



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

WELINGTON MOTHÉ DE OLIVEIRA

**PROPOSTA DE UMA METODOLOGIA DE DESENVOLVIMENTO DISTRIBUÍDO  
DE SOFTWARE PARA INSTITUTOS FEDERAIS DE EDUCAÇÃO DO BRASIL**

Recife

2022

WELINGTON MOTHÉ DE OLIVEIRA

**PROPOSTA DE UMA METODOLOGIA DE DESENVOLVIMENTO DISTRIBUÍDO  
DE SOFTWARE PARA INSTITUTOS FEDERAIS DE EDUCAÇÃO DO BRASIL**

Dissertação apresentada ao Programa de Pós Graduação em Ciência da Computação da Universidade Federal de Pernambuco, Centro Acadêmico de Informática, como requisito para a obtenção do título de Mestre em Ciências da Computação. Área de concentração: Sistemas de Informação

Orientador: Alexandre Marcos de Lins Vasconcelos.

Recife

2022

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

O48p Oliveira, Welington Mothé de  
Proposta de uma metodologia de desenvolvimento distribuído de software para institutos federais de educação do Brasil / Welington Mothé de Oliveira. – 2022.  
164 f.: il., fig., tab.

Orientador: Alexandre Marcos de Lins Vasconcelos.  
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022

Inclui referências, glossário e apêndices.

1. Sistemas de informação. 2. Engenharia de software. 3. Desenvolvimento de software. I. Vasconcelos, Alexandre Marcos de Lins (orientador). II. Título.

004 CDD (23. ed.) UFPE - CCEN 2022-164

WELINGTON MOTHÉ DE OLIVEIRA

**PROPOSTA DE UMA METODOLOGIA DE DESENVOLVIMENTO DISTRIBUÍDO  
DE SOFTWARE PARA INSTITUTOS FEDERAIS DE EDUCAÇÃO DO BRASIL**

Dissertação apresentada ao Programa de Pós Graduação em Ciência da Computação da Universidade Federal de Pernambuco, Centro Acadêmico de Informática, como requisito para a obtenção do título de Mestre em Ciências da Computação. Área de concentração: Sistemas de Informação.

Aprovado em: 20/07/2022.

**BANCA EXAMINADORA**

---

Prof. Dr. Alexandre Vasconcelos (Orientador)  
Universidade Federal de Pernambuco - UFPE

---

Prof. Dr. Vinicius Cardoso Garcia (Examinador Interno)  
Universidade Federal de Pernambuco - UFPE

---

Prof. Dr. Rafael Prikladnicki (Examinador Externo)  
Pontifícia Universidade Católica do Rio Grande do Sul

## AGRADECIMENTOS

À Universidade Federal de Pernambuco, em especial ao Centro de Informática, que nos recebeu e nos proporcionou aprendizado, capacitação, experiências científicas, e acima de tudo, a oportunidade de conquistar mais um grau acadêmico.

Ao Instituto Federal de Educação, Ciência e Tecnologia, Campus São Mateus: seguramente, podemos afirmar que todo este processo só foi possível e realizado graças a esta instituição.

Ao Professor orientador Alexandre Marcos Lins de Vasconcelos pelas orientações, mais que direcionamentos, e sim, ensinamentos que certamente contribuíram e contribuirão para o meu crescimento profissional e pessoal.

A empresa Whimsical, que possibilitou a construção de todas as representações gráficas geradas para este trabalho, seguramente, podemos afirmar que a qualidade de representação visual apresentada não seria a mesma sem essa ferramenta web sem precedentes.

Ao meu pai, exemplo de sabedoria inestimável, ensinamentos que me fizeram o homem que hoje sou.

À minha mãe, meu porto seguro, por todos os momentos em que cuidou de cada detalhe para garantir a qualidade de vida que eu e meus irmãos podemos usufruir.

À minha esposa, que me incentiva, ajuda e fornece inspiração para tudo em minha vida. E que especificamente nesse trabalho me apoiou e ajudou mais do que nunca.

Aos meus filhos Luísa e Ravi, fonte de uma inspiração e força inexplicável por meras palavras.

Ao meu irmão e irmã pelo apoio e encorajamento que sempre me deram em cada momento de minha vida.

Aos colegas de trabalho que supriram minha ausência nos momentos mais complicados.

Aos colegas do curso, principalmente, Clayton, Eliandro, Jobson, José Fernando, Leonardo, Marcelo, Rogério e Wellyson, que nos mantiveram unidos e com muito incentivo para a conclusão deste projeto.

À equipe tabajara, Carlos, Diego, Gustavo e Lucas, que me acompanharam a cada passo dessa jornada, compartilhando momentos de dificuldades e alegrias, formando uma união inesquecível, tão forte quanto uma família, estarão sempre em

minha vida.

Não existem agradecimentos especiais: a contribuição de todos que foram fundamentais para a realização deste sonho. Todos ofereceram contribuição, à sua maneira e tempo! Sem estas contribuições, este presente trabalho não teria sido realizado. A todos vocês, instituições e pessoas queridas, minha mais sincera gratidão!

Muito obrigado!

*“O fracasso deve ser encarado como um obstáculo de curto prazo para benefícios de longo prazo”. (JHA; VILARDELL; NARAYAN, 2016)*

## RESUMO

Com a crescente utilização de sistemas computacionais a demanda por desenvolvimento e manutenção de software tem se tornado cada vez maior, sobrecarregando assim pequenas equipes centralizadas. O desenvolvimento de software tem evoluído de acordo com surgimento de novas tecnologias e a exigência do mercado cada vez mais globalizado, forçando muitas empresas a distribuir suas equipes de desenvolvimento de software com o objetivo de aumentar sua produtividade. Atualmente é possível listar muitos casos em que grandes empresas multinacionais deixaram de ter uma equipe centralizada para usufruir dos benefícios de múltiplas equipes trabalhando em países com fuso horários distintos e produzindo software de maneira ininterrupta. Porém apesar de muitos benefícios o modelo de Desenvolvimento Distribuído de Software (DDS) também traz um grau de complexidade muito maior, podendo levar a elevado custo ou a perda de produtividade se mal implantado. A proposta apresentada neste trabalho foi fundamentada nos resultados de uma Revisão Sistemática da Literatura (RSL), realizada em conjunto com pesquisa ad-hoc para compreender como o tema tem sido abordado na literatura e um estudo de campo qualitativo baseado em dois questionários executados em conjunto com entrevistas não estruturadas realizadas de forma virtual para compreender o cenário de desenvolvimento de software dos Institutos Federais de Educação do Brasil (IFs). O cenário dos IFs apontou que apesar de possuir uma estrutura distribuída entre reitoria e campi, com equipes de TI em cada uma dessas localidades, não existe uma coordenação entre as equipes para um trabalho em conjunto. O que tem como consequência o problema de desenvolvimento paralelo de projetos de mesmo objetivo. Neste contexto, esta pesquisa tem como objetivo a elaboração de uma proposta de metodologia de DDS que contemple a coordenação de todos os indivíduos de forma integrada em uma única equipe institucional que possa trabalhar todos os projetos de forma totalmente remota. Neste sentido, a principal contribuição desta pesquisa é uma readequação da organização dos profissionais de TI dos IFs de modo a eliminar o desenvolvimento paralelo de projetos de mesmo objetivo. Por fim, um Grupo Focal avaliou a proposta.

**Palavras-chave:** desenvolvimento distribuído de software; institutos federais de educação; engenharia de software baseada em evidências.

## ABSTRACT

With the increasing use of computer systems, the demand for software development and maintenance has become increasing, thus overloading small centralized teams. Software development has evolved according to the emergence of new technologies and the demand of an increasingly globalized market, forcing many companies to distribute their software development teams to increase their productivity. It is currently possible to list many cases in which large multinational companies no longer have a centralized team to enjoy the benefits of multiple teams working in countries with different time zones and producing software uninterruptedly. However, despite many benefits, the Distributed Software Development (DSD) model also brings a much greater degree of complexity, which can lead to high cost or loss of productivity if poorly implemented. The proposal presented in this work was based on the results of a Systematic Literature Review (RSL), carried out together with ad-hoc research to understand how the topic has been approached in the literature and a qualitative field study based on two questionnaires carried out together with unstructured interviews carried out in a virtual way to understand the software development scenario of the Federal Institutes of Education in Brazil (IFs). The scenario of the IFs pointed out that despite having a structure distributed in rectory and campuses, with IT teams in each of these locations, there is no coordination between the teams to work together. This leads to the problem of parallel development of projects with the same objective. In this context, this research aims to develop a proposal for a DSD methodology that includes the coordination of all individuals in an integrated way in a single institutional team that can work on all projects completely remotely. In this sense, the main contribution of this research is a readjustment of the organization of the IT professionals of the IFs to eliminate the parallel development of projects with the same objective. Finally, a Focus Group evaluated the proposal.

**Keywords:** distributed software development; federal institutes of education; evidence-based software engineering.

## LISTA DE FIGURAS

Figura 1 –	Mapa nacional da Rede Federal .....	18
Figura 2 –	Estrutura da Rede Federal e Institutos Federais .....	20
Figura 3 –	Estrutura de TI dos Institutos Federais.....	20
Figura 4 –	Processo da pesquisa .....	24
Figura 5 –	Etapas da pesquisa.....	25
Figura 6 –	Camadas da engenharia de software.....	28
Figura 7 –	Ambientes e contextos .....	31
Figura 8 –	Níveis de dispersão dos indivíduos .....	34
Figura 9 –	Dispersão dos stakeholders do IFSP .....	55
Figura 10 –	Dispersão dos stakeholders do projeto COVID-19.....	59
Figura 11 –	Papéis do desenvolvimento de software .....	79
Figura 12 –	Fluxo de trabalho da metodologia DDS Soberano .....	92
Figura 13 –	Exemplo de planilha de seleção de estudos .....	119

## LISTA DE TABELAS

Tabela 1 – Equipe responsável pela RSL .....	111
Tabela 2 – Questões de pesquisa.....	112
Tabela 3 – Bases de dados .....	113
Tabela 4 – Critérios de inclusão.....	114
Tabela 5 – Critérios de exclusão.....	114
Tabela 6 – Filtros utilizados nas consultas.....	117
Tabela 7 – Quantidade de estudos retornados por base de dados .....	117
Tabela 8 – Estudos primários.....	120
Tabela 9 – Práticas propostas por cada autor .....	122
Tabela 10 – Metodologias encontradas nos estudos primários .....	129

## LISTA DE QUADROS

Quadro 1 – Classificação dos IFs respondentes.....	50
Quadro 2 – Características do desenvolvimento de software do IFSP .....	52
Quadro 3 – Ferramentas utilizadas no DDS do IFSP.....	53
Quadro 4 – Papéis e responsabilidades no IFSP.....	55
Quadro 5 – Características do projeto Rede Federal e o enfrentamento a COVID-19 .....	59
Quadro 6 – Características do projeto SUAP Colaborativo.....	61
Quadro 7 – Ferramentas planejadas para o projeto SUAP Colaborativo.....	61
Quadro 8 – Práticas coletadas nas pesquisas de campo e literatura.....	67
Quadro 9 – Descrição dos elementos PICOC da pesquisa .....	111
Quadro 10 – Termos de busca .....	113
Quadro 11 – Termos de busca após refinamento de string .....	116
Quadro 12 – String definida para a consulta.....	117
Quadro 13 – Quantitativo de estudos selecionados.....	118
Quadro 14 – Entrevistas com especialistas dos IFs.....	151

## LISTA DE GRÁFICOS

Gráfico 1 – Quantidade de IFs que responderam ao questionário .....	137
Gráfico 2 – Quantidade de respondentes que estão na Reitoria/Campi.....	137
Gráfico 3 – Modelo de desenvolvimento de software utilizado nos IFs .....	138
Gráfico 4 – Pretensão da direção dos IFs em migrar para o modelo DDS .....	139
Gráfico 5 – Melhor modelo de desenvolvimento de software para os IFs .....	139
Gráfico 6 – Institutos que já questionaram a possibilidade de usar o modelo DDS .....	140
Gráfico 7 – Institutos que disponibilizam serviço de hospedagem de sistemas ...	142
Gráfico 8 – Papel da Reitoria no desenvolvimento de software do instituto .....	142
Gráfico 9 – Campi que desenvolvem softwares independente da reitoria .....	143
Gráfico 10 – Campi que desenvolvem softwares junto a reitoria.....	144
Gráfico 11 – Institutos que possuem projeto que utiliza a força de trabalho de Alunos e Professores para melhoria de software institucional .....	144
Gráfico 12 – Frameworks/metodologias/modelos de processo utilizados nos IFs..	145
Gráfico 13 – Ferramentas utilizadas nos IFs .....	146
Gráfico 14 – Papéis de desenvolvimento de software mais utilizados nos IFs.....	147
Gráfico 15 – Institutos que possuem documentação de desenvolvimento de software .....	148
Gráfico 16 – Papéis em que os respondentes já atuaram no contexto do desenvolvimento de software da instituição .....	158
Gráfico 17 – Presentes na transição do modelo centralizado para distribuído .....	159
Gráfico 18 – Avaliação do modelo DDS no IFSP.....	160
Gráfico 19 – Benefícios percebidos na utilização do DDS.....	161
Gráfico 20 – Desafios encontrados na utilização do DDS .....	161
Gráfico 21 – Práticas adotadas pelo IFSP .....	162

## LISTA DE SIGLAS E ABREVIATURAS

CONIF	Conselho Nacional das Instituições da Rede Federal de Educação Profissional, Científica e Tecnológica
DDS	Desenvolvimento Distribuído de Software
ESBE	Engenharia de Software Baseada em Evidências
FORTI	Fórum de Gestores de Tecnologia da Informação
ICGSE	International Conference on Global Software Engineering
Institutos Federais ou IFs	Institutos Federais de Educação, Ciência e Tecnologia
MEC	Ministério da Educação
MSL	Mapeamento Sistemático da Literatura
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
Rede Federal	Rede Federal de Educação Profissional, Científica e Tecnológica
RSL	Revisão Sistemática da Literatura
TI	Tecnologia da Informação

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>18</b>
1.1	MOTIVAÇÃO .....	21
1.2	DEFINIÇÃO DO PROBLEMA .....	22
1.3	OBJETIVOS.....	22
1.4	QUESTÕES .....	23
1.5	METODOLOGIA .....	23
1.6	ESTRUTURA DA DISSERTAÇÃO .....	26
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>28</b>
2.1	PROCESSO DE DESENVOLVIMENTO DE SOFTWARE.....	28
2.2	DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE .....	30
<b>2.2.1</b>	<b>Características .....</b>	<b>31</b>
<b>2.2.2</b>	<b>Níveis de dispersão dos indivíduos .....</b>	<b>33</b>
<b>2.2.3</b>	<b>Esforço de alinhamento .....</b>	<b>34</b>
2.3	ORGANIZAÇÃO DE PESSOAS .....	34
<b>2.3.1</b>	<b>Definição de organização.....</b>	<b>35</b>
<b>2.3.2</b>	<b>Mindset .....</b>	<b>35</b>
<b>2.3.3</b>	<b>Cultura organizacional .....</b>	<b>36</b>
2.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	37
<b>3</b>	<b>REVISÃO SISTEMÁTICA DA LITERATURA.....</b>	<b>39</b>
3.1	QUESTÕES DE INVESTIGAÇÕES ESPECÍFICAS .....	39
3.2	SELEÇÃO DOS ESTUDOS PRIMÁRIOS.....	40
3.3	ANÁLISE DOS RESULTADOS.....	40
3.4	LIMITAÇÕES E DIFICULDADES ENCONTRADAS .....	44
3.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	45
<b>4</b>	<b>CENÁRIO DOS INSTITUTOS FEDERAIS .....</b>	<b>46</b>
4.1	PESQUISA DE CAMPO.....	46
<b>4.1.1</b>	<b>Processo de desenvolvimento de software nos IFs .....</b>	<b>46</b>
<b>4.1.2</b>	<b>Entrevistas não estruturadas e questionário sobre DDS nos IFs .....</b>	<b>47</b>
4.2	RESULTADOS DA PESQUISA DE CAMPO .....	48
<b>4.2.1</b>	<b>Nível de dispersão dos <i>stakeholders</i> dos IFs .....</b>	<b>49</b>
<b>4.2.2</b>	<b>Instituto Federal de São Paulo (IFSP) .....</b>	<b>52</b>
4.2.2.1	Ferramentas.....	53

4.2.2.2	Dispersão dos stakeholders.....	54
4.2.2.3	Papéis.....	55
4.2.2.4	Fluxo de desenvolvimento.....	56
4.2.2.5	Fluxo de comunicação.....	57
<b>4.2.3</b>	<b>Projeto Rede Federal e o enfrentamento a COVID-19.....</b>	<b>58</b>
<b>4.2.4</b>	<b>Projeto SUAP Colaborativo - Módulo Projetos de Ensino.....</b>	<b>60</b>
<b>4.2.5</b>	<b>Projeto SIG Colaborativo.....</b>	<b>62</b>
4.3	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	62
<b>5</b>	<b>ELABORAÇÃO DA PROPOSTA.....</b>	<b>64</b>
5.1	PRÁTICAS QUE INSPIRARAM A METODOLOGIA.....	65
<b>5.1.1</b>	<b>Categorias de práticas.....</b>	<b>65</b>
<b>5.1.2</b>	<b>Tipos de práticas.....</b>	<b>65</b>
<b>5.1.3</b>	<b>Práticas reunidas.....</b>	<b>66</b>
5.2	METODOLOGIA DE DDS SOBERANO.....	76
<b>5.2.1</b>	<b>Requisitos.....</b>	<b>76</b>
<b>5.2.2</b>	<b>Objetivos.....</b>	<b>77</b>
<b>5.2.3</b>	<b>Papéis.....</b>	<b>78</b>
5.2.3.1	Dono da Comunicação.....	79
5.2.3.2	Dono do Conhecimento.....	80
5.2.3.3	Demandantes.....	80
5.2.3.4	Desenvolvedores.....	80
5.2.3.5	Líder Técnico.....	81
5.2.3.6	Dono do Produto.....	82
5.2.3.7	Time de Projeto.....	82
5.2.3.8	Dono do Processo.....	83
5.2.3.9	Dono de Mudanças.....	83
5.2.3.10	Influenciadores.....	84
<b>5.2.4</b>	<b>Artefatos.....</b>	<b>84</b>
5.2.4.1	Regras do Jogo.....	84
5.2.4.2	Plano de Comunicação Interna.....	85
5.2.4.3	Briefing do Produto.....	85
5.2.4.4	Plano de Entregas do Produto.....	86
5.2.4.5	Backlog de Produto Priorizado.....	86
5.2.4.6	Backlog de Defeitos.....	87

5.2.4.7	Backlog de Processos.....	87
5.2.4.8	Backlog de Mudanças.....	87
<b>5.2.5</b>	<b>Cerimônias .....</b>	<b>88</b>
5.2.5.1	Dança das Cadeiras .....	88
5.2.5.2	Reunião de Idealização do Produto .....	88
5.2.5.3	Reunião de Planejamento do Produto .....	89
5.2.5.4	Reunião de Entrega do Produto.....	89
<b>5.2.6</b>	<b>Atividades.....</b>	<b>89</b>
5.2.6.1	Refinar Backlog de Produto .....	90
5.2.6.2	Desenvolver Itens Priorizados .....	90
5.2.6.3	Revisar Itens Codificados .....	90
<b>5.2.7</b>	<b>Modelo do processo .....</b>	<b>91</b>
<b>5.2.8</b>	<b>Fluxo de trabalho .....</b>	<b>91</b>
<b>5.2.9</b>	<b>Fluxo de papéis.....</b>	<b>93</b>
<b>5.2.10</b>	<b>Diretrizes de comunicação.....</b>	<b>93</b>
<b>5.2.11</b>	<b>Coordenação .....</b>	<b>95</b>
5.3	RECOMENDAÇÕES .....	96
5.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	97
<b>6</b>	<b>AVALIAÇÃO DA PROPOSTA .....</b>	<b>98</b>
6.1	CONDUÇÃO .....	98
6.2	RESULTADOS.....	99
6.3	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	102
<b>7</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>103</b>
7.1	CONTRIBUIÇÕES DA PESQUISA.....	103
7.2	LIMITAÇÕES DO ESTUDO .....	103
7.3	TRABALHOS FUTUROS.....	104
7.4	CONCLUSÕES.....	104
	<b>REFERÊNCIAS .....</b>	<b>106</b>
	<b>GLOSSÁRIO .....</b>	<b>110</b>
	<b>APÊNDICE A – REVISÃO SISTEMÁTICA – PROTOCOLO.....</b>	<b>111</b>
	<b>APÊNDICE B – REVISÃO SISTEMÁTICA – CONDUÇÃO.....</b>	<b>116</b>
	<b>APÊNDICE C – QUESTIONÁRIO SOBRE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NOS INSTITUTOS FEDERAIS.</b>	<b>131</b>
	<b>APÊNDICE D – ENTREVISTAS NÃO ESTRUTURADAS .....</b>	<b>151</b>

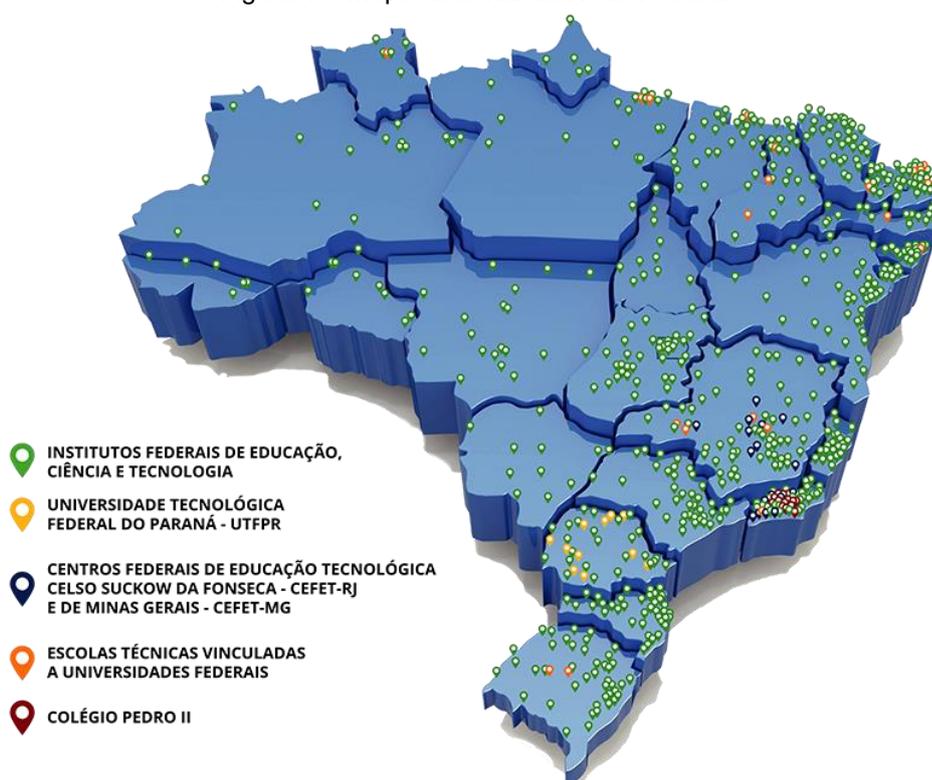
<b>APÊNDICE E – QUESTIONÁRIO SOBRE PROCESSO DE DDS NOS INSTITUTOS FEDERAIS.....</b>	<b>153</b>
<b>APÊNDICE F – ROTEIRO PARA CONDUÇÃO DO GRUPO FOCAL .....</b>	<b>163</b>

## 1 INTRODUÇÃO

Este capítulo tem a finalidade de introduzir o problema abordado para o entendimento do leitor. Para tanto, será apresentada a motivação para este estudo, a definição do problema que levou a pesquisa, os objetivos pré-estabelecidos, a metodologia elaborada para resolução deste problema e a estrutura idealizada para apresentar as informações.

Por meio da Lei 11.892 de 29 de dezembro de 2008 foi instituída a Rede Federal de Educação Profissional, Científica e Tecnológica (Rede Federal) que trata a união de várias instituições de ensino subordinadas à Secretaria de Educação Profissional e Tecnológica do Ministério da Educação (SETEC/MEC).

Figura 1 – Mapa nacional da Rede Federal



Fonte: MEC (2019).

Os Institutos Federais de Educação, Ciência e Tecnologia (Institutos Federais ou IFs), representados na Figura 1 com a cor verde, foram estabelecidos pela mesma lei, são instituições, pluricurriculares e multicampi (reitoria, campus, campus avançado, polos de inovação e polos de educação a distância), especializados na oferta de educação profissional e tecnológica (EPT) em todos os seus níveis e formas

de articulação com os demais níveis e modalidades da Educação Nacional, oferta os diferentes tipos de cursos de EPT, além de licenciaturas, bacharelados e pós-graduação *stricto sensu*. (BRASIL, 2008)

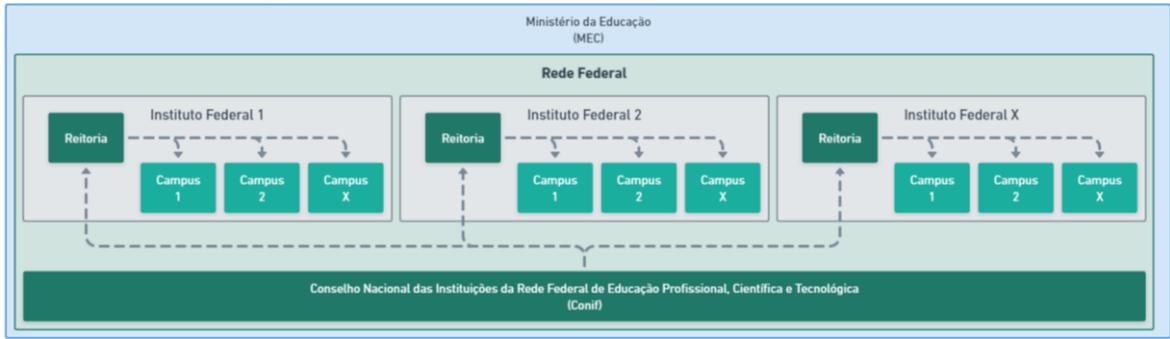
Devido ao fato da Lei 11.892 definir algumas das características estruturais dos Institutos Federais, podemos considerar a estrutura deles muito similares. Sendo que possuem uma estrutura multicampi, formada por uma sede administrativa, Reitoria, e vários Campi distribuídos por várias cidades. Esta lei forneceu ainda uma diferenciação entre os IFs e as Universidades, que é a grande autonomia dos campi perante a sua sede administrativa, tal como é a autonomia dos Institutos Federais perante o MEC.

A grande autonomia concedida, garante o direito de os campi tomarem suas decisões independente da Reitoria, tal como possuir seu próprio CNPJ e realizar suas próprias licitações. Sendo assim, podemos enxergar a Reitoria como uma unidade estratégica e gestora de políticas institucionais, em suas diversas áreas, principalmente em relação ao ensino, à pesquisa e à extensão, seu principal papel é de integração das diversas unidades (Campi) para a promoção de ações conjuntas.

Apesar de a estrutura interna dos IFs ter sido criada com uma unidade de integração (Reitoria) desde sua idealização e planejamento, não foi criada nenhuma estrutura inicial para realizar a integração entre Instituições Federais. Entretanto pouco tempo após a criação da Rede Federal, foi criado o Conselho Nacional das Instituições da Rede Federal de Educação Profissional, Científica e Tecnológica (Conif), que passou a realizar este papel. Ele não faz parte de uma hierarquia da Rede Federal, mas é uma instância de discussão, proposição e promoção de políticas de desenvolvimento da formação profissional e tecnológica, pesquisa e inovação. Com o objetivo de fortalecer e consolidar a Rede Federal de Educação como um todo através da integração de ações estratégicas que promovem o estreitamento das relações entre as instituições participantes e demais instituições parceiras. (Conif, 2020)

Para resumir o entendimento da estrutura da Rede Federal e dos Institutos Federais podemos ver a Figura 2, com a perspectiva de uma hierarquia de comunicação, cooperação e integração, não como uma estrutura de comando e controle.

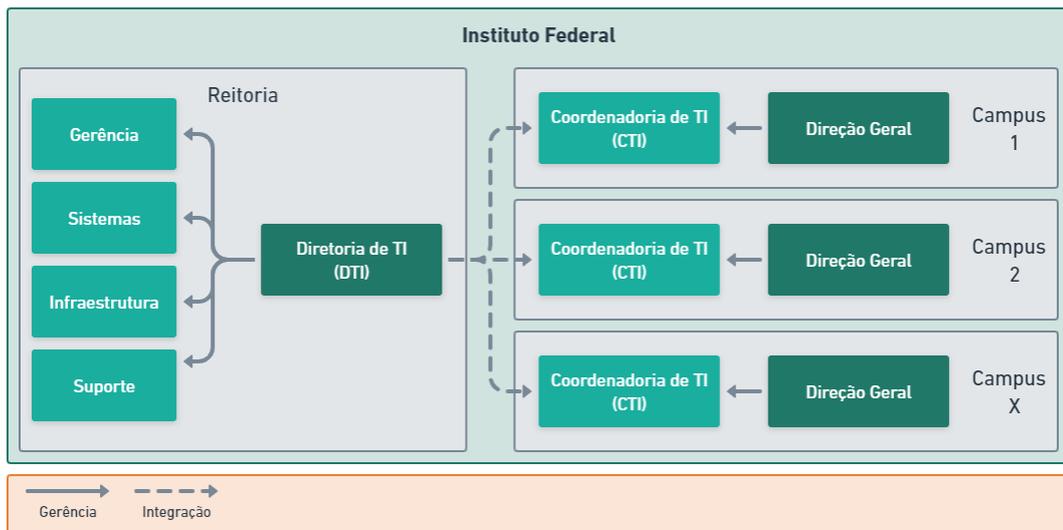
Figura 2 – Estrutura da Rede Federal e Institutos Federais



Fonte: O autor (2022).

Os profissionais de Tecnologia da Informação (TI) dos Institutos Federais, ficam divididos entre a Diretoria de Tecnologia da Informação (DTI) e Coordenadoria de Tecnologia da Informação (CTI). Existe uma CTI por cada campus, sendo cada campus localizado em uma cidade diferente. A DTI que fica localizada na Reitoria, tem o papel de integrar as ações de TI que são relevantes para todos os campi, além de manter a estrutura computacional básica do instituto. Observando a Figura 3 podemos entender melhor a relação entre estes, apesar de as CTIs atuarem diretamente ligadas a DTI de modo a integrar a TI distribuída por todas as unidades de um instituto, a única hierarquia de gerência que a CTI deve responder diretamente é a Direção Geral de cada campus.

Figura 3 – Estrutura de TI dos Institutos Federais



Fonte: O autor (2022).

O entendimento da relação entre DTI e as CTIs é de grande relevância para o estudo apresentado, uma vez que o produto desta pesquisa visa reduzir os problemas de desenvolvimento de software inerentes a esse modelo de trabalho.

## 1.1 MOTIVAÇÃO

Na última década houve um crescente número de empresas de software que passaram a distribuir seus processos de desenvolvimento de forma geográfica, criando assim o Desenvolvimento Distribuído de Software (DDS) (ROCHA, 2010).

Hoje é evidente a alta adesão ao modelo distribuído na iniciativa privada, principalmente quando as empresas são multinacionais, segundo Prado e Ferrari (2014) existe um crescimento do número de empresas que adotam abordagens ágeis juntas de abordagens distribuídas, o que pode ocasionar muitos benefícios como redução de prazos de entrega, menor burocracia no desenvolvimento, maior produtividade e mais eficiência. Todavia, projetos que empregam ambas as abordagens podem ter muitos outros desafios a enfrentar, em sua maioria relacionados a Comunicação, Coordenação e Socialização.

Projetos que empregam ambas as abordagens enfrentam algumas dificuldades como, por exemplo, gerenciar equipes distribuídas e gestão de atividades. Sendo assim, é necessário pesquisar novas formas de adaptar metodologias de desenvolvimento de software às realidades de equipes e projetos. (PRADO e FERRARI, 2014, p. 209).

As principais características que diferenciam o modelo distribuído do tradicional são: dispersão geográfica (distância física); dispersão temporal (diferenças de fuso-horário); e diferenças culturais (idioma, tradições, costumes, normas e comportamento) (AUDY; PRIKLADNICKI, 2007).

As principais características do modelo ágil, é a priorização da comunicação entre indivíduos mais que documentação, além de um dos doze princípios ágeis ser a crença de que o método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face. Sendo uma característica do modelo DDS a distância geográfica, todas as comunicações passam a ser mediadas por ferramentas, o que torna impossível a conversa face a face.

Apesar da complexa dificuldade de alinhar as duas abordagens, devido à dispersão geográfica, também foi essa característica, o principal motivador para a

busca do modelo distribuído de desenvolvimento de software para este estudo, uma vez que, os Institutos Federais possuem uma estrutura fisicamente distribuída, com unidades situadas em municípios distintos e essas unidades possuem recursos e pessoal próprio, possibilitando uma instância de desenvolvimento de softwares em cada uma delas.

## 1.2 DEFINIÇÃO DO PROBLEMA

Os IFs possuem uma constante necessidade de desenvolvimento e manutenção de software, derivada da dinâmica que uma instituição educacional apresenta.

Tomando o Instituto Federal do Espírito Santo (Ifes), como exemplo, local de trabalho do pesquisador, este utiliza o modelo co-localizado ou tradicional, ou seja, sua equipe institucional de desenvolvimento de software fica centralizada na DTI. Essa separação dos campi do processo de desenvolvimento de software, causa um grande problema que é o desenvolvimento paralelo de projetos de mesmo objetivo.

Podemos entender esse problema como uma demanda de desenvolvimento de software, que quando não suprida pela equipe de desenvolvimento de software institucional, é levada ao desenvolvimento não coordenado por parte das CTIs, o produto resultante que deveria beneficiar a todos os campi, acaba por beneficiar apenas um campus. Inclusive o fato de não divulgar uma aplicação desenvolvida por um campus, pode fazer com que outros campi que possuem a mesma necessidade também desenvolvam suas soluções de forma individual.

Podemos destacar ainda outro problema, a falta de padronização dos códigos, uma vez que não existe uma coordenação do desenvolvimento de softwares de forma institucional, os campi passam a desenvolver softwares de forma independente para a resolução de demandas locais, sem nenhuma padronização, o que dificulta a reutilização dos softwares por outros campi.

**Com base no exposto tem-se como principal problema de pesquisa:**  
Como elaborar uma metodologia de Desenvolvimento Distribuído de Software aplicável ao contexto dos Institutos Federais de Educação do Brasil?

## 1.3 OBJETIVOS

Construir uma metodologia de Desenvolvimento Distribuído de Software a partir

de práticas aplicáveis à realidade dos Institutos Federais de Educação do Brasil.

### Objetivos específicos

Os objetivos específicos que contribuirão para atingir o objetivo geral são:

1. Identificar na literatura práticas de DDS;
2. Identificar as características presentes no desenvolvimento de software dos IFs através de uma pesquisa de campos.
3. Identificar dentre as práticas de DDS encontradas, as mais relevantes para à realidade dos IFs;
4. Realizar a união de todas as informações em uma proposta que contemple uma transição simples e eficaz do modelo centralizado para o distribuído;
5. Avaliar o grau de aceitação da proposta pelos profissionais da área;
6. Analisar os pontos positivos e negativos, e relatar as lições aprendidas.

### 1.4 QUESTÕES

Este estudo objetiva responder às seguintes questões de pesquisa:

- Q1:** Quais são as práticas de DDS da literatura?
- Q2:** Quais são as práticas de DDS em utilização nos IFs?
- Q3:** Como propor uma metodologia de DDS a partir das práticas encontradas?

A resolução de **Q1** fica estabelecida a partir de análise de práticas encontradas na Revisão Sistemática da Literatura, enquanto a resolução de **Q2** fica estabelecida a partir do levantamento de dados feito junto às coordenadorias de sistemas dos IFs que já trabalham de forma distribuída.

Enquanto a resolução de **Q3** será definida a partir da análise de todas as práticas encontradas, em conjunto com a perspectiva de utilização no cenário encontrado na pesquisa de campo. Uma vez definida uma metodologia possivelmente aplicável aos IFs, essa proposta será apresentada a um grupo focal para que seja avaliada por servidores de TI de outros IFs.

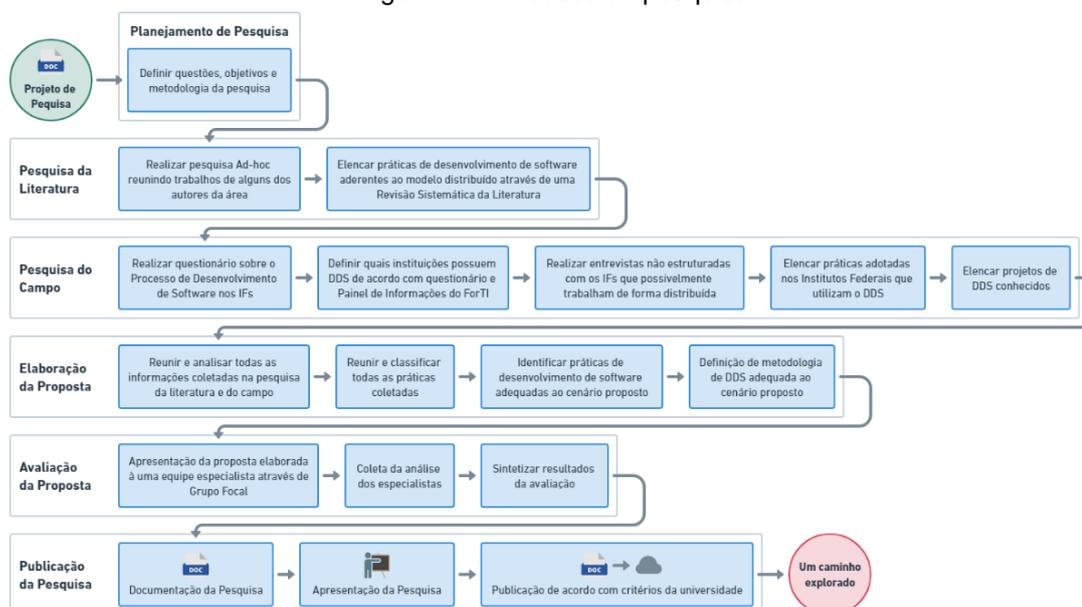
### 1.5 METODOLOGIA

Esta seção apresenta os métodos de pesquisa utilizados para a condução deste trabalho, com objetivo de tornar os resultados mais confiáveis e possíveis de serem reproduzidos por outros pesquisadores.

Esta pesquisa tem a finalidade **Aplicada**, pois busca resolver o problema da forma de trabalhar com equipes geograficamente distribuídas no cenário de desenvolvimento de software dos IFs. Seu objetivo é **Descritivo e Exploratório**, pois além de descrever um conhecimento da área de DDS encontrado na literatura são aplicados métodos de coleta de dados do mundo real no cenário proposto através de questionários e reuniões. Possui uma abordagem **Qualitativa**, pois se baseia nas evidências encontradas nas informações coletadas para elaborar uma proposta.

A Figura 4 ilustra o processo metodológico da pesquisa, desde o planejamento até sua publicação. Resume-se da seguinte forma, após o planejamento, o problema é evidenciado e suas nuances delineadas para identificar uma forma de solucioná-lo.

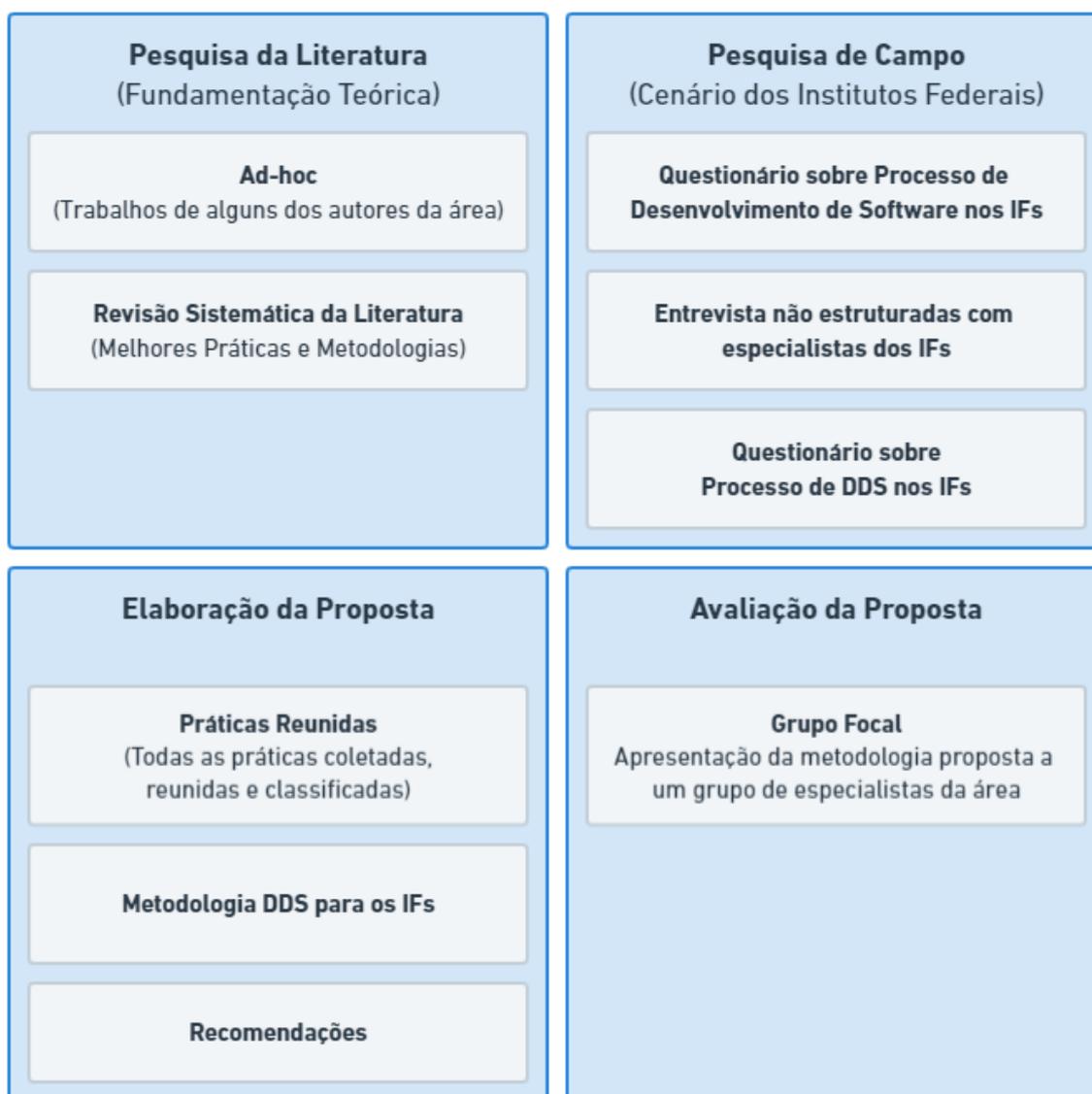
Figura 4 – Processo da pesquisa



Fonte: O autor (2022).

Com o planejamento traçado, o estudo focou nas quatro principais fases da pesquisa, que são, Pesquisa da Literatura, Pesquisa de Campo, Elaboração da Proposta e Avaliação da Proposta. Os produtos destas fases são representados na Figura 5.

Figura 5 – Etapas da pesquisa



Fonte: O autor (2022).

A **primeira etapa**, Pesquisa da Literatura, é voltada para a reunião de informações literárias a respeito das metodologias utilizadas atualmente, práticas relacionadas à área. Essa fundamentação teórica teve como base dois modelos, o *Ad hoc* e a RSL.

A pesquisa *Ad hoc* consiste basicamente na seleção de trabalhos de autores brasileiros com base em guias de referências e recomendações, alguns deles, como Rafael Prikladnicki, têm tratamento diferenciado por serem referenciados em muitos trabalhos. O seu livro “Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas” foi o maior guia para este trabalho.

A Revisão Sistemática da Literatura (RSL) é direcionada à busca de práticas de

DDS mais utilizadas, porém permite agregar um conhecimento maior a respeito de todo o tema, entender um pouco mais de todas as metodologias envolvidas.

A **segunda etapa**, Pesquisa de Campo, é voltada para a coleta de informações a respeito do cenário atual do Desenvolvimento de Software nos Institutos Federais, através de um questionário aberto a todos os profissionais de TI dos IFs, entrevistas não estruturadas e um questionário com especialistas dos IFs que trabalham com DDS.

A **terceira etapa**, Elaboração da Proposta, utiliza a primeira e a segunda etapa para reunir as práticas, que são apresentadas na seção 5.1, é então elaborada uma proposta de Metodologia de Desenvolvimento Distribuído de Software que se adequa a realidade encontrada na segunda etapa.

A **quarta etapa**, Avaliação da Proposta, onde a proposta elaborada deverá ser apresentada a um Grupo Focal, formado por especialistas da área, capazes de realizar uma avaliação profissional e imparcial.

Após a execução de toda a pesquisa é feito um processo de Publicação da Pesquisa que consiste em documentar, apresentar e adequar os documentos aos critérios predefinidos pela plataforma de publicação.

## 1.6 ESTRUTURA DA DISSERTAÇÃO

Além deste capítulo introdutório, este trabalho está estruturado da seguinte maneira:

**Capítulo 2 – Fundamentação Teórica:** Apresenta o referencial teórico do assunto proposto por este trabalho;

**Capítulo 3 – Revisão Sistemática da Literatura:** Apresenta uma pesquisa secundária sobre o tema na literatura, que busca a agregação de informações para a resolução do problema proposto de forma não tendenciosa;

**Capítulo 4 – Cenário dos Institutos Federais:** Apresenta os resultados encontrados na pesquisa de campo através de questionários e entrevistas nos IFs;

**Capítulo 5 – Práticas Reunidas:** Apresenta todas as práticas de desenvolvimento distribuído de software reunidas nos capítulos anteriores;

**Capítulo 6 – Metodologia de DDS para os IFs:** Apresenta a proposta de metodologia de DDS para os IFs;

**Capítulo 7 – Avaliação da Proposta:** Apresenta a avaliação da metodologia

com especialistas dos IFs, através de Grupo Focal;

**Capítulo 8 – Considerações Finais:** Apresenta as considerações finais do trabalho e as lições aprendidas;

Além destes capítulos, o trabalho conta com os seguintes apêndices:

**Apêndice A – Revisão Sistemática – Protocolo:** Apresenta o protocolo de planejamento para a execução da Revisão Sistemática da Literatura;

**Apêndice B – Revisão Sistemática – Condução:** Apresenta a forma com que foi conduzida a Revisão Sistemática da Literatura;

**Apêndice C – Questionário sobre Processo de Desenvolvimento de Software nos IFs:** Apresenta o questionário aplicado nos IFs;

**Apêndice D – Entrevistas não estruturadas com especialistas dos IFs:** Apresenta a coleta de dados das entrevistas aos IFs que possivelmente trabalhavam com DDS;

**Apêndice E – Questionário sobre Processo de DDS nos IFs:** Apresenta o questionário aplicado aos especialistas dos IFs que já atuam com DDS;

**Apêndice F – Roteiro para condução do Grupo Focal:** Apresenta o passo a passo utilizado para aplicar a técnica de Grupo Focal.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem a finalidade de qualificar o leitor para o entendimento deste trabalho, para tanto, serão apresentados os principais conceitos que embasam a proposta deste trabalho, buscando apresentá-los da maneira mais genérica possível.

### 2.1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

A construção de um produto de software é organizada através de um conjunto de atividades relacionadas, que é denominada de processo de desenvolvimento de software ou apenas Processo de Software (SOMMERVILLE, 2011).

Segundo Pressman (2011), afirma que a Engenharia de Software pode ser dividida em quatro camadas, ferramentas, métodos, processo e foco na qualidade (Figura 6), o processo é a camada mais importante, pois ela é a base para os métodos e ferramentas, além de possibilitar um desenvolvimento racional do software.



Fonte: Pressman (2011).

De acordo com Sommerville (2011), os processos de desenvolvimento são complexos, pois dependem de pessoas para tomar decisões e fazer julgamentos, como todos os processos intelectuais e criativos. Não existe um processo ideal, a maioria das organizações desenvolve os próprios processos de desenvolvimento de software. Para algumas organizações é necessário um processo de desenvolvimento muito bem estruturado, enquanto para outras, que possuem requisitos voláteis, provavelmente será mais eficaz um processo menos formal e mais flexível.

Um processo de desenvolvimento de software é uma instância de um modelo de processo, adequado a uma realidade específica. As metodologias estabelecem

basicamente a sequência das atividades e como elas devem se relacionar, para atender um contexto específico. Existem diversos modelos de processo de desenvolvimento de software, e cada modelo pode ter mais do que uma metodologia que o operacionaliza, definindo a forma que os métodos e as ferramentas devem ser utilizados (AUDY; PRIKLADNICKI, 2007, p. 12).

As metodologias e frameworks são artefatos resultantes de uma longa jornada de pesquisa, experimentação, tradução de exemplos e definições de conceitos, traçada por muitos indivíduos diferentes que apoiaram e transmitiram cada um desses conhecimentos que as formam. As metodologias de desenvolvimento surgiram antes mesmo do desenvolvimento de software em equipes, quando os softwares eram criados para os antigos mainframes existia um alto custo de alterações, devido ao acesso limitado aos computadores. Por isso, era necessário um grande planejamento e documentação antes da execução de uma mudança (SOARES, 2004).

As metodologias são formadas por Pessoas, Papéis, Cerimônias e Artefatos. Metodologia é a forma com a qual deve ser executado um processo para a realização de algo. O Framework deriva da mesma ideia, porém ao contrário das metodologias tenta apenas expressar as possibilidades de caminhos e deixa o utilizador responsável por escolher o que melhor se adequa à sua realidade. Neste trabalho, metodologias e frameworks serão chamados de Orientações.

O Modelo em Cascata ou Waterfall Model descrito por Royce em 1970, se tornou referência mundial e abriu as portas do desenvolvimento de software em equipe, essa organização inicial do processo possibilitou ainda uma evolução da capacidade de produção de grandes softwares, como os ERP - Enterprise Resource Planning.

O processo clássico de desenvolvimento de software foi muito útil e resolveu muitos dos problemas iniciais da área, mas com o passar do tempo, o crescimento da demanda por softwares forçou a redução do ciclo de desenvolvimento.

As metodologias mais famosas do modelo Iterativo e Incremental vieram a partir do conceito de desenvolvimento ágil de software. Para entender melhor essas orientações precisamos conhecer o movimento ágil, também conhecido como filosofia ágil.

O surgimento do Manifesto para Desenvolvimento Ágil de Software em 2001, trouxe uma nova forma de enxergar o processo de desenvolvimento de softwares, criado em uma reunião entre dezessete especialistas em processos de desenvolvimento de software representando os métodos Scrum, Extreme

Programming (XP) e outros, trouxe quatro valores e doze princípios, que eram comuns a todos esses métodos e que viriam a nortear as próximas gerações de metodologias (SOARES, 2004).

Em contraste com os modelos clássicos, o desenvolvimento de software baseado nos princípios ágeis introduz flexibilidade na organização de fases do ciclo de desenvolvimento e se concentra em como produzir software em ciclos mais curtos sem comprometimento da qualidade. Os métodos ágeis promovem a redução de tempo de desenvolvimento, a motivação do trabalho em equipe, a redução do número de erros, e a flexibilidade para alterar prioridades, entre outros aspectos.

Dentre as principais orientações encontradas nesta pesquisa, o Scrum foi o framework que mais se destacou dos outros, sendo o mais citado nos estudos analisados na revisão sistemática da literatura e o mais usado nos projetos encontrados na pesquisa de campo, a motivação para tal relevância deste framework para a área pode estar em suas características fundamentais que se aproximam muito das necessidades vividas no DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE. Desenvolvido por Jeff Sutherland, ele é um framework voltado ao trabalho em equipe, sendo assim trabalha exatamente o maior desafio do DDS que é a comunicação entre os indivíduos e equipes. Apesar de ser inicialmente projetado para o modelo co-localizado, às ferramentas de comunicação atuais possibilitam seu uso no modelo distribuído.

Outra metodologia importante para o movimento Ágil e para este trabalho foi o XP que traz práticas muito utilizadas no desenvolvimento de software co-localizado, que podem ser repensadas para o modelo distribuído.

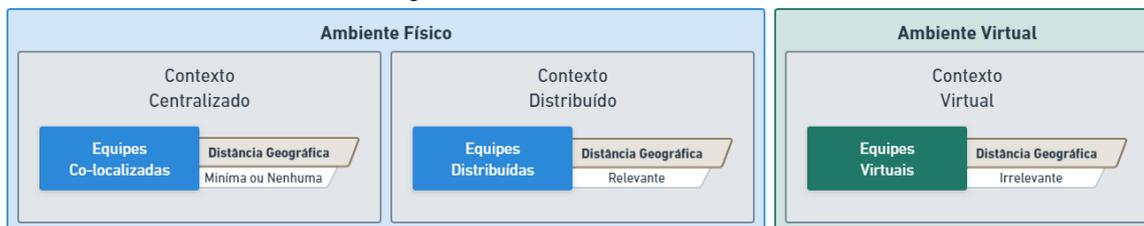
## 2.2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

O desenvolvimento de software que surgiu como uma prática individual, se tornou um processo associado a equipes, agora a necessidade de distribuir as equipes geograficamente torna esse processo ainda mais complexo, uma vez que os conflitos comuns a organização e comunicação entre vários indivíduos são acrescentados aos desafios do DDS.

Após a diferenciação do contexto centralizado dos contextos distribuídos e virtuais (ver Figura 7), foram criados os termos Desenvolvimento Distribuído de Software e Desenvolvimento Global de Software, assim ele passou a ser chamado de

## Desenvolvimento de Software Centralizado, Co-localizado ou Tradicional.

Figura 7 – Ambientes e contextos



Fonte: Adaptado de Audy e Prikladnicki (2007).

Alguns fatores foram muito importantes para o surgimento das metodologias de desenvolvimento distribuído de software, tais como, a globalização com o advento da internet, o surgimento de cada vez mais empresas multinacionais e o aprendizado adquirido da rápida expansão dos softwares código aberto (OpenSource) criados através de plataformas de desenvolvimento colaborativo. Este Desenvolvimento Colaborativo de Software mostrou a capacidade de evolução do Desenvolvimento de Software para um modelo distribuído onde as pessoas pudessem estar em locais diferentes, com horários diferentes, com culturas diferentes e mesmo assim contribuir para um mesmo projeto. As grandes empresas aproveitaram essa ideia para terceirizar a mão de obra de desenvolvimento de software para equipes que mostrassem um melhor desempenho ou realizasse o trabalho com um menor custo.

### 2.2.1 Características

A principal característica do Desenvolvimento Distribuído de Software (DDS) é dispersão geográfica das pessoas, mas esta ocasiona naturalmente outras duas características fundamentais, são estas, a diferença sociocultural dos indivíduos e a dispersão temporal entre eles, neste trabalho vamos chamá-los de “Distâncias” (AUDY; PRIKLADNICKI, 2007, p. 12).

Existir uma comunicação, cooperação ou colaboração entre os indivíduos dispersos do projeto é essencial para sua caracterização como DDS, caso contrário, torna o cenário dividido em dois projetos distintos, localizados em regiões diferentes, o que não se enquadra nas características inerentes ao tema.

As características podem ser referenciadas como fatores que diferenciam o processo de desenvolvimento de software tradicional do distribuído ou então como a razão básica para os diversos desafios que essa abordagem pode trazer.

A **Distância Geográfica** pode ser considerada a distância física entre os indivíduos de uma organização, podendo existir uma ou várias equipes, sendo elas fisicamente agrupadas ou totalmente virtuais, essa distância se torna relativa à comunicação entre os indivíduos e conseqüentemente relativa à distribuição das equipes. Por essa razão foi elaborada a classificação do nível de dispersão dos indivíduos que trata todas as variáveis referentes a forma de agrupar as pessoas.

Quando a força de desenvolvimento de software não está co-localizada, ou seja, passa a estar distribuída, outras variáveis podem passar a interferir o fluxo do processo de desenvolvimento, bem como idioma diferente ou fuso horário, de forma que algumas práticas, como a comunicação face-a-face, podem se tornar difícil ou inviável.

A **Distância Temporal** pode ser encarada como a diferença de horário em que as pessoas estão disponíveis para a comunicação síncrona “em tempo real”, o que no desenvolvimento tradicional podia ser percebido como o “chegar atrasado” de uma pessoa, no desenvolvimento distribuído se torna muito mais complexo. Uma pessoa pode estar em um país diferente do outro, então o fuso horário pode possuir uma diferença superior a 8 horas, ou então, o ritmo de trabalho pode ser definido de maneira inversa pela organização ou pelo indivíduo fazendo com que um trabalhe durante o dia enquanto o outro trabalhe durante a noite, mesmo estando na mesma cidade.

Apesar da distância temporal influenciar em muitos desafios do DDS ela também pode trazer um benefício muito buscado pelas empresas multinacionais, conhecido como Follow-The-Sun, que consiste na prática de distribuir as equipes ao redor do mundo, em diferentes fusos horários, o que permite manter uma equipe disponível para o trabalho a qualquer momento do dia (24h) (AUDY; PRIKLADNICKI, 2007, p. 47).

A **Distância Sociocultural** consiste nas diferenças entre os indivíduos, formadas por conhecimentos adquiridos através de suas vivências, isto é, valores, princípios, costumes, tradições, crenças, etc. Tudo isso influencia diretamente o comportamento e a forma de comunicação de cada indivíduo, dificultando a organização, padronização e gerenciamento do processo de desenvolvimento de software.

Este é o principal fator para a adoção da prática de alinhamento do “Mindset” e “Orientações” da organização, uma vez que a mentalidade dos indivíduos precisa ser

adaptada às necessidades da instituição para que possa ser mais fluida a comunicação da equipe, tendo em vista que todos já possuem uma certa noção do que deve esperar em cada etapa dos processos.

### 2.2.2 Níveis de dispersão dos indivíduos

As **Equipes Co-localizadas** são as equipes que estão presentes em um mesmo ambiente durante o desenvolvimento do software, tradicionalmente o ambiente de trabalho possibilita conversas face-a-face diariamente.

As **Equipes Distribuídas** são formadas quando os indivíduos de uma organização ou projeto estão fisicamente distribuídos em dois ou mais grupos, com localizações distintas, passando a existir uma dependência de algum meio de comunicação entre essas equipes.

Os projetos distribuídos podem apresentar variados níveis de dispersão geográfica (ver Figura 8) o que impossibilita a interação face a face. De acordo com Audy e Prikladnicki (2007), estes podem ser:

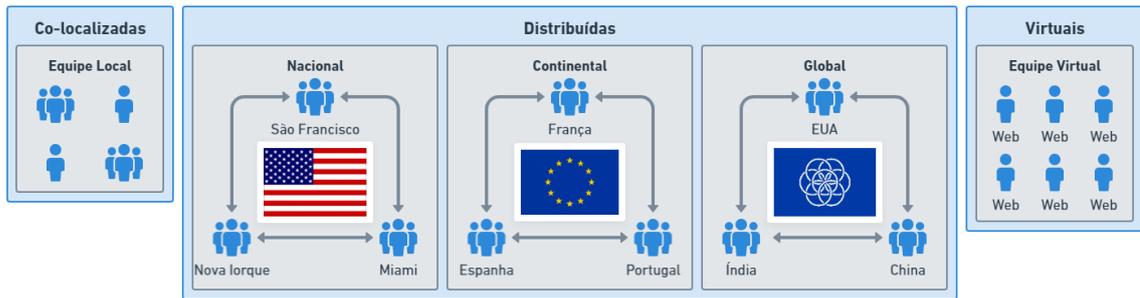
**Distância Nacional** caracteriza-se por membros localizados no mesmo país. Em alguns países há diferenças de fusos horários, no entanto as equipes podem reunir-se em um curto intervalo de tempo

**Distância Continental** equipes localizadas em países diferentes, mas em um mesmo continente. As diferenças culturais, de idioma e de fusos horários são mais acentuadas, podendo dificultar interações.

**Distância Global** é caracterizada por equipes localizadas em diferentes países e continentes. A diferença de fuso horário pode impedir interações síncronas, e as diferenças culturais podem dificultar o trabalho.

As **Equipes Virtuais** são equipes existentes apenas em nível virtual, fator que a torna totalmente dependente de determinadas ferramentas para existir, mas que possibilita uma maior capacidade de participação, uma vez que os indivíduos podem estar em qualquer localização física que possua acesso a essas ferramentas.

Figura 8 – Níveis de dispersão dos indivíduos



Fonte: Adaptado de Audy e Prikladnicki (2007).

### 2.2.3 Esforço de alinhamento

Existe um esforço comum à execução de atividades, que é a definição de qual forma será executada a tarefa, naturalmente os indivíduos definem isso sozinhos ou em discussão com o grupo. Alguns autores levantam a observação de que em muitas equipes este esforço não é contabilizado como parte do trabalho, ou não é nem percebido pelos stakeholders, fato este que pode resultar em uma grande diferença entre o tempo estimado e o tempo de execução dos projetos.

Alguns autores podem chamar este fator de “Metatrabalho”, “Trabalho extra”, ou “Trabalho de Articulação”, nesta pesquisa o definimos como “Esforço de Alinhamento”, pois entendemos que ele tem o objetivo de alinhar o entendimento de um ou mais indivíduos em relação ao objetivo da execução de determinada atividade (MATTHIESEN, BJØRN, PETERSEN, 2014; GIUFFRIDA, DITTRICH, 2014; ESBENSEN, BJORN, 2014; LOUS ET AL., 2018).

Entendemos o Esforço de Alinhamento como parte primordial do trabalho, pois influencia toda a coordenação dos indivíduos e a excelência na execução das atividades. Posto isso, entendemos que ele deve ser previsto como parte do trabalho e contabilizado em eventuais estimativas de tempo.

## 2.3 ORGANIZAÇÃO DE PESSOAS

Compreender o comportamento humano é essencial para a gestão de uma organização. A organização depende dos indivíduos, tanto quanto os indivíduos dependem da organização. Para entender a melhor forma de gerir pessoas, é imprescindível conhecer os fatores que influenciam a maneira de pensar e conseqüentemente o comportamento de cada pessoa.

### 2.3.1 Definição de organização

Podemos chamar de Organização de Pessoas, Organização de Indivíduos ou apenas de Organização, a combinação de esforços de várias pessoas, com a finalidade de agregar valor para a realização de propósitos específicos de forma coordenada.

De acordo com Santos (2015) uma organização é um grupo de pessoas ligadas por padrões estáveis de interação e que colaboram entre si para alcançar um objetivo comum.

Outras denominações comuns para a Organização de Indivíduos são “Equipes” e “Times”, que geralmente são tratados pelos autores com diferenças. Neste trabalho serão tratados apenas como sinônimos de organização.

### 2.3.2 Mindset

O **Mindset** pode ser traduzido como “Mentalidade” ou “Configuração Mental”, e significa a forma como uma pessoa pensa, o que molda o comportamento de cada um perante os desafios que enfrentará. Este conceito foi criado pela psicóloga Carol Dweck e se popularizou no mundo da Engenharia de Software.

Dweck (2017) classifica a configuração mental em Mindset Fixo e Mindset de Crescimento, basicamente essa distinção diz respeito à forma de pensar perante os desafios encontrados, enquanto o primeiro identifica a derrota como um fracasso o segundo identifica a derrota como uma oportunidade de crescimento intelectual e assim utiliza-se dela para melhorar sua performance e não fracassar novamente.

A partir de 2001, quando surgiu o “Manifesto Ágil” o mundo aprendeu, através de 4 valores e 12 princípios, que antes de qualquer mudança ou adoção de estratégia para uma organização é necessário mudar a forma como as pessoas integrantes do projeto pensam, o que posteriormente passou a ser chamado de “Mindset Ágil”.

A metodologia Lean derivada do “Lean Manufacturing” que é um sistema de produção desenvolvido pela Toyota, no Japão, introduziu, em 1950, uma forma de pensar que influenciava toda a cultura organizacional para buscar continuamente a melhoria dos processos em direção a maximização do valor percebido pelo cliente, eliminando seus desperdícios. O que hoje podemos chamar de “Mindset Lean”.

Em resumo, apesar de um termo relativamente novo o mindset sempre esteve

entre nós e sempre foi percebido de formas diferentes, podemos dizer que o mindset, trata-se das características da mente humana que vão determinar os pensamentos, comportamentos e atitudes. Ele pode ser entendido como a junção de Valores e Princípios experienciados por um indivíduo que levaram ele a adotar um determinado comportamento, de acordo com o que ele toma como verdade.

A forma de pensar é influenciada por muitos fatores ao longo da vida, uma não é necessariamente melhor que a outra, não existindo um mindset correto, temos que adaptar nossa forma de pensar ao contexto que estamos ou queremos estar. Posto isso, o **Mindset Organizacional** trata a forma de pensar que a organização adota como eficaz para o contexto em que ela atua, entretanto, o mindset da organização só se torna real a partir do momento que todos os membros passam a alinhar seu mindset ao da organização, pois são os comportamentos dos membros que representa a forma de pensar da organização.

### 2.3.3 Cultura organizacional

Naturalmente, como todas as relações entre pessoas, estas organizações passam a se tornar um ambiente propício para a propagação de comportamentos. Assim se estabelece a cultura organizacional, que pode ser entendida como o conjunto de comportamentos que distingue uma organização de pessoas das outras, ela é formada por um conjunto de costumes e crenças estabelecidos através de normas, valores e atitudes que impactam diretamente nos comportamentos e nas expectativas compartilhadas por todos os membros. Assim dizendo, um membro afeta a cultura organizacional ao mesmo tempo que é afetado por ela, essa relação de troca determina que o perfil pessoal, social e profissional de uma pessoa deve ser compatível com a cultura organizacional ao qual ela pretende fazer parte. Uma vez que esse equilíbrio não é mantido a relação tende a ser desfeita (BOULART; LANZA, 2007; SCHEIN, 2004).

A cultura como um conjunto de pressupostos básicos define para nós em que prestar atenção, o que as coisas significam, como reagir emocionalmente ao que está acontecendo e quais ações tomar em vários tipos de situações. (SCHEIN, 2004, p. 32).

A cultura de uma organização é a maneira como se espera que uma organização

vá se comportar ao utilizar seus produtos ou serviços. Existem muitos motivos para entender a cultura de uma organização. Pode ser fundamental para planejar estratégias durante grandes mudanças, como quando uma organização está em expansão ou fusão, momentos em que pode ser definido uma nova cultura para áreas geográficas distintas que se adequem melhor a localidade.

Cultura organizacional tem implicações para todos os aspectos do ambiente de trabalho. Santos (2015) lista algumas destas implicações:

A cultura determina os tipos de respostas que a organização tem frente a mudanças necessárias e novos problemas. Ela pode ajudar a prever o quão bem a organização irá lidar com a mudança.

As pessoas só podem notar sua cultura quando há uma quebra de rotina e eles têm que lidar com algo inesperado. Ter um conhecimento e compreensão da cultura pode ajudar os membros a prever como a empresa pode responder e lidar com o imprevisto.

A cultura determina o tipo de pessoas que serão atraídas para a organização e que serão bem-sucedidas. Ela também dá uma direção clara para a formação e desenvolvimento dos indivíduos, definindo o que é e o que não é importante, e quais as competências que o indivíduo precisa para ter um bom desempenho na organização.

Em uma organização que tem divisões ou subculturas pode levar à fragmentação ou perplexidade, daí a confusão, conflito e falta de cooperação.

Cultura implica estabilidade, padronização de comportamentos, e reflete todos os aspectos da vida em grupo. Ela reconhece que os nossos ambientes de trabalho são extensões de nós mesmos e dirige a atenção para um significado simbólico.

Organizações que possuem culturas que são muito fortes podem tornar-se resistentes à mudança e será muito lento para se adaptar ao seu ambiente.

Assim como a cultura organizacional é importante para entender o comportamento de toda a organização, é necessário entender o comportamento de cada indivíduo que faz parte dela, seja para avaliar seu comprometimento com os ideais da organização ou para identificar um papel que ele tenha mais domínio, obtendo um melhor desempenho e satisfação para ambas as partes.

## 2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi apresentado a fundamentação teórica utilizada durante esta pesquisa.

Foi abordado o entendimento do processo de desenvolvimento de software, explanando o pensamento de renomados autores, tais como, Pressman, Sommerville, Audy e Prikladnicki.

Foram apresentados alguns dos muitos conceitos e características relevantes do DDS, explicados por Audy e Prikladnicki.

Por fim foram mostradas características inerentes às pessoas, que são base para todos os processos que envolvem o DDS.

### 3 REVISÃO SISTEMÁTICA DA LITERATURA

A Revisão Sistemática da Literatura (RSL) é uma técnica de análise de estudos primários correlatos que tem como objetivo a síntese de dados relevantes para a resolução de uma ou mais questões específicas.

Uma revisão sistemática da literatura é um meio de identificar, avaliar e interpretar toda a pesquisa disponível relevante para uma questão de pesquisa específica, área de tópico ou fenômeno de interesse. Estudos individuais que contribuem para uma revisão sistemática são chamados estudos primários; uma revisão sistemática é uma forma de estudo secundário. (KITCHENHAM; CHARTERS, 2007).

A Engenharia de Software Baseada em Evidências (ESBE) originou-se em 2004 inspirada no sucesso do paradigma baseado em evidências empregado na medicina, adaptando as práticas baseadas em evidências para atender às características muito diferentes da Engenharia de Software e às consequências que essas características têm para estudos empíricos.

O estudo de Kitchenham (2004) é citado por vários autores como o primeiro trabalho a estabelecer um paralelo entre as áreas da Medicina e Engenharia de Software, apresentando a técnica de Revisão Sistemática da Literatura como uma abordagem de pesquisa baseada em evidências significativamente forte na Engenharia de Software. A partir deste momento cada vez mais projetos de pesquisas foram baseados na abordagem de RSL, provando ser uma técnica relevante para o avanço na área.

Nesta pesquisa foi utilizado o método de revisão sistemática da literatura como o objetivo de buscar por fundamentação teórica através da agregação de informações para a resolução do problema proposto de forma não tendenciosa. Desta forma a revisão sistemática foi executada de acordo com os critérios estabelecidos no protocolo da RSL que se encontra integralmente no **APÊNDICE A**.

#### 3.1 QUESTÕES DE INVESTIGAÇÕES ESPECIFICAS

Com o objetivo de investigar meios de mitigar os desafios comuns a projetos de desenvolvimento distribuído de software, este trabalho busca responder às seguintes questões de pesquisa:

**Q1:** Quais são as metodologias apresentadas na literatura?

**Q2:** Quais são as práticas adotadas nos processos de desenvolvimento distribuído de software apresentados na literatura?

A resposta de **Q1**, nem sempre está explícita, muitas vezes a metodologia é descrita nos artigos de forma implícita, com a utilização de uma prática ou outra derivada de uma determinada metodologia.

A resposta de **Q2** representa o foco principal deste trabalho, uma vez que as práticas são consideradas pelo autor, como a manifestação de todos os valores, princípios, conceitos, implícitos no mindset dos desenvolvedores descritos pelo caso.

### 3.2 SELEÇÃO DOS ESTUDOS PRIMÁRIOS

A partir da String de busca e das fontes definidas, as buscas primárias retornaram um total de 245 trabalhos, nos repositórios IEEE Xplore Digital Library e ACM Digital Library. Após a aplicação dos passos descritos como “Escolha dos estudos” no Protocolo que pode ser encontrado no **APÊNDICE A**, foi escolhido um conjunto final de 34 trabalhos para serem analisados em detalhes.

O processo de seleção ocorreu através da aplicação dos critérios de seleção de forma sequencial com a leitura do título, resumo, introdução, conclusão, e por fim, o estudo na íntegra.

### 3.3 ANÁLISE DOS RESULTADOS

A análise dos dados extraídos foi elaborada levando em consideração as práticas citadas com as narrativas de cada autor, porém foi necessário a realização de uma normalização, devido ao fato de cada autor expor as informações de uma maneira diferente, de acordo com as perspectivas que cada um possui. Após a normalização os dados se tornaram mais consistentes e a análise ficou coerente e coesa.

Todas as práticas extraídas da RSL foram reunidas com as demais práticas encontradas e analisadas em uma única planilha, seus resultados podem ser vistos na seção 5.1.

A utilização de modelos de desenvolvimento iterativo e incremental é uma prática comum, citada pela maioria dos artigos. Dentre estas metodologias, as mais citadas são o Scrum e o XP, entretanto existe uma dificuldade de migração entre qualquer modelo, que naturalmente gera um certo medo, esse medo é inerente ao ser humano e está presente em toda mudança.

O artigo Jha, Vilardell, Narayan (2016) apresenta uma excelente abordagem de migração de modelo de desenvolvimento, com a adoção gradual de práticas e a constante melhoria do processo, mantendo um Modelo Híbrido com qualidade e performance. Um exemplo citado é a abordagem de Engenharia de Requisitos, utilizando conceitos Ágeis e Lean para redução do tempo de entrega. Isso prova que é possível realizar uma transição gradual entre modelos de processo sem perder o desempenho da equipe.

A abordagem Stage-Gate, apresentada por Laukkanen et al. (2016), é um modelo para auxiliar a gestão do processo de desenvolvimento de novos produtos, porém ao contrário do trabalho anterior, o autor chega à conclusão de que este modelo em específico atrapalhou a adoção de metodologias ágeis, particularmente, a prática de Entrega Contínua. Esse caso mostrou que uma prática pode impedir a transição para o modelo DDS, o que ressalta a importância de conhecer as características que podem fazer uma prática não ser aderente ao modelo DDS.

Lous et al. (2018) apresenta um caso de sucesso que vai além do modelo iterativo e incremental, explica como foi feita a integração do desenvolvimento ágil com o DDS de forma equilibrada, mantendo a velocidade e simplicidade do ágil, mesmo em ambiente virtual, a nova metodologia apresentada neste trabalho abre portas para uma forma de desenvolvimento sem o peso do tempo sobre a equipe de desenvolvimento. Assim esse modelo pode ser visto como uma forma contínua de desenvolvimento independente do tempo relacionado ao projeto, pois as equipes não são pressionadas por ele, e sim encorajadas a realizar o trabalho com a menor quantidade de impedimentos possíveis. O papel de “Ágil Servo-Líder” foi criado, para facilitar mudanças que a equipe entende que possam melhorar o trabalho e garantir que esses impedimentos encontrados sejam resolvidos.

As ferramentas são essenciais para organizações que utilizam o DDS, é papel da gestão prover e manter ferramentas adequadas para cada demanda, todos os trabalhos pesquisados citam algum tipo de ferramenta direta ou indiretamente.

Ferramentas de design e desenvolvimento são importantes. Ferramentas pobres e um ambiente pobre podem levar a um projeto ruim. Forneça as ferramentas e treinamento certos. (KUHRMANN; FERNÁNDEZ, 2015).

Segundo Krüger et al. (2017) a organização deve construir confiança entre seus colaboradores para o qual o fator mais importante é a comunicação, em ambientes de DDS não existe comunicação adequada sem ferramentas de qualidade. Por isso é extremamente importante a sábia escolha de uma ferramenta de gerenciamento de comunicação eficaz e adequada a todos os formatos de comunicação.

Sendo a comunicação o fator mais relevante em um modelo distribuído de trabalho, devemos valorizar as práticas citadas no estudo referentes a ela. Segundo sete autores a comunicação deve ser estruturada em ferramenta única e centralizada, que possua comunicação síncrona e assíncrona e permita interações individuais, coletivas e globais, via texto, áudio ou vídeo.

Expandimos a ideia de Berczuk, de que "ágil é sobre pessoas, mas a ágil distribuída requer boas ferramentas para ajudar as pessoas a se comunicarem efetivamente a distâncias" enfatizando o fato de que as ferramentas utilizadas devem estar facilitando todas as dimensões da cooperação e devem apoiar práticas ad-hoc. (LOUS et al., 2018).

Assim como a comunicação, as informações são um bem intangível de alto valor para a organização, devido a isso segue a mesma necessidade de ferramentas de armazenamento, única, centralizada e que promova visibilidade e transparência. Informações podem tomar diversas formas, assim é possível que sejam tratadas de forma diferente, assim havendo a necessidade de um sistema para cada tipo de informação ser armazenada e gerenciada, Arquivos, Problemas, Tarefas, Feedback, Conhecimento, etc. O mais significativo é que essas informações sejam organizadas e todos os stakeholders concordem com essa forma de organização.

Não podemos deixar de salientar as diferenças de tratamento que o conhecimento deve ter em relação à informação, o conhecimento é muito mais complexo e difícil de transmitir do que os outros tipos de informações. Assim sendo, muitas práticas encontradas na RSL mostram essa forma diferenciada de armazenar e transmitir esse conhecimento. O primeiro fator a se levar em consideração ao armazenamento de conhecimento é a confiança, apesar de outras informações também precisarem ser confiáveis, o conhecimento requer um custo muito maior para identificar e corrigir informações não confiáveis.

Um dos maiores desafios das organizações é encontrar o conhecimento certo no momento certo devido à fragmentação do conhecimento, sobrecarga e descontextualização. (STRAY, 2018).

Talvez a forma de armazenamento mais relevante para o processo de desenvolvimento de software é o código, que podemos considerar o produto desse processo. As práticas da literatura são unânimes quanto à forma de armazenamento desta informação, sendo de forma centralizada com capacidade de versionamento e distribuída entre os pares, durante a utilização do código para o desenvolvimento. Entretanto existem variações nas práticas relacionadas ao desenvolvimento que orientam a forma como código deve ser ramificado, refatorado, integrado ou entregue. Independente da forma como o desenvolvimento ocorre é fundamental que a plataforma de versionamento de código seja organizada e gerida por todos os desenvolvedores de forma coordenada.

Gupta, Venkatachalapathy, Jeberla (2019) ressalta a importância de escolher ferramentas de Arquitetura de Software de menor custo de implantação, ele apresenta o exemplo de implantação de arquitetura “Sem servidor” no caso de sucesso da Siemens, e destaca os seguintes benefícios dessa abordagem:

Os desenvolvedores podem codificar a solução na linguagem de programação escolhida. Poucos recursos são uma combinação de micro serviços separadamente implantáveis, como funções lambda. Assim, a mudança de função lambda é suficiente para uma versão de versão de liberação incremental. Assim, poderíamos alcançar uma implantação mais rápida em comparação com a implantação demorada com componentes monolíticos e mamutes.

A escalabilidade da solução para os cenários de produção já está embutida. Conseguimos nos integrar com serviços de terceiros superiores, quando necessário, por exemplo, serviço de autenticação.

Não há servidores físicos ou lógicos a serem administrados.

Essas características mostram como a arquitetura “sem servidor” pode reduzir o custo de manutenção a longo prazo, mas é importante considerar o custo de migração de uma arquitetura para outra, o mais relevante é o fato que existem muitas arquiteturas disponíveis para melhorar o fluxo de trabalho e de implantação de software. Escolher a arquitetura adequada pode mudar toda a estrutura da organização.

Podemos considerar a automatização uma técnica necessária para a abstração

de trabalho dentro do processo de desenvolvimento de software, uma prática muito relevante para aumentar a performance deste trabalho é a Automatização do Provisionamento de Ambientes, que pode fornecer uma estrutura única para os diversos ambientes de Desenvolvimento, Teste, Homologação e Produção.

Segundo Roopa, Kumar, Mani (2018) a Multinacional Siemens atua com Equipes de Desenvolvimento Multifuncionais trabalhando de forma Remota, porém apesar de serem multidisciplinares, elas não concluíam os fluxos de trabalho de forma independente, o que gerava uma grande quantidade de interações entre fusos horários, que estava aumentando as despesas. Foi decidido então prover maior autonomia às Equipes Remotas, para tanto ela definiu um plano de transferência de conhecimento entre as Equipes de Desenvolvimento através de reuniões. Nessas reuniões, novas técnicas ou abordagens, novas ferramentas ou infraestrutura de codificação eram compartilhadas. As equipes compartilhavam porque algumas abordagens funcionaram e porque outras não, o que os ajuda a implantar as práticas na equipe. Além das reuniões foram utilizadas técnicas como Reverse Shadowing e Mentoria.

O estudo de Jha, Vilardell, Narayan (2016) apresenta uma abordagem de Transferência de Conhecimento através da atualização dos registros, assim não existe necessidade de reuniões específicas. O registro ocorre durante a documentação do sistema ou atualização das últimas alterações para todos os stakeholders.

Segundo Lous et al. (2018) as informações do projeto devem estar disponíveis com detalhes para todos os membros, uma vez que elas podem influenciar aspectos importantes do desenvolvimento de software, como confiança e compartilhamento de conhecimento, esses aspectos podem se tornar um problema quando as equipes não estão se sentindo unidas.

### 3.4 LIMITAÇÕES E DIFICULDADES ENCONTRADAS

Em relação a estratégia da revisão sistemática da literatura adotada, Kitchenham (2004) afirma que a condução da pesquisa por apenas uma pessoa configura uma ameaça à validade, uma vez que a pesquisa sistemática busca o levantamento de informações de forma imparcial.

Mesmo realizando um processo bem rigoroso para determinar as palavras

chaves e refinar a string de pesquisa e selecionar os estudos de forma criteriosa, não é possível afirmar que estudos relevantes não tenham sido omitidos.

A tarefa de cruzamento de dados de diversos trabalhos diferentes, acrescido pelo esforço de análise e tratamento dos dados descritos de formas diferentes, formam uma combinação perigosa para a tarefa de normalização dos dados. Esta foi considerada a etapa mais complexa e difícil de se realizar sozinho.

Uma dificuldade encontrada foi a falta de conhecimento de alguns termos comuns a área de DDS que foram entendidos ao decorrer da análise, de fato, estes não podem ser considerados limitantes a esta pesquisa, pois ao se deparar com a diferença de tratamento da análise, eu, pesquisador, optei por reiniciar toda a análise.

Fica como uma proposta de mitigação destes problemas a realização de uma etapa de checagem superficial dos trabalhos coletados, essa etapa teria a função de proporcionar uma visão geral para o pesquisador, antes de realizar uma análise, para tanto o pesquisador deve baixar todos os artigos e visualizar alguns trechos superficialmente, de forma aleatória, buscando termos desconhecidos.

### 3.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

A Revisão Sistemática possibilitou identificar vários casos de sucesso de práticas ágeis dentro do contexto distribuído, foram identificados modelos de desenvolvimento de software híbridos, assim como as características que possibilitam ou não a integração de práticas ágeis a uma organização que já possui um modelo de desenvolvimento próprio. Explorar essas alternativas facilita a proposição de novas formas de encarar os desafios comuns do DDS.

Apesar de muitos autores citarem a comunicação como o fator mais impactado pelas dificuldades inerentes ao DDS, o aspecto mais impactado pelas práticas encontradas foi a Coordenação, que apesar de depender diretamente da Comunicação é um fator que se refere a forma como se organizam as pessoas dentro do processo. Assim é importante levar em consideração nas próximas pesquisas a relevância das diferentes formas de comunicação, que levam a Socialização e a Coordenação de pessoas.

Foram elencadas ainda as metodologias mais citadas pelos autores que revelaram um conjunto básico de metodologias comumente utilizadas em conjunto no mercado, Scrum, XP e Kanban.

## 4 CENÁRIO DOS INSTITUTOS FEDERAIS

Este capítulo tem a finalidade de apresentar a estrutura do cenário estudado para execução da proposta e explorar as informações relevantes advindas da pesquisa de campo a fim de expor as características do cenário levadas em consideração na construção da proposta.

### 4.1 PESQUISA DE CAMPO

Uma ferramenta muito útil para a pesquisa foi o Painel de informações do FORTI encontrado no portal do CONIF (Conif, 2020), que além de possibilitar o planejamento de ações que deveriam ser executadas nesta etapa da pesquisa, também possibilitou a identificação dos institutos que não desenvolvem de forma centralizada.

A fim de identificar as instituições participantes da Rede Federal que possui força de desenvolvimento de software geograficamente distribuída, foram utilizadas as respostas da pergunta “Equipe de desenvolvimento é toda centralizada na Reitoria?” que retornou uma possível descentralização dos que responderam “Não”, que foram, IFC, IFF, IFMT, IFRR, IFSP, IFMG, IFSUDESTEMG e IFSULDEMINAS.

Para a validação dessa informação foi planejada a utilização de uma reunião estruturada, mas esta não foi possível devido ao início da pandemia, que exigiu muito de todos os profissionais de TI e impossibilitou a reunião de toda a equipe de desenvolvimento de cada instituição, então a pesquisa de campo ficou estruturada de acordo com as seções a seguir.

#### 4.1.1 Processo de desenvolvimento de software nos IFs

Para a etapa de análise do Processo de Desenvolvimento de Software no contexto dos Institutos Federais, foi aplicado um questionário com profissionais de TI das Instituições Federais, com o propósito de identificar o nível de conhecimento em relação ao Desenvolvimento Distribuído de Software do ponto de vista dos stakeholders.

Para realização dos questionários foi utilizado o Google Forms, por ser uma ferramenta completa, fácil de usar e que possui um mecanismo nativo para a tabulação dos dados obtidos. Ele ficou disponível para respostas entre 20/04/2020 e

30/07/2020 e obteve 36 respostas.

O resultado desta pesquisa de campo surpreendeu as expectativas, uma vez que as respostas confirmam a existência do desenvolvimento paralelo entre Reitoria e Campi em outros institutos, além de expressar uma grande vontade de mudar esse cenário para um modelo distribuído de forma organizada e estruturada. No capítulo de resultados poderemos analisar de maneira aprofundada todos os aspectos resultantes dessa etapa. Ele pode ser visualizado na íntegra no **APÊNDICE C**.

#### **4.1.2 Entrevistas não estruturadas e questionário sobre DDS nos IFs**

Após a junção dos dados obtidos com o **painel de informações do FORTI** e o **questionário sobre o processo de desenvolvimento de software nos IFs**, foi possível selecionar os IFs que tinham maior probabilidade de apresentar dados relevantes sobre o DDS nos IFs. Estes foram convidados a participar de uma entrevista, inicialmente planejada para seguir um roteiro previamente estruturado, porém, entendeu-se que por se tratar de um tema muito complexo e dinâmico, a escolha por seguir de forma não estruturada alcançaria resultados mais relevantes.

Ainda na execução das entrevistas não estruturadas houve outra mudança de planejamento, devido a pandemia, não foi possível reunir as equipes de desenvolvimento de software, sendo assim, optou-se por realizar um questionário em paralelo para toda a equipe, enquanto eram realizadas as entrevistas não estruturadas com os coordenadores do setor de desenvolvimento de software.

As primeiras entrevistas foram focadas em identificar as maneiras de trabalho remoto utilizadas e como eles se organizavam. Uma vez feito isso, apenas o IFSP foi levado a um novo nível de discussão, com o intuito de absorver um conteúdo mais focado em suas experiências com o DDS.

Em paralelo as entrevistas, foi aplicado o questionário aos membros da equipe de desenvolvimento de software do IFSP. O questionário teve o propósito de identificar as práticas utilizadas, as principais dificuldades enfrentadas no modelo DDS do ponto de vista da equipe e a partir dessas dificuldades as ações tomadas para resolução ou mitigação.

Para realização dos questionários foi utilizado o Google Forms, por ser uma ferramenta completa, fácil de usar e que possui um mecanismo nativo para a tabulação dos dados obtidos. Ele ficou disponível para respostas entre 18/03/2020 e

30/03/2020 e foram obtidas 6 respostas.

O resultado desta pesquisa de campo foi de extrema importância, uma vez que as respostas apresentam a realidade da utilização do modelo distribuído de desenvolvimento de software nos IFs e as características de sua implantação.

Foi possível identificar as dificuldades enfrentadas no momento de transição de um modelo centralizado para um modelo distribuído. Um dos participantes com experiência em DDS, apontou os seguintes problemas: a dificuldade de organizar demandas como tarefas, a distribuição delas entre a equipe e o nivelamento de conhecimento dos integrantes da equipe.

As pessoas que participaram de ambos os modelos, centralizado e distribuído, puderam destacar as diferenças notadas entre ambos, como o crescimento de conhecimento da equipe e o aumento da utilização de ferramentas de trabalho em equipe on-line.

O relatório das entrevistas não estruturadas pode ser visualizado na íntegra no **APÊNDICE D**, e o questionário no **APÊNDICE E**.

## 4.2 RESULTADOS DA PESQUISA DE CAMPO

Nesta seção são apresentadas as informações mais relevantes encontradas na pesquisa de campo, levando em consideração a união de todos os dados levantados. Todas as questões expostas são analisadas de forma imparcial, apenas agregando os dados apresentados nas etapas anteriores e apontando as evidências encontradas.

Todas as práticas extraídas da pesquisa de campo foram reunidas com as demais práticas encontradas e analisadas em uma única planilha, seus resultados podem ser vistos na seção 5.1.

As interações experienciadas nesta pesquisa de campo demonstraram que existe uma baixa maturidade de desenvolvimento de software no geral, uma vez que a maioria dos indivíduos demonstraram um certo nível de desorganização entre o desenvolvimento realizado na Reitoria e nos Campi, além de uma baixa utilização de metodologias de desenvolvimento de software. Entretanto os participantes demonstraram indiretamente conhecer e utilizar algumas das técnicas ágeis, em especial Scrum, XP e Kanban.

Alguns têm consciência e praticam as metodologias de forma transparente,

outros praticam apenas algumas técnicas, sem considerar a metodologia como um todo. Apesar de alguns IFs trabalharem com Indivíduos Remotos, o IFSP foi usado como principal exemplo, uma vez que foi constatado a utilização consciente e bem sucedida de metodologia de desenvolvimento com equipes remotas, ele utiliza metodologias Scrum, XP, Kanban e Scrum de Scrums. Uma análise geral sobre o IFSP é apresentada na seção 4.2.2.

#### **4.2.1 Nível de dispersão dos *stakeholders* dos IFs**

Apesar de a estrutura dos IFs serem semelhantes, sabe-se que o nível de dispersão dos *stakeholders* dos institutos é definido de forma individual devido a sua autonomia, isso resulta numa grande diversidade de cenários que pode ser percebida em todas as pesquisas realizadas.

As características mais importantes observadas neste contexto são, a inclusão ou não dos profissionais dos campi na Equipe Institucional de Desenvolvimento de Software, o desenvolvimento de projetos com membros externos a Equipe Institucional de Desenvolvimento de Software, a existência ou não de desenvolvimento de software independente da Equipe Institucional de Desenvolvimento de Software e pôr fim a organização do desenvolvimento de software de todo o instituto possuir um padrão ou não. Desta forma foram criadas as seguintes classificações para cada cenário encontrado na pesquisa de campo:

##### **Cenário 1 - Equipe Institucional Centralizada, campi independentes**

- Equipe Institucional de Desenvolvimento de Software Centralizada
- Todos os projetos centralizados na Equipe Institucional
- Profissionais dos campi desenvolvem de forma independente e sem padrões institucionais

##### **Cenário 2 - Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes**

- Equipe Institucional de Desenvolvimento de Software Centralizada
- Alguns projetos são organizados de forma distribuída com Equipes Virtuais

- Profissionais dos campi desenvolvem de forma independente e sem padrões institucionais

**Cenário 3 - Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi participa de alguns projetos**

- Equipe Institucional de Desenvolvimento de Software Centralizada
- Alguns projetos são organizados de forma distribuída com Equipes Virtuais
- Profissionais dos campi desenvolvem de forma independente e participam de alguns projetos institucionais.

**Cenário 4 - Equipe Institucional Distribuída, com projetos através de Equipes Virtuais**

- Equipe Institucional de Desenvolvimento de Software Distribuída
- Todos os projetos são geridos pela Equipe Institucional Distribuída através de Equipes Virtuais para cada projeto
- Profissionais dos campi desenvolvem de forma remota, mas apenas dentro da Equipe Institucional de acordo com os padrões institucionais

Assim como essas classificações, os dados apresentados no Quadro 1, foram elaborados, a partir do conjunto dos dados, do painel de informações do FORTI, do questionário sobre o processo de desenvolvimento de software nos IFs e das entrevistas não estruturadas, que foram elaborados com o intuito de fornecer a maior quantidade de dados possíveis para que pudesse ser realizada uma análise de alto nível e uma classificação inicial, uma vez que não foram encontrados estudos anteriores que trouxessem uma classificação dos cenários de desenvolvimento de software dos Institutos Federais.

Quadro 1 – Classificação dos IFs respondentes

Instituto	Painel de Informações do FORTI	Questionário Processo de DS nos IFs	Conclusão
IFC	Não é centralizado	Distribuído	Equipe Institucional Centralizada, com projetos distribuídos através de

			Equipes Virtuais, campi participa de alguns projetos (Cenário 3)
<b>IFF</b>	Não é centralizado	-	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFMT</b>	Não é centralizado	-	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFRR</b>	Não é centralizado	Passou de centralizado para distribuído devido ao estado de quarentena	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFSP</b>	Não é centralizado	-	Equipe Institucional Distribuída, com projetos através de Equipes Virtuais (Cenário 4)
<b>IFMG</b>	Não é centralizado	-	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFSUDESTEMG</b>	Não é centralizado	-	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFSULDEMINAS</b>	Não é centralizado	Distribuído	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFMS</b>	Centralizado	Distribuído	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFSUL</b>	Centralizado	Distribuído	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)

<b>IFPE</b>	Centralizado	É misto, na Reitoria as soluções sistêmicas e devido aos processos não normatizados, os campi também desenvolvem	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFBAIANO</b>	Centralizado	Passou de centralizado para distribuído devido ao estado de quarentena	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFRS</b>	Centralizado	Passou de centralizado para distribuído devido ao estado de quarentena	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)
<b>IFES</b>	Centralizado	Passou de centralizado para distribuído devido ao estado de quarentena	Equipe Institucional Centralizada, com projetos distribuídos através de Equipes Virtuais, campi independentes (Cenário 2)

Fonte: O autor (2022).

#### 4.2.2 Instituto Federal de São Paulo (IFSP)

O Instituto Federal de São Paulo (IFSP) é o principal caso de sucesso dentre as instituições pesquisadas, analisando de forma comparativa com os estudos anteriores aqui mostrados. O IFSP possui as características mais indicadas para o cenário das IFs (ver Quadro 2). Atualmente possui um modelo de desenvolvimento de software Distribuído com Equipes virtuais Dinâmicas.

Quadro 2 – Características do desenvolvimento de software do IFSP

<b>Modelo de Desenvolvimento de Software</b>	Distribuído com Equipes Virtuais
<b>Data de Implantação</b>	Junho de 2017
<b>Tempo médio de implantação</b>	4 meses
<b>Sistema integrado de gestão</b>	Sistema Unificado de Administração Pública (SUAP)
<b>Stakeholders</b>	22 Pessoas, sendo destes, 5 profissionais polivalente, 15 Desenvolvedores e 2 testers
<b>Dispersão</b>	Reitoria e 7 Campi

<b>Quantidade de papéis</b>	5 Papéis
-----------------------------	----------

Fonte: O autor (2022).

Segundo o Coordenador de Desenvolvimento, que esteve presente durante a transição do modelo centralizado para o atual, a implementação do sistema atual teve início em junho de 2017 e durou aproximadamente 4 meses, com o treinamento das pessoas e preparação das ferramentas. As principais dificuldades relatadas foram a necessidade de organizar demandas com a equipe e o nivelamento de conhecimento dos membros. Entretanto possibilitou muitas melhorias para os envolvidos, tais como, evolução da utilização das ferramentas, crescimento técnico da equipe e melhoria da qualidade de vida e satisfação dos colaboradores.

#### 4.2.2.1 Ferramentas

Apesar de não ser considerado o fator mais relevante no desenvolvimento de software, as ferramentas têm um papel importante na forma como as equipes vão atuar. No modelo Distribuído elas têm um papel ainda mais importante, pois todas as comunicações, a troca de informações e os mecanismos de colaboração, dependerá das ferramentas disponíveis, devido, entre outras coisas, aos membros estarem geograficamente distantes. Exposto a relevância das ferramentas, o Quadro 3 apresenta o demonstrativo de ferramentas utilizadas no desenvolvimento de software do IFSP.

Quadro 3 – Ferramentas utilizadas no DDS do IFSP

<b>Gerenciamento de Chamados</b>	GLPI
<b>Comunicação Síncrona</b>	Mattermost
<b>Comunicação Assíncrona</b>	Mattermost e E-Mail
<b>Gerenciamento de Projetos</b>	OpenProject
<b>Gerenciamento de Tarefas</b>	Gitlab
<b>Versionamento de Arquivos</b>	Gitlab
<b>Integração Contínua</b>	Hudson
<b>Aplicação como Serviço (Containers)</b>	Docker

Fonte: O autor (2022).

O fato de possuir um sistema integrado de gestão, o Sistema Unificado de Administração Pública (SUAP), faz com que a maioria dos projetos desenvolvidos pelo IFSP sejam voltados aos módulos e funções dele, fazendo com que sua plataforma de gestão e versionamento de código e tarefas fique organizada com apenas um projeto, apesar de abranger várias equipes trabalhando em módulos e funções diferentes do sistema integrado.

#### 4.2.2.2 Dispersão dos stakeholders

Os stakeholders do IFSP se destacam pela capacidade dinâmica de adaptar-se ao desenvolvimento de novos projetos com a designação de pessoas para equipes virtuais de forma independente de localização física ou tamanho do projeto. Sua adaptabilidade se dá principalmente pelo fator de dispersão dos stakeholders, inerente ao cenário dos IFs, que é ilustrado pela Figura 9, que apresenta uma estrutura adaptativa em relação às dimensões de distância física, distância temporal, estrutura de projeto, necessidade de força de trabalho, necessidade de papéis e fluxo de desenvolvimento.

Sendo assim, cada novo projeto pode trazer uma quantidade de integrantes diferente, fluxo de desenvolvimento próprio e principalmente indiferença da localização física de cada indivíduo, uma vez que todos os projetos são geridos através de equipes virtuais dinâmicas e adaptativas.

Figura 9 – Dispersão dos stakeholders do IFSP



Fonte: O autor (2022).

#### 4.2.2.3 Papéis

A Figura 9 utiliza bandeiras atreladas às pessoas como representação dos papéis assumidos por cada integrante da equipe, a legenda identifica os papéis de acordo com a letra e a cor da bandeira, as definições de responsabilidades de cada um dos 5 papéis são apresentadas no Quadro 4 para um melhor entendimento da forma de trabalho adotada.

Quadro 4 – Papéis e responsabilidades no IFSP

<b>Analista de requisitos</b>	Responsável por levantar as necessidades de cada projeto.
<b>Coordenador de Desenvolvimento</b>	Responsável por definir os critérios de dimensionamento e dispersão dos integrantes das equipes de desenvolvimento de cada projeto, realizar o planejamento das equipes, gerenciar aspectos contratuais e andamento dos projetos.
<b>Desenvolvedor</b>	Responsável por buscar todas as habilidades necessárias para desenvolver um software.
<b>Líder de equipe</b>	Responsável por coordenar o esforço de alinhamento das equipes.
<b>Testador</b>	Responsável por garantir determinado nível de qualidade do software entregue.

Fonte: O autor (2022).

Uma vez descrita a forma dinâmica de atuação dos stakeholders no desenvolvimento de software do IFSP, é fundamental explicar que toda essa capacidade só é possível graças ao perfil polivalente dos profissionais identificados genericamente na Figura 9 como “Pessoa 1 à Pessoa 5”. A maioria dos stakeholders possuem apenas um papel, mas esses são chamados de polivalentes por se prepararem para exercer tarefas de complexidades diferentes e assumir um papel diferente para cada necessidade que o projeto apresentar. Desta forma, é possível traçar as diretrizes quanto ao projeto e a equipe virtual se adaptar para cada demanda.

As pessoas podem mudar de papel de acordo com suas características, vontade ou necessidade, mas geralmente elas já têm um perfil adequado a um papel específico, muitas vezes já possuem competência técnica para exercer esse papel.

Caso as pessoas não tenham a competência adequada ao papel, é necessário que se preparem para exercer esse papel através de capacitações disponibilizadas pela instituição. Então, após ser identificada uma vontade ou necessidade de exercer um novo papel, o Coordenador de Desenvolvimento prepara um planejamento de treinamento para a pessoa, de acordo com o papel que será assumido. Após a pessoa estar apta a assumir o papel, é definido como uma atribuição desta pessoa, tornando possível a utilização nos próximos projetos a serem executados.

As tarefas são definidas pelo Coordenador de Desenvolvimento e os Líderes de Equipe através de reuniões de equipes no início e durante o desenvolvimento do projeto. Após a definição das demandas e seus responsáveis, elas são adicionadas, em forma de tarefas, no painel “Kanban” na etapa “To do” em português “Por Fazer”. Após a listagem das tarefas no painel fica a cargo de cada membro atribuir a si as tarefas que têm condições de executar no momento.

#### 4.2.2.4 Fluxo de desenvolvimento

As principais influências no fluxo de desenvolvimento de software vêm da Metodologia Ágil, do Framework Scrum, da Metodologia Extreme Programming (XP) e da Técnica Kanban, a combinação dessas técnicas, possibilitaram a criação de equipes virtuais adaptativas, integradas e que se adequam às particularidades de cada projetos. Com ciclos de desenvolvimento rápidos, chamados de sprints no Scrum, são levados resultados visíveis para os stakeholders e feedback são

estimulados para melhoria no próximo ciclo de desenvolvimento. A documentação passou a ser minimalista, focada em funcionalidades, como orienta a filosofia ágil. O gerenciamento do fluxo de desenvolvimento se torna visível em um painel de post its com a técnica do Kanban.

Outra metodologia utilizada no IFSP é o Scrum de Scrums, que tem o objetivo de escalar o Scrum para várias equipes que precisam trabalhar juntas para entregar soluções complexas, se baseia na delegação de poder para o Líder de Equipe, deste modo, são realizadas reuniões com todos os Líderes de Equipe onde o Coordenador de Desenvolvimento passa as diretrizes que devem ser seguidas e, posteriormente, cada Líder de Equipe realiza uma reunião para transmitir essas diretrizes a todos os membros de sua equipe.

Eles citaram dois fatores de que dificultam a execução do DDS na instituição:

O primeiro é a alta influência das ferramentas no processo de desenvolvimento, o que é uma característica padrão do DDS, mas no contexto da administração pública têm os agravamentos da dificuldade de compra de ferramentas profissionais e, conseqüentemente, de testes de várias ferramentas gratuitas para encontrar a que melhor atende.

O segundo fator é a estrutura organizacional (Gestão), que foi muito reportada pelos stakeholders do IFSP como a maior e mais complexa barreira de se lidar. Deve ser trabalhado o entendimento de todas as pessoas da gestão, para que possam ser alteradas as normas vigentes a ponto de reduzir a burocracia ou prover uma maior capacidade de tomada de decisão diretamente do setor de desenvolvimento de software.

#### 4.2.2.5 Fluxo de comunicação

A comunicação tem sido apontada como o principal fator determinante de fracasso ou sucesso de projetos de DDS, usando o Scrum como modelo, a comunicação foi estabelecida com reuniões periódicas como determinado pelo Scrum.

O fluxo da comunicação ocorre de forma cíclica, baseada em caixa temporal de 15 minutos, possui uma reunião diária com todos os membros de cada equipe e uma reunião diária dos Líderes de equipes virtual com o Coordenador de Desenvolvimento, desta forma a informação pode circular entre todas as esferas de forma rápida e sem uma interrupção demasiadamente longa.

### **4.2.3 Projeto Rede Federal e o enfrentamento a COVID-19**

O projeto “Rede Federal e o enfrentamento a COVID-19” foi proposto pelo CONIF em uma das reuniões do Fórum de TI (FORTI) com o objetivo de disponibilizar uma plataforma on-line para que todos os institutos pudessem centralizar as informações públicas de projetos de enfrentamento a COVID-19, a plataforma possibilitou a visualização em tempo real de todas as iniciativas de enfrentamento a pandemia realizadas pelos Institutos Federais em um mapa. Suas principais características podem ser visualizadas no Quadro 5.

Quadro 5 – Características do projeto Rede Federal e o enfrentamento a COVID-19

<b>Proposição</b>	CONIF
<b>Instituições Desenvolvedoras</b>	IFSP
<b>Dispersão</b>	Reitoria e 2 Campi
<b>Tempo médio de desenvolvimento</b>	3 semanas
<b>Stakeholders</b>	3 Pessoas, sendo destes, 1 Coordenador de projetos e Líder de equipe virtual e 2 Desenvolvedores e Testers

Fonte: O autor (2022).

A rápida absorção desta demanda de maneira emergencial, com proporção nacional só foi possível graças à estrutura de desenvolvimento de softwares do Instituto Federal de São Paulo (IFSP), descrita na seção 4.2.2, que se prontificou a realizar todo o desenvolvimento e disponibilização da plataforma em sua estrutura.

A força de desenvolvimento de software deste projeto contou com 3 pessoas em uma Equipe virtual do IFSP distribuída regionalmente, sendo que destes, 1 assumiu os papéis de Coordenador de projetos e Líder de equipe virtual e 2 assumiram os papéis de Desenvolvedor e Testador.

Figura 10 – Dispersão dos stakeholders do projeto COVID-19



Fonte: O autor (2022).

A dispersão geográfica da equipe virtual, representada pela Figura 10, contou

com a participação da Reitoria de São Paulo e mais 2 Campi, Campinas e São João da Boa Vista, que atuaram no formato de home-office, uma vez que devido ao cenário de crise sanitária foi proibido a aglomeração de pessoas. É importante ressaltar que devido ao fato de o IFSP já estar totalmente adequado ao modelo distribuído de desenvolvimento de software, a necessidade de os integrantes trabalharem em suas residências não alterou sua performance significativamente.

Por se tratar do desenvolvimento de um projeto dentro da força de trabalho do IFSP as especificações de Ferramentas, Papéis, Tarefas, Fluxo de Desenvolvimento e Fluxo de Comunicação são as mesmas relatadas na seção 4.2.2.

#### **4.2.4 Projeto SUAP Colaborativo - Módulo Projetos de Ensino**

O Sistema Unificado de Administração Pública (SUAP) é um Sistema Integrado de Gestão (SIG) que visa centralizar as informações e processos da Instituição em uma única plataforma, ele foi desenvolvido pelo Instituto Federal do Rio Grande do Norte (IFRN), e disponibilizado gratuitamente a outros IFs do país. O sistema dispõe de vários módulos com aplicações para diferentes atividades administrativas, tais como, Protocolo, Almoxarifado, Patrimônio, Orçamento e Financeiro.

Hoje os aprimoramentos realizados no sistema por parte do IFRN são liberados para todos os IFs assim que validados e as melhorias realizadas por outros IFs são enviadas ao IFRN, para serem implantadas, desde que atendam às demandas de toda a Rede Federal. De certa forma já possui um cenário disperso, mas não o consideramos um desenvolvimento distribuído, uma vez que não existe uma organização entre esse desenvolvimento. Para que passe a ser um DDS será necessário a separação do código principal do SUAP do repositório do IFRN, ficando um repositório com o código de produção do IFRN e outro repositório com um código genérico para todas as IFs, sendo assim será necessário ainda uma equipe que coordene a integração dos códigos no repositório genérico.

Como uma iniciativa rumo ao desenvolvimento distribuído de software interligando todas as instituições federais, foi idealizado na reunião dos fóruns do CONIF a realização deste projeto piloto de desenvolvimento colaborativo do SUAP, que conta com a articulação entre as instituições IFMA, IFG, IFRN, IFAP, IFF, IFGoiano, IFRO e IFPB. O objetivo dele é a criação de um Módulo de Gerenciamento de Projetos de Ensino que foi definido a partir da demanda levantada pelo Fórum de

Dirigentes de Ensino (FDE) de controle das informações dos projetos de ensino da Instituição. Suas principais características podem ser visualizadas no Quadro 6.

Quadro 6 – Características do projeto SUAP Colaborativo

<b>Proposição</b>	CONIF
<b>Instituições Desenvolvedoras</b>	IFG, IFRN, IFMA, IFAP, IFF, IFGoiano, IFRO, IFPB
<b>Dispersão</b>	IFG, IFRN, IFMA, IFAP, IFF, IFGoiano, IFRO, IFPB
<b>Tempo médio de desenvolvimento</b>	6 meses (Planejado)
<b>Stakeholders</b>	11 Pessoas, sendo destes, 1 Coordenador, 1 Supervisor da Comissão de Sistemas do FORTI, 1 Supervisor da equipe Master SUAP, 3 Representantes do FDE, 1 Mapeamento de Processos e Manuais, 4 Desenvolvedores

Fonte: O autor (2022).

Uma vez definido o objetivo do projeto, os participantes do Workshop construíram um documento com um planejamento bem detalhado que estabeleceu o Grupo de Trabalho, Cronograma, Ferramentas (ver Quadro 7), Papéis, Tarefas, Plano de comunicação, Forma de gerenciamento, dentre outros.

Quadro 7 – Ferramentas planejadas para o projeto SUAP Colaborativo

<b>Gerenciamento de Arquivos (Nuvem)</b>	Google Drive
<b>Comunicação Síncrona</b>	WhatsApp
<b>Comunicação Assíncrona</b>	WhatsApp e E-Mail
<b>Gerenciamento de Tarefas</b>	Trello
<b>Gerenciamento de Status</b>	Status Report Mensal

Fonte: O autor (2022).

As ferramentas escolhidas, Trello e Google Drive, possibilitam respectivamente a centralização do Gerenciamento de Tarefas e Arquivos, enquanto a comunicação Síncrona e Assíncrona contou com o apoio das ferramentas WhatsApp e E-Mail, é importante ressaltar que hoje a ferramenta WhatsApp além de permitir a troca de textos simples, áudio, e arquivos de mídia que podem ser respondidos imediatamente ou a posteriori, também possibilita ligações em tempo real por áudio e vídeo, o que a torna como uma ferramenta muito útil para equipes distribuídas.

#### 4.2.5 Projeto SIG Colaborativo

O “Planejamento Colaborativo no Desenvolvimento dos Sistemas de Gestão das Instituições da Rede Federal” é uma proposta para o Desenvolvimento Colaborativo dos Módulos do Sistema Integrado de Gestão (SIG) utilizado pelas 16 (dezesesseis) Instituições da Rede Federal, foi elaborado e apresentado pelo Instituto Federal do Pará (IFPA) na 85ª Reunião Ordinária do CONIF, realizada em 13 de junho de 2018, em Foz do Iguaçu – PR.

A partir de uma reunião integrada entre os fóruns do CONIF, foi idealizada a criação de um Grupo de Trabalho (GT) com o objetivo de elaborar um projeto completo, com as ações necessárias para buscar soluções a curto prazo e longo prazo.

Para médio prazo, uma solicitação à Escola Superior de Redes (ESR), unidade de serviço da Rede Nacional de Ensino e Pesquisa (RNP), que disponibilize em sua grade de curso uma trilha específica de Desenvolvimento de Sistemas voltado para as tecnologias do SIG.

Para longo prazo, uma solicitação ao CONIF de formalização, junto ao Fórum de Pró-Reitores de Pesquisa, Inovação e Pós-Graduação – FORPOG a criação de um GT para elaboração de um Mestrado Profissional em rede na Área de Ciência da Computação com ênfase em Desenvolvimento de Sistemas e Redes e Sistemas Distribuídos.

Dessa forma, os recursos repassados pela SETEC/MEC seriam investidos em um Programa de Pós-graduação na própria rede, estabelecendo, assim, que as dissertações sejam soluções que pudessem ser implantadas na Rede Federal de Educação Profissional e Tecnológica, essa proposta foi aprovada pelo CONIF e o IFPA colocou-se à disposição para coordenar o projeto.

### 4.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foi apresentado a estrutura dos Institutos Federais e a pesquisa de campo sobre o Desenvolvimento de Software realizado neles atualmente.

O fato da estrutura dos Institutos Federais ter sido concebida com uma independência organizacional dos campi em relação a reitoria, não existindo assim uma obrigação de seguir as orientações da Reitoria, possibilitou essa segregação dos

stakeholders naturalmente proporcionando o desenvolvimento paralelo de projetos de mesmo objetivo.

A pesquisa de campo comprovou que o fato ocorre em outras instituições e que os servidores da área não estão preparados para reconhecer o fator e tratá-lo, o que prova a relevância do estudo e da disseminação do conhecimento entre estes profissionais.

Os dados levantados nesta pesquisa com o objetivo de explicar as possibilidades de estruturas de DDS nos IFs, foram considerados efetivos, uma vez que conseguiram apresentar a necessidade de uma metodologia de desenvolvimento de software mais dinâmica que as atuais, para tratar a complexidade de desenvolver softwares para uma organização pública de educação multicampi.

O fato mais relevante a se considerar na pesquisa de campo talvez seja a indicação de que outros projetos existentes (SIG Colaborativo e SUAP Colaborativo) visam não apenas trabalhar o DDS no contexto de cada Instituto Federal, mas também a nível de Rede Federal. Juntar todos os IFs em uma única forma de desenvolvimento pode trazer muitos benefícios de produtividade, mas a complexidade de tratar o esforço de alinhamento de tantos desenvolvedores simultâneos pode não compensar.

Muitos estudos devem ser feitos antes de levar qualquer proposta a nível de Rede Federal para uma coordenação eficaz.

## 5 ELABORAÇÃO DA PROPOSTA

Este capítulo apresenta a Metodologia de Desenvolvimento Distribuído de Software (DDS) para as Instituições Federais (IFs), derivada de práticas descritas na literatura, pesquisa de campo, metodologias e frameworks amplamente usados no mercado, Scrum e XP. Elaborado em resposta às necessidades encontradas nos IFs, que mesmo com uma grande quantidade de profissionais habilitados para o desenvolvimento, não utilizam dessa capacidade para suprir a crescente demanda de desenvolvimento de software.

Esta proposta busca transmitir o conhecimento das metodologias e frameworks da área com uma abordagem pragmática, voltada aos contextos encontrados nos IFs. Preserva a essência do desenvolvimento de software e apresenta uma forma inovadora de tratar os conceitos comuns da área. Com isso a proposta visa não apenas introduzir o DDS nas Instituições Federais, mas também contribuir para a manutenção da harmonia entre as necessidades institucionais e as necessidades pessoais, promovendo assim um equilíbrio natural entre a força de trabalho e as tarefas que precisam ser executadas em tempo hábil.

As principais inspirações desta metodologia foram o IFSP na pesquisa de campo e o estudo de caso publicado por Lous et al. (2018). O IFSP participou ativamente, compartilhando toda sua experiência com duas entrevistas e um formulário survey, eles mostraram uma atuação dinâmica realizando um desenvolvimento distribuído com práticas ágeis e uma organização baseada em Equipes Virtuais. Já o estudo de caso da empresa multinacional que atua em três países, mostra com riquezas de detalhes como conseguiram desenvolver de forma distribuída sem perder os preceitos ágeis, também com uma abordagem de Equipes Virtuais, unindo Equipes Remotas e Indivíduos Remotos de forma colaborativa e auto-organizada.

Segundo Lous et al. (2018) a implantação do ágil foi correta e bem-sucedida, entretanto algumas observações levaram a melhoria do processo e a concepção de uma nova proposta de processo. Esse novo processo trata o time de desenvolvimento com total autonomia e delega toda a coordenação aos próprios membros, assim é possível remover a “ideia de iterações”, ou a pressão que a corrida contra o relógio causava na equipe. Isso afeta vários outros mecanismos do processo, tais como a "Reunião de Planejamento da Iteração" e as “Estimativas de Atividades”, assim deixamos de controlar o “Tempo” e passamos a controlar apenas a “Carga de

Trabalho” que a equipe suporta, com a priorização das atividades.

A metodologia construída a partir desse raciocínio também busca manter o desenvolvimento de software mais dinâmico que os casos citados, separando completamente a noção de tempo dos indivíduos da noção de tempo do projeto. Assim consideramos esta metodologia aderente aos critérios de grande dinamismo do contexto encontrado nas Instituições Federais de Educação do Brasil.

## 5.1 PRÁTICAS QUE INSPIRARAM A METODOLOGIA

Essa seção apresenta a reunião de todas as práticas úteis ao modelo distribuído de desenvolvimento de software, coletadas nas etapas anteriores, devidamente classificadas de acordo com seus possíveis impactos.

Práticas são uma descrição detalhada da execução de um processo com o auxílio ou não de um método ou ferramenta. Por muitas vezes elas acabam por ser responsáveis pelo emprego de vários dos princípios e valores determinados pela organização. As práticas aqui reunidas serviram como a principal fonte de inspiração para a construção da metodologia proposta, apresentada na seção 5.2.

### 5.1.1 Categorias de práticas

Após reunir todas as práticas em uma planilha e identificar algumas de suas características, de forma empírica, definiu-se que, a melhor forma de categorizá-las seria por áreas de impacto, uma vez que toda prática gera um impacto em uma área que visa melhorar de alguma forma. Uma vez percebido isso, as práticas receberam as seguintes categorias: Comunicação, Conhecimento, Coordenação, Informação, Modelo de Processo, Planejamento, Processo e Socialização.

### 5.1.2 Tipos de práticas

Durante a análise dos dados coletados, foi identificado algumas práticas que afetam a organização de uma forma diferente de outras, então foi elaborada a divisão por tipos de práticas, que representam o nível de impacto da prática na estrutura da organização ou no modelo de desenvolvimento.

As práticas do Tipo Estrutural, ou **Práticas Estruturantes**, são práticas que

impactam recursos básicos da organização, financeiro, pessoal, ferramentas, enfim qualquer recurso da estrutura da organização. Elas visam estabelecer recursos necessários para a execução de um modelo de processo dentro do contexto atual da organização. Geralmente essas são práticas que vão depender de outros setores para a adoção.

As práticas do Tipo Organizacional, ou **Práticas Organizacionais**, são práticas que apesar de não afetarem os recursos básicos da organização, geram mudança na organização das pessoas, seja na comunicação, socialização, transferência de conhecimento, coordenação ou até mesmo em todo o processo de desenvolvimento de software definido. Elas visam organizar os recursos de acordo com o modelo de processo.

As práticas do Tipo Estratégico, ou **Práticas Estratégicas**, são práticas que geram benefícios para atividades específicas, sem impactar a organização das pessoas ou a estrutura da organização. Elas visam estabelecer a forma como deve ser executado o processo.

### 5.1.3 Práticas reunidas

Nesta seção são apresentadas todas as práticas encontradas na pesquisa de campo e na pesquisa da literatura, além da inclusão das práticas definidas pelas metodologias mais citadas em ambas as pesquisas, que são, o **Scrum**, o **XP** e o **Kanban**.

Todas as práticas úteis ao modelo distribuído de desenvolvimento de software, que foram listadas no Quadro 8, estão devidamente classificadas de acordo com os tipos e categorias descritas nas seções anteriores, além de possuírem o campo “Origem” para determinar onde o pesquisador as encontrou.

Quadro 8 – Práticas coletadas nas pesquisas de campo e literatura

ID	Nome	Tipo	Categoria	Origem	Estudos
P001	Comunicação Interna Síncrona e Assíncrona centralizada em ferramenta única	Estruturante	Comunicação	RSL	E13,E14, E17,E21, E22,E29, E33
P002	Plataforma de Conhecimento Organizacional única, centralizada e confiável	Estruturante	Conhecimento	RSL	E13,E20, E22,E25
P003	Plataforma de Armazenamento de Dados com suporte a versionamento e permissões de acesso por grupo	Estruturante	Informação	RSL	E16,E24, E22
P004	Prover Ferramenta de Gerenciamento de Tarefas, centralizada	Estruturante	Informação	RSL	E13,E15, E21
P005	Sistemas de Informação que promovam visibilidade e transparência para cada tipo de informação de forma centralizada	Estruturante	Informação	RSL	E05,E14, E28
P006	Prover Ferramenta de Gerenciamento de Problemas, centralizada	Estruturante	Informação	RSL	E24
P007	Prover Ferramentas de Gerenciamento de Feedback	Estruturante	Informação	RSL	E13
P008	Utilizar Modelo de Desenvolvimento Iterativo e Incremental	Estruturante	Modelo de Processo	RSL	E13,E22, E28
P009	Realizar uma Transição Gradual entre Modelos de Processo	Estruturante	Modelo de Processo	RSL	E28
P010	Utilizar Modelo de Desenvolvimento Incremental sem Iterações	Estruturante	Modelo de Processo	RSL	E21
P011	Versionamento do Código	Estruturante	Processo	RSL, XP	E10,E21, E22,E24
P012	Prover Ferramenta de Provisionamento de Ambientes, automatizada	Estruturante	Processo	RSL	E15
P013	Utilizar Arquitetura de Software de menor custo de implantação	Estruturante	Processo	RSL	E16
P014	Prover Ferramentas de Colaboração	Estruturante	Socialização	RSL	E13,E15

P015	Comunicação direta com Canal Aberto entre todos Stakeholders	Organizacional	Comunicação	RSL	E10,E03, E15
P016	Elaborar Plano de Comunicação Interna	Organizacional	Comunicação	RSL	E14,E28
P017	Definir Nível de Maturidade da Comunicação na Organização	Organizacional	Comunicação	RSL	E17
P018	Definir Normas de Comunicação	Organizacional	Comunicação	RSL	E14
P019	Definir responsabilidades e limites entre a comunicação de Equipes Remotas	Organizacional	Comunicação	RSL	E04
P020	Padronizar os Termos Linguísticos Utilizados	Organizacional	Comunicação	RSL	E09
P021	Remoção do sistema de e-mail como comunicação interna	Organizacional	Comunicação	RSL	E21
P022	Utilizar Histórias de Usuário para expressar tarefas	Organizacional	Comunicação	RSL	E16
P023	Cliente presente	Organizacional	Comunicação	XP	
P024	Metáfora	Organizacional	Comunicação	XP	
P025	Programação em Pares	Organizacional	Conhecimento	RSL, XP	E03,E08, E15,E21, E33
P026	Elaborar Plano de Transferência de Conhecimento	Organizacional	Conhecimento	RSL	E32
P027	Planejar estratégia de Criação, Armazenamento, Gestão e Transmissão de Conhecimento	Organizacional	Conhecimento	RSL	E25
P028	Transferência de Conhecimento via Registro	Organizacional	Conhecimento	RSL	E28
P029	Reunião Diária	Organizacional	Coordenação	RSL, IFs, Scrum, XP	E03,E08, E16,E21, E22,E26, E28,E32, E33,E34, E22
P030	Reunião de Planejamento da Iteração	Organizacional	Coordenação	RSL, IFs,	E16,E21

				Scrum, XP	
P031	Reunião de Revisão da Iteração	Organizacional	Coordenação	RSL, IFs, Scrum	E08,E16, E15
P032	Definir valores necessários para Equipes se tornarem Auto-organizáveis	Organizacional	Coordenação	RSL	E10,E11, E14,E21, E28
P033	Distribuir Tarefas por sequência ou dependência	Organizacional	Coordenação	RSL	E19,E22, E23,E07
P034	Planejar Timebox Sobreposto de ao menos uma hora entre Indivíduos Remotos	Organizacional	Coordenação	RSL	E03,E14, E15,E22
P035	Define papéis e responsabilidades detalhadamente	Organizacional	Coordenação	RSL	E14,E30
P036	Definir equipes com diversidade de habilidades técnicas, vivências, experiências, culturas e perfis comportamentais.	Organizacional	Coordenação	RSL	E32,E28
P037	Definir responsáveis pelo Esforço de Alinhamento	Organizacional	Coordenação	RSL	E01,E04
P038	Prever o Esforço de Alinhamento	Organizacional	Coordenação	RSL	E01,E04
P039	Definir Mindset Organizacional	Organizacional	Coordenação	RSL	E05
P040	Definir um Dono da Comunicação	Organizacional	Coordenação	RSL	E02
P041	Padronizar método de trabalho das Equipes Remotas	Organizacional	Coordenação	RSL	E14
P042	Permitir transferência de tarefas entre Equipes Remotas	Organizacional	Coordenação	RSL	E14
P043	Planejamento da Iteração Ad-hoc	Organizacional	Coordenação	RSL	E21
P044	Planejamento da Iteração de baixo para cima	Organizacional	Coordenação	RSL	E28
P045	Ritmo Sustentável	Organizacional	Coordenação	XP	
P046	Explicação dos processos de forma visual, detalhada e centralizada	Organizacional	Informação	RSL	E10,E13

P047	Status do Projeto com meios de monitoramento	Organizacional	Informação	RSL	E14
P048	Reunião de Retrospectiva da Iteração	Organizacional	Processo	RSL, IFs, Scrum	E16,E21, E29,E33
P049	Integração Contínua do Código	Organizacional	Processo	RSL, XP	E03,E16, E21,E28, E29,E32
P050	Propriedade Coletiva	Organizacional	Processo	RSL, XP	E03,E33
P051	Padronização do Código	Organizacional	Processo	RSL, XP	E22
P052	Refatoração do Código	Organizacional	Processo	RSL, XP	E03
P053	Entrega Contínua do Código	Organizacional	Processo	RSL	E04,E11, E16,E28, E03,E29
P054	Melhoria Contínua (Kaizen)	Organizacional	Processo	RSL	E16,E29, E32,E33
P055	Processos modelados de forma Ágil	Organizacional	Processo	RSL	E03
P056	Entregas Frequentes	Organizacional	Processo	Ágil	
P057	Testes constantes	Organizacional	Processo	XP	
P058	Versões Pequenas	Organizacional	Processo	XP	
P059	Realizar reuniões de forma presencial como mecanismo de socialização, aumenta a confiança	Organizacional	Socialização	RSL	E08,E14, E32,E16
P060	Diferenças culturais devem ser identificadas e gerenciadas	Organizacional	Socialização	RSL	E14
P061	Estabelecer protocolos sociais comuns	Organizacional	Socialização	RSL	E24
P062	Incentivar os stakeholders a liderarem socializações que promovam a quebra de barreiras socioculturais e colaborativas	Organizacional	Socialização	RSL	E24

P063	Realizar reuniões por demanda, stakeholders podem solicitar quando necessário	Organizacional	Socialização	RSL	E21
P064	Tratar pontos cegos culturais	Organizacional	Socialização	RSL	E01
P065	Comunicação Simplificada com Modelos Visuais	Estratégica	Comunicação	RSL	E09,E15
P066	Brainstorming	Estratégica	Comunicação	RSL	E29
P067	Comunicação de texto com Modelos de mensagens	Estratégica	Comunicação	RSL	E22
P068	Feedback direto ao Dono do Produto	Estratégica	Comunicação	RSL	E08
P069	Reunião Entrega de Produto com demonstração e Feedback dos Stakeholders	Estratégica	Comunicação	RSL	E08
P070	Reuniões devem ter as pautas comunicadas detalhadamente para garantir que os participantes estejam preparados	Estratégica	Comunicação	RSL	E14
P071	Tarefas concluídas devem ser informadas a todos stakeholders	Estratégica	Comunicação	RSL	E13
P072	Designar um mentor para transferir conhecimento à novos membros	Estratégica	Conhecimento	RSL	E14,E32
P073	Incentivar o conhecimento de novas tecnologias	Estratégica	Conhecimento	RSL	E14
P074	Preferencialmente utilize compartilhamento de tela para transferência de conhecimento	Estratégica	Conhecimento	RSL	E22
P075	Preferencialmente utilize comunicação síncrona para transferência de conhecimento	Estratégica	Conhecimento	RSL	E22
P076	Reverse Shadowing	Estratégica	Conhecimento	RSL	E32
P077	Utilizar a comunicação assíncrona como Base de Conhecimento	Estratégica	Conhecimento	RSL	E13
P078	Utilizar as Reuniões frequentes para orientações metodológicas	Estratégica	Conhecimento	RSL	E29

P079	Incentivar o comprometimento com o objetivo geral	Estratégica	Coordenação	RSL, XP, Scrum	E28
P080	Produto Mínimo Viável (MVP)	Estratégica	Coordenação	RSL, XP, Lean	E32
P081	Refinamento do Backlog do Produto	Estratégica	Coordenação	RSL, Scrum	E16,E28, E30,E15
P082	Histórias de Usuário bem detalhadas (baixa granularidade)	Estratégica	Coordenação	RSL, XP	E22,E29, E32
P083	Gestão visual do fluxo de trabalho de forma organizada e centralizada	Estratégica	Coordenação	RSL, Kanban	E03
P084	Elaboração do Backlog do Produto	Estratégica	Coordenação	RSL, Scrum	
P085	Reservar 20% do tempo para os colaboradores explorarem ideias inovadoras relacionadas aos produtos da organização, experimentar e aprender em Ambiente de Teste	Estratégica	Coordenação	RSL	E16,E21, E33
P086	Reunião de Alinhamento de Equipes Remotas	Estratégica	Coordenação	RSL	E28,E34
P087	Carga de trabalho da Equipe de Desenvolvimento sempre estável	Estratégica	Coordenação	RSL	E28
P088	Coordenação eficaz mantendo o equilíbrio entre os processos sociais, produtividade da equipe e o bem-estar individual	Estratégica	Coordenação	RSL	E26
P089	Coordenação melhorada a partir de discursões de alinhamento do trabalho	Estratégica	Coordenação	RSL	E24
P090	Dono do Produto deve conhecer de forma próxima cada Equipe Remota	Estratégica	Coordenação	RSL	E14
P091	Incentivar o desenvolvimento de habilidades colaborativas	Estratégica	Coordenação	RSL	E14
P092	Integrar especialistas em decisões	Estratégica	Coordenação	RSL	E20

P093	Iterações não mudam as regras durante execução	Estratégica	Coordenação	RSL	E15
P094	Manter a diversidade cultural em equipes	Estratégica	Coordenação	RSL	E29
P095	Planejar os horários das reuniões de acordo com o Dono do Produto	Estratégica	Coordenação	RSL	E08
P096	Reuniões devem ter os participantes definidos de acordo com os assuntos em pauta	Estratégica	Coordenação	RSL	E14
P097	Reuniões devem ter seus requisitos determinados	Estratégica	Coordenação	RSL	E14
P098	Rotação de Papéis	Estratégica	Coordenação	RSL	E33
P099	Utilizar Equipes Remotas apenas para Desenvolvimento e Testes	Estratégica	Coordenação	RSL	E22
P100	Status do Projeto visível a todos os Stakeholders	Estratégica	Informação	RSL	E10,E14
P101	Catálogo de alterações dos testes	Estratégica	Informação	RSL	E15
P102	Documentação enxuta	Estratégica	Informação	RSL	E28
P103	Gerir informações em ferramenta que facilite alterações rápidas (Planilhas)	Estratégica	Informação	RSL	E15
P104	Informações de Projeto publicadas em site com alto nível de abstração	Estratégica	Informação	RSL	E14
P105	Reuniões registradas em ATA, Áudio e Vídeo	Estratégica	Informação	RSL	E24
P106	Status do Projeto com fluxo de informações rastreável	Estratégica	Informação	RSL	E14
P107	Planejar reuniões fixas para reduzir as desvantagens individuais	Estratégica	Planejamento	RSL	E28,E14
P108	Calendário de Entregas sempre atualizado	Estratégica	Planejamento	RSL	E22
P109	Metas do Projeto com reavaliações frequentes	Estratégica	Planejamento	RSL	E28
P110	Planejamento de Lançamento	Estratégica	Planejamento	RSL	E04

P111	Implantação Contínua do Código	Estratégica	Processo	RSL	E16,E21, E33,E15
P112	Revisão do Código por demanda feita pela Equipe de Desenvolvimento	Estratégica	Processo	RSL	E08,E21, E33
P113	Definição de Pronto para Organização ou Projeto	Estratégica	Processo	RSL	E15,E34
P114	Análise estática do Código	Estratégica	Processo	RSL	E21
P115	Atribuir autoridade para o Dono do Produto exigir a conclusão das tarefas delegadas	Estratégica	Processo	RSL	E14
P116	Atualização de Histórias de Usuário a partir de registro de reuniões	Estratégica	Processo	RSL	E15
P117	Comunicação aberta durante a execução de atividades	Estratégica	Processo	RSL	E13
P118	Definir o tempo para um registro de atividade ser considerado "Defeito"	Estratégica	Processo	RSL	E15
P119	Definir uma estimativa de esforço das Histórias de Usuário	Estratégica	Processo	RSL	E06
P120	Gerenciar Dívida Técnica	Estratégica	Processo	RSL	E32
P121	Implementação gradual de mudanças	Estratégica	Processo	RSL	E16
P122	Incentivar identificação de falhas	Estratégica	Processo	RSL	E28
P123	Manter o equilíbrio entre implementação de recursos e qualidade	Estratégica	Processo	RSL	E28
P124	Novas Histórias de Usuário devem ser informadas a todos stakeholders	Estratégica	Processo	RSL	E13
P125	Orientar os stakeholders a executar uma atividade por vez	Estratégica	Processo	RSL	E28
P126	Padronização dinâmica, baseada em rotinas sem hierarquias	Estratégica	Processo	RSL	E27
P127	Pare a Iteração	Estratégica	Processo	RSL	E32
P128	Ramificação do Código	Estratégica	Processo	RSL	E12
P129	Remover estimativas de tamanho de tarefas	Estratégica	Processo	RSL	E21

P130	Toda a equipe deve interromper o desenvolvimento para se concentrar na resolução de problemas quando um número de problemas de qualidade pré-definido for atingido	Estratégica	Processo	RSL	E32
P131	Visita a Equipes Remotas	Estratégica	Socialização	RSL	E14,E08, E29,E34
P132	Realizar Reuniões Individuais (One-on-one)	Estratégica	Socialização	RSL	E21,E26, E33
P133	Incentivar a exploração de novas formas de construir confiança entre indivíduos remotos	Estratégica	Socialização	RSL	E20
P134	Incentivar a socialização através de conversas informais	Estratégica	Socialização	RSL	E24
P135	Incentivar a socialização no início do projeto	Estratégica	Socialização	RSL	E14
P136	Permitir socialização por canais informais	Estratégica	Socialização	RSL	E24
P137	Prover agenda centralizada (Publicizar registro de reunião)	Estratégica	Socialização	RSL	E24
P138	Realizar reuniões baseadas em slides	Estratégica	Socialização	RSL	E21
P139	Reunião de Retrospectiva da Iteração Gamificada	Estratégica	Socialização	RSL	E15
P140	Utilização de Áudio e Vídeo em reuniões remotas	Estratégica	Socialização	RSL	E08
P141	Utilizar o humor para aumentar a socialização	Estratégica	Socialização	RSL	E14
P142	Utilizar técnica de Modelos culturais para simplificar a visualização dos problemas comuns	Estratégica	Socialização	RSL	E18
P143	Utilizar videoconferência para socialização informal	Estratégica	Socialização	RSL	E14
P144	Utilize reuniões para a construção de confiança entre os stakeholders	Estratégica	Socialização	RSL	E22

Fonte: O autor (2022).

## 5.2 METODOLOGIA DE DDS SOBERANO

Desenvolvimento Distribuído de Software (DDS) Soberano é uma metodologia de desenvolvimento de software ágil de forma distribuída, e escalável, que tem o propósito de orientar a coordenação de pessoas em ambientes totalmente virtuais e dinâmicos. Apesar de tratar de desenvolvimento de software, seu foco são as questões de coordenação das pessoas em ambientes virtuais.

Esta metodologia é formada por Diretrizes, Papéis, Artefatos, Cerimônias e Atividades. Conta com um Fluxo de Trabalho para facilitar o entendimento da metodologia, apresentando a direção que segue as principais cerimônias e atividades.

### 5.2.1 Requisitos

Um requisito fundamental desta metodologia é o conhecimento e prática de utilização de desenvolvimento de software ágil, em especial as orientações propostas pelo Framework Scrum e a Metodologia XP.

Em relação à adoção de um Time Soberano com total autonomia, é necessário que a organização tenha um nível de maturidade capaz de prover a liberdade e a estrutura de informações necessária para construir uma equipe auto-organizada. O cerceamento da autonomia dos desenvolvedores é um alto risco à eficácia desta metodologia.

Para a correta aplicação das orientações definidas nesta metodologia é necessário que a organização já esteja executando de forma satisfatória as seguintes práticas:

- Padronização do Código;
- Propriedade coletiva do Código;
- Versionamento do Código;
- Gestão visual do fluxo de trabalho de forma organizada e centralizada (Kanban);
- Comunicação síncrona e assíncrona centralizada em ferramenta única.

## 5.2.2 Objetivos

Esta metodologia tem como principais objetivos:

**Desenvolvimento Distribuído de Software Ágil:** Elevar uma organização que desenvolve de forma ágil para um modelo de Desenvolvimento Distribuído de Software Ágil totalmente virtual.

**Localização geográfica dos desenvolvedores Irrelevante:** O desenvolvimento totalmente virtual possibilita a independência de localização física dos colaboradores, entretanto possui desafios extras, apesar de existir muitas ferramentas que suportam essa nova forma de trabalho, espera-se que ocorra a construção de novas soluções virtuais que reduzam ainda mais os desafios inerentes ao modelo DDS, uma vez que toda a comunicação se torna mediada por ferramentas web e deixa de existir a comunicação presencial rotineira.

**Time de Desenvolvimento Soberano:** Manter os desenvolvedores com foco total nas atividades, blindar a equipe de intervenções externas e possibilitar a manutenção das necessidades internas através da auto-organização de uma equipe multidisciplinar.

**Com o conceito de “Time Soberano” passa a ser possível separar o tempo dos indivíduos do tempo do projeto:** O Time Soberano blinda os direitos individuais ao mesmo tempo que promove o atendimento às demandas de forma dinâmica, assim é possível realizar o desenvolvimento de um projeto rapidamente com todos os desenvolvedores disponíveis ou atender a vários projetos de forma gradual, enquanto alguns desenvolvedores desfrutam de férias ou licenças.

**Adaptação dinâmica:** A capacidade única de adaptação constante é a parte mais complexa desta metodologia, pensada com o objetivo de facilitar a resolução rápida de novas demandas que podem surgir, consiste na designação dos desenvolvedores do Time Soberano para atendimento das demandas organizadas em forma de projetos.

**Gerência de tempo dinâmica:** A fim de facilitar a adesão ao novo e garantir uma adaptação dinâmica, o tempo não é limitado ao que é planejado, a qualquer momento, uma nova tarefa pode ser priorizada e passar a frente de todas as planejadas.

**Comunicação como base:** Sendo a comunicação a base de todos os processos sociais, seja para reduzir ou aumentar, ela sempre terá um grande impacto sobre

todos os fatores.

**Comunicação como conhecimento:** Tratá-la como conhecimento facilita a organização de experiências para reduzir o tempo de solução de novas demandas.

**Comunicação por demanda:** Uma vez que a grande quantidade de reuniões pode reduzir o desempenho dos desenvolvedores, foi necessário implantar uma forma de comunicação mais eficiente, a comunicação síncrona passa a ser por demanda e a comunicação assíncrona passa a ser padrão.

**Reduzir o impacto do esforço de alinhamento sobre os Desenvolvedores:** A coordenação de pessoas é a chave de um processo de DDS, porém a comunicação necessária para o alinhamento das atividades pode consumir um tempo precioso para a organização, encontrar o equilíbrio certo do nível de comunicação necessária passa a ser um dever de todos Stakeholders.

### 5.2.3 Papéis

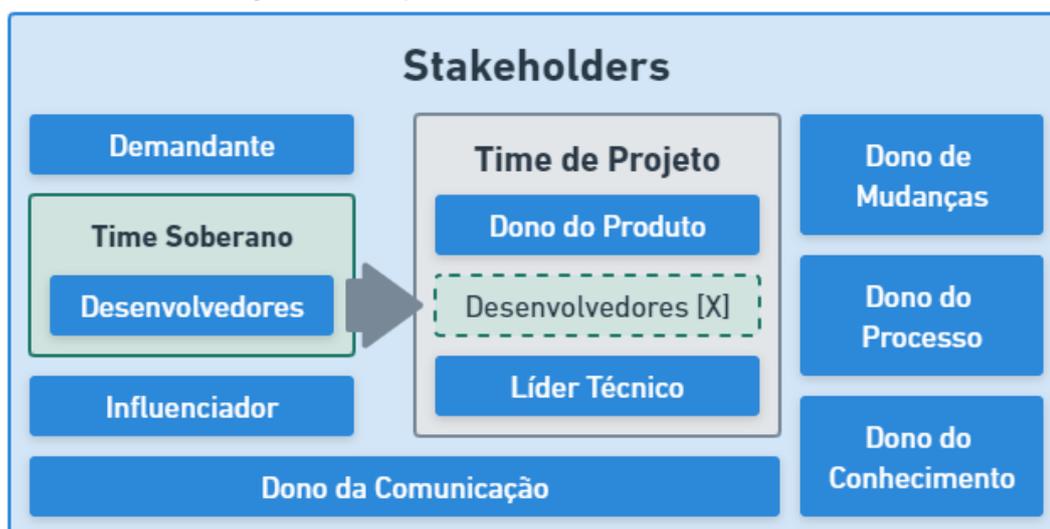
As pessoas assumem muitos papéis ao longo de suas vidas, no âmbito familiar os papéis são definidos pelo nível de parentesco, por exemplo “filho”, “irmão”, “pai”, “avô”, no âmbito do trabalho geralmente são definidos pelo status social, “o gestor”, “o empresário”. No desenvolvimento de software os papéis são definidos pelos tipos de responsabilidades atribuídas a eles, cada metodologia pode ter papéis com responsabilidades e atividades diferentes.

Nesta metodologia a construção destes papéis foram baseadas nas metodologias e frameworks ágeis consolidados no mercado e nos relatos encontrados na literatura. O Scrum é referência dos papéis chave, todos os papéis e responsabilidades definidas por ele são seguidas, com algumas pequenas alterações, responsabilidades acrescentadas, removidas ou compartilhadas, devido a mudança do fluxo de trabalho e a possibilidade de designação dos desenvolvedores para variados Times de Projeto.

Dentre os relatos, foram encontrados três Bens Intangíveis comuns às organizações, estes foram determinados como fundamentais para o processo de evolução das organizações, devido a grande quantidade de fatores ligados a eles e a necessidade de acompanhar e gerenciar esses fatores. Considerado o exposto, ficou determinada a necessidade de criar três papéis: **Dono da Comunicação, Dono do Conhecimento e Dono de Mudanças.**

Segundo o PMBOK (PMI, 2017), **Stakeholders** de um projeto são todos os indivíduos, grupo ou organização que pode afetar, ser afetado ou sentir-se afetado por uma decisão, atividade ou resultado de um projeto. Os Stakeholders do projeto podem ser internos ou externos ao projeto, e podem estar envolvidos ativamente ou passivamente, ou não estar cientes do projeto. Eles podem ter um impacto positivo ou negativo no projeto, ou ser impactados de forma positiva ou negativa pelo projeto.

Figura 11 – Papéis do desenvolvimento de software



Fonte: O autor (2022).

### 5.2.3.1 Dono da Comunicação

O **Dono da Comunicação** se compromete a garantir o entendimento de como cada tipo de informação deve ser transmitida dentro e fora da organização e incentivar as relações de comunicação, socialização, cooperação e colaboração. Ele deve zelar pela remoção de barreiras de comunicação e socioculturais entre todos stakeholders.

#### Responsável por:

- Estruturar modelos de comunicação, textual e visual que facilitem a compreensão de informações complexas;
- Definir o nível de maturidade da comunicação interna da organização;
- Identificar, analisar e encontrar formas de manter um bom clima organizacional;
- Registrar e publicar de forma simples, clara e efetiva o Mindset Organizacional e a Cultura Organizacional;

- Encontrar formas de fortalecer a Cultura Organizacional;
- Encontrar formas manter um Mindset Organizacional positivo;
- Elaborar um Plano de Comunicação Interna, considerando que toda comunicação deve ser veiculada por canais virtuais;
- Estabelecer protocolos sociais.

Toda cerimônia requer alguém para coordenar a execução, apesar de ser seu dever guiar a forma como as cerimônias devem ser elaboradas, não é necessariamente o responsável pela execução.

Desejável que possua especialização em Comunicação.

#### 5.2.3.2 Dono do Conhecimento

O **Dono do Conhecimento** tem a responsabilidade de prover, estruturar e definir a forma como devem ser armazenados os dados, informações e conhecimento da organização. Tem o dever de orientar a forma como o conteúdo deve entrar e sair deste armazenamento.

Deve buscar formas de orientar todos os stakeholders sobre os deveres de todos em zelar pela integridade e confiabilidade dos dados que guarda, assim como pela eficácia, legalidade e segurança da transmissão deles.

Não é de sua responsabilidade o backup dos dados, porém deve monitorar e checar sua integridade e segurança.

Desejável que possua especialização em Ciência dos Dados.

#### 5.2.3.3 Demandantes

**Demandantes** são todos que de alguma forma motivaram uma demanda para os desenvolvedores, seja criação ou manutenção do software.

#### 5.2.3.4 Desenvolvedores

Os **Desenvolvedores** são todas as pessoas que fazem parte da execução do desenvolvimento de produtos, criação e manutenção do software, assim como definido no Framework Scrum para “Developers”. As pessoas que assumem o papel

de desenvolvedor, se comprometem a buscar todas as habilidades necessárias para concluir as entregas definidas em cada projeto.

Todos os Desenvolvedores da Organização são tratados como um “**Time Soberano**”, pois tem total autoridade para decidir os caminhos necessários para concluir o produto, neste caso tendo os outros papéis como suporte para as tomadas de decisão relacionadas ao desenvolvimento. Nesta perspectiva, é interessante evidenciar que a tentativa de prover autoridade total do desenvolvimento de software para os desenvolvedores, em um Time Soberano, não leva a uma inversão do poder, mas sim a uma compreensão de igualdade. Assim como o Dono do Produto zela pelo produto e o Dono do Processo zela pelo processo, o Time Soberano tem a responsabilidade de zelar pela execução dos projetos.

São pessoas que estão comprometidas em criar qualquer aspecto de um Incremento utilizável, com múltiplas habilidades, essenciais para a Comunicação, Socialização, Coordenação e Auto-organização, eles decidem internamente quem faz o quê, quando e como. Posto isto, temos apenas um time de desenvolvimento, independentemente do tamanho, essa equipe tem autoridade e autonomia para ser total responsável pela tradução dos desejos e dores dos Demandantes em produto utilizável.

#### 5.2.3.5 Líder Técnico

Um desenvolvedor designado a liderar os demais desenvolvedores nos aspectos técnicos dentro de um projeto específico. Ao se tornar líder de um projeto, o desenvolvedor deixa de estar disponível para tarefas de outros projetos. Sua principal responsabilidade é a de revisar todo o código do produto.

##### **Responsável por:**

- Garantir a padronização do código estabelecida;
- Garantir coesão na integração do código do projeto;
- Prover e manter ambientes da aplicação;
- Auxiliar o Dono do Produto no alinhamento dos demais desenvolvedores com o objetivo do projeto;
- Auxiliar o Dono do Processo com o gerenciamento dos impedimentos do projeto;

- Incentivar a auto-organização dos colaboradores do projeto.

#### 5.2.3.6 Dono do Produto

O **Dono do Produto**, “Product Owner” do Scrum, se compromete em entregar o maior valor possível para cada produto, dentro dos recursos de mão de obra, financeiro e tempo disponíveis. Ele é responsável por identificar o escopo do produto e planejar as estratégias necessárias para alcançar o resultado desejado, manter os Desenvolvedores alinhados com o escopo e principalmente atualizar o escopo constantemente, de maneira a prover uma entrega de maior valor.

Deve buscar a melhor forma de apresentar o projeto. Prezar pela transparência, visibilidade e compreensão de todos os detalhes. Assim como sanar as dúvidas dos desenvolvedores.

Um projeto deve conter, minimamente, uma meta clara e objetiva (**Objetivo do Produto**) e uma lista de tarefas priorizadas (**Backlog do Produto Priorizado**).

Para que um Time Soberano seja viável é fundamental que todos os projetos tenham suas tarefas priorizadas e é obrigação do Dono do Produto assegurar isso, entretanto não é necessário, nem recomendável que execute essa função sozinho, o Dono do Produto deve manter a comunicação com todos os Stakeholders a fim de identificar a melhor priorização para o projeto.

#### **Responsável por:**

- Identificar as necessidades do projeto;
- Elaborar e manter o Backlog do Produto Priorizado;
- Conduzir a Reunião de Planejamento do Produto;
- Conduzir a Reunião de Entrega do Produto;
- Auxiliar os Desenvolvedores a entender e manter o foco no Objetivo do Produto.

#### 5.2.3.7 Time de Projeto

O **Time de Projeto** são todas as pessoas que se comprometeram com o objetivo geral proposto pelo projeto, Dono do Produto, Líder Técnico e os Desenvolvedores [X].

Assim como definido no Framework Scrum para o “Scrum Team”, ela é uma

unidade de profissionais focados em um objetivo de cada vez, o Objetivo do Produto. Essa equipe é responsável por todas as atividades relacionadas ao produto, desde a colaboração com stakeholder, verificação, manutenção, operação, experimentação, pesquisa e desenvolvimento, e qualquer outra coisa que possa ser necessária. Entretanto, uma grande diferença é a condição de que os desenvolvedores não são necessariamente exclusivos de um projeto.

Os **Desenvolvedores [X]** são solicitados pelo Time do Projeto de acordo com a demanda de tarefas priorizadas como urgentes ou importantes, o quantitativo de desenvolvedores pode variar de acordo com o decorrer do projeto.

#### 5.2.3.8 Dono do Processo

O **Dono do Processo** é responsável por garantir o entendimento e a execução correta de todo o processo, ele se compromete a manter todos os envolvidos cientes de como e quando devem interagir com cada envolvido no fluxo do processo. Pode ser comparado ao “Scrum Master” do Framework Scrum.

Ele é responsável pela eficácia do desenvolvimento de software em todos os projetos da organização, bem como o equilíbrio do quantitativo de desenvolvedores por projeto da organização. Esse alinhamento pode ocorrer semanalmente ou até diariamente, a depender da demanda de cada projeto.

Deve realizar o gerenciamento de impedimentos de tarefas de todos os projetos, com o auxílio do Líder Técnico designado para cada projeto.

O Dono do Processo deve garantir a melhoria contínua de todos os processos da organização, assim como encontrar novas técnicas organizacionais e estratégicas ou sugerir mudanças estruturais/organizacionais ao Dono de Mudanças.

O processo de desenvolvimento de software é orgânico, em constante evolução, tendo isso em mente é responsabilidade do Dono do Processo acompanhar essa evolução, treinar e orientar a organização na adoção de processos, planejar e aconselhar implementações de técnicas dentro da organização.

Ele deve acompanhar o nível do Esforço de Alinhamento necessário para a execução do processo e realizar as mudanças necessárias para reduzir esse esforço.

#### 5.2.3.9 Dono de Mudanças

O **Dono de Mudanças** é um papel ocioso, que é ativado apenas quando um projeto de mudança é aprovado pela gestão. Normalmente quem será designado para

este papel é um Desenvolvedor que já atuou como Dono de Processo e possui as características adequadas para a realização de um trabalho de longa duração e alto custo.

Normalmente este papel ficará ativo poucas vezes durante a vida da organização, mas é fundamental em uma grande mudança de paradigma, como a proposta por essa metodologia.

Ele tem a complexa tarefa de liderar os projetos de mudança que exigem a reestruturação da organização, guiar a adoção gradual de cada mudança de forma a manter a estabilidade e eficácia de todo o processo atual, ao fim dos projetos de mudança o papel de Dono de Mudança fica ocioso, porém o Dono do Processo fica a cargo de reunir novas ideias que promovam um novo projeto de mudanças.

**Responsável por:**

- Realizar análise de impacto das mudanças propostas;
- Identificar, analisar e preparar táticas de mitigação de riscos;
- Planejar a execução gradual das mudanças;
- Informar, preparar, orientar e apoiar os stakeholders impactados.

#### 5.2.3.10 Influenciadores

**Influenciadores** são todos que mesmo não criando uma demanda, influenciam uma alteração de como as coisas devem ser feitas, um exemplo, pode ser as forças políticas, de leis ou incentivos.

#### 5.2.4 Artefatos

Os artefatos são a centralização de informações que promove transparência e organização para o trabalho. São considerados nesta metodologia a principal forma de comunicação para coordenação do trabalho. Esta metodologia busca trabalhar a organização de artefatos baseada em modelos e exemplos, os modelos devem ser dinamicamente definidos para cada caso de uso.

##### 5.2.4.1 Regras do Jogo

Definido em comum acordo com o Time Soberano, tem como fundamento

garantir ao Time Soberano, o poder de definir as regras do processo de forma dinâmica.

**Gerente:** Dono do Processo

**Deve definir:**

- Critérios de definição do Dono de Produto;
- Critérios de definição do Desenvolvedor Líder;
- Critérios de designação de Desenvolvedores para projeto;
- Definição de Pronto da organização;
- Critérios para substituir a Definição de Pronto da organização por uma específica do projeto.

#### 5.2.4.2 Plano de Comunicação Interna

Um Plano de Comunicação Interna não pode ser muito grande ou muito engessado, ele deve ser simples, direto e tão fácil de modificar quanto a tecnologia evolui. Ele tem como objetivo garantir que a informação chegue da forma correta para as pessoas corretas.

**Gerente:** Dono da Comunicação

**Deve definir:**

- Os meios de comunicação que devem ser utilizados;
- Quando os meios de comunicação devem ser utilizados;
- Qual tipo de informação deve circular em cada canal.

#### 5.2.4.3 Briefing do Produto

Este artefato deve ser o mais simples possível, a sua função é apresentar a ideia do produto de forma clara, concisa e transparente. Ele deve conter o mínimo de informações necessárias para o entendimento do projeto.

**Gerente:** Dono do Produto

**Deve definir:**

- O que o demandante deseja;
- O objetivo do produto;
- Qual o menor tamanho viável para o produto;

- A Visão do Produto em uma frase.

#### 5.2.4.4 Plano de Entregas do Produto

Este artefato consiste no agrupamento de histórias de usuário que potencialmente formaram um incremento utilizável, a entrega desse incremento é feita através da Reunião de Entrega do Produto, que chamamos de Marco de Entrega.

O plano pode possuir quantos marcos de entrega precisar, os marcos podem ser estipulados por tempo, quantidade de itens ou por funcionalidades relevantes para o demandante.

O primeiro marco de entrega também marca o fim da etapa de Criação de Produto, após esse marco o ciclo de desenvolvimento se repete durante toda a etapa de Manutenção de Produto, o fim desta etapa pode ser representado no plano com um objetivo final, mas geralmente não possui um tempo determinado.

Este plano é elaborado durante a Reunião de Idealização de Produto e refinado durante a Reunião de Planejamento do Produto, entretanto pode ser atualizado pelo Dono do Produto a qualquer momento.

**Gerente:** Dono do Produto

#### 5.2.4.5 Backlog de Produto Priorizado

Lista de tarefas sobre o que é necessário para melhorar o produto. A priorização do Backlog do Produto é essencial para que esta metodologia funcione, apesar de ser razoavelmente priorizada no início do projeto, sua priorização deve ser contínua, até o fim do projeto. A priorização das tarefas, fazem parte do Refinamento do Backlog de Produto.

Todo Backlog de Produto deve conter o **Objetivo do Produto**, que deve servir como parâmetro de satisfação, mesmo que não se tenha concluído todo o trabalho previsto no backlog o objetivo deve ser minimamente cumprido.

**Gerente:** Dono do Produto

**Seus itens devem:**

- Ser claramente definidos, em linguagem padronizada e inteligível por todos os stakeholders;
- Ser descritos no formato de “**Histórias de Usuário**”;
- Possuir Tarefas menores que um dia de trabalho;

- Possuir apenas o detalhamento necessário para o entendimento.

#### 5.2.4.6 Backlog de Defeitos

Lista de tarefas que apresentaram algum defeito que impediu sua conclusão. Tem o objetivo de sinalizar quais tarefas precisam de revisão, fica em responsabilidade do Líder Técnico revisar ou designar um desenvolvedor de conhecimento específico do problema para solucioná-lo.

Ela pode ser visualizada de diversas formas diferentes, a depender da ferramenta usada, caso a ferramenta disponha de filtro por etiquetas (Tags) utilize desta forma, mantendo o Backlog de Defeitos, junto ao Backlog do Produto.

Importante ressaltar que “Defeitos” são diferentes de “Dívida Técnica”, uma Dívida Técnica é uma escolha da equipe em relação às prioridades do Objetivo do Produto, segundo Ward Cunningham a equipe escolhe assumir essa dívida, sabendo que se o projeto continuar esse código terá que ser refatorado em algum momento para “pagar a dívida”. Uma Dívida Técnica não deve entrar no Backlog de Defeitos, um caminho para tratá-la pode ser definido no artefato “Regras do Jogo”, tal como marcá-la com uma etiqueta.

**Gerente:** Líder Técnico

#### 5.2.4.7 Backlog de Processos

Lista de ideias de possíveis melhorias de processo, detalhando possíveis problemas e soluções.

Como não existe uma reunião de retrospectiva do processo, todas as informações referentes a mudanças do processo atual são listadas no Backlog de Processos, como uma “Brainstorming Assíncrona”, todos podem contribuir com as ideias listadas em diferentes níveis de detalhamento, contribuindo com as ideias postadas e sem poder remover ou denegrir as ideias anteriores.

**Gerente:** Dono do Processo

#### 5.2.4.8 Backlog de Mudanças

Lista de orientações de melhorias do processo que demandam grande esforço de mudança, seja em coordenação de indivíduos ou recursos financeiros e estruturais.

Segue a mesma lógica do Backlog de Processos, inclusive normalmente será derivada de ideias já relacionadas no Backlog de Processos, ela passa a existir quando ideias de mudanças tomam proporções que afetam a estrutura organizacional, impactam financeiramente ou afetam aos modelos de processo utilizados. Geralmente essas mudanças serão enriquecidas com alto grau de detalhamento para a construção de um projeto de longo prazo.

**Gerente:** Dono do Processo e Dono de Mudanças

### 5.2.5 Cerimônias

Todas as cerimônias são virtuais e contam com o auxílio de ferramentas de comunicação síncronas ou assíncronas para existirem. As cerimônias aqui descritas não são necessariamente uma reunião, meios de comunicação assíncronos podem simplificar os métodos definidos.

#### 5.2.5.1 Dança das Cadeiras

A cerimônia “Dança das Cadeiras” ocorre sempre que é necessário designar um membro do Time Soberano para a atuação em um papel que está vago. Ela pode ser feita de diversas formas diferentes, pode ser uma reunião ou um jogo, a depender das regras definidas no artefato “Regras do Jogo”.

**Ela está prevista para ocorrer:**

- Quando surgir um novo projeto, onde é necessário a designação de um Dono de Produto e um Líder Técnico para compor o Time do Projeto;
- Quando um Dono de Produto solicitar mais desenvolvedores para integrar seu projeto;
- Quando for necessário a atuação de um Dono de Mudanças;
- Quando algum papel integral (Dono do Processo, Dono da Comunicação e Dono do Conhecimento) ficar vago.

#### 5.2.5.2 Reunião de Idealização do Produto

Esta cerimônia tem o objetivo de prover todas as informações pertinentes para o desenvolvimento do produto, esse conjunto de informações deve seguir o modelo definido para o artefato Briefing do Produto. Recomendável que esta cerimônia seja gravada em áudio e vídeo, para consulta posterior.

**As informações coletadas devem ser o suficiente para o Dono do Produto:**

- Preparar o Briefing do Produto;
- Preparar o esboço do Plano de Entregas do Produto;
- Preparar o esboço do Backlog do Produto Priorizado;
- Preparar a apresentação do projeto.

**5.2.5.3 Reunião de Planejamento do Produto**

Esta cerimônia é utilizada para apresentar o projeto ao máximo de desenvolvedores disponíveis. Recomendável que esta cerimônia seja feita via webconferência, com registro em áudio e vídeo, publicado para todo o Time Soberano.

**Seus objetivos são:**

- Avaliar o projeto;
- Alinhar a equipe ao Objetivo do Produto;
- Definir a melhor estratégia de execução do trabalho;
- Avaliar e atualizar a Priorização do Backlog do Produto;
- Atualizar o artefato Plano de Entregas do Produto;
- Definir os desenvolvedores que inicialmente integrarão a Equipe de Projeto, de acordo com as Regras do Jogo.

**5.2.5.4 Reunião de Entrega do Produto**

Nesta cerimônia é feita a entrega dos incrementos do produto que possibilitam a demonstração de algum valor aos stakeholders, chamados de incremento utilizável.

Seu principal objetivo é validar as entregas e retornar o feedback, para manter uma melhoria contínua do produto e reavaliar a priorização do Backlog do Produto.

Na última vez que a Entrega do Produto for realizada, para a entrega final do produto, recomenda-se que a reunião seja aberta a convidados.

**5.2.6 Atividades**

Todas as atividades são executadas por um indivíduo remoto, sendo irrelevante sua localização física, uma vez que todas as ferramentas necessárias para as

atividades se encontram acessíveis via web.

#### 5.2.6.1 Refinar Backlog de Produto

A atividade de Refinamento do Backlog do Produto é o pilar do fluxo iterativo que ocorre entre os Itens do Backlog, ela é realizada pelo Dono do Produto, que fica constantemente buscando informações com todos os Stakeholders que possam influenciar a melhoria do produto.

Ela recebe seu primeiro Feedback da cerimônia de Planejamento do Produto, durante toda a execução do desenvolvimento podem surgir novos fatores que levam o Dono do Produto a refinar o item novamente, entretanto sempre que surge um novo Incremento Utilizável é feita uma cerimônia de Entrega do Produto para demonstrar o novo incremento ao Demandante, assim receber um novo Feedback.

Esse ciclo continua até a entrega final do produto.

#### 5.2.6.2 Desenvolver Itens Priorizados

Esta atividade é o centro de todo o trabalho, onde acontece a codificação dos desejos descritos nos Itens de Backlog do Produto. Os Desenvolvedores designados a participar do projeto escolhem dentre os Itens Priorizados e vão executando item a item de forma autônoma.

Nesta atividade recomenda-se a utilização de técnicas de trabalho em duplas para incentivar a transferência de conhecimento e manter o equilíbrio de conhecimento do Time Soberano, técnicas úteis quando um novato precisa aprender a forma de trabalho do time ou uma tarefa complexa que demanda mais de uma pessoa para explorar as melhores soluções.

#### 5.2.6.3 Revisar Itens Codificados

O Líder Técnico fica constantemente ativo ao revisar todas as atividades relacionadas ao projeto, nesta atividade ele revisa os itens codificados a fim de garantir que eles estão dentro dos padrões estabelecidos e estão passando nos testes necessários para integração do código ao repositório principal.

Nesta atividade pode ser identificado um Incremento Utilizável, então é preparada uma demonstração para que o Dono do Produto apresente ao

Demandante.

### **5.2.7 Modelo do processo**

Este modelo de processo de software é formado por um ciclo de desenvolvimento, onde efetivamente é construído o software, é uma etapa de Criação de Produto, que determina o que deve ser feito e como deve ser feito.

O ciclo é totalmente voltado ao Backlog de Produto Priorizado, uma vez que ele inicia através da priorização dos itens e se conclui através da entrega dos itens como um incremento utilizável.

#### **Criação de Produto**

- Idealização do Produto (Compreender e Definir o Problema)
- Planejamento do Produto (Análise e Proposição de Soluções)
- Execução do primeiro Ciclo de Desenvolvimento de Produto

#### **Ciclo de Desenvolvimento de Produto**

- Backlog de Produto Priorizado (Lista de tarefas priorizadas e atualizadas de acordo com Feedback dos stakeholders);
- Desenvolvimento de Itens (Codificação de nova versão com alterações propostas);
- Revisão de Itens (Validação, Integração e Implantação do Código, Disponibilização de Incremento Utilizável);
- Entrega de Produto (Feedback dos stakeholders com descrição das mudanças propostas);
- Refinamento do Backlog de Produto (Atualização da lista de tarefas).

### **5.2.8 Fluxo de trabalho**

Uma vez que essa metodologia busca tratar todas as informações por meios assíncronos de comunicação, todo o modelo de fluxo de trabalho teve que ser pensado de forma a transmitir a informação via ferramentas web, devido a isso foram criadas as seguintes representações visuais:



**Sistema de Informação:** Representa a nuvem de informações que mantém todos os artefatos acessíveis via web;

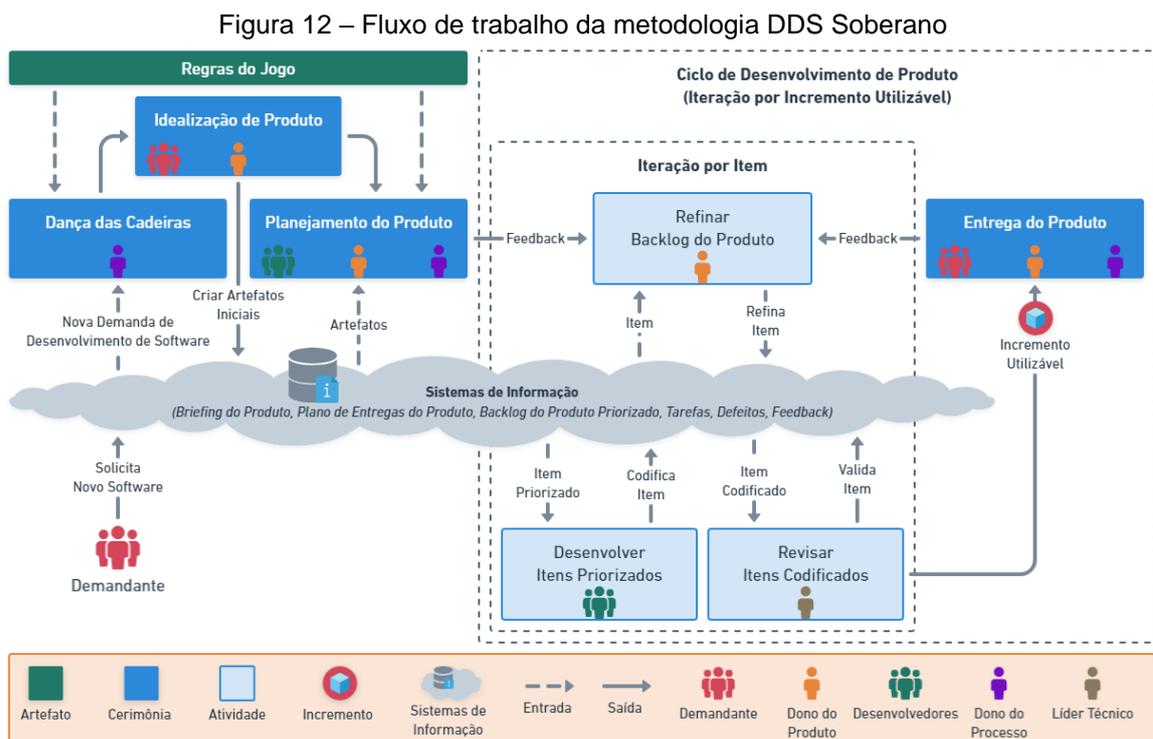


**Seta “Entrada”:** Representa a entrada de informações nas cerimônias ou atividades;



**Seta “Saída”:** Representa a saída de informações das cerimônias ou atividades.

O fluxo de trabalho apresentado na Figura 12, tenta representar da forma mais simples possível todas as relações entre Cerimônias, Atividades, Artefatos e Papéis, que constituem o modelo de processo definido.



Fonte: O autor (2022).

O fluxo é iniciado com a solicitação de um novo software por um demandante, promovendo a criação de um projeto que é designado a dois desenvolvedores, membros do Time Soberano, que devem ocupar os papéis de Dono de Produto e Líder Técnico do Time do Projeto.

Esse Time do Projeto, ainda sem desenvolvedores, tem o papel de Compreender e Definir o Problema especificado pelo demandante, através de uma Reunião de

Idealização do Produto, também deve construir os artefatos iniciais e preparar tudo que for necessário para o projeto.

O Time de Projeto passa a ter desenvolvedores designados a partir da Reunião de Planejamento do Produto, onde também serão refinados os artefatos básicos do produto.

Após a conclusão dos artefatos iniciais é realizada a construção do primeiro incremento utilizável, através do Ciclo de Desenvolvimento de Produto. A entrega deste primeiro incremento marca o fim da etapa de criação do produto, assim o Feedback do demandante, determina se o projeto definido está correto ou não.

Caso o demandante aprove, o Ciclo de Desenvolvimento de Produto se repete de acordo com o Plano do Produto até o fim do ciclo de vida do software.

### 5.2.9 Fluxo de papéis

No DDS Soberano as designações de papéis ocorrem periodicamente, os principais gatilhos para a execução dessas designações são:

**O surgimento de uma nova demanda:** Promove a designação do Dono do produto e do Líder Técnico, que vão formar o Time de Projeto da nova demanda;

**A aprovação do Plano de Entregas do Produto:** Promove a designação dos Desenvolvedores, que farão parte do Time de Projeto inicialmente;

Quando a quantidade de sugestões no Backlog de Mudanças atingir um nível estipulado nas Regras do Jogo: Promove a designação de um Dono de Mudanças.

Observa-se que a qualquer momento, qualquer stakeholders, pode solicitar ao Dono do Processo, o aumento ou diminuição da quantidade de desenvolvedores designados para os Times dos Projetos.

### 5.2.10 Diretrizes de comunicação

A comunicação é a base para o trabalho em equipe, sua qualidade influencia diretamente a qualidade do projeto, processo, produto, enfim de todo o trabalho em equipe. Considerado o fator chave das metodologias ágeis, a comunicação direta também é o principal desafio do Desenvolvimento Distribuído de Software, ao dizer isso pode parecer ilógico juntar as duas metodologias em um único processo, pois é isso que muitos pesquisadores e empresas têm tentado fazer, visto que muitos

benefícios podem ser conquistados com a união delas.

Sendo uma tendência juntar essas duas metodologias, o DDS Soberano busca utilizar as formas de comunicação que obtiveram sucesso nos casos encontrados nesta pesquisa para conseguir unir o modelo ágil ao modelo distribuído de desenvolvimento de software. Para tanto a primeira diretriz quanto a comunicação é:

### **1. Utilizar comunicação assíncrona sempre que possível**

A comunicação assíncrona possibilita uma troca de informações com o tempo mais rápido possível sem a interferência no trabalho do receptor da mensagem. Ela também possibilita o registro da mensagem como uma base de conhecimento para problemas futuros semelhantes.

Algumas pessoas podem se adaptar mais rápido que outras a um modelo assíncrono, à primeira vista pode parecer que a comunicação deixou de existir quando estamos remotos e toda informação que recebemos, está escrita ou desenhada, sem um rosto para atribuir a uma fala. Tendo isso em vista, a segunda diretriz diz:

### **2. Toda comunicação síncrona deve utilizar áudio e vídeo**

Exibir expressões faciais, mesmo que por meios virtuais pode facilitar a compreensão das informações mais rapidamente, além de fomentar a socialização entre os indivíduos, a socialização dentre muitos outros fatores pode ser uma grande aliada da comunicação, facilitando a solidariedade e o espírito de equipe.

Uma vez que a maior dificuldade da comunicação assíncrona é garantir que a informação como um emissor descreve é interpretada corretamente por um receptor, a padronização da forma de comunicação se torna nossa maior aliada, assim a terceira diretriz diz:

### **3. Toda comunicação prevista deve possuir um modelo a seguir**

Uma vez que uma forma de comunicação se torna constante, rotineira, deve ser pensado uma forma adequada de se transmitir, assim criando um modelo a ser utilizado toda vez que for realizar esse tipo de comunicação, tal como é feito com o modelo de tarefas “História de Usuário”, ele se tornou um modelo tão bem aceito que muitas metodologias o adotam para facilitar a comunicação do time de desenvolvimento.

Entretanto um fator relevante e muitas vezes ignorado pelas organizações é a orientação do objetivo de comunicação de cada canal criado, assim as pessoas passam a utilizar os canais de forma incorreta o que gera grandes transtornos, tanto para quem espera encontrar uma informação, quanto para quem precisa gerenciar

esses meios. Tendo em vista manter uma organização dos meios de comunicação e determinar uma estrutura que se adeque a necessidade de cada organização a quarta diretriz diz:

#### **4. Criar um Plano de Comunicação Interna aderente às necessidades encontradas na organização**

Um Plano de Comunicação Interna não pode ser muito grande ou muito engessado, ele deve ser simples, direto e tão fácil de modificar quanto a tecnologia evolui. Ele tem como objetivo garantir que a informação chegue da forma correta para as pessoas corretas. Para tanto é necessário que ele tenha a definição de qual tipo de informação deve circular em cada canal, quais meios de comunicação devem ser utilizados e principalmente quando devem ser utilizados.

Uma grande dificuldade em se definir um Plano de Comunicação Interna adequado se dá referente a quantidade de ferramentas e possibilidades de comunicação existentes, assim a **quinta diretriz** define:

#### **5. Comunicação interna deve ser centralizada em uma única ferramenta**

A total centralização da comunicação interna em apenas uma ferramenta, que possibilite a comunicação síncrona e assíncrona, também possibilite o registro das informações como um mecanismo de banco de conhecimento promove uma confiança e sensação de organização da informação, o que influencia as pessoas a seguirem as normativas de comunicação, uma vez que visualizam claramente o padrão definido.

Observa-se que para isso ocorrer é necessário eliminar a troca de informações por qualquer outra via, não utilizar e-mails, chats pessoais ou comunicação face a face para tratar assuntos de interesse de um grupo.

##### **5.2.11 Coordenação**

Uma vez que todo processo de comunicação, transferência de informações e conhecimento, passam a ser totalmente virtuais, a localização física dos indivíduos se torna irrelevante, todavia o Esforço de Alinhamento dos stakeholders tem um aumento considerável.

Quanto a organização de equipes, em outras metodologias é orientado a divisão dos desenvolvedores em equipes pequenas, de 3 a 9 integrantes, assim metodologias que visam escalar Scrum para projetos maiores agrupam várias equipes pequenas e tentam organizar o Esforço de Alinhamento dessas equipes com outra reunião entre

os líderes das equipes e o Dono do Produto.

Nesta metodologia, não existe a divisão dos desenvolvedores, em equipes pequenas, todos são unidos em um único time nomeado "Time Soberano". Eles se auto-organizam para atender as demandas de todos os projetos, sendo levado em consideração para isso a priorização dos itens de backlog de cada produto.

O Dono do Produto e o Líder Técnico têm todo o entendimento do projeto e assim a responsabilidade de guiar o entendimento dos Desenvolvedores que assumirem tarefas de seus projetos. Assim como os Desenvolvedores escolhem as tarefas que devem executar de acordo com a ordem de prioridade no Backlog de Produto, eles devem atender aos Desenvolvedores de acordo com a ordem de solicitações no canal de comunicação interna. A comunicação assíncrona funciona como um "backlog dos impedimentos de execução" para atendimento rápido aos desenvolvedores, entretanto impedimentos que impossibilitem a execução da tarefa em menos de um dia, devem ser registrados no sistema de gerenciamento de tarefas, ou seja, no item do Backlog de Produto.

Todas as tratativas de conflitos e dependências são tratadas individualmente e organizadas para que sejam solucionadas o mais rápido possível.

### 5.3 RECOMENDAÇÕES

Mude o sistema, para possibilitar a mudança dos hábitos, apenas assim será possível alcançar uma mudança da cultura organizacional e do mindset organizacional.

Segundo Gupta, Venkatachalapathy, Jeberla (2019), a transição para um modelo distribuído deve ocorrer gradualmente, por isso é importante que seja definida uma ordem de implantação de novas práticas para que o "Modelo Híbrido" seja consistente enquanto o processo de migração ocorrer.

Comece pequeno, adote uma prática, depois outra, pensando na melhor forma de não perder o desempenho e a confiança do Time Soberano.

Apesar de não haver previsão de estimativas, é possível estimar os projetos de acordo com indicadores de capacidade de desempenho de projetos anteriores, utilizando indicadores.

A migração do modelo centralizado para o modelo distribuído requer que as práticas de organização e estratégia de ações viabilizem o entendimento de todos os

colaboradores do peso e a importância que o processo de comunicação tem para o sucesso da organização.

#### 5.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Foram levantadas as práticas de desenvolvimento distribuído de software da literatura através de uma Revisão Sistemática, alinhadas às práticas já existentes em demais metodologias e frameworks, foi apresentada uma proposta de Metodologia de DDS com aplicabilidade ao contexto dos IFs.

Esta metodologia foi totalmente construída para o contexto das Instituições Federais de Educação do Brasil (IFs), entretanto existe a possibilidade de aplicações em contextos similares, com as devidas adequações. Por se tratar de uma metodologia de equipes virtuais, é passível de implementação em contextos que tenham uma capacidade de utilização de todas as ferramentas web necessárias.

Seu ponto forte, é também sua principal dificuldade de aplicação em outros contextos, uma vez que ela requer uma habilidade de gerenciar as “Regras do Jogo” de forma a prever dificuldades nas dependências causadas pela adaptação dinâmica que ela provê, principalmente em relação a gerência de tempo dinâmica.

Outra complexidade vem do trabalho de alteração do mindset organizacional para tornar os desenvolvedores, uma equipe única, multidisciplinar, auto-organizáveis e que trabalhe de forma orgânica.

## 6 AVALIAÇÃO DA PROPOSTA

Este capítulo tem a finalidade de apresentar as informações referentes à avaliação da proposta de metodologia de DDS para os IFs, realizada através do método de avaliação de Grupo Focal.

Segundo Kontio, Bragge e Lehtola (2004), a técnica de grupo focal consiste em discussões cuidadosamente planejadas, projetadas para obter as percepções dos membros do grupo sobre uma área de interesse definida. Geralmente são selecionados entre 3 e 12 participantes especialistas na área e a discussão é guiada e facilitada por um moderador, que segue um roteiro predefinido para que a discussão se mantenha focada.

Levando em consideração que Kontio, Bragge e Lehtola (2004) descreve detalhadamente a técnica com a visão voltada para a área de Engenharia de Software, os passos descritos por ele foram utilizados como base para a criação do roteiro e execução do Grupo Focal.

O grupo focal ocorreu por meio de webconferência, no dia 07/07/2022 às 15h (Horário de Brasília), contou com a participação de 8 profissionais especialistas em Desenvolvimento de Software que trabalham atualmente nos IFs.

Seu principal objetivo foi o de revelar as percepções dos participantes sobre a metodologia proposta e promover discussões acerca das perguntas norteadoras, como forma de identificar se a proposta faz sentido e se é aplicável aos IFs. Além de prover um espaço para proposição de ideias que possam refinar a proposta apresentada.

De acordo com o roteiro, descrito no 0, foram executadas as três etapas, a divulgação e seleção dos participantes, a apresentação da metodologia e posteriormente a indução da discussão sobre as questões colocadas como norteadoras.

### 6.1 CONDUÇÃO

**Primeira Etapa:** Seis dias antes da data estipulada para a realização do Grupo Focal foram enviados convites por diversas plataformas (WhatsApp, E-mail, Discord) para todos Desenvolvedores de Software dos IFs do Brasil, os participantes puderam entrar em um grupo de WhatsApp para acompanhar as informações sobre o Grupo

Focal e puderam confirmar a participação marcando o horário disponível em uma ferramenta de agendamento de eventos por meio de votação, chamada Doodle, assim foi possível definir o melhor horário para a realização da conversa. Quando confirmado o horário, foi novamente comunicado em todas as plataformas o horário definido, a possibilidade de confirmar presença via Grupo de WhatsApp e o link de acesso da sala de webconferência.

**Segunda Etapa:** Após a entrada de todos os participantes, iniciou-se os trabalhos, o qual durou cerca de 40 minutos. Foi apresentado o objetivo do grupo focal, o fato de que a conversa está sendo gravada para posterior consulta e transcrição, enfim, foi apresentada a proposta.

**Terceira Etapa:** A maioria dos participantes expressaram ter entendido e gostado da metodologia proposta, várias dúvidas surgiram durante as falas, o que já era esperado, devido à complexidade do modelo, enfim foram identificadas as respostas de todas as perguntas, entretanto alguns dos participantes continuaram na conversa expondo algumas de suas ideias de melhorias ou visões futuras. Todas as informações relevantes desta etapa foram transcritas e organizadas na seção a seguir.

## 6.2 RESULTADOS

Os Institutos Federais representados neste grupo focal, dentre os 8 participantes, foram, IFMT, IFPE, IFRO, IFRR e IFTM.

O modelo proposto faz sentido?

Foi unânime a compreensão da proposta como sendo válida, necessária e aderente às necessidades encontradas nos IFs. Inclusive o problema que motivou esta pesquisa “O desenvolvimento paralelo de projetos de mesmo objetivo”, descrito na seção 1.2, foi citado por um dos participantes (P8) que disse “Existem ferramentas e funcionalidades que foram desenvolvidas pelo campus, porém sem aplicação de modelo ou metodologia que estão órfãs pois o servidor que desenvolveu entregou e não participa mais, nem de suporte, nem de desenvolvimento, alguns até em linguagens não dominadas pela equipe oficial de desenvolvimento”.

Outro participante (P3), alegou que “Com toda certeza, acredito que seria muito válido para o IFPE, principalmente o ponto de evitar o desenvolvimento de várias ferramentas iguais, uma em cada campus isolada”.

O modelo proposto atende às necessidades dos Institutos Federais?

Quando questionados sobre a possibilidade de aplicação da metodologia nos IFs, todos concordaram que atende e seria aplicável, entretanto dois participantes levantaram a questão de que o quantitativo de pessoas em alguns institutos poderia impossibilitar a adesão.

Considera que seja possível aplicar a metodologia em sua instituição?

Inevitavelmente ao se perguntar a viabilidade do modelo a realidade dos IFs vários dos participantes levaram em consideração seus institutos e quase todos assinalaram que apesar de a metodologia ser conceitualmente aplicável, seu instituto não atenderia aos requisitos no momento. Apenas um dos participantes (P6) afirmou que sim e não apresentou nenhum fator que constasse dentre os requisitos previstos pelo modelo.

Este participante (P6) foi o primeiro a relatar fator político como um limitante para a adoção da proposta, logo após sua afirmação, outros três participantes concordaram com o fato. Eu, no papel de moderador, os informei que a metodologia não trata as questões políticas, apenas as relações de coordenação das atividades entre os indivíduos e suas implicações.

Ideia de expansão da metodologia como projeto de implantação em toda a Rede Federal

O mesmo participante (P6) determinou ainda que apesar de o fator financeiro ser um limitante para a adoção da proposta em nível institucional, ele seria um impulsionador caso fosse estruturada uma proposta de adesão de toda a Rede Federal com apoio do CONIF. Essa afirmativa foi corroborada por vários participantes, sendo um fato desejado por muitos deles, foi levado adiante com uma ideia inovadora da possibilidade de implantação desta metodologia em âmbito de toda a Rede Federal, com o apoio de uma fundação.

O participante (P4) afirmou que esse modelo seria muito interessante e que a fundação poderia romper os problemas relacionados à política inerente às instituições participantes da Rede Federal de Educação, sendo totalmente necessária para o efetivo funcionamento de uma dinâmica de trabalho entre IFs.

O participante (P8) ressaltou ainda que o modelo proposto não só atenderia a um trabalho conjunto de instituições, de toda a Rede Federal, como seria muito mais indicado, devido ao quantitativo de pessoal que cada IF possui em comparação com o quantitativo que seria disponibilizado em uma cooperação entre instituições.

Então os participantes (P4, P6, P8), lançaram como sugestão de um trabalho futuro a possibilidade de levar a proposta do modelo para o âmbito de toda a Rede Federal, com o apoio de uma Fundação, para o amparo da implantação desta metodologia entre os Institutos Federais.

Algum fator impediria a aplicação desta metodologia em seu instituto?

Como já citado na discussão da pergunta anterior o grupo ressaltou inúmeras vezes a dificuldade política, inerente aos IFs, o participante (P4) afirmou que à lei de formação dos IFs que determina uma grande autonomia para cada campus, acaba por dificultar a padronização de todo o instituto uma vez que alguns gestores podem decidir não cooperar com um padrão definido institucionalmente. Afirmou ainda que existe naturalmente uma certa disputa em relação a priorização das demandas de cada campus. Outro fator relacionado à política, citado por três participantes (P3, P4 e P7), é o apoio da alta gestão em mudanças estruturais como a proposta por essa metodologia.

O participante (P5) relaciona o fator político aos fatores socioculturais, evidenciando o fato de a atual organização propiciar o poder de priorização das demandas para a alta gestão sem um controle adequado, o que gera o problema de priorização sem consideração de fatores técnicos de desenvolvimento. Segundo o participante é necessário que exista um papel e uma cultura que permita aos desenvolvedores negar algumas prioridades que podem desarranjar todo o processo de desenvolvimento.

O participante (P2) corrobora com a relevância do fator sociocultural e enfatiza a qualidade da metodologia proposta em colocar a comunicação como base e tentar tratar detalhes fundamentais para a adoção de um modelo de processo distribuído. Ele ressalta a problemática relacionada a comunicação falha ou inexistente quando o projeto fica dividido em indivíduos remotos e co-localizados, uma vez que conversas face a face podem ocorrer sem o devido registro em ferramenta web, sendo um ponto forte da metodologia trabalhar com todos os indivíduos totalmente virtual e toda a comunicação ser via ferramenta web. Ele também indica a possível problemática de acúmulo de papéis em casos que os institutos possuam pouco quantitativo de desenvolvedores e sugere então a adição do requisito de quantitativo mínimo de pessoas para a adoção da metodologia.

### 6.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo descreveu a avaliação da metodologia de desenvolvimento distribuído de software proposta neste trabalho, que foi realizada através do método de grupo focal. Descreveu também o conceito do grupo focal, como foi sua condução e os resultados obtidos. O roteiro utilizado na condução do grupo focal pode ser encontrado no 0.

As principais considerações sobre a relevância do estudo e a validação da proposta podem ser tomadas devido ao reconhecimento dos membros do grupo focal de que a proposta é eficaz na resolução do principal problema que motivou este trabalho, o desenvolvimento paralelo de projetos de mesmo objetivo, que contou ainda com diversos depoimentos que relatam a vivência do problema pelos participantes.

O grupo focal além de confirmar que a proposta é aplicável aos Institutos Federais, afirmou que é extremamente necessário a adoção de uma metodologia que possibilita unificar o desenvolvimento de software entre reitoria e campi de forma organizada e padronizada.

Uma sugestão que trouxe uma nova perspectiva para trabalhos futuros foi a possibilidade de adoção da metodologia proposta no âmbito de toda a Rede Federal, uma vez que ela representa um dinamismo e adaptabilidade que é característico da organização de grande autonomia que os Institutos Federais possuem.

Com as afirmações apontadas entende-se que o Grupo Focal determina a metodologia como válida e como necessária para toda a Rede Federal de Educação, tanto dentro dos IFs quanto entre os IFs.

## 7 CONSIDERAÇÕES FINAIS

O principal objetivo desta pesquisa foi desenvolver uma metodologia capaz de atribuir os conceitos ágeis a um ambiente distribuído de desenvolvimento de software, que adequasse as necessidades encontradas na estrutura inerente aos IFs. A relevância desta pesquisa foi justificada pela falta de uma metodologia adequada ao cenário proposto.

Este capítulo apresenta as principais contribuições, limitações encontradas e recomendações para trabalhos futuros.

### 7.1 CONTRIBUIÇÕES DA PESQUISA

A principal contribuição do trabalho foi a elaboração da Metodologia de Desenvolvimento Distribuído de Software Soberano. Tem-se a expectativa de que ela propicie um ambiente de desenvolvimento de software ágil e distribuído de qualidade para os IFs.

Para a comunidade científica da área de desenvolvimento de software, este trabalho permite que novas pesquisas sejam desenvolvidas a partir dos resultados e das possíveis lacunas na área de investigação obtidas.

Para os gestores de TI dos IFs, o trabalho contribui com informações, argumentos e lições aprendidas para justificar uma futura adoção de metodologia de desenvolvimento de software.

### 7.2 LIMITAÇÕES DO ESTUDO

Apesar de mudar a metodologia inicialmente proposta para reduzir as limitações, algumas continuaram a existir. Entre elas, podemos destacar as seguintes:

A principal limitação deste estudo é a pouca informação sobre IFs que utilizam o modelo distribuído de desenvolvimento de software atualmente, sendo que foi identificado apenas 1 de 41 IFs que apresenta dados relevantes para este estudo;

Mesmo com extensa pesquisa de campo, não foi possível obter dados dos cenários de todos os IFs;

A falta de documentação dos projetos nos IFs, apesar de ser uma prática adotada por muitos, senão todos os IFs, a área de sistemas ainda não tem muita

documentação de projetos disponíveis;

Falta de avaliação em um contexto real, impossibilitando uma análise mais ampla sobre a efetividade da metodologia, fazendo com que os especialistas avaliassem a metodologia apenas com base em suas experiências.

Houve um baixo nível de participação na avaliação feita pelo Grupo Focal, seria necessária uma nova avaliação com maior participação, o que não foi possível realizar.

Embora com algumas limitações, podemos considerar os resultados satisfatórios.

### 7.3 TRABALHOS FUTUROS

A pesquisa alcançou todos os objetivos almejados no planejamento, entretanto existem muitas oportunidades de trabalhos relacionadas ao Desenvolvimento Distribuído de Software que podem derivar deste estudo, tais como:

- A experimentação e análise do comportamento da Metodologia DDS-Soberano em utilização nos IFs;
- O aperfeiçoamento da Metodologia DDS-Soberano através de indicadores de desempenho;
- Desenvolvimento de uma plataforma em nuvem que suporte a Metodologia DDS-Soberano;
- Construção de uma plataforma de divulgação e treinamento da Metodologia DDS-Soberano;
- Construção de uma proposta de adoção da Metodologia DDS-Soberano por toda a Rede Federal de Educação, através do apoio de uma fundação.

### 7.4 CONCLUSÕES

A primeira etapa, Pesquisa da Literatura, em especial a técnica de Revisão Sistemática da Literatura, reuniu muitas informações literárias a respeito das metodologias utilizadas atualmente e práticas relacionadas à área.

As práticas coletadas foram classificadas e organizadas de acordo com seus impactos. Mas a principal contribuição da RSL sem dúvida foram os dois artigos que

foram considerados as principais influências para a construção da Metodologia DDS Soberano.

A segunda etapa, Pesquisa de Campo, contou com três métodos para coletar as informações necessárias para entender o cenário atual do Desenvolvimento Distribuído de Software nos Institutos Federais.

Seus resultados provaram que já existem iniciativas de desenvolvimento distribuído de software dentre os IFs e foi tido como referência o desenvolvimento de software apresentado pelo IFSP por se tratar de uma equipe dinâmica e efetiva.

Iniciativas entre institutos federais como as apresentadas pelos projetos SIG Colaborativo e SUAP Colaborativo, foram provas que a própria Rede Federal tem tentado elaborar iniciativas do tipo.

A terceira etapa, Elaboração da Proposta, só se tornou possível após a absorção de todas as informações levantadas em todas as etapas anteriores e a concepção de uma proposta que alinhasse as práticas encontradas às dificuldades enfrentadas pelos Institutos Federais.

A realidade encontrada em uma Instituição Federal de Educação é tão complexa e dinâmica que a única alternativa que fez sentido para atacar o problema foi a construção de uma proposta totalmente baseada em ambiente virtual com comunicação totalmente centralizada em plataformas web. Assim se concluiu a concepção da Metodologia DDS Soberano.

Com o objetivo de avaliar a capacidade de aderência da metodologia às necessidades dos Institutos Federais, foi realizado ainda um Grupo Focal que apontou indícios de que a proposta atende a estrutura interna comum aos Institutos Federais, mas que devido a quantidade de desenvolvedores disponíveis em alguns dos IFs talvez não seria viável, sendo mais provável a utilização no âmbito da Rede Federal, entre os IFs, do que internamente.

Diante do exposto, acreditamos que a pesquisa atendeu seu objetivo e a metodologia também atendeu a necessidade para qual foi idealizada, proporcionando uma nova forma de adequação do processo de desenvolvimento de software de forma geograficamente distribuída, melhorando a capacidade de coordenação de projetos remotos no âmbito dos Institutos Federais de Educação do Brasil.

## REFERÊNCIAS

Agile Manifesto. **Manifesto for Agile Software Development**. Disponível em: <http://agilemanifesto.org>. Acesso em: 10 jul. 2022.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27001**: tecnologia da informação: técnicas de segurança: sistemas de gestão da segurança da informação: requisitos. Rio de Janeiro, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27002**: tecnologia da informação: técnicas de segurança: código de prática para controles de segurança da informação. Rio de Janeiro, 2013.

AUDY, J.; PRIKLADNICKI, R. **Desenvolvimento Distribuído de Software**: Desenvolvimento de software com equipes distribuídas. Rio de Janeiro: Elsevier, 2007.

BOULART, I.; LANZA, M. **Identidade das Pessoas e das Organizações**. Revista de Administração e Diálogo, v. 9, n. 1, p. 1–18, 2007.

BRASIL. Lei nº 11.892, de 29 de dezembro de 2008. **Institui a Rede Federal de Educação Profissional, Científica e Tecnológica, cria os Institutos Federais de Educação, Ciência e Tecnologia, e dá outras providências, Brasília, DF, dez 2008**. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_Ato2007-2010/2008/Lei/L11892.htm](http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Lei/L11892.htm). Acesso em: 03 out. 2019.

BURKE, P.; STETS, J. **Identity Theory**. New York: Oxford, 2009.

Conif. 2019. **Conselho Nacional das Instituições da Rede Federal de Educação Profissional, Científica e Tecnológica (Conif)**. Disponível em: [http://portal.conif.org.br/br/?option=com\\_sppagebuilder&view=page&id=8&Itemid=577](http://portal.conif.org.br/br/?option=com_sppagebuilder&view=page&id=8&Itemid=577). Acesso em: 03 out. 2019.

COSTA, J. **Mapeamento Sistemático de Gerenciamento de Projetos no Desenvolvimento Distribuído de Software**. Dissertação de Mestrado – Centro de Informática, Universidade Federal de Pernambuco, Recife - PE, 2014.

DWECK, C. **Mindset**: A nova psicologia do sucesso. Rio de Janeiro: SCHWARCZ S.A., 2017.

ESBENSEN, M.; BJØRN, P. **Routine and standardization in global software development**. Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, p. 12–23, 2014. Disponível em: <https://doi.org/10.1145/2660398.2660413>. Acesso em: 12 jul. 2022.

GIUFFRIDA, R.; DITTRICH, Y. **How social software supports cooperative practices in a globally distributed software project**. 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2014 -

Proceedings, p. 24–31, 2014. Disponível em:  
<https://doi.org/10.1145/2593702.2593716>. Acesso em: 12 jul. 2022.

GUPTA, R. K.; VENKATACHALAPATHY, M.; JEBERLA, F. K. **Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team**: A Case Study of a Healthcare Organization. Proceedings - 2019 ACM/IEEE 14th International Conference on Global Software Engineering, ICGSE 2019, p. 30–34, 2019. Disponível em: <https://doi.org/10.1109/ICGSE.2019.00020>. Acesso em: 12 jul. 2022.

JHA, M. M.; VILARDELL, R. M. F.; NARAYAN, J. **Scaling agile scrum software development**: Providing agility and quality to platform development by reducing time to market. Proceedings - 11th IEEE International Conference on Global Software Engineering, ICGSE 2016, p. 84–88, 2016. Disponível em:  
<https://doi.org/10.1109/ICGSE.2016.24>. Acesso em: 12 jul. 2022.

KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Joint Technical Report, Software Engineering Group, Keele University, and Empirical Software Eng., Nat'l ICT Australia, 2004.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University, 2007.

KRUGER, J. et al. **Daedalus or icarus? Experiences on follow-the-sun**. Proceedings - 2017 IEEE 12th International Conference on Global Software Engineering, ICGSE 2017, p. 31–35, 2017. Disponível em:  
<https://doi.org/10.1109/ICGSE.2017.4>. Acesso em: 12 jul. 2022.

KUHRMANN, M.; FERNANDEZ, D. M. **Systematic Software Development**: A State of the Practice Report from Germany. Proceedings - 2015 IEEE 10th International Conference on Global Software Engineering, ICGSE 2015, p. 51–60, 2015. Disponível em: <https://doi.org/10.1109/ICGSE.2015.11>. Acesso em: 12 jul. 2022.

LAUKKANEN, E. et al. **Bottom-up Adoption of Continuous Delivery in a Stage-Gate Managed Software Organization**. International Symposium on Empirical Software Engineering and Measurement, v. 08-09-Sept, 2016. Disponível em:  
<https://doi.org/10.1145/2961111.2962608>. Acesso em: 12 jul. 2022.

LOUS, P. et al. **From scrum to agile**: A journey to tackle the challenges of distributed development in an agile team. ACM International Conference Proceeding Series, p. 11–20, 2018. Disponível em: <https://doi.org/10.1145/3202710.3203149>. Acesso em: 12 jul. 2022.

LOUS, P. et al. **Virtual by design**: How a work environment can support agile distributed software development. Proceedings - International Conference on Software Engineering, p. 102–111, 2018. Disponível em:  
<https://doi.org/10.1145/3196369.3196374>. Acesso em: 12 jul. 2022.

MATTHIESEN, S.; BJØRN, P.; PETERSEN, L. M. **“Figure out how to code with the hands of others”**: Recognizing cultural blind spots in global software development. Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW, p. 1107–1119, 2014. Disponível em: <https://doi.org/10.1145/2531602.2531612>. Acesso em: 12 jul. 2022.

MCTIC. 2018. **Estratégia Brasileira para a Transformação Digital (E-Digital)**. Disponível em: <http://www.mctic.gov.br/mctic/export/sites/institucional/estrategiadigital.pdf>. Acesso em: 06 out. 2019.

MEC, Expansão da Rede Federal. 2019. **Expansão da Rede Federal**. Disponível em: <http://portal.mec.gov.br/setec-programas-e-aco-es/expansao-da-rede-federal>. Acesso em: 03 out. 2019.

MEC, Instituições. 2019. **Instituições da Rede Federal**. Disponível em: <http://portal.mec.gov.br/rede-federal-inicial/instituicoes>. Acesso em: 03 out. 2019.

MEPDG. 2019. **Estratégia de Governança Digital (EGD)**. Disponível em: <http://www.planejamento.gov.br/EGD>. Acesso em: 06 out. 2019.

Plataforma Nilo Peçanha. 2019. **Rede Federal de Educação Profissional, Científica e Tecnológica SETEC/MEC**. Disponível em: <http://plataformanilopecanha.mec.gov.br/2019.html>. Acesso: em 03 out. 2019.

PMI. **Guia PMBOK**: Um Guia do Conhecimento em Gerenciamento de Projetos. 6. ed. Pensilvânia: Project Management Institute, Inc., 2017.

PRADO, G.; FERRARI, F. **Avaliação do impacto do desenvolvimento distribuído de software em um projeto adotando o scrum**: Um estudo comparativo. Revista TIS, v. 2, n. 3, 2014.

PRESSMAN, R. **Engenharia de Software**: Uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

PRIKLADNICKI, R. **Desenvolvimento Distribuído de Software e Processos de Desenvolvimento de Software**. Trabalho Individual II, Disciplina de Mestrado – PUCRS, Porto Alegre - RS. 2002.

PRIKLADNICKI, R. **MuNDDoS - Um modelo de referência para desenvolvimento distribuído de software**. Dissertação de Mestrado – PUCRS, Porto Alegre - RS. 2003.

Rede Federal, Expansão. 2019. **Expansão da Rede Federal**. Disponível em: <http://redefederal.mec.gov.br/expansao-da-rede-federal>. Acesso em: 03 out. 2019.

Rede Federal, Instituições. 2019. **Instituições da Rede**. Disponível em: <http://redefederal.mec.gov.br/instituicoes>. Acesso: em 03 out. 2019.

ROCHA, R. et al. **Modelos de Colaboração no Desenvolvimento Distribuído de Software**: uma Revisão Sistemática da Literatura. Artigo – IV Workshop de Desenvolvimento Distribuído de Software, 2010.

ROCHA, R. **Fatores que Afetam Negativamente os Modelos de Desenvolvimento Distribuído de Software**. Dissertação de Mestrado – Universidade Federal de Pernambuco. Recife - PE, 2010.

ROOPA, M. S.; KUMAR, R.; MANI, V. S. **Transitioning from plan-driven to lean in a global software engineering organization**: A practice-centric view. Proceedings - International Conference on Software Engineering, p. 1–5, 2018. Disponível em: <https://doi.org/10.1145/3196369.3196395>. Acesso em: 12 jul. 2022.

SANTOS, I. **Influência dos fatores organizacionais no sucesso de projetos no setor automotivo**. Dissertação de Mestrado – Universidade Estadual Paulista, São Paulo - SP, 2015.

SCHEIN, E. **Organizational Culture and Leadership**. 3. ed. San Francisco: Jossey-Bass. 2004.

SIQUEIRA, F. **O Desenvolvimento Distribuído de Software Características e Recomendações para a Gerência de Projetos**. Dissertação de Mestrado – Escola Politécnica, Universidade de São Paulo, São Paulo - SP, 2005.

SOARES, F. et al. **Adoção de Scrum em uma Fábrica de Desenvolvimento Distribuído de Software**. Centro de Informática, Universidade Federal de Pernambuco, Recife - PE, 2007.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STRAY, V. **Planned and unplanned meetings in large-scale projects**. ACM International Conference Proceeding Series, v. Part F1477, 2018. Disponível em: <https://doi.org/10.1145/3234152.3234178>. Acesso em: 12 jul. 2022.

## GLOSSÁRIO

**Processo:** Conjunto sequencial e particular de ações com objetivo comum.

**Prática:** Uma descrição detalhada da execução de um processo, construída a partir de conhecimentos experienciados, ou seja, a partir da forma como as pessoas executam uma ação habitualmente.

**Abordagem:** Modo de tratar determinada situação, maneira de agir ou estratégia adotada para a realização de uma prática.

**Método:** Conjunto de procedimentos, regras e operações previamente fixados que permitem chegar à determinada meta, fim ou conhecimento. Mostra como se deve conduzir o pensamento, assim como a ação, para que determinado objetivo seja atingido com eficiência. Além de trazer um conjunto sequencial e particular de ações com objetivo comum, determina uma forma de execução deles. Isso produz um resultado de processo mais específico, sendo assim um processo pode possuir muitos métodos baseados nele.

**Técnica:** Um método, que além de trazer um conjunto sequencial e particular de ações com objetivo comum e determinar uma forma específica de execução deles, possui também comprovações de um determinado resultado esperado. Em uma técnica existe uma forma pré-estabelecida de se chegar ao resultado esperado. É usada para atingir uma mesma meta, repetidamente, com a menor perda de consistência ou qualidade do produto.

**Modelo:** Um padrão utilizável para a execução de uma ação.

**Modelo Conceitual:** Um conjunto de suposições baseadas no mundo real que indicarão as regras de negócio de um sistema.

**Framework:** Um conjunto de modelos para criação de uma metodologia adequada à realidade de uma organização.

**Metodologia:** Um sistema de métodos, princípios e regras que regulam uma determinada disciplina em ciências ou artes. Geralmente, métodos, processos e técnicas constituem uma metodologia, as quais são apoiadas pela utilização de ferramentas.

**Filosofia:** Um modo de problematizar o mundo de maneira crítica e conceitual. Ela é utilizada pelo ser humano para desbravar os confins da racionalidade e do conhecimento.

## APÊNDICE A – REVISÃO SISTEMÁTICA – PROTOCOLO

Esta Revisão Sistemática da Literatura (RSL) tem como objetivo analisar e sintetizar a literatura existente sobre Desenvolvimento Distribuído de Software, com objetivo de relacionar as práticas de desenvolvimento de software adotadas nos projetos da literatura.

Foi observada a necessidade de se realizar uma RSL devido ao fato de não encontrar uma relação de práticas utilizadas no DDS mesmo existindo uma grande quantidade de estudos que apresentam casos de DDS.

Este protocolo promove a documentação íntegra da fase de planejamento executada com o intuito de se preparar de forma adequada para as fases posteriores de uma RSL e avaliar a real necessidade da realização dela perante toda a equipe. Ele descreve as delimitações da pesquisa, tais como, metodologia, objetivo, questões, termos de busca e critérios de seleção dos estudos definidos.

A equipe responsável pela RSL é apresentada no Tabela 1.

Tabela 1 – Equipe responsável pela RSL

Nome	Papel	Afiliação
Alexandre Marcos Lins de Vasconcelos	Orientador	UFPE
Welington Mothé de Oliveira	Pesquisador	UFPE

Fonte: O autor (2022).

### QUESTÕES DE INVESTIGAÇÕES ESPECIFICAS

Para a definição das questões foi utilizada a técnica PICOC (ver Quadro 9) sugerida por Kitchenham e Charters (2007) que serve como uma linha guia de raciocínio para a delimitação da pergunta chave, facilitando a especificação com foco no resultado esperado.

Quadro 9 – Descrição dos elementos PICOC da pesquisa

<b>População</b>	Estudos que apresentam projetos de Desenvolvimento Distribuído de Software.
<b>Intervenção</b>	Processos.
<b>Contexto</b>	Estudos industriais e acadêmicos.
<b>Resultado</b>	Práticas utilizadas nos processos.

<b>Comparação</b>	Não é aplicável.
-------------------	------------------

Fonte: O autor (2022).

Inicialmente foi proposto encontrar apenas as práticas adotadas no desenvolvimento distribuído de software (Q1 na Tabela 2), mas ao decorrer da pesquisa foi entendido a necessidade de encontrar as metodologias mais utilizadas (Q2 na Tabela 2), com o intuito de relacionar suas práticas, por serem consideradas pelo autor como validadas pelo mercado.

Tabela 2 – Questões de pesquisa

<b>ID</b>	<b>Descrição</b>
<b>Q1</b>	Quais são as práticas adotadas nos processos de desenvolvimento distribuído de software?
<b>Q2</b>	Quais são as metodologias utilizadas nos casos apresentados na literatura?

Fonte: O autor (2022).

## METODOLOGIA DA RSL

Inicialmente os pesquisadores em conjunto executam as Strings de Busca adaptadas em cada uma das bases de dados. A partir da lista de estudos retornados, no mínimo dois pesquisadores leem o título de cada um dos estudos, descartando aqueles que claramente não estejam relacionados à estratégia de busca, ou que não preencham os Critérios de Inclusão, Critérios de Qualidade ou estejam relacionados aos Critérios de Exclusão definidos. Estudos excluídos nesta fase não são armazenados.

A lista de estudos é dividida entre os pesquisadores participantes, de forma que cada estudo seja lido por pelo menos dois pesquisadores. Neste segundo passo, cada pesquisador lê o resumo e as conclusões de cada estudo. A partir desta leitura, são avaliados os critérios de Inclusão, Exclusão e Qualidade novamente. Estudos excluídos nesta fase são armazenados para posterior consulta na planilha de coleta de dados (Anexo I). Como resultado se obtém a lista completa dos estudos.

Cada pesquisador então lê por completo os estudos que lhe foram alocados, avalia a qualidade dos estudos e extrai os dados básicos de caracterização do estudo, bem como os dados específicos relacionados às perguntas de pesquisa, preenchendo a planilha de coleta de dados (Anexo I).

Em uma seção conjunta os pesquisadores apresentam as dúvidas relacionadas à extração de dados ou inclusão/exclusão de estudos. Havendo discrepâncias entre pesquisadores, a decisão final cabe ao pesquisador sênior ou professor orientador.

Após finalizada a extração dos dados, os passos seguintes correspondem à análise, interpretação e documentação dos resultados da RSL.

### ***Bases de dados***

As bases de dados que serão utilizadas nesta RSL, estão listadas na Tabela 3, elas foram escolhidas devido à grande quantidade de estudos relacionados à área de tecnologia de informação contido nelas.

Tabela 3 – Bases de dados

<b>Nome</b>	<b>Acesso</b>
<b>IEEE Xplore Digital Library</b>	<a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a>
<b>ACM Digital Library</b>	<a href="https://dl.acm.org/">https://dl.acm.org/</a>

Fonte: O autor (2022).

### ***Termos de busca***

A partir da análise PICOC e das questões levantadas foi possível relacionar uma lista de possíveis palavras chaves, ao comparar com outros estudos de RSL alguns termos foram acrescentados à lista.

Foi identificada a necessidade de realizar a busca por estudos em inglês devido a diferença de quantidade de publicações em relação ao português nos repositórios utilizados. Os termos foram então traduzidos e associados aos seus sinônimos para uma pesquisa mais eficaz, conforme apresentado no Quadro 10.

Quadro 10 – Termos de busca

<b>Termo</b>	<b>Tradução em inglês</b>	<b>Sinônimos em inglês</b>
Desenvolvimento Distribuído de Software	Distributed Software Development	Global software development, Collaborative software development, Global software engineering, Distributed teams, Geographically distributed software development, Dispersed teams
Melhores Práticas	Best Practices	Lesson, Learned, Success Factor

Fonte: O autor (2022).

### ***Cr terios de sele o***

Os estudos ser o selecionados a partir de uma an lise dos t tulos, resumo e conclus o. De acordo com os cr terios de inclus o definidos na Tabela 4 e os cr terios de exclus o definidos na Tabela 5.

Tabela 4 – Cr terios de inclus o

<b>Identifica�o</b>	<b>Descri�o</b>
CI1	Estudos escritos em ingl�s.
CI2	Estudos recentes (publicados a partir de 2010) que j� possuem aprova�o pela comunidade cient�fica.
CI3	Estudos prim�rios.
CI4	Estudos aplicados na academia ou na ind�stria.
CI5	Estudos publicados e dispon�veis integralmente em bases de dados cient�ficas.
CI6	Estudos que identifiquem as pr�ticas em projetos de DDS.
CI7	Estudos que identifiquem as pr�ticas para redu�o de um desafio encontrado no DDS.
CI8	Possui o termo ou sin�nimos de "DDS" no t�tulo, resumo ou corpo do trabalho.
CI9	Possui o termo ou sin�nimos de "Pr�ticas" no corpo do trabalho.

Fonte: O autor (2022).

Tabela 5 – Cr terios de exclus o

<b>Identifica�o</b>	<b>Descri�o</b>
CE1	Estudos duplicados.
CE2	Estudos incompletos, rascunhos, slides ou resumos.
CE3	Estudos publicados antes de 2010.
CE4	Estudos que n�o est�o dispon�veis gratuitamente para download nos ambientes institucionais do CIn/UFPE ou do IFES.
CE5	Estudos que n�o est�o escritos no idioma ingl�s.
CE6	Estudos que n�o identifiquem pr�ticas no DDS.

Fonte: O autor (2022).

## REFERÊNCIAS

KITCHENHAM, B. **Procedures for Performing Systematic Reviews**. Joint Technical Report, Software Engineering Group, Keele University, and Empirical Software Eng., Nat'l ICT Australia, 2004.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University, 2007.

## APÊNDICE B – REVISÃO SISTEMÁTICA – CONDUÇÃO

Este documento promove a documentação íntegra da condução da RSL que se refere às etapas de Coleta de dados e Análise de dados respectivamente. O objetivo deste documento é apresentar a forma com que foi executada a RSL para uma posterior validação e utilização para a realização de novas RSLs.

### ***Coleta de estudos***

A string foi elaborada baseando-se nos padrões de string dos repositórios escolhidos, os termos em inglês derivados da questão de pesquisa foram concatenados com operadores booleanos “OR” e “AND” formatando dois grupos de pesquisa um para o “desenvolvimento distribuído software” e outro para “práticas”.

Após a construção da primeira string foi realizado o refinamento da string através de um ciclo de tentativas e erros para a escolha dos melhores termos e melhor formatação da string de acordo com as bases de dados escolhidas.

### ***Refinamento de string***

A string foi definida a partir do processo de refinamento de string, que acontece com a verificação do resumo dos artigos encontrados e a inclusão ou exclusão de termos de buscas (Quadro 11).

Quadro 11 – Termos de busca após refinamento de string

Palavra-chave	Tradução em inglês	Sinônimos em inglês
Desenvolvimento Distribuído de Software	Distributed Software Development	Global software development, Collaborative software development, Global software engineering, Globally distributed work, Collaborative software engineering, Distributed development, Distributed teams, Global software teams, Globally distributed development, Globally distributed project, Geographically distributed software development, Dispersed teams, Dispersed Software Development, Dispersed Development, Dispersed Software Engineering, Geographically Dispersed Development Teams
Prática	Practice	Lesson, Learned, Success Factor

Fonte: O autor (2022).

A string resultante (Quadro 12) foi a junção de dois grupos de termos “Desenvolvimento Distribuído de Software” e “Prática”, respectivamente divididos pelo comando “AND”, nesses grupos são adicionados todos os possíveis sinônimos encontrados, utilizando o comando “OR”.

Quadro 12 – String definida para a consulta

String definida
("Distributed software development" OR "Global software development" OR "Collaborative software development" OR "Global software engineering" OR "Globally distributed work" OR "Collaborative software engineering" OR "Distributed development" OR "Distributed teams" OR "Global software teams" OR "Globally distributed development" OR "Globally distributed project" OR "Geographically distributed software development" OR "Dispersed teams" OR "Dispersed Software Development" OR "Dispersed Development" OR "Dispersed Software Engineering" OR "Geographically Dispersed Development Teams") AND ("Practice" OR "Lesson" OR "Learned" OR "Success Factor")

Fonte: O autor (2022).

Após o Refinamento de string é necessário a determinação dos filtros de buscas de acordo com critérios pré-determinados definidos na Tabela 6. Logo iniciou-se efetivamente a Coleta de Estudos utilizando a string definida no

Quadro 12 – e os filtros definidos na Tabela 6.

Tabela 6 – Filtros utilizados nas consultas

Filtros	
Filtro 1	Estudos publicados entre janeiro de 2010 e dezembro de 2019.

Fonte: O autor (2022).

A realização da consulta obteve êxito em todas as bases de dados com a mesma string e filtros definidos trazendo o quantitativo de estudos apresentado na Tabela 7.

Tabela 7 – Quantidade de estudos retornados por base de dados

Base de Dados	Data de consulta	Estudos retornados
ACM Digital Library ( <a href="https://dl.acm.org">https://dl.acm.org</a> )	17/12/2019	144
IEEE Xplore Digital Library ( <a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a> )	17/12/2019	101
<b>Total</b>		<b>245</b>

Fonte: O autor (2022).

Seus resultados são uma lista de artigos de cada base de dados, que devem ser unidos em uma planilha para a verificação de artigos duplicados. Com a remoção destes artigos duplicados ficam legíveis para a etapa de Seleção de estudos apenas 216 artigos únicos.

### **Seleção de estudos**

A seleção dos estudos foi realizada levando em consideração a análise do título, resumo, introdução, conclusão, enfim a obra na íntegra.

Foram selecionados 34 estudos dos 216 após remoção de duplicados, sendo que houve 5 etapas de seleção para a obtenção destes estudos, os quantitativos de cada uma destas etapas podem ser identificados no Quadro 13.

Quadro 13 – Quantitativo de estudos selecionados

Etapas	Quantidade	
	REJEITADO	ACEITO
Estudos captados	245	
	REJEITADO	ACEITO
Remoção de duplicados	29	216
Seleção por Título	98	118
Seleção por Resumo	37	81
Seleção por Introdução	38	43
Seleção por Conclusão	2	41
Seleção por Obra Completa	7	<b>34</b>

Fonte: O autor (2022).

Durante as etapas de remoção por Título e Resumo foi utilizada uma planilha (ver Figura 13) com termos a serem rejeitados ou aceitos para facilitar a seleção, essa técnica consiste em avaliar alguns itens e definir o termo presente que deveria ser rejeitado ou aceito imediatamente, assim encontrado, o termo é adicionado a planilha e ela seleciona os demais itens da lista automaticamente.

Figura 13 – Exemplo de planilha de seleção de estudos

Título (Português)	118 títulos aceitos	Leitura Título	Agile	Systematic Literature Review
Escala Ágil em toda a Organização Global: Uma Auto-avaliação de SAFe Industrial em Início de Etapa	ACEITO		ACEITO	
Siga o Desenvolvimento da Lua: Escrevendo uma Revisão Sistemática da Literatura sobre Educação em Engenharia Global de Software	REJEITADO			REJEITADO

Fonte: O autor (2022).

Para seguir na seleção por Introdução, por Conclusão e por Obra Completa, se faz necessário o Download dos arquivos completos, assim é possível avaliar os 3 de uma só vez caso prefira. Ao se realizar esta etapa pode ser tentador aproveitar a mesma para a Extração dos dados, porém não é aconselhável realizar a extração junto a seleção, devido ao fato de cada uma exigir uma forma diferente de ver o mesmo texto. Caso seja feita desta forma há um grande risco de a extração ser superficial e pobre.

### ***Extração de dados***

A extração dos dados requer inicialmente uma nova organização dos estudos referenciando-os como E[X] (E01, E02, ...) onde X vai de 1 a quantidade de estudos encontrados. Requer ainda uma análise básica de cada artigo onde pode ser feita uma simples resenha de cada um, uma classificação do tipo de artigo e o método usado em cada um. Também é interessante separar alguns trechos lidos para citação em seu relatório.

Uma vez preparada, a extração de dados deve ocorrer de forma clara, e ininterrupta, pois se trata da fase mais complexa da RSL, onde se ela for realizada de forma incorreta toda a análise fica comprometida. É muito fácil desviar o foco durante a extração, por isso tenha o objetivo da extração sempre descrito de forma clara e alinhe esse objetivo a sua forma de ler o artigo. Uma vez iniciada a leitura de um artigo é necessário concluir toda a extração do mesmo de forma ininterrupta, pois a retomada em uma leitura deste tipo pode consumir muito tempo.

Para realizar a extração dos dados nesta RSL foi colocado como objetivo encontrar as práticas do DDS citadas pelos autores, porém ficou evidente uma grande quantidade de práticas descritas de forma não explícitas, o que fez com que eu (pesquisador), refizesse toda a etapa com uma visão mais detalhada para extrair esses dados implícitos.

A etapa de extração de dados da RSL foi feita de modo a buscar a maior quantidade de dados independentemente de sua futura utilização, por esse motivo uma parte dela é descartada durante a análise de dados.

### **Análise dos dados**

A extração dos dados requer inicialmente uma nova organização dos estudos

### **Estudos primários**

O conjunto final de estudos primários pode ser visualizado na Tabela 8, foram totalizados 34 trabalhos, para se chegar a esses estudos foram realizadas as seguintes etapas de seleção: Remoção de duplicados, Seleção por Títulos, Seleção por Resumos, Seleção por introdução e conclusão e Seleção por obra completa.

Tabela 8 – Estudos primários

ID	Título	Referência
E01	"Figure out How to Code with the Hands of Others": Recognizing Cultural Blind Spots in Global Software Development	Matthiesen, Bjørn, Petersen (2014)
E02	A Scrum-based Process to Distributed Projects in Multidisciplinary Teams: A Case Study	Dos Santos et al. (2018)
E03	Agile Cultural Challenges in Europe and Asia: Insights from Practitioners	Ayed et al. (2017)
E04	Agile Practices for the Global Teaming Model	NOLL et al. (2014)
E05	An Approach for Enabling Effective and Systematic Software Reuse: In a Globally Distributed Software Engineering Team That Uses a Lean Development Methodology	Mani, Halwas (2016)
E06	An Empirical Investigation on Effort Estimation in Agile Global Software Development	Britto et al. (2017)
E07	An Empirical Study of Search-Based Task Scheduling in Global Software Development	Kroll et al. (2017)
E08	An Empirical Study on Scrum Application Patterns in Distributed Teams	Seckin, Ovatman (2018)

- Behavior-driven development as an approach to improve software quality and communication across remote business stakeholders, developers and QA: two case studies Scandaroli et al. (2019)
- E09
- E10 Blueprint Model: A new Approach to Scrum Agile Methodology Godoy et al. (2019)
- E11 Bottom-up Adoption of Continuous Delivery in a Stage-Gate Managed Software Organization Laukkanen et al. (2016)
- E12 Branch Use in Practice: A Large-Scale Empirical Study of 2,923 Projects on GitHub Zou et al. (2019)
- E13 Breaking Collaboration Barriers through Communication Practices in Software Crowdsourcing Machado et al. (2016)
- E14 Case study on risk management practice in large offshore-outsourced Agile software projects Sundararajan et al. (2014)
- E15 Challenges Faced by Agile Testers: A Case Study Padmini et al. (2018)
- E16 Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization Gupta, Venkatachalapathy, Jeberla (2019)
- E17 Communication in Distributed Software Development: A Preliminary Maturity Model Farias et al. (2016)
- E18 Cultural Models and their Interplay in Global Software-Engineering Practice Shah, Nersessian (2015)
- E19 Daedalus or Icarus? Experiences on Follow-the-sun Krüger et al. (2017)
- E20 Domino: A Descriptive Framework for Hybrid Collaboration and Coupling Styles in Partially Distributed Teams Neumayr et al. (2018)
- E21 From Scrum to Agile: A Journey to Tackle the Challenges of Distributed Development in an Agile Team Lous et al. (2018)
- E22 FTS-SPM: A Software Process Model for Follow the Sun Development: Preliminary Results Kroll, Richardson, Audy (2014)
- E23 Global Software Engineering: Evolution and Trends Ebert et al. (2016)
- E24 How Social Software Supports Cooperative Practices in a Globally Distributed Software Project Giuffrida, Dittrich (2014)
- E25 Knowledge Management in Globally Distributed Agile Projects -- Lesson Learned Razzak, Smite (2015)
- E26 Planned and Unplanned Meetings in Large-scale Projects Stray (2018)
- E27 Routine and Standardization in Global Software Development Esbensen, Bjorn (2014)

E28	Scaling Agile Scrum Software Development: Providing Agility and Quality to Platform Development by Reducing Time to Market	Jha, Vilardell, Narayan (2016)
E29	Software development using agile and scrum in distributed teams	Khmelevsky, Li, Madnick (2017)
E30	Streamlining Value in a FOSS Project	Müller et al. (2019)
E31	Systematic Software Development: A State of the Practice Report from Germany	Kuhrmann, Fernández (2015)
E32	Transitioning from Plan-driven to Lean in a Global Software Engineering Organization: A Practice-centric View	Roopa, Kumar, Mani (2018)
E33	Virtual by Design: How a Work Environment Can Support Agile Distributed Software Development	Lous et al. (2018)
E34	Why Closely Coupled Work Matters in Global Software Development	Jensen (2014)

---

Fonte: O autor (2022).

### **Práticas**

A Tabela 9 traz de forma integral as 168 práticas encontradas nos estudos primários após a normalização.

Tabela 9 – Práticas propostas por cada autor

ID	Nome	Estudos
P134	Reunião Diária	E03,E08,E16,E21, E22,E26,E28,E32, E33,E34,E22
P104	Prover Ferramenta de Comunicação, única, centralizada, síncrona e assíncrona, que permita interação individual, em grupo ou global, com texto, áudio e vídeo	E13,E14,E17,E21, E22,E29,E33
P039	Entrega Contínua	E04,E11,E16,E28, E03,E29
P030	Definir valores necessários para Equipes se tornarem Auto-organizáveis, Informação, Responsabilidade, Confiança, Autonomia e Colaboração	E10,E11,E14,E21, E28
P092	Programação em Pares	E03,E08,E15,E21, E33
P033	Distribuir Tarefas por sequência ou dependência	E19,E22,E23,E07

---

P049	Implantação Contínua	E16,E21,E33,E15
P061	Integração Contínua	E03,E16,E21,E28
P068	Melhoria Contínua	E16,E29,E32,E33
P085	Planejar Timebox Sobreposto de ao menos uma hora entre Indivíduos Remotos	E03,E14,E15,E22
P101	Prover Ferramenta de Armazenamento de Conhecimento, única, centralizada, confiável	E13,E20,E22,E25
P108	Prover Ferramenta de Versionamento de Código, centralizada	E10,E21,E22,E24
P115	Realizar reuniões de forma presencial como mecanismo de socialização, aumenta a confiança	E08,E14,E32,E16
P130	Reunião de Retrospectiva da Iteração	E16,E21,E29,E33
P168	Visita a Equipes Remotas	E14,E08,E29,E34
P100	Prover Ferramenta de Armazenamento de Arquivos, centralizada	E16,E24,E22
P102	Prover Ferramenta de Armazenamento de Informação, única, centralizada, que promova visibilidade e transparência.	E05,E14,E28
P106	Prover Ferramenta de Gerenciamento de Tarefas, centralizada	E13,E15,E21
P116	Realizar Reuniões Individuais (One-on-one)	E21,E26,E33
P126	Reservar 20% do tempo para os colaboradores explorarem ideias inovadoras relacionadas aos produtos da organização, experimentar e aprender em Ambiente de Teste	E16,E21,E33
P144	Revisão de Código por demanda feita pela Equipe de Desenvolvimento	E08,E21,E33
P163	Utilizar Modelo de Desenvolvimento Iterativo e Incremental	E13,E22,E28
P013	Comunicação direta com Canal Aberto entre Equipe de Desenvolvimento e Cliente	E03,E15
P018	Define papéis e responsabilidades detalhadamente	E14,E30
P019	Definição de Pronto para Organização ou Projeto	E15,E34
P020	Definir equipes com diversidade de habilidades técnicas, vivências, experiências, culturas e perfis comportamentais.	E32,E28
P026	Definir responsáveis pelo Esforço de Articulação	E01,E04
P031	Designar um mentor para transferir conhecimento à novos membros	E14,E32
P037	Elaborar Plano de Comunicação	E14,E28
P041	Explicação dos processos de forma visual, detalhada e centralizada	E10,E13

---

P047	Histórias de Usuário bem detalhadas (baixa granularidade)	E22,E32
P084	Planejar reuniões fixas para reduzir as desvantagens individuais	E28,E14
P089	Prever o Esforço de Articulação	E01,E04
P097	Promover reuniões frequentes de Transferência de Conhecimento	E32,E16
P103	Prover Ferramenta de Colaboração	E13,E15
P121	Refinamento do Backlog do Produto	E16,E28
P127	Reunião de Alinhamento de Equipes Remotas	E28,E34
P128	Reunião de Planejamento da Iteração	E16,E21
P129	Reunião de Planejamento da Iteração com Planning Poker	E06,E21
P132	Reunião de Revisão da Iteração	E08,E16
P150	Status do Projeto visível a todos os Stakeholders	E10,E14
P001	Análise estática de código	E21
P002	Ao expressar ideias mais complexas, prefira utilizar recursos visuais	E15
P003	Aprendizado contínuo	E16
P004	Atividades visíveis de forma organizada e centralizada	E03
P005	Atribuir autoridade para o Dono do Produto exigir a conclusão das tarefas delegadas	E14
P006	Atualização de Histórias de Usuário a partir de registro de reuniões	E15
P007	Brainstorming	E29
P008	Calendário de Entregas sempre atualizado	E22
P009	Carga de trabalho da Equipe de Desenvolvimento sempre estável	E28
P010	Catálogo de alterações dos testes	E15
P011	Comunicação aberta durante a execução de atividades	E13
P012	Comunicação de texto com Modelos de mensagens	E22
P014	Comunicação direta com Canal Aberto entre Stakeholders	E10
P015	Comunicação Simplificada com Modelos Visuais	E09
P016	Coordenação eficaz mantendo o equilíbrio entre os processos sociais, produtividade da equipe e o bem-estar individual	E26
P017	Coordenação melhorada a partir de discursões de articulação do trabalho	E24
P021	Definir Mindset Organizacional	E05

---

P022	Definir Nível de Maturidade de Comunicação na Organização	E17
P023	Definir Normas de Comunicação	E14
P024	Definir o tempo para um registro de atividade ser considerado "Defeito"	E15
P025	Definir responsabilidades e limites entre a comunicação de Equipes Remotas	E04
P027	Definir um Coordenador de Cerimônia para as reuniões	E14
P028	Definir um Dono da Comunicação	E02
P029	Definir uma estimativa de esforço das Histórias de Usuário	E06
P032	Diferenças culturais devem ser identificadas e gerenciadas	E14
P034	Divisão de tarefas de forma modular entre Equipes Remotas	E14
P035	Documentação enxuta	E28
P036	Dono do Produto deve conhecer de forma próxima cada Equipe Remota	E14
P038	Elaborar Plano de Transferência de Conhecimento	E32
P040	Estabelecer protocolos sociais comuns	E24
P042	Feedback direto ao Dono do Produto	E08
P043	Ferramenta de gestão de dados que facilite alterações rápidas (Planilhas)	E15
P044	Gamificar o desenvolvimento	E15
P045	Garantir a confiabilidade e estabilidade dos entregáveis	E16
P046	Gerenciar Dívida Técnica	E32
P048	Histórias de Usuário com documentação técnica adicional	E29
P050	Implantação gradual das ferramentas propostas	E16
P051	Implementação gradual de mudanças	E16
P052	Incentivar a exploração de novas formas de construir confiança entre indivíduos remotos	E20
P053	Incentivar a socialização através de conversas informais	E24
P054	Incentivar a socialização no início do projeto	E14
P055	Incentivar identificação de falhas	E28
P056	Incentivar o comprometimento com o objetivo geral	E28
P057	Incentivar o conhecimento de novas tecnologias	E14
P058	Incentivar o desenvolvimento de habilidades colaborativas	E14

---

P059	Incentivar os stakeholders a liderarem socializações que promovam a quebra de barreiras socioculturais e colaborativas	E24
P060	Informações de Projeto publicadas em site com alto nível de abstração	E14
P062	Integração Diária	E29
P063	Integração do Código com diretrizes de qualidade requeridas bem definidas	E32
P064	Integrar especialistas em decisões	E20
P065	Iterações não mudam as regras durante execução	E15
P066	Manter a diversidade cultural em equipes	E29
P067	Manter o equilíbrio entre implementação de recursos e qualidade	E28
P069	Metas do Projeto com reavaliações frequentes	E28
P070	Novas Histórias de Usuário devem ser informadas a todos stakeholders	E13
P071	Orientar os stakeholders a executar uma atividade por vez	E28
P072	Padronização dinâmica, baseada em rotinas sem hierarquias	E27
P073	Padronização do Código	E22
P074	Padronizar método de trabalho das Equipes Remotas	E14
P075	Padronizar os Termos Linguísticos Utilizados	E09
P076	Pare a Iteração	E32
P077	Permitir socialização por canais informais	E24
P078	Permitir transferência de tarefas entre Equipes Remotas	E14
P079	Planejamento da Iteração Ad-hoc	E21
P080	Planejamento da Iteração de baixo para cima	E28
P081	Planejamento de Lançamento	E04
P082	Planejar estratégia de Criação, Armazenamento, Gestão e Transmissão de Conhecimento	E25
P083	Planejar os horários das reuniões de acordo com o Dono do Produto	E08
P086	Preferencialmente utilize compartilhamento de tela para transferência de conhecimento	E22
P087	Preferencialmente utilize comunicação síncrona para transferência de conhecimento	E22

---

P088	Prevenção de problemas baseado em análise de dependências das tarefas	E27
P090	Processos modelados de forma Ágil	E03
P091	Produto Mínimo Viável (MVP)	E32
P093	Projeto com no máximo três Equipes Remotas	E22
P094	Projeto deve possuir uma identidade para seu objetivo	E14
P095	Promover a conscientização cultural	E22
P096	Promover a transferência de conhecimento entre indivíduos	E15
P098	Propriedade Coletiva	E03
P099	Prover agenda centralizada	E24
P105	Prover Ferramenta de Gerenciamento de Problemas, centralizada	E24
P107	Prover Ferramenta de Provisionamento de Ambientes, automatizada	E15
P109	Prover Ferramentas de Gerenciamento de Feedback	E13
P110	Publicizar registro de reunião	E24
P111	Qualidade de Código de responsabilidade da Equipe de Desenvolvimento	E33
P112	Ramificação de Código	E12
P113	Rastreamento de Código via anotações	E13
P114	Realizar reuniões baseadas em slides	E21
P117	Realizar reuniões por demanda, stakeholders podem solicitar quando necessário	E21
P118	Realizar uma Transição Gradual entre Modelos de Processo	E28
P119	Refatoração de Código	E03
P120	Refinamento de Backlog do Produto baseado no tamanho das Histórias de Usuário	E16
P122	Refinamento do Backlog do Produto com Priorização das Histórias de Usuário	E30
P123	Refinamento do Backlog do Produto gravadas em áudio e vídeo	E15
P124	Remoção do sistema de e-mail como comunicação interna	E21
P125	Remover estimativas de tamanho de tarefas	E21
P131	Reunião de Retrospectiva da Iteração Gamificada	E15

---

P133	Reunião de Revisão da Iteração com tempo de discussão do Feedback com a Equipe de Desenvolvimento	E15
P135	Reunião Diária via telefone, videoconferência ou e-mail	E22
P136	Reunião Entrega de Produto com demonstração e Feedback dos Stakeholders	E08
P137	Reuniões devem ter as pautas comunicadas detalhadamente para garantir que os participantes estejam preparados	E14
P138	Reuniões devem ter os participantes definidos de acordo com os assuntos em pauta	E14
P139	Reuniões devem ter seus requisitos determinados	E14
P140	Reuniões registradas em ATA, Áudio e Vídeo	E24
P141	Reutilização de Código	E05
P142	Reverse Shadowing	E32
P143	Revisão de Código formal e obrigatória feita pela Equipe de Qualidade dedicada	E21
P145	Rotação de Papéis	E33
P146	Solicitar testes de aceitação antecipados para os recursos fechados	E28
P147	Stage-Gate	E11
P148	Status do Projeto com fluxo de informações rastreável	E14
P149	Status do Projeto com meios de monitoramento	E14
P151	Tarefas concluídas devem ser informadas a todos stakeholders	E13
P152	Testes automatizados no turno inverso do desenvolvimento	E32
P153	Toda a equipe deve interromper o desenvolvimento para se concentrar na resolução de problemas quando um número de problemas de qualidade pré-definido for atingido	E32
P154	Transferência de Conhecimento via Registro	E28
P155	Tratar pontos cegos culturais	E01
P156	Utilização de Áudio e Vídeo em reuniões remotas	E08
P157	Utilizar a comunicação assíncrona como Base de Conhecimento	E13
P158	Utilizar Arquitetura de Software de menor custo de implantação	E16
P159	Utilizar as Reuniões frequentes para orientações metodológicas	E29
P160	Utilizar Equipes Remotas apenas para Desenvolvimento e Testes	E22

---

P161	Utilizar Histórias de Usuário para expressar tarefas	E16
P162	Utilizar Modelo de Desenvolvimento Incremental sem Iterações	E21
P164	Utilizar o humor para aumentar a socialização	E14
P165	Utilizar técnica de Modelos culturais para simplificar a visualização dos problemas comuns	E18
P166	Utilizar videoconferência para socialização informal	E14
P167	Utilize reuniões para a construção de confiança entre os stakeholders	E22

Fonte: O autor (2022).

## **Metodologias**

Ao decorrer do estudo foi reparado uma grande relação das práticas citadas e das metodologias/frameworks citadas pelos autores dos estudos primários, muitos deles trazem informações referentes a essas metodologias na explicação de utilização de suas práticas, tendo isso em mente viu-se a necessidade de listar na Tabela 10, quais estudos citam cada metodologia.

Tabela 10 – Metodologias encontradas nos estudos primários

ID	Nome	Estudos
M01	Scrum	E27,E34,E25,E04,E01,E02,E32,E24,E26,E29,E31,E20,E15,E14,E11,E09,E10,E16,E28,E22,E21,E23,E03,E08,E07,E06
M02	XP	E25,E26,E29,E31,E30,E15,E11,E09,E21,E06,E33,E04,E22,E03,E28,E16,E14,E12
M03	Kanban	E25,E04,E01,E31,E15,E09,E10,E21,E03,E06
M04	SAFe	E04,E32,E11,E16,E21,E18,E03,E06
M05	CMMI	E25,E04,E31,E15,E14,E17
M06	FDD	E25,E31,E15,E06
M07	TDD	E30,E15,E09,E21
M08	Adaptive Software Development (ASD)	E06,E33
M09	DSDM	E15,E06
M10	Open-Source Software Development (OSS)	E24,E12

M11	GTM	E04
M12	C2M	E17
M13	Wave	E17
M14	SCAMPI	E17
M15	Agile Modeling (AG)	E03
M16	RUP	E23

---

Fonte: O autor (2022).

## **APÊNDICE C – QUESTIONÁRIO SOBRE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NOS INSTITUTOS FEDERAIS**

Este questionário faz parte da pesquisa desenvolvida pelo mestrando em Ciências da Computação do Centro de Informática da Universidade Federal de Pernambuco (CIn-UFPE), Welington Mothé de Oliveira, orientado pelo Professor Alexandre Marcos Lins de Vasconcelos. Não será feito nenhum tipo de identificação, responda as perguntas abaixo de acordo com a perspectiva que tem de seu local de trabalho. As perguntas que não souber responder deixe em branco ou clique na opção “Não sei informar”.

Foi construído o objetivo de analisar o Processo de Desenvolvimento de Software no contexto dos Institutos Federais de Educação com o propósito de identificar o nível de conhecimento em relação ao Desenvolvimento Distribuído de Software do ponto de vista dos stakeholders.

A motivação deste questionário se deu durante a dificuldade de encontrar os IFs que já realizam o Desenvolvimento Distribuído de Software (DDS) atualmente, uma informação parecida aparece no painel de informações do FORTI, mas ao entrar em contato com os representantes do setor de desenvolvimento de software dos institutos respondentes foi identificado uma complexidade de especificação dos cenários muito maior do que o previsto, o que acarretou na necessidade de buscar mais informações sobre os demais IFs, com o intuito de identificar se outros institutos possuíam tais características.

Foi utilizada a ferramenta Google Forms para criar e disponibilizar o questionário digital que ainda possibilitou a tabulação dos dados levantados, ele ficou disponível para respostas entre 20/04/2020 e 30/07/2020 e obteve 36 respostas.

As pessoas questionadas neste formulário são servidores dos IFs que conhecem o desenvolvimento de software de sua instituição do ponto de vista de um stakeholder, seja o desenvolvedor ou o cliente demandante.

### **Questões**

As duas perguntas da primeira seção têm a função de identificação a nível institucional, sem revelar dados pessoais. Enquanto a segunda seção trata as informações a respeito do ponto de vista do respondente em relação ao atual formato

de Desenvolvimento de Software em sua Instituição.

## **SEÇÃO DE IDENTIFICAÇÃO**

### **Q1. Instituição**

- CEFET-MG
- CEFET-RJ
- CPII
- IFAC
- IFAL
- IFAM
- IFAP
- IFB
- IFBA
- IFBAIANO
- IFC
- IFCE
- IFES
- IFF
- IFFar
- IFG
- IFGOIANO
- IFMA
- IFMG
- IFMS
- IFMT
- IFNMG
- IFPA
- IFPB
- IFPE
- IFPI
- IFPR
- IFRJ
- IFRN

- IFRO
- IFRR
- IFRS
- IFS
- IFSC
- IFSERTÃO-PE
- IFSP
- IFSudesteMG
- IFSUL
- IFSULDEMINAS
- IFTM
- IFTO

**Q2. Nome do seu Campus**

Campo de texto curto.

Ex.: Campus X, Reitoria ou Unidade X

**SEÇÃO DE DESENVOLVIMENTO DE SOFTWARE**

**Q3. Qual modelo de desenvolvimento de software é utilizado em seu instituto?**

- Centralizado (Todos os membros podem comunicar-se face a face diariamente)
- Distribuído (Ao menos um membro não consegue se comunicar face a face diariamente)
- Passou de centralizado para distribuído devido ao estado de quarentena
- Outros [O respondente pode escrever]

**Q4. Existe alguma pretensão da direção do seu instituto em migrar para o modelo de Desenvolvimento Distribuído de Software?**

- Sim
- Não
- Não sei informar

**Q5. Qual modelo de desenvolvimento de software você acredita que seja melhor para os Institutos Federais?**

- Centralizado
- Distribuído

**Q6. Alguém já questionou a possibilidade de usar o modelo distribuído de desenvolvimento de software no instituto? Descreva o ocorrido.**

Campo de texto aberto.

**Q7. A reitoria do seu instituto disponibiliza serviço de hospedagem de sistemas?**

- Sim
- Não
- Não sei informar

**Q8. Qual o papel da Reitoria no desenvolvimento de software do instituto?**

- Desenvolvimento de projetos sem colaboração
- Mantenedora dos serviços e ferramentas de desenvolvimento de software ou integração de equipes (Git, Kanban, Rede Social, Groupware)
- Ofertante de cursos de desenvolvimento de software
- Gerenciamento de todos os projetos
- Desenvolvimento de projetos com colaboração
- Coordenação de todas as equipes
- Desenvolvimento de projetos com equipes virtuais (Fisicamente Distribuída)
- Integração do código produzido por todas as equipes
- Orientadora de padrões de projetos
- Ofertante de cursos de desenvolvimento de software voltados a padrões institucionais
- Orientadora de docentes na preparação de cursos de desenvolvedores de software voltados a padrões institucionais
- Outros [O respondente pode escrever]

**Q9. Seu campus desenvolve softwares independente da reitoria?**

- Sim
- Não
- Sou da Reitoria

**Q10. Seu campus desenvolve algum software junto a reitoria?**

- Sim
- Não
- Sou da Reitoria

**Q11. Algum projeto no seu instituto utiliza a força de trabalho de Alunos e Professores para melhoria de software institucional?**

- Sim
- Não
- Sou da Reitoria

**Q12. Quais frameworks/metodologias/modelos de processo são utilizados?**

- CMMI
- GTM
- MPS.BR
- XP
- RUP
- SAFe
- Scrum
- Kanban
- Outros [O respondente pode escrever]

**Q13. Quais ferramentas são utilizadas?**

- Gitlab (Versionamento)
- Hudson (Integração Contínua)
- OpenProject (Gerenciamento de Projetos)
- GLPI (Gerenciamento de Chamados)
- Mattermost (Comunicação Síncrona e Assíncrona)

- Docker (Aplicações como Serviço)
- Behaven
- Outros [O respondente pode escrever]

**Q14. Em quais papéis você já atuou no desenvolvimento de software da instituição?**

- Líder de Equipe
- Desenvolvedor
- Analista de Projeto
- Líder de Projeto
- Atendimento/Suporte
- Administrador de Banco de Dados
- Testador
- Gerente de Projeto
- Outros [O respondente pode escrever]

**Q15. Observações**

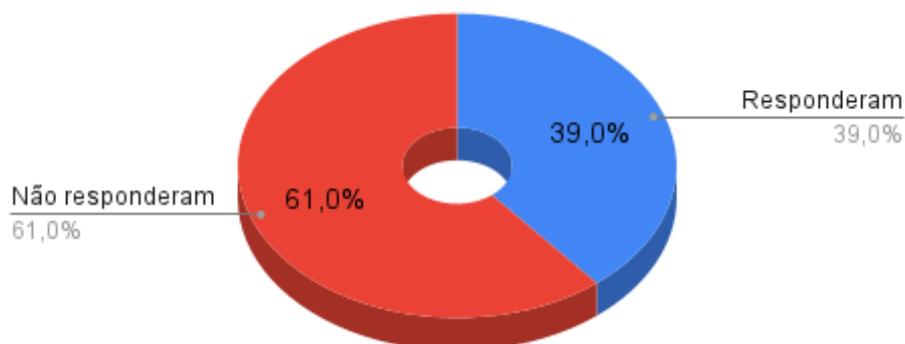
Campo de texto aberto.

***Síntese das Respostas***

Este questionário teve uma taxa de participação de 39,02% (16/41) dos Institutos, devido a pequena taxa podemos imaginar que não seja parâmetro para uma afirmação a respeito de toda a Rede Federal, assim sendo recomenda-se sempre utilizar estes dados coletados em conjunto com outra pesquisa, tal como o Painel de Informações do FORTI, o qual tem uma taxa de participação 95,12% (39/41) dos IFs.

A seção de identificação do formulário contém duas questões, “Instituição” e “Campus”, que possibilitaram identificar a quantidade de IFs que participaram da pesquisa e quantos respondentes estão lotados na Reitoria ou Campi, apresentados nos gráficos a seguir.

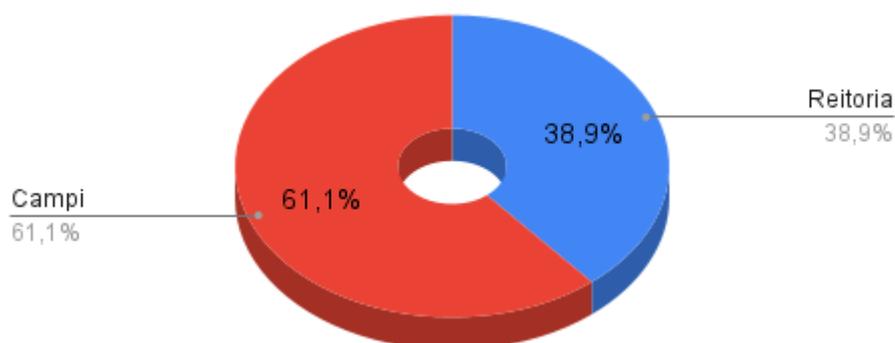
Gráfico 1 – Quantidade de IFs que responderam ao questionário



Fonte: O autor (2022).

Os 16 institutos que responderam ao questionário foram IFAM, IFBA, IFBAIANO, IFC, IFCE, IFES, IFMT, IFPB, IFPE, IFRR, IFRS, IFSC, IFSP, IFSudesteMG, IFSULDEMINAS e IFTM.

Gráfico 2 – Quantidade de respondentes que estão na Reitoria/Campi



Fonte: O autor (2022).

A maioria dos respondentes não são da Reitoria, levando em consideração que a maioria dos IFs trabalhe o Desenvolvimento de Software de forma centralizada na reitoria, o grau de conhecimento dos respondentes em relação ao assunto proposto pode ser considerado não especializado. O que torna importante a separação dos perfis que atuam diretamente no desenvolvimento de software da Reitoria. Próximas pesquisas devem levar em consideração essa separação para a captura de dados mais relevantes.

Todos os gráficos abaixo foram normalizados, desconsiderando os que não souberam informar ou que não responderam.

## SEÇÃO DE DESENVOLVIMENTO DE SOFTWARE

### RQ3. Qual modelo de desenvolvimento de software é utilizado em seu instituto?

Gráfico 3 – Modelo de desenvolvimento de software utilizado nos IFs



Fonte: O autor (2022).

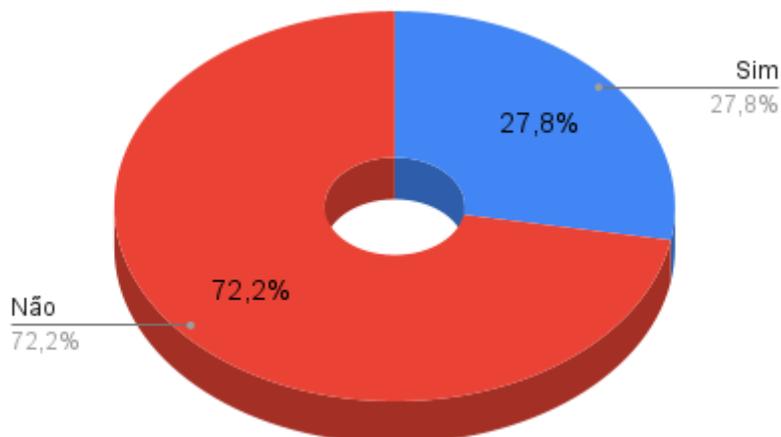
Na questão de modelo de desenvolvimento de software utilizado nos IFs, houve 35 respostas, foi encontrado uma taxa de 64,4% utilizando o modelo Centralizado e apenas 22,2% o modelo Distribuído, deixando claro que a maioria ainda se encontra trabalhando de forma Centralizada em suas Reitorias.

Outra informação relevante foi o fato que 5 pessoas responderam que “Passou de centralizado para distribuído devido ao estado de quarentena” o que indica a Pandemia como um novo fator de incentivo a adoção de metodologias de DDS.

Um respondente acrescentou a resposta “É misto, na Reitoria as soluções sistêmicas e devido aos processos não normatizados, os campi também desenvolvem”, isso indica que seu instituto além de possuir um desenvolvimento de software centralizado na Reitoria, também possui desenvolvimento nos campi de forma independente, uma realidade também vivenciada por mim no Ifes.

### RQ4. Existe alguma pretensão da direção do seu instituto em migrar para o modelo de Desenvolvimento Distribuído de Software?

Gráfico 4 – Pretensão da direção dos IFs em migrar para o modelo DDS



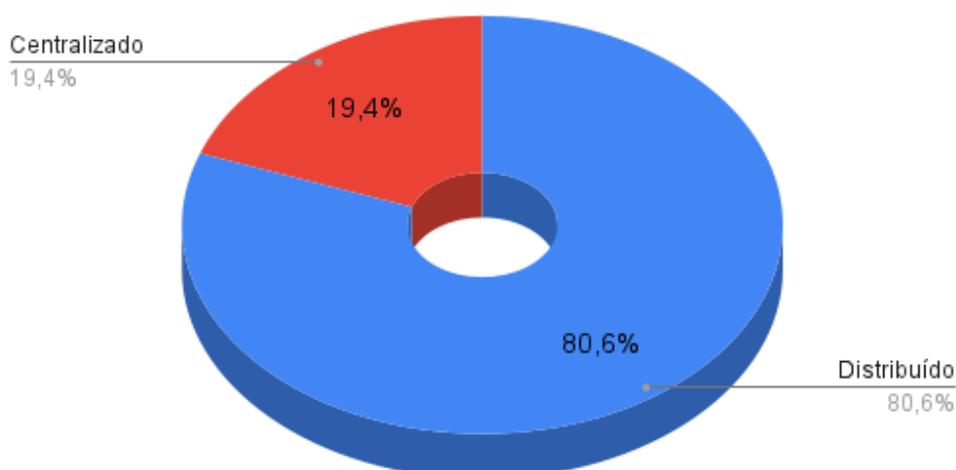
Fonte: O autor (2022).

Dentre 19 respostas a grande maioria respondeu a esta questão dizendo que a direção não possui intenção de migrar para o modelo de Desenvolvimento Distribuído de Software, mas é importante considerar que essa intenção possa não ter chegado ao conhecimento do respondente.

Um dos respondentes acrescentou que a possibilidade de usar o modelo DDS no instituto já foi questionada, mas devido a rotatividade na coordenação, o planejamento nunca chegou a ser oficial.

#### **RQ5. Qual modelo de desenvolvimento de software você acredita que seja melhor para os Institutos Federais?**

Gráfico 5 – Melhor modelo de desenvolvimento de software para os IFs

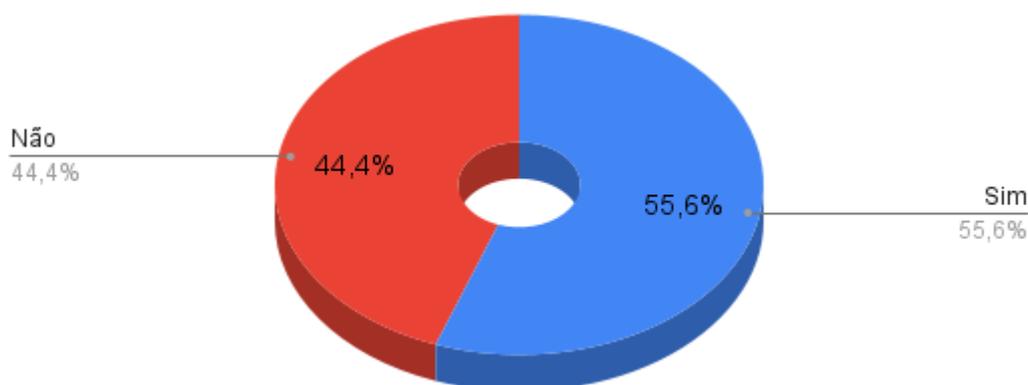


Fonte: O autor (2022).

Esta questão deixa evidente a perspectiva de que o modelo de desenvolvimento distribuído de software é uma opção relevante para a maioria dos respondentes, sendo eles 80,6% das 36 respostas, no entanto a possibilidade de utilização deste modelo não é bem vista por 19,4%, o que apesar de ser uma taxa pequena do ponto de vista de comparação deste gráfico, pode representar uma alta taxa de falha para um projeto de migração do modelo.

#### **RQ6. Alguém já questionou a possibilidade de usar o modelo distribuído de desenvolvimento de software no instituto?**

Gráfico 6 – Institutos que já questionaram a possibilidade de usar o modelo DDS



Fonte: O autor (2022).

Dentre as 36 respostas desta questão, 55,6% confirmaram que alguém já questionou a possibilidade de usar o modelo DDS no instituto, enquanto 44,4% não confirmou, mas é importante levar em consideração a possibilidade de não ter chegado ao conhecimento do respondente.

As descrições das respostas apresentam um cenário onde a questão é levantada pelo próprio respondente, ou outros próximos a ele, em visitas ou reuniões técnicas, dentre eles são informados que não foi levado a discussão, está em análise ou não foi aceito.

Outros cenários onde existe uma descrição de discussão mais aprofundada que apresenta os seguintes fatores.

**Resposta 3:** “Sim, mas não há confiança nem força de trabalho alinhada que tenha conseguido unir as pontas”

Talvez seja o principal ponto de sucesso para implantação de qualquer

mudança, o alinhamento de todos os envolvidos, por isso essa resposta pode representar um nível muito baixo de maturidade da força de desenvolvimento de software para a adoção de uma mudança de paradigma tão grande.

**Resposta 9:** “Internamente temos projetos no Gitlab, mas entre outros campi ainda não vejo o DDS de forma concreta”

Esta resposta mostra um problema muito comum de comunicação entre a Reitoria e os Campi, onde cada unidade acaba desenvolvendo software de forma independente.

**Resposta 11:** “Sim, mas devido a rotatividade na coordenação, o planejamento nunca chegou a ser oficial”

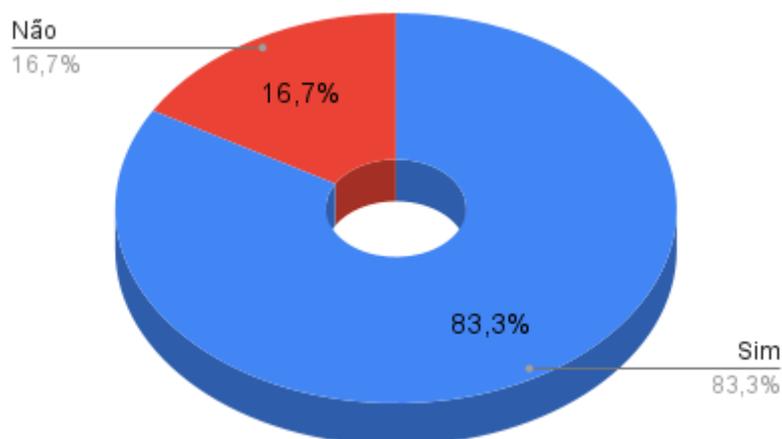
Esta resposta traz outra problemática frequente para os setores de TI dos IFs, a falta de continuidade de projetos, isso demonstra a necessidade de melhoria da maturidade em gerenciamento de projetos da instituição antes da migração para um modelo DDS.

**Resposta 13:** “Sim. Eu mesmo várias vezes em grupos de TI e em eventos com professores da área. Porém o mindset é de obter retorno financeiro para instituição, centralizando e exigindo Termos de Cooperação”

Este fator apresentado pode ser discutido muito mais amplamente, pois a distribuição da força de desenvolvimento de software não leva necessariamente a perda da propriedade do código construído. Outro ponto é a capacidade de o Termo de Cooperação ser alterado, para que as partes envolvidas tenham um retorno em porcentagens dos lucros obtidos.

**RQ7. A reitoria do seu instituto disponibiliza serviço de hospedagem de sistemas?**

Gráfico 7 – Institutos que disponibilizam serviço de hospedagem de sistemas



Fonte: O autor (2022).

Esse questionamento demonstra uma força muito grande da estrutura das Reitorias quanto ao provimento de suporte aos sistemas dos Campi com 83,3% dos 35 respondentes possuindo uma Reitoria que disponibiliza serviço de hospedagem de sistemas. Isso demonstra um ponto positivo para o suporte ao modelo DDS uma vez que alguns testes de softwares demandaram deste espaço provido pelas Reitorias.

#### RQ8. Qual o papel da Reitoria no desenvolvimento de software do instituto?

Gráfico 8 – Papel da Reitoria no desenvolvimento de software do instituto



Fonte: O autor (2022).

Esta questão pode se tornar muito complexa de se debater, portanto esta análise vai abarcar apenas os 4 pontos mais citados.

Era esperado uma taxa ainda mais alta de “Desenvolvimento de projetos sem colaboração” uma vez que a maioria dos responderam que os IFs desenvolvem software de maneira Centralizada, entretanto ela contempla apenas 23/36 respostas, o que pode ser considerado baixo, mesmo sendo o mais citado.

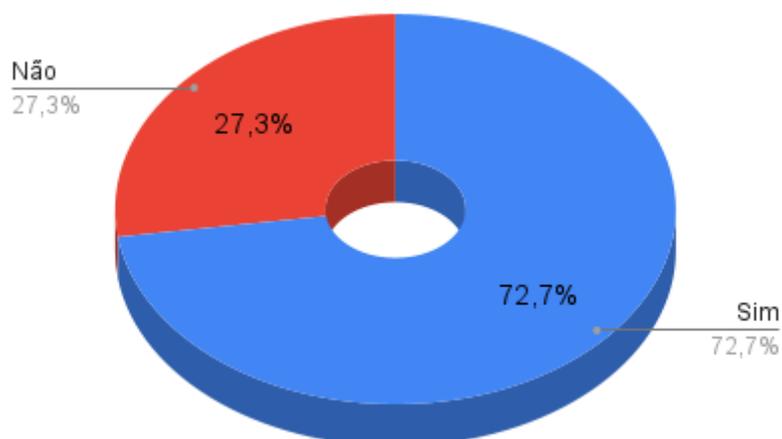
As alternativas “Desenvolvimento de projetos com colaboração” e “Desenvolvimento de projetos sem colaboração” podem ser mal interpretadas, tendo isso em mente, se faz necessário intender que a criação de projetos pode ocorrer com ou sem colaboração, independentemente de a instituição trabalhar de forma Centralizada ou Distribuída.

As alternativas “Gerenciamento de todos os projetos”, “Desenvolvimento de projetos com colaboração” e “Mantenedora dos serviços e ferramentas de desenvolvimento de software ou integração de equipes (Git, Kanban, Rede Social, Groupware)” existindo em uma instituição, demonstram uma facilidade de migração para o Distribuído, uma vez que preenche alguns dos requisitos básicos para a migração.

A análise dessa questão será mais útil quando comparada com as outras perguntas.

#### **RQ9. Seu campus desenvolve softwares independente da reitoria?**

Gráfico 9 – Campi que desenvolvem softwares independente da reitoria

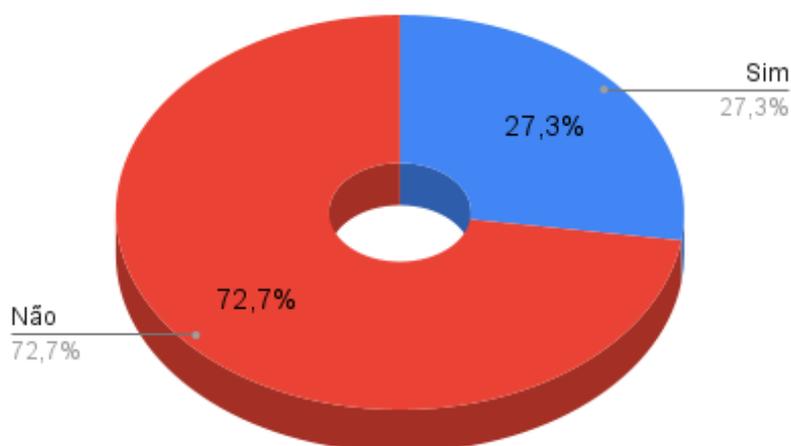


Fonte: O autor (2022).

Apenas 22 respondentes eram de campus, 72,7% destes confirmaram que seu campus possui desenvolvimento de software de forma independente da Reitoria, enquanto 27,3% afirmaram não possuir.

#### RQ10. Seu campus desenvolve algum software junto a reitoria?

Gráfico 10 – Campi que desenvolvem softwares junto a reitoria

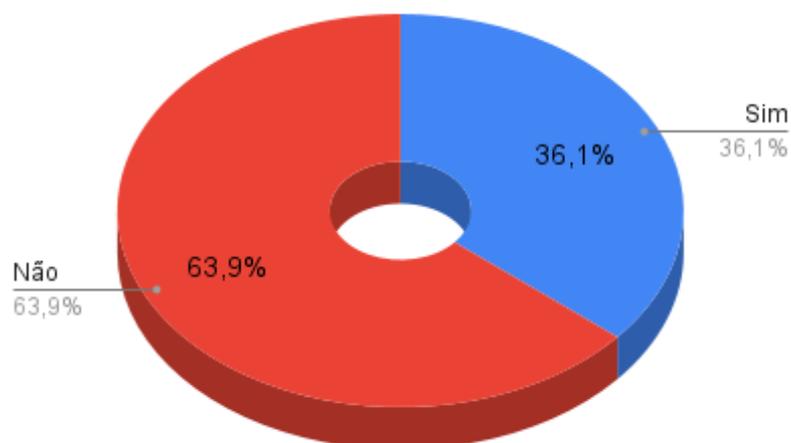


Fonte: O autor (2022).

Apenas 22 respondentes eram de campus, 27,3% destes confirmaram que seu campus possui desenvolvimento de software junto a Reitoria, enquanto 72,7% afirmaram não possuir.

#### RQ11. Algum projeto no seu instituto utiliza a força de trabalho de Alunos e Professores para melhoria de software institucional?

Gráfico 11 – Institutos que possuem projeto que utiliza a força de trabalho de Alunos e Professores para melhoria de software institucional



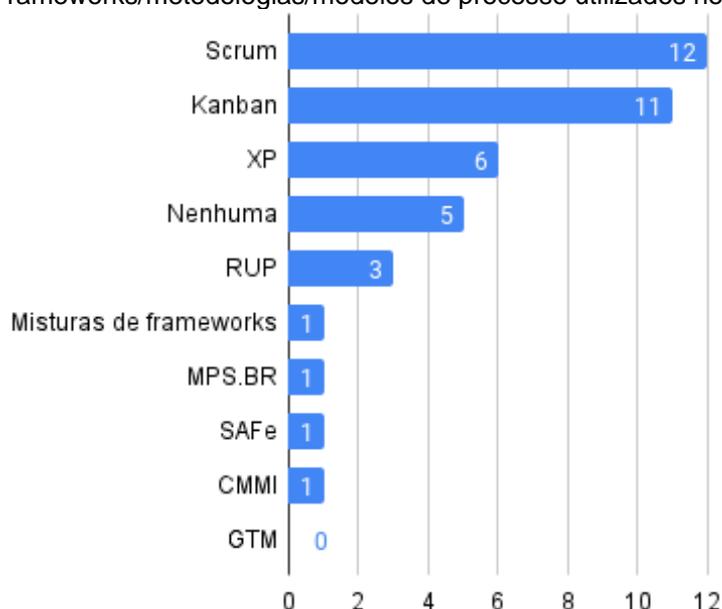
Fonte: O autor (2022).

Dentre os 26 respondentes, apenas 36,1% destes confirmaram que seu instituto utiliza a força de trabalho de Alunos e Professores para melhoria de software institucional, enquanto 63,9% afirmaram não utilizar.

Esta questão é muito importante para entender o nível de maturidade no desenvolvimento de software da instituição, o resultado mostra uma realidade diferente da esperada, uma vez que mais de 36% apresentam uma capacidade de trabalho colaborativo dos professores e alunos com a instituição.

### RQ12. Quais frameworks/metodologias/modelos de processo são utilizados?

Gráfico 12 – Frameworks/metodologias/modelos de processo utilizados nos IFs



Fonte: O autor (2022).

A resposta demonstra uma alta taxa de utilização do Scrum, Kanban e XP, assim como acontece na iniciativa privada que conheço, talvez seja devido ao fato de serem metodologias muito divulgadas em nossa região.

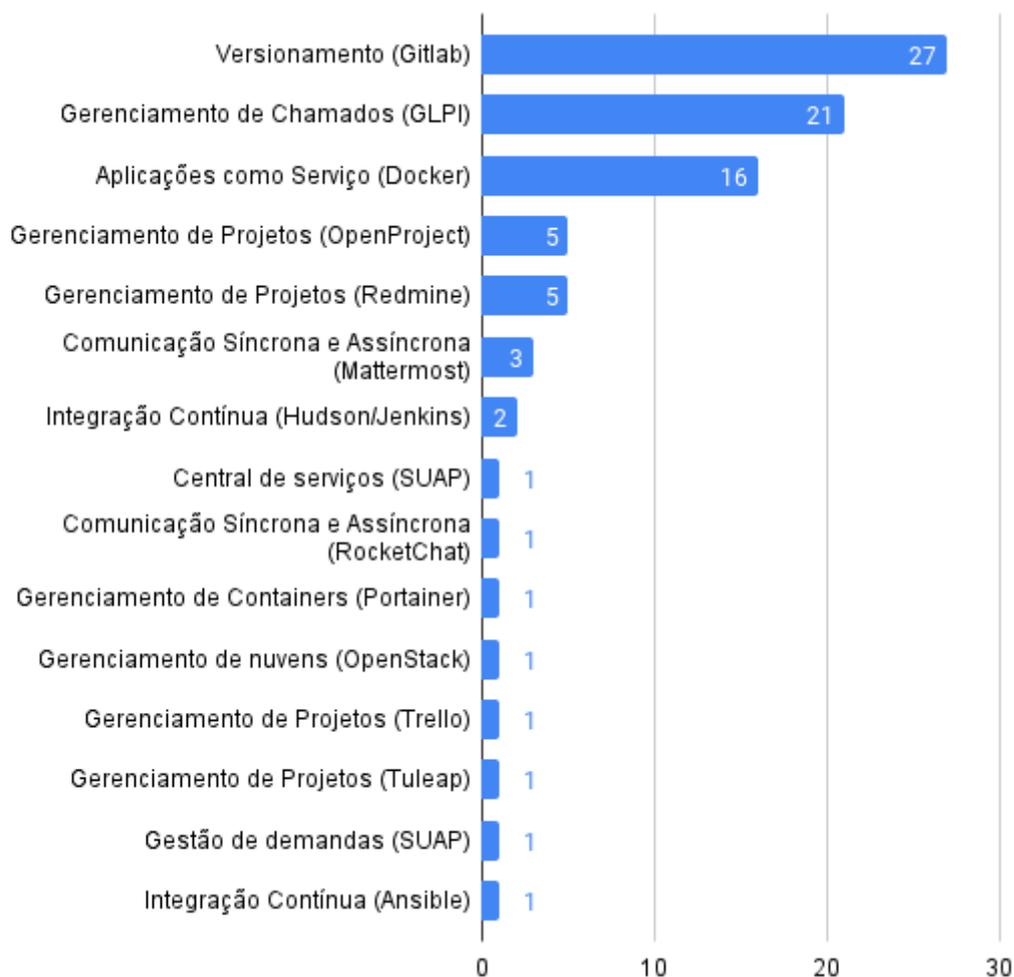
Outro fator importante nesta questão é considerar o fato de que as metodologias quase nunca são utilizadas em sua integralidade, uma vez que vários fatores como o ambiente ou a cultura podem interferir na maneira como elas são implantadas, o que faz com que a maioria das instituições utilize uma Mistura de Frameworks/Metodologias, assim sendo um respondente adicionou uma alternativa “Misturas de frameworks”, poderia ser anulada, uma vez que o objetivo da pergunta era comparar o grau de utilização das metodologias, porem acredito ser importante mostrar esse fator.

O Framework SAFe mostra esse caminho de mistura de frameworks sendo levado a sério, uma vez que ele traz de forma oficial um framework que deixa claro a utilização da junção de vários frameworks e metodologias, incluindo Scrum, XP e Kanban.

Outro fator importante a se notar nesta questão é que a maioria dos respondentes utilizam as metodologias voltadas a gestão de projetos ou equipes, mas poucos utilizam as metodologias voltadas a melhoria de processos, tais como MPS.BR ou CMMI.

### RQ13.Quais ferramentas são utilizadas?

Gráfico 13 – Ferramentas utilizadas nos IFs



Fonte: O autor (2022).

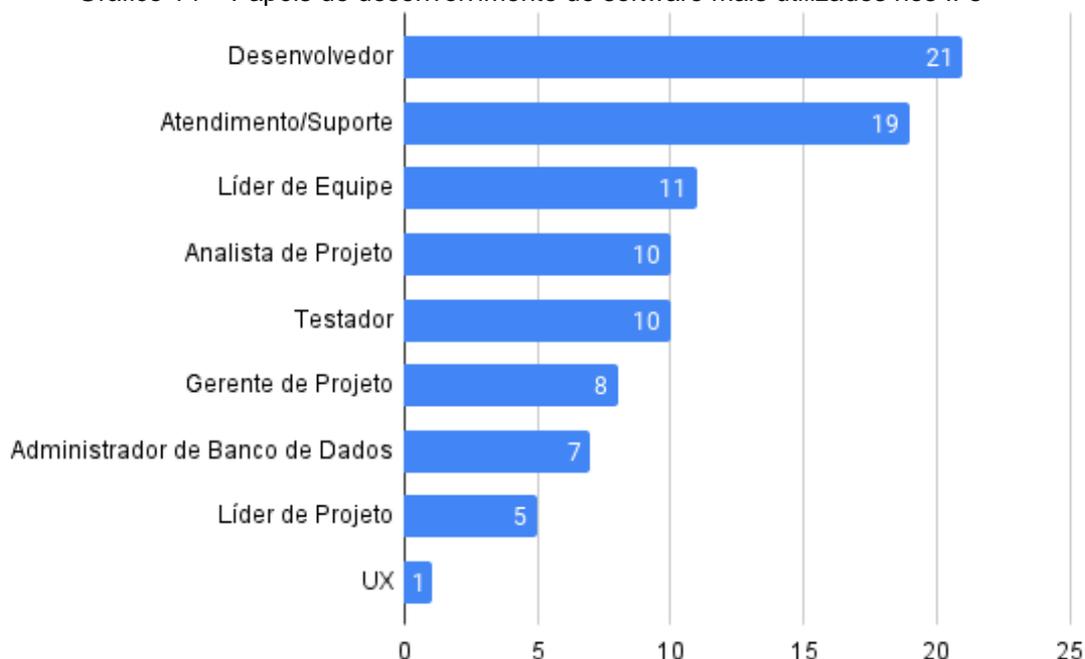
Das 35 respostas, 27 utilizam uma ferramenta de versionamento (Gitlab), o que demonstra um bom nível de conhecimento da equipe de desenvolvimento de software e uma capacidade de trabalhar de forma colaborativa.

Também foi notado um alto índice de utilização das ferramentas GLPI e Docker o que pode indicar um bom nível de gerenciamento das demandas internas e a capacidade de lidar com desenvolvimento de software em forma de serviços.

Outra característica importante a se notar é a baixa utilização de ferramentas de Gerenciamento de Projetos, uma vez que juntando as quatro citadas (OpenProject, Redmine, Trello e Tuleap) chega-se a uma taxa de utilização de apenas 34,28%, isso pode mostrar um baixo nível de capacidade de gestão.

#### RQ14. Em quais papéis você já atuou no desenvolvimento de software da instituição?

Gráfico 14 – Papéis de desenvolvimento de software mais utilizados nos IFs



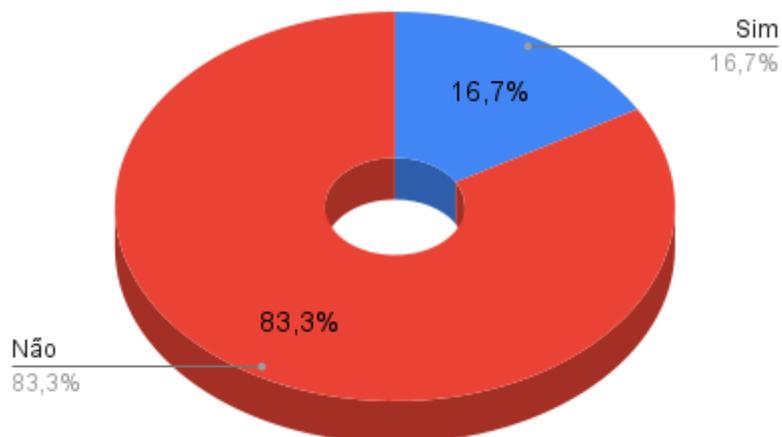
Fonte: O autor (2022).

Esta questão foi elaborada com o intuito de conhecer o perfil dos respondentes, como se trata de uma pesquisa voltada a todos os stakeholders houve 7 participantes que não participavam ativamente do desenvolvimento de software, o que prove uma taxa de 80,6% de participação.

A grande maioria dos respondentes possuem mais de um papel, geralmente sendo “Desenvolvedor” um dos papéis.

### RQ15. Possui alguma documentação de desenvolvimento de software?

Gráfico 15 – Institutos que possuem documentação de desenvolvimento de software



Fonte: O autor (2022).

Apenas 80,6% dos respondentes souberam responder à questão, destes apenas 16,7% informaram a existência de algum tipo de documentação, fato que demonstra um alto índice de despreparo para a migração de um modelo centralizado para um modelo distribuído, uma vez que uma das principais características do modelo é a disponibilidade de informações dos projetos on-line.

### RQ16. Observações

A maioria das observações trazem pontos de vistas, muito ricos em informações, apesar de serem pessoais, todas são relatadas por pessoas muito experientes, todos já atuaram em mais de três papéis.

**Resposta 27:** “90% dos projetos são desenvolvidos por mim sem ajuda.” (IFCE, Campus Fortaleza, 8 papéis)

O desenvolvimento de software centralizado possui um problema muito grande em relação ao modelo distribuído que é dificuldade de transferência de conhecimento de um indivíduo para outro, uma vez que a maioria das informações são passadas por via face a face, não deixando documentado as ocorrências.

O cenário descrito torna ainda mais crítico o problema de transferência de conhecimento, devido a apenas uma pessoa deter um conhecimento muito abrangente, esse conhecimento levaria muito tempo para ser transferido para uma

pessoa nova, o que culmina em uma parada total do processo de manutenção dos sistemas existentes durante o tempo necessário para o aprendizado.

**Resposta 5:** “A documentação dos projetos de desenvolvimento fica no Redmine (tarefas, prazos, atas, etc.) e a documentação técnica no GitLab (bugs, roteiro de instalação, etc.). Seria interessante a definição de um processo de desenvolvimento distribuído no Ifes, que conte com a participação dos servidores de TI dos campi e da Reitoria. ” **(IFES, Campus Serra, 3 papéis)**

Apesar de o respondente ter uma propensão a adoção do DDS caso exista um processo bem definido e ainda sugerir a utilização de um modelo distribuído focado no desenvolvimento de software colaborativo entre os servidores de TI dos campi e da Reitoria, a resposta descreve com riqueza de detalhes um cenário onde o campus desenvolve de forma centralizada e separada da reitoria, o que indica uma possível dificuldade de comunicação e organização em um projeto de migração para o modelo DDS.

Devido ao fato de eu, pesquisador, fazer parte IFES, sinto a necessidade de acrescentar o fato de que o cenário descrito pelo respondente é encontrado em vários outros campi e que apesar de alguns já manifestarem publicamente a vontade de atuar junto a Reitoria no desenvolvimento de software, não existe nenhuma iniciativa planejada para tal.

**Resposta 12:** “O desenvolvimento de software pode ser estratégico inclusive no âmbito da inovação e Empreendedorismo. A partir de 2014 é possível que servidores e alunos prestem serviços remunerados para instituições parceiras. Num modelo OpenSource essas equipes poderiam receber por manutenção e customizações. ” **(IFRS, Campus Sertão, 4 papéis)**

A resposta do servidor do IFRS, Campus Sertão, traz uma informação importantíssima para a perspectiva de construção de um processo de desenvolvimento de software colaborativo nos IFs, utilizar a força de trabalho de Alunos e Professores para a melhoria dos sistemas institucionais ou para a construção de projetos para a comunidade pode trazer benefícios imensuráveis, uma vez que eles estão mais próximos da necessidade que os sistemas pretendem atender.

**Resposta 16:** “A direção de TI quer novamente jogar na mão de empresa a customização, o que alguns servidores são totalmente contra. Vejo isto como uma desvalorização da qualidade e do conhecimento da equipe. ” **(IFMT, Reitoria, 3 papéis)**

O respondente da Reitoria do IFMT possui uma visão contrária a terceirização da manutenção dos softwares, porém o fator de desvalorização da qualidade e do conhecimento da equipe de desenvolvimento de software mencionado por ele é muito complexo para se tratar independente de outros fatores, como por exemplo a comunicação, organização e a transparência entre os stakeholders do projeto. A desvalorização não ocorre pelo fato de se terceirizar as tarefas, sim pelo fato de não organizar as outras demandas, que poderiam ser atribuídas aos já experientes desenvolvedores internos. Na prática a terceirização de tarefas deveria existir para fortalecer e valorizar a equipe interna, uma vez que ao terceirizar tarefas corriqueiras, estes ficariam livres para executar tarefas de maior importância, complexidade ou que por questões de segurança devem ser mantidas com acesso restrito.

Porém é prudente salientar que o processo de implantação de um modelo distribuído com apoio de terceiros, exige maior maturidade e capacidade de discursão de muitos detalhes inerente a segurança dos dados expostos aos terceiros, talvez seja o modelo de distribuição de maior complexidade, por isso cabe a necessidade de avaliar o nível de maturidade do desenvolvimento de software da instituição antes da tomada de decisão.

**Resposta 19:** “A documentação referente à projetos em desenvolvimento fica na wiki do Redmine, de acesso interno aos colaboradores” **(IFRR, Reitoria, 3 papéis)**

O respondente lotado na Reitoria do IFRR, apenas usou o campo de observações para afirmar que existe um sistema de gerenciamento de projetos (Redmine) onde é mantido toda a documentação dos projetos do instituto.

## APÊNDICE D – ENTREVISTAS NÃO ESTRUTURADAS

Este relatório apresenta a forma como foi estruturada as entrevistas realizadas para coleta de informações acerca do Desenvolvimento Distribuído de Software (DDS) nos IFs que trabalham com esse modelo, seu maior foco foi separar o tipo de desenvolvimento realizado por eles, uma vez que as informações coletadas até o momento não identificavam com clareza a forma como estes viam o modelo distribuído e como utilizavam ele na prática.

Para conseguir chegar ao objetivo proposto foi necessário realizar as entrevistas de uma forma mais aberta e dinâmica, com o intuito de deixar os participantes falarem abertamente sobre como eles desenvolvem software em seus institutos e gradualmente introduzir as perguntas chaves que demonstraram a qual categoria de DDS eles pertencem.

Isso proporcionou uma evolução natural da entrevista de acordo com o conhecimento apresentado por cada participante e possibilitou não apenas alcançar o objetivo proposto, como aumentar o conhecimento do pesquisador em relação as possibilidades de Desenvolvimento de Software.

### ***Participantes***

Os participantes foram convidados através de e-mail e as entrevistas foram agendadas de acordo com suas disponibilidades e a necessidade de maiores esclarecimentos. As datas de cada entrevista podem ser visualizadas no Quadro 14.

Quadro 14 – Entrevistas com especialistas dos IFs

Instituto	Data	Participantes
IFSP	18/03/2020	Coordenador do Desenvolvimento de Software
	10/06/2020	Coordenador do Desenvolvimento de Software
IFSULDEMINAS	18/03/2020	Coordenador do Desenvolvimento de Software
IFRR	18/04/2020	Coordenador do Desenvolvimento de Software
IFSUDESTEMG	04/05/2020	Coordenador do Desenvolvimento de Software
IFC	19/05/2020	Coordenador do Desenvolvimento de Software

Fonte: O autor (2022).

Foram realizadas 2 entrevistas com o Instituto Federal de São Paulo (IFSP), devido ao fato de ser o único instituto a apresentar uma considerável experiência em lidar com o esforço de alinhamento de equipes remotas e utilizar um Fluxo de Trabalho totalmente virtual, com auxílio de ferramentas web. Sua disponibilidade de cooperar ativamente com esta pesquisa, possibilitou o levantamento de muitas informações relevantes para o tema. Os demais apresentaram um cenário de baixa utilização do DDS, na maioria dos casos, pode ser considerado como “Centralizado, com alguns Indivíduos Remotos”, porém a carência de dados pode ter comprometido esta análise.

## **CONSIDERAÇÕES FINAIS**

As entrevistas obtiveram um resultado satisfatório, uma vez que elucidaram a experiência de cada instituto entrevistado com o modelo distribuído de desenvolvimento de software. Isso possibilitou um maior aprofundamento nos conhecimentos que experienciaram.

## **APÊNDICE E – QUESTIONÁRIO SOBRE PROCESSO DE DDS NOS INSTITUTOS FEDERAIS**

Este questionário faz parte da pesquisa desenvolvida pelo mestrando em Ciências da Computação do Centro de Informática da Universidade Federal de Pernambuco (CIn-UFPE), Welington Mothé de Oliveira, orientado pelo Professor Alexandre Marcos Lins de Vasconcelos. Não será feito nenhum tipo de identificação, responda as perguntas abaixo de acordo com a perspectiva que tem de seu local de trabalho. As perguntas que não souber responder deixe em branco ou clique na opção “Não sei informar”.

Foi construído com o objetivo de identificar as práticas de desenvolvimento de software em IFs que trabalham de forma distribuída.

A motivação deste questionário se deu durante a realização da revisão sistemática da literatura, onde encontrei muitas práticas na literatura e todas muito baseadas no mercado privado, onde percebi que era necessária a busca por uma realidade mais próxima da vivida no Instituto Federal do Espírito Santo, para tornar possível uma maior taxa de sucesso da proposta.

Foi utilizada a ferramenta Google Forms para criar e disponibilizar o questionário digital que ainda possibilitou a tabulação dos dados levantados, ele ficou disponível para respostas entre 18/03/2020 e 30/03/2020 e obteve 6 respostas.

As pessoas questionadas neste formulário são servidores do Instituto Federal de São Paulo que trabalham dentro dos critérios do Desenvolvimento Distribuído de Software (DDS) e que conhecem o desenvolvimento de software de sua instituição a fundo, do ponto de vista de um membro da equipe de desenvolvimento.

### **QUESTÕES**

As questões foram divididas em três seções, a primeira com três perguntas de identificação do respondente, a segunda com três perguntas sobre a “Diferenças entre os modelos” e a terceira com quatro perguntas referente ao “Desenvolvimento Distribuído de Software”.

## **SEÇÃO DE IDENTIFICAÇÃO**

### **Q1. Instituição**

- IF-Farroupilha
- IF-São Paulo
- IF-Roraima
- IF-Catarinense
- IF-Sul de Minas
- IF-Sudeste de Minas Gerais
- IF-Minas Gerais
- IF-Mato Grosso

### **Q2. Em quais papéis você já atuou no desenvolvimento de software da instituição?**

- Desenvolvedor
- Analista de Projeto
- Administrador de Banco de Dados
- Gerente de Projeto
- Líder de Projeto
- Testador
- Atendimento e Suporte
- Outros [O respondente pode escrever]

### **Q3. Você estava presente na transição do modelo centralizado para distribuído?**

- Sim
- Não

## **SEÇÃO DE DIFERENÇAS ENTRE OS MODELOS**

### **Q4. Quais foram as dificuldades da transição?**

Campo de texto aberto.

**Q5. Quais diferenças foram notadas entre o formato Centralizado e Distribuído?**

Campo de texto aberto.

**Q6. Se pudesse escolher um caminho hoje, por qual dos dois modelos optaria?**

- Desenvolvimento Distribuído de Software
- Desenvolvimento Centralizado na Reitoria

## **SEÇÃO DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE**

**Q7. Qual sua avaliação a respeito do modelo Distribuído de Desenvolvimento de Software?**

Escala de 1 a 10

**Q8. Quais foram os benefícios percebidos na utilização deste modelo?**

- Melhoria no entendimento da operacionalização dos processos dos campi
- Equipe maior e profissional com melhor bem-estar
- Estar próximo do usuário final de diversos setores
- Outros [O respondente pode escrever]

**Q9. Quais foram os desafios encontrados na utilização deste modelo?**

- Falha na comunicação
- Atraso de comunicação
- Demasiada formalização dos processos, documentos e comunicação.
- Desconforto com os meios de comunicação
- Mal-entendidos
- Sensação de estar trabalhando juntos, mas não necessariamente como uma equipe unificada
- Falta de Confiança
- Equipe dispersa não tomar decisões
- Burocracia na tomada de decisões
- Política interna

- Outros [O respondente pode escrever]

**Q10. Quais práticas foram adotadas?**

- Identificar a maturidade da comunicação
- Implementar uma estratégia de comunicação
- Realizar reuniões presenciais entre equipes dispersas
- Incentivar o aumento da frequência de comunicação
- Incentivar métodos de comunicação informal
- Incentivar a comunicação pessoal promovendo estreitamento de relações
- Reter tarefas que exigem comunicação frequente entre grupos dentro de equipes Co-localizadas.
- Utilização de comunicação síncrona e assíncrona
- Estabelecer pontos de interface de comunicação entre os membros da equipe
- Definir a capacidade e a produtividade potencial das equipes
- Designar uma pessoa para responder as perguntas das equipes dispersas
- Determinar tarefas complexas para as equipes Co-localizadas
- Utilizar ferramentas wikis para documentação
- Entendimento comum sobre como cada uma das partes toma decisões
- Estabelecer um horário prolongado para conversar com equipes dispersas
- Fornece e repetir instruções detalhadas com antecedência
- Garantir a existência de uma infraestrutura, processos e procedimentos de gerenciamento adequados para ajudar a estabelecer cooperação e coordenação entre os locais
- Sincronizar horas de trabalho
- Planejar e acordar marcos alcançáveis
- Incentivar o relato de capacidade de execução de determinada tarefa no tempo estimado
- Incentivar o relato de problemas que possam vir a ocorrer
- Monitorar projetos com referência a custo, tempo, produtividade, qualidade e risco
- Estabelecer relatórios formais regulares para ajudar o gerente de projetos a permanecer ciente de como o projeto está progredindo
- Planejar e comentar reuniões sprint

- Planejar tarefas (iterações) curtas
- Reuniões diárias com equipes distantes separadas da Co-localizadas
- Realizar entregas (builds) frequentes e contínuas
- Determinar a abordagem para alocação de tarefas entre locais
- Identificar tarefas de gerenciamento de projetos
- Estabelecer procedimentos de cooperação e coordenação entre locais
- Estabelecer procedimentos de relatório entre locais
- Definir gerenciamento entre locais
- Identifique metas, objetivos e recompensas
- Estabelecer e manter colaborativamente limites de propriedade de produtos de trabalho
- Estabelecer e manter colaborativamente interfaces e processos
- Desenvolver, comunicar e distribuir planos de trabalho em colaboração
- Os requisitos regulamentares de controle de qualidade são altos, o que implica encargos adicionais para documentação e testes.
- Adoção de métodos ágeis
- Ferramentas colaborativas
- Jogos sérios educativos
- Delegar trabalho em pacotes de trabalho
- Outros [O respondente pode escrever]

### ***Síntese das Respostas***

Este questionário contou com a participação de 6 respondentes da força de desenvolvimento de software do Instituto Federal de São Paulo (IFSP), único instituto dentre os que obtive a oportunidade de interagir que apresentou todas as características de Desenvolvimento Distribuído de Software, assim sendo, recomenda-se buscar outras fontes de informações para possibilidade de comparação com os dados coletados.

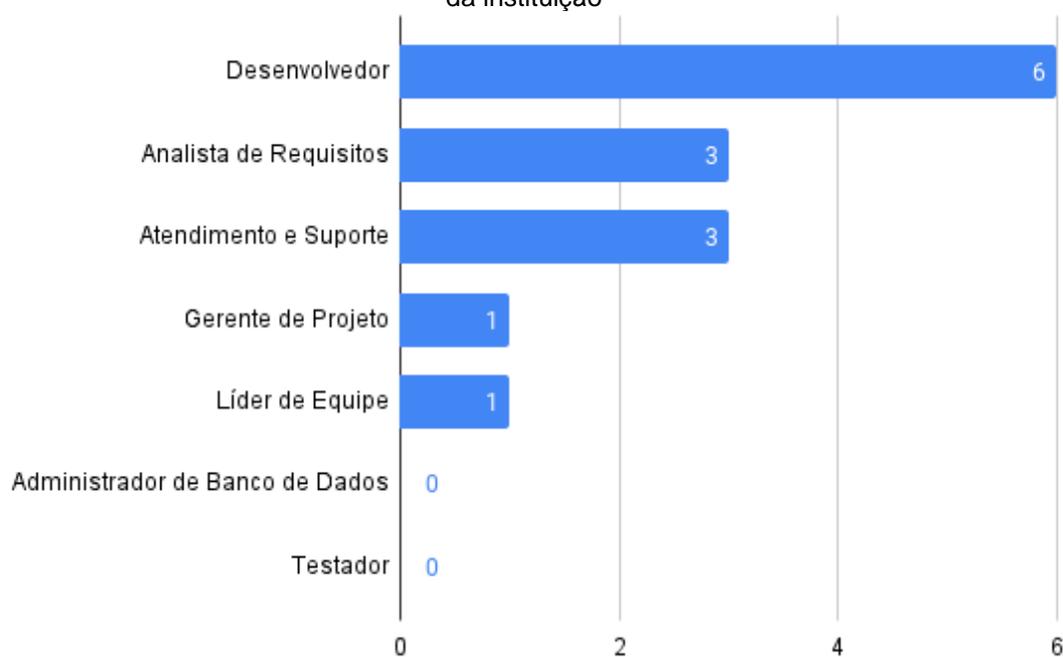
## SEÇÃO DE IDENTIFICAÇÃO

### RQ1. Instituição

Apenas o Instituto Federal de São Paulo participou deste questionário.

### RQ2. Em quais papéis você já atuou no desenvolvimento de software da instituição?

Gráfico 16 – Papéis em que os respondentes já atuaram no contexto do desenvolvimento de software da instituição

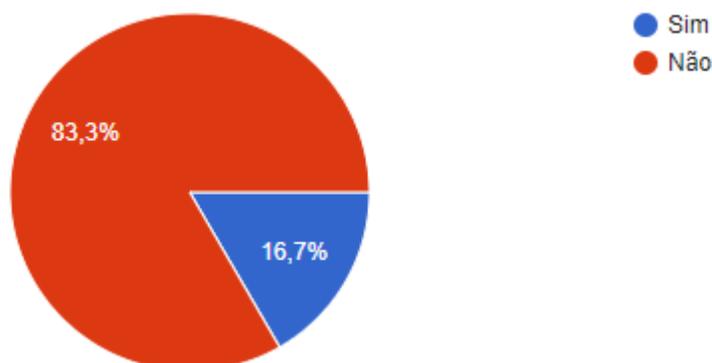


Fonte: O autor (2022).

Foi identificado que todos respondentes são desenvolvedores, 50% são Analista de Projeto e 50% realizam atendimento e suporte diretamente ao demandante.

### RQ3. Você estava presente na transição do modelo centralizado para distribuído?

Gráfico 17 – Presentes na transição do modelo centralizado para distribuído



Fonte: O autor (2022).

Dentre os 6 respondentes, apenas o coordenador do setor de desenvolvimento estava presente na transição entre os modelos, essa resposta leva automaticamente a seção de “Diferenças entre os modelos”.

## **SEÇÃO DE DIFERENÇAS ENTRE OS MODELOS**

Esse questionário tentou avaliar a percepção dos respondentes sobre a diferença entre os dois modelos identificando quem havia participado da transição com a pergunta “Você estava presente na transição do modelo centralizado para distribuído?”, mas apenas um participante esteve presente durante a transição, sendo a única resposta das questões referentes a transição, todas as respostas serão apresentadas de forma direta em forma de parágrafos.

Importante ressaltar que apesar de ser o único respondente que estava presente na migração do modelo centralizado para distribuído, este é o atual coordenador do setor e já atuou em 5 papéis.

### **RQ4. Quais foram as dificuldades da transição?**

As dificuldades da transição foram organizar as demandas com a equipe e o nivelamento de conhecimento.

### **RQ5. Quais diferenças foram notadas entre o formato Centralizado e Distribuído?**

As diferenças notadas entre o formato Centralizado e Distribuído foram o crescimento de conhecimento da equipe e a evolução da utilização de ferramentas de

gerenciamento e desenvolvimento de software.

#### RQ6. Se pudesse escolher um caminho hoje, por qual dos dois modelos optaria?

Se pudesse escolher um caminho hoje, ele optaria pelo Desenvolvimento Distribuído de Software.

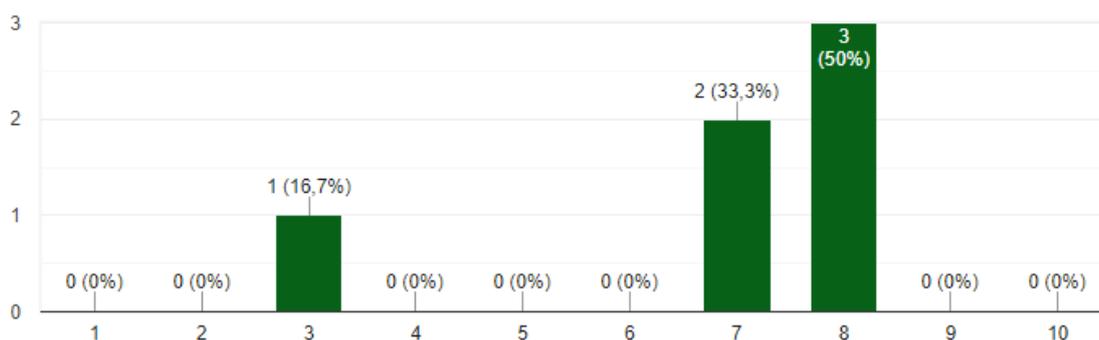
Suas respostas apresentam uma certa confiança de que o modelo Desenvolvimento Distribuído de Software seja melhor que o modelo de Desenvolvimento Centralizado de Software, apresenta dificuldades comum a qualquer transição e utiliza palavras como “crescimento” e “evolução” ao se referir a diferença dos dois.

Apesar da riqueza de informações apresentadas, a experiência de apenas um respondente é pouco para qualquer conclusão, é necessário a busca de outras fontes de informações para obter uma melhor percepção a respeito da migração dos modelos.

### SEÇÃO DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

#### RQ7. Qual sua avaliação a respeito do modelo Distribuído de Desenvolvimento de Software?

Gráfico 18 – Avaliação do modelo DDS no IFSP

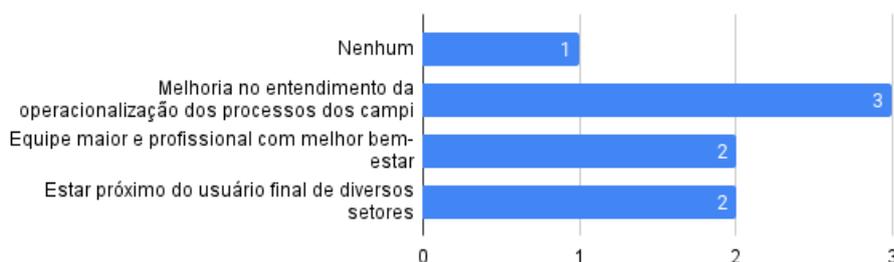


Fonte: O autor (2022).

Na questão de modelo de desenvolvimento de software utilizado nos IFs foi encontrado uma taxa de 64,4% utilizando o modelo Centralizado e apenas 22,2% o modelo Distribuído, deixando claro que a maioria ainda se encontra trabalhando de forma Centralizada em suas Reitorias.

### RQ8. Quais foram os benefícios percebidos na utilização deste modelo?

Gráfico 19 – Benefícios percebidos na utilização do DDS



Fonte: O autor (2022).

De acordo com os respondentes foram citados três benefícios, são eles:

- Melhoria no entendimento da operacionalização dos processos dos campi
- Equipe maior e profissional com melhor bem-estar
- Estar próximo do usuário final de diversos setores

Houve ainda um dos respondentes que diz que não existe nenhum benefício no modelo DDS, o mesmo que avaliou com uma pontuação baixa o modelo na RQ7, infelizmente não havia um campo de texto aberto para permitir que o respondente expressasse a motivação do descontentamento com o DDS.

### RQ9. Quais foram os desafios encontrados na utilização deste modelo?

Gráfico 20 – Desafios encontrados na utilização do DDS



Fonte: O autor (2022).

Foram levantados 10 desafios do DDS e 7 deles foram indicados pela força de desenvolvimento de software do IFSP. O fator mais relevante desta questão é que a maioria dos desafios estão relacionados a comunicação, inclusive os 2 mais citados que são o “Atraso de comunicação” e a “Falha na comunicação”.

### RQ10.Quais práticas foram adotadas?

Gráfico 21 – Práticas adotadas pelo IFSP



Fonte: O autor (2022).

As práticas tendem a ser adotadas em consonância com os desafios enfrentados, talvez devido a isso, assim como a questão anterior as práticas mais citadas estão relacionadas a comunicação, inclusive a mais citada entre elas “Incentivar o aumento da frequência de comunicação”.

## APÊNDICE F – ROTEIRO PARA CONDUÇÃO DO GRUPO FOCAL

Este roteiro foi utilizado para aplicação do Grupo Focal, com objetivo de avaliar a proposta desta pesquisa, uma Metodologia de Desenvolvimento Distribuído de Software para os Institutos Federais de Educação do Brasil. Foi elaborado com o propósito de alinhar as expectativas de todos os envolvidos e publicizar um passo a passo a ser seguido, reduzindo potenciais vieses.

Ele é estruturado através de Papéis, Perguntas norteadoras e Etapas planejadas.

### ***Papéis***

Os papéis utilizados na condução do Grupo Focal foram os seguintes:

- O **moderador**, que é responsável por alinhar o conhecimento dos participantes antes do debate e provocar um debate entre os participantes abordando as questões relacionadas à pesquisa realizada, previstas na seção Perguntas Norteadoras, com base no roteiro planejado. Fundamentalmente o moderador deve apenas captar a discursão sem inferir com opiniões, sua única participação no discurso deve ser para manter o foco dos participantes nas perguntas norteadoras.
- O **participante**, que são as pessoas que foram selecionadas para discutir as questões propostas, eles devem se limitar ao tema abordado com o propósito de provocar o foco nas perguntas norteadoras, sua principal função é fornecer dados de forma simples e descontraída, como em uma conversa informal com amigos.

No contexto desta pesquisa, espera-se que os participantes além de discutir e avaliar o produto apresentado, proponham melhorias ao produto que possibilitem um refinamento da proposta por parte do pesquisador.

### ***Perguntas norteadoras***

Esta seção apresenta as principais questões a serem abordadas.

- O modelo proposto faz sentido?
- O modelo proposto atende às necessidades dos Institutos Federais?
- Considera que seja possível aplicar a metodologia em sua instituição?
- Algum fator impediria a aplicação desta metodologia em seu instituto?

### ***Etapas Planejadas***

Para a **primeira etapa**, espera-se que sete dias antes da data estipulada seja enviado o convite para todos os Desenvolvedores de Software dos IFs do Brasil, o convite deve contar com um link de seleção de horários disponíveis em uma ferramenta de agendamento de eventos por meio de votação e um link de grupo de WhatsApp para acompanhar as informações sobre o Grupo Focal. Quando confirmado o horário, deve ser novamente comunicado a todos os participantes o horário definido, junto ao link de acesso da sala de webconferência.

Para a **segunda etapa**, espera-se que, após a entrada de todos os participantes, seja explicitado o objetivo do grupo focal e o fato de que a conversa deverá ser gravada para posterior consulta e transcrição, assim inicia-se a apresentação da proposta.

Para a **terceira etapa**, espera-se que, todos os participantes exponham suas perspectivas sobre o tema seguindo as perguntas norteadoras, enquanto o moderador participa apenas tirando dúvidas e conduzindo a discussão através das perguntas norteadoras.