEJ*. 2022. Disponível em: <https://brasiljunior.org.br/conheca-o-mej>. Acesso em: 06 mar 2022. Citado na página 32.

KANBANIZE. *Ferramentas de Gestão de Projeto Kanban - Por que e Como Começar*. 2022. Disponível em: <https://kanbanize.com/pt/recursos-kanban/guia-do-software-kanban/ferramentas-de-gestao-de-projeto-kanban>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.

KANG, K. et al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Pittsburgh, PA, USA, 1990. Citado na página 19.

KIANI, A. A. et al. A dynamic variability management approach working with agile product line engineering practices for reusing features. *Springer*, p. 8391–8432, 2021. Citado 2 vezes nas páginas 13 e 24.

KLÜNDER, J.; HOHL, P.; SCHNEIDER, K. Becoming agile while preserving software product lines. *ICSSP*, p. 26–27, 2018. Citado na página 26.

LAPORTE, C. Y.; O'CONNOR, R. V.; PAUCAR, L. H. G. Software engineering standards and guides for very small entities implementation in two start-ups. In: *2015 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*. [S.l.: s.n.], 2015. p. 5–15. Citado 2 vezes nas páginas 28 e 29.

MAJCHROWSKI, A. et al. Software development practices in small entities: an iso29110-based survey. *Journal of Software: Evolution and Process*, p. 990–999, 2016. Citado na página 12.

- MÁRCIO, V. d. S.; WESLEY, K. G. A.; IVONEI, F. d. S. Engenharia de software em empresas de pequeno e médio porte: Um mapeamento sistemático. In: *Anais da IV Escola Regional de Engenharia de Software*. Porto Alegre, RS, Brasil: SBC, 2020. p. 134–143. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/eres/article/view/13724>>. Citado na página 30.
- MERMAID. *About Mermaid*. 2022. Disponível em: <<https://mermaid-js.github.io/mermaid/#/>>. Acesso em: 10 jul 2022. Citado 3 vezes nas páginas 109, 111 e 113.
- MOLL, R. *Being prepared - a bird's eye view of smes and risk management*. 2013. Disponível em: <<https://www.iso.org/files/live/sites/isoorg/files/news/magazine/ISO>>. Acesso em: 12 mar 2022. Citado na página 12.
- NETO, J. S. *Brazilian Software Market*. 2021. Disponível em: <https://abesssoftware.com.br/wp-content/uploads/2021/08/Jorge_Sukarie_ABES-Estudo-Mercado-Brasileiro-de-Software-2021.pdf>. Acesso em: 2022-03-01. Citado na página 30.
- NUNEZ, L. T. *Software Engineering in Small Projects: The Most Essential Processes*. Tese (Tese de Doutorado) — University of Miami, Coral Gables, Florida, May 2014. Citado na página 28.
- O'CONNOR, R.; COLEMAN, G. Ignoring “best practice”: Why irish software smes are rejecting cmmi and iso 9000. *Australasian Journal of Information Systems*, 2009. Citado na página 12.
- POHL, K.; BÖCKLE, G.; LINDEN, F. J. *Software Product Line Engineering*. Secaucus, NJ, USA: Springer-Verlag, 2005. Citado 3 vezes nas páginas 17, 20 e 45.
- QUMER, A.; HENDERSON-SELLERS, B. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Elsevier*, p. 280–295, 2007. Citado na página 40.
- ROUSSEFF, D. *LEI Nº 13.267, DE 6 DE ABRIL DE 2016*. 2016. Disponível em: <https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/21173868/do1-2016-04-07-lei-n-13-267-de-6-de-abril-de-2016-21173742>. Acesso em: 06 mar 2022. Citado na página 31.
- SÁNCHEZ-GORDÓN, M.; O'CONNOR, R. V. Understanding the gap between software process practices and actual practice in very small companies. *Springer Science+Business Media*, p. 549–570, 2015. Citado na página 28.
- SÁNCHEZ-GORDÓN, M. L.; O'CONNOR, R. V. Understanding the gap between software process practices and actual practice in very small companies. *Software Quality Journal*, p. 549–570, 2016. Citado na página 12.
- SANCHEZ-GORDON, S.; SÁNCHEZ-GORDÓN, M.-L.; LUJÁN-MORA, S. Towards an engineering process for developing accessible software in small software enterprises. In: *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering*. Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda, 2016. (ENASE 2016), p. 241–246. ISBN 9789897581892. Disponível em: <<https://doi.org/10.5220/0005900702410246>>. Citado na página 31.
- SCHAEFER, M. F.; DALSSASSO, C. P.; MATOS, M. G. *Estatuto Social*. 2016. Documento interno. Citado 2 vezes nas páginas 32 e 33.

SCHMID, K. A comprehensive product line scoping approach and its validation. In: *Proceedings of the 24th International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2002. (ICSE '02), p. 593–603. ISBN 158113472X. Disponível em: <<https://doi.org/10.1145/581339.581415>>. Citado na página 26.

SCHWABER, K.; SUTHERLAND, J. *SCRUM Development Process*. London: Springer London, 1995. 117-134 p. ISBN 978-1-4471-0947-1. Citado na página 40.

SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An empirical methodology for introducing software processes. In: *Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2001. (ESEC/FSE-9), p. 288–296. ISBN 1581133901. Disponível em: <<https://doi.org/10.1145/503209.503248>>. Citado na página 49.

SILVA, I. F. da. An agile approach for software product lines scoping. *SPLC*, p. 225–228, 2012. Citado na página 52.

SILVA, I. F. da. *A Scrum-inspired Process for Software Product Lines Scoping*. Tese (Tese de Doutorado) — Federal University of Pernambuco, Recife, Pernambuco, October 2013. Citado 21 vezes nas páginas 3, 12, 13, 15, 19, 21, 22, 24, 28, 35, 36, 38, 39, 40, 41, 42, 44, 46, 52, 136 e 144.

SNORRE, G. *Practices of Agile Software ProductLine Engineering: A qualitative assessment of empirical studies*. Dissertação (Dissertação de Mestrado) — Norwegian University of Science and Technology, February 2009. Citado na página 27.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. Rio de Janeiro: Pearson Education, Inc., 2011. Citado 5 vezes nas páginas 13, 14, 15, 16 e 17.

VALE, T. et al. Splice: A lightweight software product line development process for small and medium size projects. *IEEE Computer Society*, p. 42–52, 2014. Citado 2 vezes nas páginas 13 e 23.

VALENTE, M. T. *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. [S.l.]: UmLivro, 2020. Citado 4 vezes nas páginas 40, 41, 43 e 144.

WOHLIN, C.; AURUM, A. Towards a decision-making structure for selecting a research design in empirical software engineering. *Springer Science+Business Media New York*, p. 1427–1455, 2014. Citado na página 47.

WORDPRESS. *Conheça o WordPress*. 2022. Disponível em: <<https://br.wordpress.org/>>. Acesso em: 06 jul 2022. Citado 2 vezes nas páginas 54 e 55.

Glossário

Os especialistas da abordagem RISE-DA estão presentes tanto na parte 1, quanto na parte 2 do processo e são essenciais para a construção de todos os artefatos.

Desenvolvedores Compostos por todos os especialistas necessários para que o projeto seja executado eficientemente, incluindo desenvolvedores *front-end* e *back-end* que tomam decisões técnicas sobre o projeto (VALENTE, 2020). Dentro da BIT - Empresa Júnior de Ciência da computação, o papel de desenvolvedor foi atribuído a todos os membros que não faziam parte da diretoria executiva da empresa ou que estavam como *trainees*.

Especialista do Domínio Responsável por conhecer o domínio que o produto foi ou será construído. Possui uma visão abrangente sobre os produtos e conhecimento técnico relacionado a elas (SILVA, 2013). Dentro da BIT - Empresa Júnior de Ciência da computação, o papel de especialista do domínio foi atribuído ao desenvolvedor/membro da diretoria que possuía mais conhecimento técnico e dos requisitos sobre a aplicação analisada.

Especialista do negócio Possui conhecimento sobre o mercado em que a organização atua, incluindo suas atividades internas, além de seus objetivos e estratégias de negócio (SILVA, 2013). Dentro da BIT - Empresa Júnior de Ciência da computação, o papel de especialista do negócio foi atribuído a membro que possuía o maior tempo de serviço dentro da empresa ou para algum dos tutores responsáveis pela empresa.

Especialista em análise do domínio Responsável por conduzir todas as atividades do processo de análise de domínio e auxilia na definição do escopo de domínio das aplicações da organização (SILVA, 2013). Dentro da BIT - Empresa Júnior de Ciência da computação, o papel de especialista em análise de domínio foi atribuído ao presidente da empresa, pois possuía o conhecimento da abordagem, tinha acesso as documentações e o repositório privado, além de coordenar as reuniões semanais da empresa.

Scrum Master Responsável por controlar as regras do método Scrum durante o processo e treinar os membros do time de desenvolvimento, além de continuamente facilitar qualquer impedimento que aconteça no projeto (VALENTE, 2020). Dentro da BIT - Empresa Júnior de Ciência da computação, o papel de Scrum Master foi atribuído ao presidente da empresa, pois era ele quem criava as pautas semanais e distribuía os trabalhos entre os times da empresa.