

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

WILLIAM SIMÃO DE DEUS

**SOLUÇÃO DE SOFTWARE PARA O AUXÍLIO NA GESTÃO DE
PRODUÇÃO EM PROJETOS DE DESENVOLVIMENTO
DISTRIBUÍDO**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

WILLIAM SIMÃO DE DEUS

**SOLUÇÃO DE SOFTWARE PARA O AUXÍLIO NA GESTÃO DE
PRODUÇÃO EM PROJETOS DE DESENVOLVIMENTO
DISTRIBUÍDO**

Trabalho de Conclusão de Curso apresentada à disciplina de Trabalho de Conclusão de Curso II, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná como requisito para obtenção do título de tecnólogo

Orientador: Prof. Dr. Alexandre L'Erario

CORNÉLIO PROCÓPIO

2014

Dedico este trabalho aos meus pais, Valmor Simão de Deus e Edelzina da Aparecida Baptista de Deus.

AGRADECIMENTOS

Agradeço primeiramente à Deus, que sempre esteve comigo em todas as etapas da minha vida.

Ao meu orientador, prof. Dr. Alexandre L'Erario que sempre me norteou em diversos momentos da graduação, desde o período de iniciação científica até o conclusão deste trabalho.

Ao prof. José Antônio e aos meu amigos Luan, Vitor, Igor, Lecir e Noda que me auxiliaram imensamente para a conclusão deste trabalho.

Aos meus colegas e familiares, que de alguma forma contribuíram construtivamente.

A minha irmã, Adeline Simão de Deus por oferecer atenção e um norte durante as nossas conversas. Muito obrigado.

Aos meus pais, Valmor Simão de Deus e Edelvina Aparecida Baptista de Deus por terem me apoiado na realização de mais essa etapa. Muito obrigado.

”Tenha talento, trabalhe como um condenado, sue sangue, e você conseguirá tudo sem esforço.” - Humberto Gessinger.

RESUMO

SIMÃO, William. SOLUÇÃO DE SOFTWARE PARA O AUXÍLIO NA GESTÃO DE PRODUÇÃO EM PROJETOS DE DESENVOLVIMENTO DISTRIBUÍDO 63 f. Trabalho de Conclusão de Curso – TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS , Universidade Tecnológica Federal do Paraná . Cornélio Procópio, 2014.

Este trabalho apresenta o desenvolvimento de uma solução de software para auxiliar o gerenciamento de produção em um ambiente distribuído. É exposto o contexto geral sobre características que permeiam este ambiente, entre benefícios e desafios de sua aplicação. São apresentados os principais focos de estudo sobre o gerenciamento de produção, que trouxeram qualidade ao desenvolvimento do produto final. Complementando com a definição de tecnologias e ferramentas utilizadas para seu desenvolvimento, bem como os processos que foram ser aplicados. Também é apresentado o método utilizado para a sua validação e a coleta sobre os resultados gerados. O produto final deste trabalho auxilia a coordenação de atividades, organização de equipes e papéis no ambiente de desenvolvimento distribuído.

Palavras-chave: Desenvolvimento Distribuído de Software, Gestão de Produção, Processo de Produção

ABSTRACT

SIMÃO, William. SOFTWARE SOLUTION FOR AID IN PRODUCTION MANAGEMENT FOR DISTRIBUTED DEVELOPMENT PROJECTS. 63 f. Trabalho de Conclusão de Curso – TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS , Universidade Tecnológica Federal do Paraná . Cornélio Procópio, 2014.

This document presents the development of a software solution to assist production management in a distributed environment. It exposed the general context of characteristics that permeate this environment, between benefits and challenges of your application. Are the main focus of study on production management, which brought quality to the development of the final product. Complementing the definition of technologies and tools used for its development and the processes that were being applied. It also presents the method used for its validation and the collection of the results generated. The final product of this work helps the coordination of activities, organization of teams and roles in distributed development environment.

Keywords: Development Distributed Software, Process of Production, Management of Production

LISTA DE FIGURAS

FIGURA 1	– Relação entre empresas e distância geográfica sobre o DDS	16
FIGURA 2	– Esfera de ação dos diversos profissionais nos processos decisórios	18
FIGURA 3	– Arquitetura do Sistema	23
FIGURA 4	– Detalhamento da arquitetura do Sistema	23
FIGURA 5	– Esquematização do Processo de Desenvolvimento Scrum	25
FIGURA 6	– Relação de Atores do Sistema	30
FIGURA 7	– Diagrama de Atividades Referente ao Fluxo de Trabalho	31
FIGURA 8	– Caso de Uso Geral	32
FIGURA 9	– Diagrama de Classes Bean	35
FIGURA 10	– Diagrama de Classes Dao	36
FIGURA 11	– Ambiente de Versionamento	38
FIGURA 12	– Diagrama de Sequência - Cadastro de Processos de Desenvolvimento ..	40
FIGURA 13	– Diagrama de Sequência - Cadastro de Processos de Desenvolvimento ..	41
FIGURA 14	– Diagrama de Sequência - Registro de <i>Log</i>	42
FIGURA 15	– Interface de Cadastro de Ciclo de Vida	62
FIGURA 16	– Interface de <i>Log's</i>	62
FIGURA 17	– Interface de Relatório	63

LISTA DE TABELAS

TABELA 1	– Manifesto CHAOS, entre os anos de 2004 - 2012	12
TABELA 2	– Comparação entre trabalhos relacionados e a solução desenvolvida	20
TABELA 3	– Maturidade do Projeto	26
TABELA 4	– Cronograma do Projeto	28
TABELA 5	– Configuração da Equipe Responsável pela Validação	45
TABELA 6	– Requisitos Funcionais	54
TABELA 7	– Requisitos Não Funcionais	55
TABELA 8	– Ata da Reunião 1	57
TABELA 9	– Ata da Reunião 2	57
TABELA 10	– Ata da Reunião 3	58
TABELA 11	– Ata da Reunião 4	58
TABELA 12	– Ata da Reunião 5	59
TABELA 13	– Ata da Reunião 6	59
TABELA 14	– Ata da Reunião 7	59
TABELA 15	– Ata da Reunião 8	60

LISTA DE SIGLAS

DDS	Desenvolvimento Distribuído de Software
EAP	Estrutura Analítica do Projeto
API	<i>Application Programming Interface</i>
JPA	Java <i>Persistence</i> API
CSS	Cascading Style Sheets
XHTML	<i>Extensible HyperText Markup Language</i>
HTML	HyperText Markup Language
HTTP	<i>Hypertext Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
JSF	<i>JavaServer Faces</i>
IDE	<i>Integrated Development Environment</i>
UML	<i>Unified Modeling Language</i>
MVC	<i>Model View Controller</i>
PO	Product Owner
CRUD	<i>Create Read Update e Delete</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	CONTEXTO GERAL	11
1.2	OBJETIVO	12
1.2.1	Pontos Significantes	12
1.3	ORGANIZAÇÃO DO TEXTO	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE	16
2.2	PROCESSO DE PRODUÇÃO	17
2.3	GESTÃO DE PRODUÇÃO	18
2.4	TRABALHOS RELACIONADOS	19
3	METODOLOGIA	21
3.1	TECNOLOGIAS	21
3.2	FERRAMENTAS	22
3.3	ARQUITETURA DO SISTEMA	23
3.4	PROCESSO DE DESENVOLVIMENTO	24
4	EXECUÇÃO DO PROCESSO	26
4.1	DESENVOLVIMENTO	26
4.1.1	Primeira <i>Sprint</i>	27
4.1.2	Segunda <i>Sprint</i>	29
4.1.3	Terceira <i>Sprint</i>	37
4.1.4	Quarta <i>Sprint</i>	39
4.1.5	Quinta <i>Sprint</i>	41
4.1.6	Sexta <i>Sprint</i>	43
5	RESULTADOS E DISCUSSÕES	44
5.1	VALIDAÇÃO	44
5.2	RESULTADOS	45
5.3	DISCUSSÕES	46
6	CONCLUSÃO	48
6.1	TRABALHOS FUTUROS	49
	REFERÊNCIAS	50
	Apêndice A – DOCUMENTO DE REQUISITOS	53
	Apêndice B – ATAS DAS REUNIÕES	56
	Apêndice C – INTERFACES DO SISTEMA	61

1 INTRODUÇÃO

Este capítulo apresenta a conceituação teórica sobre o Desenvolvimento Distribuído de Software (DDS), bem como as características de sua aplicação, os problemas encontrados em torno deste ambiente, a justificativa do tema elucidado e os pontos significantes que o desenvolvimento deste trabalho proporcionou.

1.1 CONTEXTO GERAL

O DDS ocorre quando equipes e/ou empresas cooperam e colaboram para desenvolver um mesmo projeto ou parte dele (L'ERARIO, 2009). O desenvolvimento distribuído é uma estratégia que procura evidenciar redução de tempo e de gastos econômicos.

Segundo Oliveira et al. (2013), a adoção do DDS é uma realidade aplicada em grandes empresas e não somente uma tendência. Os engenheiros de software estão reconhecendo a grande influência dessa forma de trabalho e buscam modelos que facilitem o desenvolvimento com equipes geograficamente distantes (AUDY; PRIKLADINICKI, 2007)

O ambiente de desenvolvimento distribuído envolve uma série de desafios. Os projetos em ambientes DDS apresentam falhas em aspectos sociais (na comunicação) e técnicos (de documentação). Estas falhas ocorrem em função da distância física e dispersão temporal das equipes/sites alocados em projeto.

A Tabela 1 apresenta uma pesquisa realizada pelo *Standish Group* sobre os riscos em projetos de desenvolvimento de software. Esta pesquisa organiza a classificação dos projetos analisados em: bem sucedido (concluído dentro do prazo e no custo estipulado), fracassados (cancelados antes de serem concluídos) e desafiadores (concluídos, com gastos superiores ao previsto e com prazo maior).

Tabela 1: Manifesto CHAOS, entre os anos de 2004 - 2012

Status	2004	2006	2008	2010	2012
Bem sucedidos	29%	35%	32%	37%	39%
Fracassados	18%	19%	24%	21%	18%
Desafiadores	53%	46%	44%	42%	43%

Fonte: Group (2014)

A Tabela 1 constata que menos da metade dos projetos são iniciados e concluídos de forma bem sucedida. Os projetos nomeados desafiadores obtiveram menor incidência desde 2004, contudo, ocorrem com frequência e superam 40% do mercado de desenvolvimento.

Alguns motivos que justificam as falhas no desenvolvimento decorrem da falta de compreensão dos requisitos do sistema e da finalidade do projeto. Estas falhas ganham proporções ainda maiores em ambientes DDS, devido à dispersão de equipes e empresas, (L'ERARIO, 2009).

Visando reverter ou diminuir as falhas no desenvolvimento de projetos são utilizadas soluções que oferecem maior exequibilidade ao DDS. Segundo Marckzak (2013), muitas soluções se baseiam em características específicas, como a comunicação, deixando de lado outros fatores, como a preocupação com coesão e consistência de informações, em todas fases de desenvolvimento de um projeto DDS.

1.2 OBJETIVO

Este trabalho apresenta como objetivo disponibilizar uma solução de software para auxiliar o gerenciamento da produção em um ambiente de desenvolvimento distribuído. Esta solução permite minimizar os desafios sobre a produção em ambientes distribuídos, oferecendo ao gerente de um projeto DDS opções de organização e gerenciamento sobre a produção de colaboradores e equipes distribuídas utilizando atividades e artefatos produzidos.

1.2.1 PONTOS SIGNIFICANTES

O desenvolvimento deste trabalho procurou reduzir desafios referentes ao gerenciamento de produção no ambiente DDS. A utilização da solução final oferece ao Administrador de um projeto DDS a personalização sobre o processo de desenvolvimento, organizando fases e atividades que devem ser desenvolvidas. Além disso, permite verificar o fluxo de trabalho atual, autenticando Gerentes responsáveis por monitorar cada equipe do projeto.

Os gerentes autenticados em um projeto possuem a capacidade de organizar quais

atividades serão desenvolvidas e designar os respectivos colaboradores responsáveis pela sua execução. Também podem supervisionar sua equipe a partir do controle individual de produção, verificando se existe uma sobrecarga ou folga de trabalho para determinados colaboradores.

Já os Colaboradores alocados em uma equipe podem verificar as suas atividades, consultar suas especificações e ao fim deste processo, enviar o artefato produzido. Desta forma, este trabalho enfatizou os seguintes pontos:

- **Ciclo de Vida:** Capacidade de personalização de um ciclo de vida (processo de desenvolvimento) adequando-se a realidade de cada projeto. Permitindo a construção de uma estrutura com fases, atividades e artefatos envolvidos no desenvolvimento deste, abrangendo a vida do sistema, desde a definição de requisitos até a conclusão do desenvolvimento.
- **Gerenciamento de Fases:** Capacidade em organizar todas as etapas do desenvolvimento de um projeto DDS. Oferecendo auxílio durante a concepção, análise, desenvolvimento, teste, implantação e manutenção de um projeto.
- **Gerenciamento de Atividades:** Eficiência para definir em cada Fase uma coleção de Atividades. Aonde cada atividade possui um conjunto de informações para auxiliar o seu desenvolvimento, gerando tarefas para colaboradores e artefatos para o projeto.
- **Gerência de Equipe:** Permite a divisão hierárquica dos colaboradores do projeto. Oferecendo uma estrutura de permissão e monitoramento sobre a produção individual de cada equipe lotada em um projeto. Com essa divisão, um Administrador pode controlar e verificar o fluxo total do projeto, enquanto os Gerentes coordenam atividades e artefatos em sua equipe.
- **Gerência de Colaboradores:** Capacidade para autenticar somente os colaboradores que participam do desenvolvimento do projeto. Permitindo o escalonamento de equipes e gerentes.
- **Gerência de Papéis:** Possibilidade em alocar os colaboradores em funções diferentes em cada equipe de um projeto.
- **Gerenciamento de Projetos:** Permite a elaboração de um planejamento sobre um objetivo particular de cada projeto. Desta forma, é possível mensurar dados técnicos, como o controle de colaboradores, equipes e gerentes alocados neste. Além de oferecer informações relevantes, tal como o marco inicial do projeto, total de dias gastos e possibilidade em

verificar atividades que ficaram ociosas ou não foram desenvolvidas, destacando a capacidade de otimizar o ciclo de vida para projetos futuros.

- Gerência de Status de Projeto: Capacidade em controlar o estado atual do projeto. Contendo possibilidades que englobam a conclusão, o desenvolvimento ou o cancelamento deste.
- Registro de Atividades: Capacidade para que cada colaborador verifique as atividades que foram alocadas para si, contendo instruções para a sua realização e opção para o envio de artefatos para os Gerentes alocados no projeto.
- Controle de Qualidade e Status de Atividades: Permite que os Gerentes verifiquem a qualidade de cada Artefato desenvolvido pelos colaboradores de sua equipe. Além disso, permite verificar se os Artefatos apresentam um resultado satisfatório, modificando o status da Atividade para concluída ou em desenvolvimento.
- Acompanhamento de *Log's*: Possibilidade em verificar o desenvolvimento individual em relação as atividades de cada colaborador. Estimando informações relevantes sobre a atividade desenvolvida, se foi produzido algum artefato, o tempo gasto e total de dias trabalhados.
- Relatórios: Opção de gerar relatórios referentes a produção no ambiente DDS. Os relatórios apresentam informações sobre a produção de cada colaborador alocado em uma equipe, desta forma, são associados os dados referentes ao desenvolvimento de atividades, do tempo utilizado para a realização de uma tarefa tarefa e do total dias gastos.

1.3 ORGANIZAÇÃO DO TEXTO

Este trabalho encontra-se estruturada em seis capítulos. Neste primeiro capítulo foi apresentado o contexto geral referente aos problemas e ao objetivo deste trabalho, por fim são demonstrados os pontos significantes que o seu desenvolvimento atingiu. No segundo capítulo é apresentado a conceituação teórica sobre o DDS, os processos, o gerenciamento de produção e a pesquisa referente aos trabalhos similares. No terceiro capítulo estão concentradas as pesquisas referente aos métodos e processos utilizados para o desenvolvimento da solução final. No quarto capítulo é apresentada uma descrição sobre o processo de desenvolvimento adotado, as atividades e artefatos produzidos. No quinto capítulo é exposta a forma como ocorreu a validação desta solução, apresentando os resultados finais coletados. No último capítulo estão

apresentados os pontos referentes ao conhecimento gerado por este projeto e a expectativa sobre trabalhos futuros.

Ao fim deste documento encontram-se os apêndices referentes aos artefatos produzidos e o registro de cada reunião realizada. Cabe salientar que todas as Figuras utilizadas para a elaboração deste trabalho que possuem autoria própria não foram referenciadas. De modo que somente imagens ou tabelas referente aos trabalhos de outros autores foram creditadas para a consulta.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica sobre os principais tópicos abordados para aumentar a qualidade do produto final deste trabalho. São apresentados conceitos referentes as características sobre o DDS, Processos de Produção e a Gestão de Produção.

2.1 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

O DDS é uma realidade aplicada em diversas organizações tecnológicas, fruto da globalização entre o relacionamento de empresas, que acaba gerando um ambiente com novos e grandes desafios para o processo de desenvolvimento (AUDY; PRIKLADINICKI, 2007).

A aplicação do DDS pode ocorrer a partir da classificação entre o relacionamento de empresas ou a partir da distância geográfica. A Figura 1 ilustra este cenário.

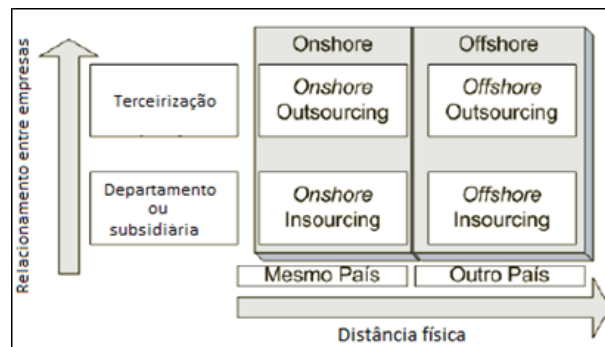


Figura 1: Relação entre empresas e distância geográfica sobre o DDS.

Fonte: (ROBINSON M, 2009)

Conforme é apresentado na Figura 1, o DDS pode ser organizado a partir de dois modelos. No modelo *Onshore* ocorre quando empresas que desenvolvem um mesmo projeto estão localizadas no mesmo país, já no modelo *Offshore* as empresas estão localizadas em países diferentes. No *Offshore* ocorre maior dificuldade em gerenciar projetos devido à dispersão, porém evidencia a redução de custos e despesas, além de maximizar a eficiência operacional (ROBINSON M, 2009).

O *Outsourcing* (ou terceirização) ocorre quando uma empresa delega atividades para serem desenvolvidas em outras empresas. No *Insourcing* (ou departamento/subsidiário) as atividades são delegadas para a mesma empresa, porém em setores diferentes.

O DDS permite que diferentes equipes, sites ou empresas dispersas geograficamente ou temporalmente, desenvolvam um mesmo projeto, evidenciando o ganho de benefícios econômicos, produtivos e qualitativos. Contudo, a aplicação apresenta diversos riscos, segundo Prikladnicki et al. (2010) podem ocorrer problemas na coordenação de equipes, divisão de tarefas, identificação de papéis e responsabilidades.

Demais conflitos e desafios decorrem no gerenciamento e monitoramento das atividades entre empresas ou equipes; falta de processos e regras comuns; comunicação; e infraestrutura de suporte as atividades de cada equipe (OLIVEIRA et al., 2013).

2.2 PROCESSO DE PRODUÇÃO

O processo de produção representa um conjunto de operações relacionadas ao processo de fabricação de um produto. Segundo Oliveira (2003), o processo de produção deve ocorrer com qualidade, procurando identificar o quanto antes não conformidades no desenvolvimento de um bem, produto ou serviço.

Existem dois focos no processo de produção, o simples (somente um produto é produzido) e o múltiplo (em que diversos produtos são produzidos). Segundo Luder et al. (2013), ambos possuem um gerenciamento complexo devido aos diversos métodos de eficiência que devem ser abordados.

Os métodos de eficiência procuram evidenciar a menor aplicação possível de materiais para a produção de um produto, (VASCONCELLOS; GARCIA, 2004). Sendo assim, o custo e o tempo para a sua produção serão menores, porém deve-se atentar para a qualidade final apresentada.

Segundo Carvalho e Paladine (2005), o conceito de qualidade em um produto deve ser compreendido como oferecimento de valor ao cliente e deve reforçar a posição competitiva da empresa em seu mercado de atuação. Os processos de produção devem trabalhar em conjunto com a gestão de qualidade (CARVALHO; PALADINE, 2005).

2.3 GESTÃO DE PRODUÇÃO

A gestão de produção é uma atividade que atinge todos os ramos das entidades industriais, comerciais e de serviços. Ela é definida como todo conjunto de recursos organizados para obter produtos ou serviços de modo sistemático (MIGUEL, 2012). Subentende-se que a gestão de produção é uma forma de organizar todos os recursos (artefatos, matéria-prima, serviço, etc) utilizados para produção de determinado produto ou serviço, de forma que ocorra o menor índice de perda possível.

De acordo com Paladini (1995), a gestão de produção controla a conformidade entre as especificações do projeto, atributos desejáveis no produto e o valor final oferecido. Também se verifica a necessidade de tratar de forma conjunta e simultânea o custo, a flexibilidade, a quantidade, o atendimento, o tempo de resposta e a inovação em um produto (ANTUNES, 2008). Em decorrência destes agentes, a esfera de decisões inerente ao trabalho de profissionais de gestão de produção é ampla, ilustrada na Figura 2.

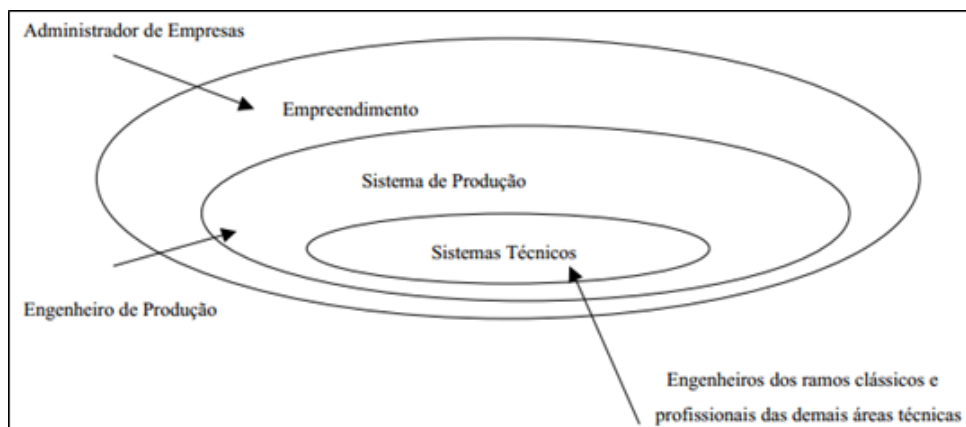


Figura 2: Esfera de ação dos diversos profissionais nos processos decisórios.

Fonte: (CUNHA, 2002)

A ênfase deste trabalho diz respeito aos Sistemas de Produção, com abordagem na área de controle e planejamento agregada da produção. Segundo Goulart (2000), esta área é responsável pelos níveis gerais e a capacidade de produção por um período de tempo. Para tal mensuração, serão utilizados métodos e técnicas organizacionais sobre o desenvolvimento e execução de atividades, que procuram aperfeiçoar o emprego dos recursos existentes no próprio sistema de produção (MIGUEL, 2012).

2.4 TRABALHOS RELACIONADOS

Os trabalhos relacionados contemplam soluções dispostas atualmente que procuram resolver problemas similares deste trabalho, porém não existem sistemas que possam auxiliar completamente a aplicação do DDS, sobretudo da sua gerência de produção. Desta forma existem soluções que contemplam apenas algumas perspectivas. Tal como o *OpenProject*¹, que é uma ferramenta com o foco na colaboração e gestão de atividades em projetos de software. A sua utilização se baseia na criação de Gráficos de Gantt, gráficos PERT e a esquematização da estrutura analítica do projeto (EAP). Desta forma, são apresentados de maneira horizontal as diferentes atividades e os responsáveis pelo seu desenvolvimento. Porém, o *OpenProject* não permite a utilização de um Processo de Desenvolvimento, que organize as diferentes fases que um projeto possui. Com isso, a gerência de um projeto a partir de outros fatores acaba subutilizada.

Além deste, existe o *Bug Tracking System*, utilizado para procurar erros e organizar a manutenção na execução de atividades (ZIMMERMANN et al., 2009). Segundo os autores, este sistema serve como um repositório central de informações. Sua principal função é providenciar de forma centralizada uma visão sobre as diferentes questões do desenvolvimento de projeto. Hodiernamente, este sistema ganhou um novo conceito de usabilidade, ao invés de gerenciar erros decorrentes das atividades, procura-se gerenciar as próprias atividades durante a sua execução. Sua utilização se baseia em uma listagem aonde são atribuídos os diferentes níveis de prioridade de cada atividade, ou correção de erro, que devem ser desenvolvidos. Entretanto, este sistema não permite designar um determinado colaborador para efetuar a correção do projeto. Sendo assim, o colaborador que deverá desenvolver a atividade pode não ser a pessoa mais qualificada para a execução desta tarefa.

Existem também ambientes de versionamento e de repositórios online, que buscam oferecer uma base de meta dados e informações sobre o desenvolvimento de um projeto, tal como o *SourceForge*². Este é um repositório online, utilizado por empresas e pela comunidade acadêmica para o armazenamento de artefatos de um projeto. Porém, não implementa um controle de permissão, impossibilitando a criação e organização de times/equipes virtuais, oferecendo uma barreira para a sua utilização em ambientes DDS.

Os trabalhos apresentados procuram minimizar alguns riscos sobre o ambiente DDS. Embora, nota-se a existência de uma lacuna sobre a área de Gerenciamento de Produção Distribuída. Desta forma, a Tabela 2 apresenta uma comparação entre os trabalhos relacionados e

¹OpenProject <https://community.openproject.org/>

²SourceForge <https://sourceforge.net/>

Tabela 2: Comparação entre trabalhos relacionados e a solução desenvolvida

	OpenProject	SourceForge	Bug Tracking	Solução Desenvolvida
Ciclo de Vida				X
Fase				X
Atividades	X		X	X
Equipes				X
Colaboradores	X	X	X	X
Papéis		X		X
Projetos	X	X	X	X
Status do Projeto			X	X
Controle de Qualidade	X	X	X	X
Status de Atividades	X		X	X
<i>log's</i> individuais				X
Relatório	X	X	X	X

o sistema desenvolvido, de forma a demonstrar as características de cada ambiente.

A Tabela 2 foi organizada a partir do mapeamento entre pontos considerados significantes de todos os trabalhos. Pode ser destacando que entre os três trabalhos relacionados existem algumas lacunas que interferem no gerenciamento de produção DDS. Nenhum deles possui a capacidade de utilizar um ciclo de vida para os projetos, desta forma existe grandes dificuldades em garantir coesão nas diferentes fases de desenvolvimento. Além disso nenhum trabalho relacionado oferece a possibilidade de visualizar os *log's* que demonstram a produção dos colaboradores.

O *OpenProject* atende algumas demandas sobre atividades, status e colaboradores, porém não permite a criação de equipes/times virtuais, além de não permitir o controle sobre o status do projeto.

Já o *SourceForge* por ser um repositório de dados sobre projetos não permite uma eficácia de controle sobre diversas atividades, além de não possuir os diferentes status que uma atividade ou o projeto possa possuir.

O *BugTracking* permite um controle mais geral sobre projetos e seu status, permitindo empregar um controle de qualidade mais satisfatório sobre as atividades. Porém não permite a organização de papéis, maximizando alguns desafios sobre gerenciamento.

Entretanto todos os trabalhos relacionados oferecem suporte aos colaboradores, procuram aumentar a qualidade do produto final e permitem a geração de relatórios. Embora possam ser percebido os pontos falhos existentes que dificultam o gerenciamento de produção nos sistemas analisados.

3 METODOLOGIA

Neste capítulo são apresentadas as tecnologias e ferramentas utilizadas para o desenvolvimento da solução apresentada. Também é exposto a organização da arquitetura do sistema e a descrição sobre o processo de desenvolvimento utilizado.

3.1 TECNOLOGIAS

Esta seção descreve e justifica a utilização das tecnologias que foram adotadas para o desenvolvimento do produto final deste trabalho. Apresentando as diferentes linguagens e padrões adotados.

Foi adotada a linguagem de programação orientada a objetos Java que é amplamente utilizada, recebendo um apoio significativo da indústria de computação (ILHAM; MURAKAMI, 2011). Sua utilização foi necessária para implementação das classes responsáveis pela dinamicidade do sistema, enviando e recebendo o processamento de informações.

Também foi utilizado o *Java Persistence API 2.0 (JPA)*, uma especificação da linguagem Java utilizado para a persistência de dados, que define um mapeamento objeto-relacional, transformando o modelo utilizado em banco de dados para objetos Java. Este padrão de persistência procura maximizar o desenvolvimento de uma aplicação, pois apresenta melhores características de organização e produtividade (BOCK, 2011).

O *Cascading Style Sheets (CSS)* é um mecanismo simples para adicionar diferentes estilos em páginas eletrônicas, foi utilizado para aperfeiçoar o visual da aplicação.

O *Extensible HyperText Markup Language (XHTML)* oferece uma nova adição de recursos ao HTML, proporcionando um caminho mais sofisticado de desenvolvimento *web* (MUSCIANO; KENNEDY, 1997). Foi utilizado para criação de páginas eletrônicas do sistema.

Como servidor foi adotado o Apache TomCat versão 8.0.3.0, utilizado amplamente para aplicações *web* baseado em *servlets*. Foi utilizado para processar todos os pedidos HTTP dos clientes do sistema.

Foi utilizado a *Structured Query Language* (SQL) para realizar a manipulação no banco de dados, testando algumas validações da solução.

Para a construção dos artefatos do sistema foi utilizado o *Unified Modeling Language* (UML), linguagem gráfica de visualização, especificação e documentação de artefatos de software, (BOOCH, 2004).

O *Bootstrap* versão 3.1.1 foi aplicado nesta solução para adicionar ícones e referências em páginas XHTML.

O *PrimeFaces* versão 4.0.2 é um *framework JavaServer Faces* para o desenvolvimento *front-end*, que conta com mais de 100 componentes otimizados, como estilos, validações com o cliente, entre outros. Seus componentes foram aplicados para auxiliar o desenvolvimento das páginas XHTML.

Foi utilizado o *JavaServer Faces* (JSF) versão 2.2, um *framework* baseado em Java, com arquitetura MVC para a construção de interfaces.

3.2 FERRAMENTAS

As ferramentas representam a utilização de ambientes integrados de desenvolvimento (IDE) para a construção desta solução. Desta forma, foram utilizadas as seguintes IDEs de programação:

O PostgreSQL versão 9.3.4, sistema gerenciador de banco de dados objeto relacional, de código aberto, com alta capacidade de processamento e otimização (STONEBRAKER; ROWE, 1986). Foi aplicado para o armazenamento de informações na base de dados do projeto.

O NetBeans versão 8.0 que é uma IDE amplamente utilizada por grandes empresas, oferecendo alta capacidade de desempenho e processamento na mais diversas operações (BOCK, 2011). O NetBeans foi empregado neste trabalho pois oferece suporte para todos padrões e linguagens de programação utilizados no desenvolvimento desta solução.

O Astah Professional, um software de modelagem UML utilizado para a criação dos artefatos da UML. Segundo Alencar et al. (2013) sua utilização é intuitiva e simples, facilitando a construção de diagramas que permitem compreender melhor o comportamento do sistema.

3.3 ARQUITETURA DO SISTEMA

A arquitetura do sistema utilizada segue o padrão de projeto denominado *Model View Controller* (MVC). A qual tem como base a divisão em camadas que são separadas em classes e layouts tornando-as independentes e que podem ser alteradas sem se afetarem (SILVA; MACHADO, 2010).

Atualmente, a arquitetura MVC é utilizada em aplicações *web* para otimizar estratégias de desenvolvimento e alcançar alta usabilidade (HASSAN; ISSAC., 2011). Foi adotada neste projeto por aproximar a realidade do paradigma de orientação a objetos e oferecer grande flexibilidade.

Na arquitetura MVC a entrada de dados, as respostas de requisições da aplicação e a interface visual são explicitamente separadas em ordenadas em diferentes objetos. Sua estrutura pode ser compreendida sobre uma *View* (visão) que é responsável pela dispersão gráfica e/ou textual dos elementos; O *Controller* (controlador) interpreta os comandos e requisições do sistema e o *Model* (modelo) que gerencia o comportamento de dados e domínio da aplicação, respondendo a pedidos ou instruções. A arquitetura deste projeto é apresentada na Figura 3.

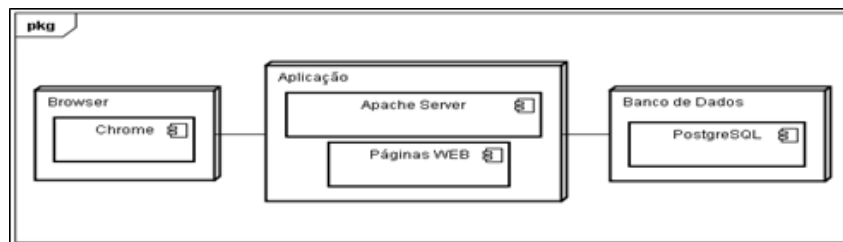


Figura 3: Arquitetura do Sistema

A Figura 3 apresentou a estrutura básica da arquitetura geral desta solução. O *Browser* refere-se ao navegador utilizado durante o desenvolvimento deste trabalho. O pacote da aplicação é formado pelo servidor, páginas e classes do sistema. A base de dados é representada pelo PostgreSQL. O detalhamento da arquitetura do sistema é apresentado na Figura 4.

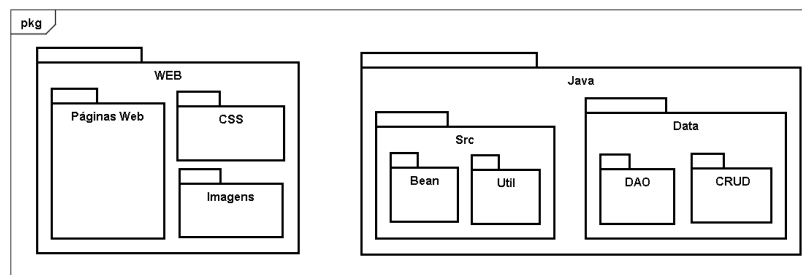


Figura 4: Detalhamento da arquitetura do Sistema

A Figura 4 apresentou a representação sobre o detalhamento da arquitetura da solução desenvolvida. O Pacote *Web* é responsável por implementar a camada *View* da arquitetura MVC, e nele estão compreendidas os arquivos responsáveis pelo estilo, estrutura e organização das páginas eletrônicas. No pacote *Páginas Web* estão as classes XHTML do sistema. No pacote *CSS* encontram-se os códigos responsáveis por adicionar estilos à solução. E no pacote *imagens* as figuras utilizadas no sistema.

O Pacote *Java* compreende as classes responsáveis por efetuarem mapeamento, persistência e algumas validações de dados, corresponde a camada *Model* da arquitetura MVC. No pacote *SRC* estão armazenadas classes Java, divididas em *Classes Managed Beans* que contem métodos e a lógica de validação para entrada de dados nas páginas *Web* e estão armazenadas no pacote *Bean*. Já no pacote *Util* existem classes com métodos estáticos utilizados pelo sistema.

A camada *Controller* da arquitetura MVC foi implementado no pacote *Data*, e apresenta o conjunto de classes responsáveis pela persistência de dados e mapeamento objeto-relacional do *JPA*. No pacote *DAO* estão as classes responsáveis pela construção de métodos de persistência e de consulta. No pacote *CRUD* as classes responsáveis por implementar o *JPA*.

3.4 PROCESSO DE DESENVOLVIMENTO

O processo de desenvolvimento adotado foi uma personalização do *Scrum*. O *Scrum* é tipicamente iterativo e incremental, apresentando uma metodologia ágil de qualidade e produtividade para a gestão e o planejamento de projetos de software (MUNDRA et al., 2013). O método ágil encontrado no *Scrum* é uma forma de desenvolvimento adaptativa e flexível, que sugere uma hierarquia entre os participantes de um projeto.

Existem três papéis principais: O *Product Owner* (PO), responsável por priorizar, definir e adicionar tarefas em um projeto. O *Development Team* (equipe de desenvolvimento), responsável pela entrega do produto. E o *Scrum Master*, que representa a ligação entre a equipe de desenvolvimento e o PO.

Este processo de desenvolvimento permite a organização de um projeto em diferentes ciclos ou iterações denominadas *Sprints* (CARVALHO; MELLO, 2009). Segundo Shwaber e Beedle (2002) a finalidade de uma *Sprint* é gerar resultados que irão agregar valor ao produto final desenvolvido. Recomenda-se que cada *Sprint* possua uma duração máxima de trinta dias e que após cada dia de trabalho ocorra uma reunião entre os colaboradores afim de que sejam apresentados as ações desenvolvidas. Esta organização está apresentada na Figura 5.

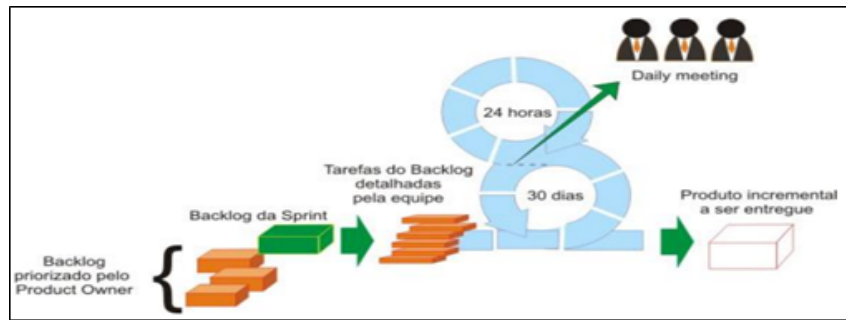


Figura 5: Esquematização do Processo de Desenvolvimento Scrum

Fonte: (PEREIRA et al., 2007)

A Figura 5 apresentou a organização do *Scrum*. Inicialmente o *Product Owner* realiza o *Product Backlog*, que consiste em selecionar e priorizar o trabalho a ser desenvolvido em uma *Sprint*. Estas tarefas são detalhadas e assume-se um ciclo de tempo para que sejam desenvolvidas, denominado *Sprint Backlog*. Ao final de uma *Sprint* é apresentado o produto final, que é revisado e entregue.

Foi adotado que o colaborador do projeto é o discente responsável pelo produto final deste trabalho, e o *Product Owner*, o orientador, formando assim a equipe de desenvolvimento. Devido ao tamanho reduzido da equipe não foi definido um papel para o *Scrum Master*, e adotou-se uma personalização do *Scrum*. Ao invés de reuniões diárias, estas ocorreram a cada 15 dias. Durante cada reunião foi apresentado o *backlog* de tarefas realizadas. O *Product Owner* revisava o que foi entregue, efetuava as correções necessárias e especificava quais tarefas seriam desenvolvidas para a próxima reunião.

4 EXECUÇÃO DO PROCESSO

Este capítulo apresenta a descrição sobre a aplicação do Processo de Desenvolvimento para a concepção da solução final, objetivo deste trabalho. Desta forma, são expostas as diferentes etapas de maturidade de desenvolvimento, bem como as atividades e artefatos gerados.

4.1 DESENVOLVIMENTO

Durante o desenvolvimento deste trabalho foram realizadas seis *Sprints*, notou-se que as características de dinamicidade e iteratividade do *Scrum* demonstraram ser oportunas, e foram adaptando-se a realidade do projeto.

Com a personalização adotada no processo, a realização de reuniões a cada quinze dias de trabalho permitiram direcionar um fluxo de atenção sobre cada etapa de desenvolvimento; aplicar correções necessárias a fim de aprimorar a qualidade final do projeto e supervisionar a maturidade do projeto.

O conteúdo gerado durante cada reunião foi armazenado em uma ata. Esta documentação encontra-se no apêndice B deste documento para ser consultada. Durante a realização de cada reunião, verificou-se os aspectos evolutivos deste trabalho. A Tabela 3 apresenta os estágios de maturação do projeto, classificados em inicial, média e alta, conforme a conclusão de cada *Sprint*.

Tabela 3: Maturidade do Projeto

Sprints	Esforço	Estágio	Características
Primeira <i>Sprint</i>	30 dias	Maturidade inicial	Concepção do Projeto, definição do tema e tecnologias e levantamento de requisitos.

Continua

Tabela 3: Maturidade do Projeto

Sprints	Esforço Estágio	Características
Segunda <i>Sprint</i>	30 dias Maturidade inicial	Pesquisa e escrita sobre fundamentação teórica, metodologia diagramas de arquitetura e elaboração do documento de proposta
Terceira <i>Sprint</i>	30 dias Maturidade média	Correção e definição de regras de negócio do sistema, implementação de interfaces iniciais.
Quarta <i>Sprint</i>	30 dias Maturidade média	O projeto já apresentava características para realização de cadastros.
Quinta <i>Sprint</i>	30 dias Maturidade alta	Todas as funcionalidades foram implementadas e o sistema ficou pronto para fase de teste.
Sexta <i>Sprint</i>	30 dias Maturidade Alta	Documentação do Projeto e desenvolvimento da Monografia

A Tabela 3 apresentou a disposição entre a duração de cada *Sprint* e a maturação deste trabalho. As *Sprints* com estágio de maturidade inicial representam etapas de concepção do sistema e foram responsáveis por alinhar diversos fatores para iniciar a fase de codificação, como definição de regras de negócios, atores e funcionalidades do sistema.

As *Sprints* com estágio de maturidade média representam que todos os pré-requisitos para a inicialização de codificação do projeto haviam sido atendidos. Assim, os requisitos funcionais e não funcionais do sistema haviam sido definidos, junto com a utilização de tecnologias, métodos e processos de desenvolvimento.

Nas duas últimas *Sprints*, o sistema foi considerado com alta maturidade, pois todas as funcionalidades para satisfazer as necessidades declaradas no documento de proposta haviam sido atendidas. A seguir serão detalhadas as atividades e artefatos de cada *Sprint*.

4.1.1 PRIMEIRA SPRINT

A realização da primeira *Sprint* foi responsável por apresentar a concepção do projeto, definir o escopo do tema e realizar o levantamento de requisitos iniciais. Ela foi realizada

entre os dias 20 de março e 20 de abril de 2014. Houveram duas reuniões entre a equipe de desenvolvimento. A primeira foi responsável por definir o tema e os objetivos a serem atingidos com o desenvolvimento do produto final. Na segunda houve a apresentação dos requisitos coletados e a correção sobre a estrutura do documento de proposta.

O Detalhamento dos artefatos gerados durante a realização da Primeira *Sprint* são detalhadas a seguir:

a) **Entrada:** Definição do tema

Nesta etapa, foi definido o tema geral do projeto, bem como algumas instâncias de problemas a serem tratados. Junto a sua definição, foi gerada a documentação sobre os requisitos do sistema.

a.1) Saída: Documento de especificação de requisitos

Este documento compreende os requisitos do sistema que são tratados como funcionalidades desejadas para o desenvolvimento do produto final. Ver a seção de Apêndices deste trabalho.

b) **Entrada:** Documento de especificação de requisitos

A partir dos requisitos documentados na etapa anterior, foi gerado o cronograma.

b.1) Saída: Cronograma do projeto

O Cronograma do Projeto apresenta o tempo para a execução de cada *Sprint*, nele está detalhado as datas de realização bem como os artefatos gerados durante a execução do Ciclo de Vida adotado. O cronograma do projeto esta apresentado na Tabela 4.

Tabela 4: Cronograma do Projeto

<i>Sprint</i>	Início	Fim	Artefatos/Tarefas
Primeira	20/03	20/04	Definição do Tema
			Documento de Requisitos
			Cronograma do Projeto
Segunda	21/04	20/05	Diagrama de Caso de Uso Geral
			Diagrama de Classes (Implementado)
			Modelo de Entidade Relacionamento
			Documento de Proposta

Continua

Tabela 4: Cronograma do Projeto

<i>Sprint</i>	Início	Fim	Artefatos/Tarefas
Terceira	27/07	26/08	Correção da Proposta Atualização de Funcionalidades Ambiente de Versionamento Atores do Sistema Correção Diagrama de Classes Codificação de Interfaces
Quarta	27/08	28/09	Diagramas de Sequência Revisão de Interfaces (<i>Sprint 3</i>)
Quarta	27/08	28/09	Codificação de Interfaces Cadastros do Sistema
Quinta	29/09	28/10	Diagramas de Sequência Interface de <i>Log, Upload e Download</i> Interfaces de Alteração de Registros Interfaces de Remoção de Registros
Sexta	29/10	30/11	Testes da Solução desenvolvida Escrita da Monografia Defesa da Monografia

A Tabela 4 apresentou a duração de cada *Sprint* deste Projeto, detalhando a relação de tempo, os artefatos gerados e tarefas executadas. Com a conclusão do Cronograma foi iniciado a Segunda *Sprint* do Projeto.

4.1.2 SEGUNDA SPRINT

Na segunda *Sprint* houve o desenvolvimento sobre os diagramas de arquitetura do sistema, como o diagrama de Caso de Uso Geral, Diagrama de Classes e o Mapeamento de Classes JPA. Foi realizada entre os dias 21 de abril e 20 de maio de 2014. Nesta *Sprint* também houve duas reuniões, na primeira foi apresentado e validado o contexto geral e a fundamentação teórica do documento de proposta. Na segunda reunião foi definido que deveria ocorrer a construção de diagramas iniciais baseados na coleta de requisitos da *Sprint* anterior.

- a) **Entrada:** Documento de especificação de requisitos

A partir da documentação de requisitos da iteração anterior, ocorreu a definição de regras de negócio da Solução.

a.1) Saída: Regras de Negócios

As regras de negócio do sistema dizem respeito as permissões e deveres de cada usuário dentro do sistema. Com isso, foi definido que o sistema possui três tipos de Usuários, sendo eles o Administrador, Gerente e Colaborador.

a.2) Saída: Atores do Sistema

Cada ator no sistema possui um conjunto de permissões e deveres, que são explorados no Caso de Uso Geral. A Figura 6 apresenta a relação dos atores do sistema.

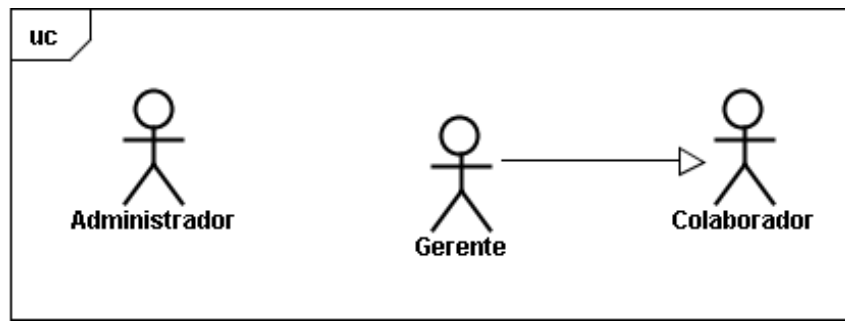


Figura 6: Relação de Atores do Sistema.

O Administrador é responsável pela coordenação de um projeto DDS, sendo sua responsabilidade cuidados referentes ao Ciclo de Vida e o cadastro de Gerentes deste projeto. Já o Colaborador possui o menor nível de permissão dentro do sistema, porém, este pode ser escolhido como Gerente de um projeto, e com isso recebe maiores permissões. Passando a ser Gerente, ele possui a capacidade de alterar o Ciclo de Vida deste projeto, administrar e gerenciar a produção dos colaboradores de sua equipe.

a.3) Saída: Diagrama de Atividades

O Diagrama de Atividades descreve o fluxo de controle realizado pelos atores e suas permissões dentro do sistema. O diagrama de atividades está apresentado na Figura 7.

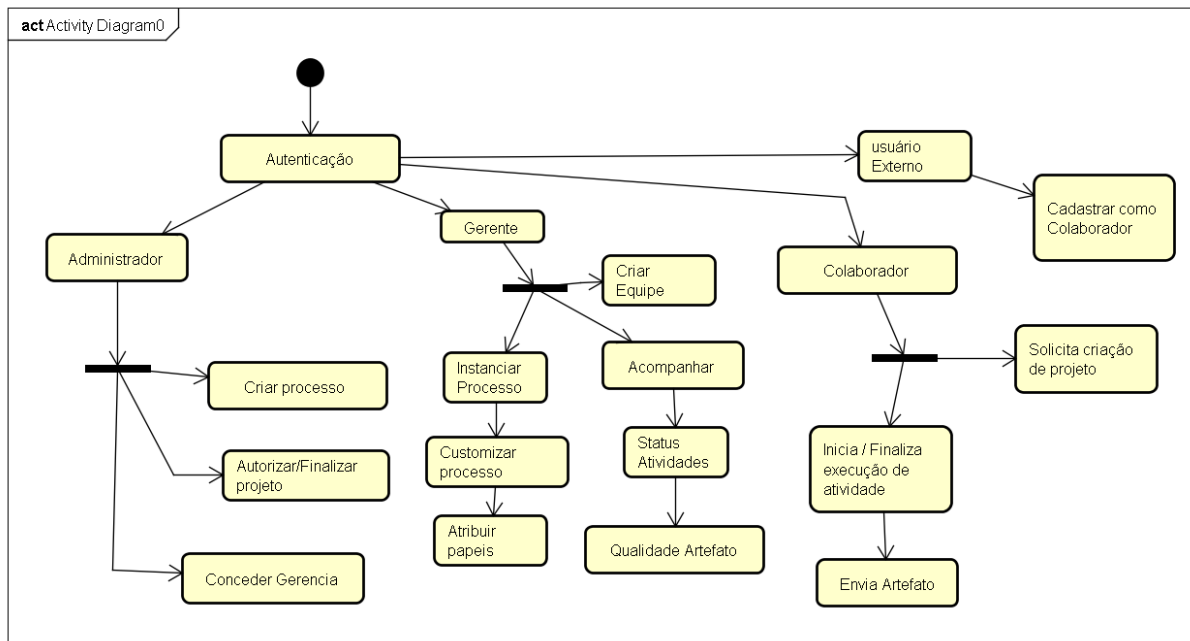


Figura 7: Diagrama de Atividades Referente ao Fluxo de Trabalho.

A Figura 7 apresentou o diagrama de atividades da solução desenvolvida. Inicialmente o usuário deve realizar a autenticação no sistema, caso não possua acesso ele é considerado um usuário externo e pode então cadastrar-se como colaborador.

Após a realização da autenticação, o usuário Administrador poderá cadastrar e gerenciar processos de desenvolvimento, autorizar o início ou fim de projetos e conceder a colaboradores permissões de Gerentes de Equipes.

Caso o usuário seja um Gerente, poderá cadastrar sua equipe, acompanhar o fluxo de atividades desenvolvidas e os colaboradores responsáveis pela execução e verificar a qualidade final do artefato produzido. Além disto, um Gerente de Equipe pode instanciar o processo de desenvolvimento adotado no projeto, customizando-o com a inserção de novas atividades para a sua equipe, atribuindo papéis aos colaboradores responsáveis.

Caso o usuário seja Colaborador, poderá enviar a solicitação de gerência para o administrador de um projeto ou então iniciar, desenvolver e finalizar atividades atribuídas para si, podendo enviar o artefato produzido.

a.4) Saída: Caso de Uso Geral

O Caso de Uso Geral demonstra um cenário sobre as funcionalidades da solução desenvolvida, atribuindo aos atores do sistema um conjunto de operações para a realização de tarefas. O Caso de Uso está apresentado na Figura 8.

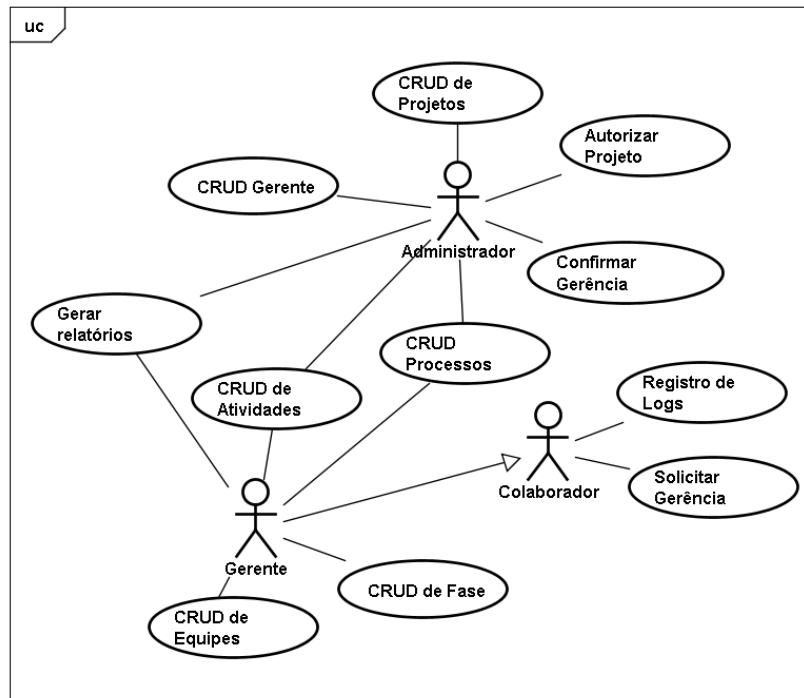


Figura 8: Caso de Uso Geral.

A Figura 8 apresentou a relação entre os atores do sistema e suas interações. A sua implementação foi necessária para que a solução desenvolvida alcançasse o objetivo de auxiliar o gerenciamento de produção DDS. O detalhamento de cada interação é apresentado a seguir

- *Create Read Update e Delete* (CRUD) de Usuários: Permite ao usuário autenticado no sistema executar operações de criação, leitura, atualização e remoção sobre sua conta. Foi implementado para oferecer aos usuários o gerenciamento sobre seus dados pessoais que estarão cadastrados.
- CRUD de Projetos: Permite ao usuário autenticado no e com permissão de Administrador no sistema a execução de operações sobre criação, leitura, atualização e remoção sobre seus projetos.
- CRUD de Atividades: Permite ao usuário autenticado no sistema, com permissão de Administrador ou Gerente executar operações de criação, leitura, atualização e remoção sobre as atividades de seus projetos. Esta funcionalidade foi implementada a fim de armazenar informações sobre a construção dos artefatos gerados durante o desenvolvimento de cada projeto.
- CRUD de Processo: Permite ao usuário autenticado no sistema, com permissão de Administrador executar operações de leitura, criação, atualização e remoção sobre um pro-

cesso. Este representa o ciclo de vida adotado, e pode ser customizado para adequar-se a realidade de cada projeto.

- **CRUD de Fase:** Permite ao usuário autenticado no sistema, com permissão de Administrador, executar operações de criação, leitura, atualização e remoção sobre as fases de desenvolvimento em seu projeto. Esta funcionalidade organiza sistematicamente os artefatos e atividades desenvolvidas por cada equipe durante determinada fase.
- **Gerenciamento de Equipes:** Permite ao usuário autenticado no sistema, com permissão de Gerente, executar operações sobre o gerenciamento de sua equipe. Oferecendo opções sobre o cadastro, alteração e remoção sobre as informações referentes à equipe.
- **Registro de *Logs*:** Permite ao usuário autenticado no sistema, com permissão de colaborador, realizar a execução das operações de criação e leitura sobre o histórico de suas atividades em um projeto. Esta funcionalidade foi implementada para fornecer ao usuário o tempo gasto e a data de realização das atividades.
- **Relatório de *Logs*:** Permite ao usuário autenticado e com permissão de Administrador ou Gerente realizar a construção de relatórios sobre *Logs* individuais dos colaboradores ou de informações de um projeto. Tal funcionalidade foi implementada a fim de oferecer uma função para a organização de relatórios utilizando parâmetros programados para auxiliar a sua construção.
- **CRUD de Gerentes:** Permite ao usuário autenticado e com permissão de Administrador, realizar operações de criação, leitura, atualização e remoção dos gerentes de um projeto. Esta funcionalidade foi implementada a fim de oferecer uma forma de organizar os gerentes alocados em projeto.
- **Autorizar Projeto:** Permite ao usuário autenticado e com permissão de Administrador autorizar o início de seus projetos. Com isso, podem ser realizadas operações de criação e alteração do status atual em um projeto.
- **Solicitar Gerência:** Permite ao usuário autenticado e com permissão de Colaborador realizar a solicitação de gerência em projeto. Essa funcionalidade foi implementada para que os Administradores possam definir quais Colaboradores devem ser Gerentes em um projeto.

b) **Entrada:** Documento de especificação de requisitos, Regras de Negócios e Diagrama de Caso de Uso Geral.

b) Saída: Diagrama de Classes

Este diagrama apresenta o conjunto de classes utilizadas para o desenvolvimento da solução final, foi construído a partir da organização de pacotes apresentada na Figura 4. O Diagrama de Classes foi dividido em dois pacotes. O primeiro pacote representam as classes *bean* do sistema, estas classes utilizam métodos comuns que estão armazenado no pacote *Util*. Já a Figura 10 apresenta as classes responsáveis por conter os métodos de persistência do sistema

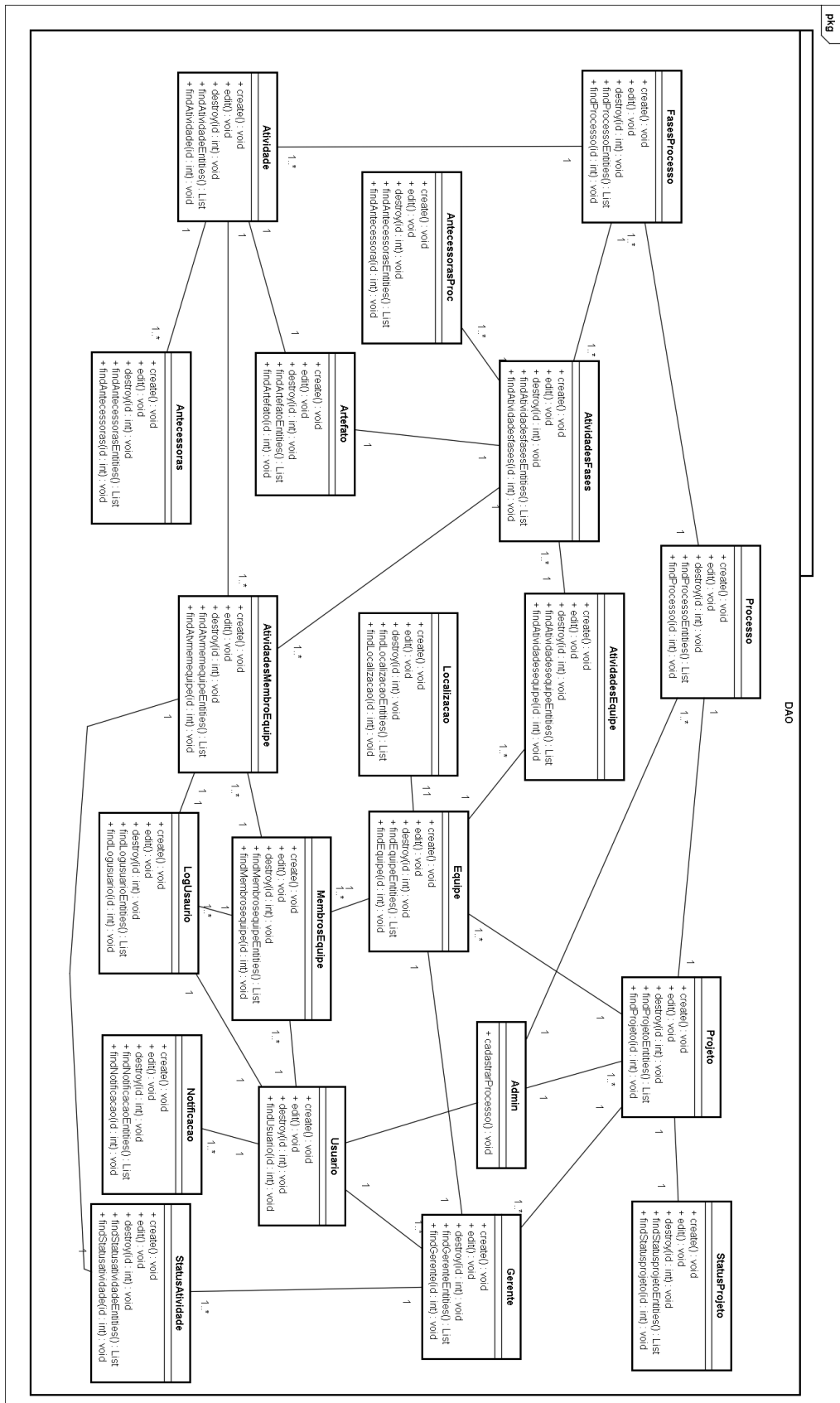


Figura 10: Diagrama de Classes Dao.

a.4) Saída: Ferramentas e Tecnologias.

Durante a execução da Segunda *Sprint* ocorreu a definição sobre as ferramentas e tecnologias que foram empregadas para o desenvolvimento da solução final deste projeto. A descrição referente a cada tecnologia ou ferramenta empregada foi apresentado no capítulo 3 deste trabalho.

4.1.3 TERCEIRA *SPRINT*

A Terceira *Sprint* foi responsável por definir e organizar o ambiente de versionamento do projeto, efetuar a correção do documento de proposta e aperfeiçoamento das regras de negócio do sistema. Com isso, a *Sprint* iniciou-se dia 27 de Julho de 2014 e foi finalizada dia 26 de Agosto de 2014.

Durante a primeira reunião, foi definido e organizado o ambiente para o versionamento do projeto e da monografia. Optou-se pelo ambiente *Source Forge* e foi organizada uma estrutura de pastas para o armazenamento sobre a documentação, páginas *web*, classes e *scripts* relacionados ao projeto e os artefatos desenvolvidos. Como tarefas para a próxima reunião recomendou-se a modelagem dos *CRUD's* iniciais do Sistema e correção da proposta.

Na segunda reunião, realizada dia 04 de Agosto de 2014, foi definido então o conjunto de regras sobre as permissões entre Administrador e Gerente do projeto. Optou-se pelo Administrador realizar a gerência sobre um projeto e a definição de um processo (ciclo de vida), derivando ao Gerente a tarefa de especificação desse processo, bem como as suas atividades. Também foi definido que um Colaborador (membro/gerente) poderá estar em diversos projetos simultaneamente. Como tarefa para a próxima reunião recomendou-se que a refinação dos *CRUD's* sobre Gerentes e projetos.

a) **Entrada:** Correção do Documento de Proposta.

O Documento de Proposta apresenta o conceito inicial do sistema, e a apresentação de métodos e processos utilizados para seu desenvolvimento. Dentre as principais alterações, ocorreram a inserção de novas funcionalidades e a correção do Diagrama de Classes.

a.1) Saída: Atualização de funcionalidades.

Foi definido a inserção de uma nova funcionalidade na solução, sendo ela a geração de relatórios referentes ao tempo gasto nas atividades realizadas pelos Colaboradores. Esta e todas as funcionalidades foram detalhadas no Diagrama de Caso de Uso geral, apresentado na seção anterior.

a.2) Saída: Correção Diagramas de Classes

O Diagrama de Classes foi reformulado, ocorrendo a rotulação de relacionamento e inserção de novas Classes do Sistema. O Diagrama de Classes foi apresentado na seção 4.1.2 deste documento.

b) **Entrada:** Definição do Ambiente de Versionamento

Foi adotado a utilização do ambiente de versionamento *Source Forge*, aonde o *Product Owner* e o Colaborador possuem acesso e permissão para alterações. Neste ambiente toda a documentação e codificação referentes a construção do sistema estão armazenadas e podem ser consultadas por qualquer usuário.

b.1) Saída: Construção do ambiente de Versionamento¹

O ambiente de versionamento foi estruturado em três pacotes, o primeiro pacote denominado *Docs* é responsável por armazenar a data de cada reunião, as atividades desenvolvidas e artefatos gerados. O segundo pacote denominado *Scripts* contém o *script* para gerar o banco de dados. E o terceiro pacote denominado *Source* contém as classes responsáveis pelo mapeamento, persistência de dados e páginas *web* do sistema. Este ambiente é apresentado na Figura 5.










Name ↕	Modified ↕	Size ↕	Downloads / Week ▾
↑ Parent folder			
📁 Docs	2014-07-28		1   
📁 Scripts	2014-08-04		2   
📁 Source	2014-07-28		28   
Totals: 3 Items			

Figura 11: Ambiente de Versionamento.

A Figura 11 apresentou o ambiente de versionamento do projeto, bem como as datas de criação de cada pacote.

d) **Entrada:** Codificação de *CRUD's* iniciais

d.1) Saída: Interfaces de Cadastro de Ciclo de Vida, Atividades, Página Inicial de Gerentes e Administradores.

Ao fim da terceira *Sprint* foram codificados algumas interfaces do sistema. Primeira-

¹ProDDS <https://sourceforge.net/projects/tccprodds/>

mente optou-se por realizar o Cadastro de um Ciclo de Vida, por ser de grande importância para o desenvolvimento da solução final. Com isso, também foi desenvolvido a interface referente ao cadastro de atividades, além da criação de interfaces relacionadas a Página Inicial dos usuários do sistemas que possuíam permissão de gerenciamento sobre o Ciclo de Vida.

4.1.4 QUARTA SPRINT

A Quarta *Sprint* foi responsável por revisar as interfaces implementadas do sistema, sobretudo, a interface responsável pelo cadastramento de um Ciclo de Vida. Teve início no dia 27 de Agosto de 2014 e término em 28 de Setembro de 2014.

Na primeira reunião as observações feitas pelo *Product Owner* dizem respeito ao refinamento da regra de negócios sobre o cadastramento e utilização de um processo de desenvolvimento em um projeto. Desta forma ficou definido que um Administrador, poderá cadastrar um novo processo, este deverá obrigatoriamente possuir um nome, uma descrição e o total de fases. Também será necessário que o Administrador cadastre cada fase, definindo as atividades desenvolvidas, bem como as antecessoras e sucessoras. Com isso ele pode criar um projeto e selecionar os Gerentes de Equipe. Já os Gerentes possuem a responsabilidade de instanciar o processo, podendo optar por remodelar o processo [modificando fases/atividades/sucessoras/antecessoras] e também definir qual membro da sua equipe deverá desenvolver determinada atividade. Como tarefas para a próxima reunião ficou definido que a nova regra de negócios sobre Ciclo de Vida seria implementada.

Na segunda reunião foi apresentada a nova implementação das interfaces referentes ao cadastro de processo, projeto, solicitação de gerência e implementação da regra de negócios referente ao Ciclo de Vida. O *product owner* falou ser interessante implementar a interface do gerente como uma forma de visualizar suas atribuições ou pendências dentro do sistema, tal como cadastrar uma equipe, verificar se existe atividade sem atribuição a um membro. Por último foi sugerido que a implementação de uma tabela que ele possa estimar o total de horas gastas em uma determinada atividade. Como tarefas para a próxima reunião foi definido que fosse desenvolvida a interface de Gerentes do Sistema.

a) **Entrada:** Regras de Negócio

a.1) Saída: Refinamento de Regras de Negócios referentes ao Ciclo de Vida.

Ficou definido que o Ciclo de Vida deverá possuir fases e atividades. O Ciclo de Vida e suas fases deverão, obrigatoriamente, serem cadastradas por um usuário do sistema com permissão de Administrador. Este também será o responsável pelo cadastro e seleção dos Gerentes

de seu Projeto. Ao Gerente, cabe a ação de instanciar esse Ciclo de Vida, podendo realizar alterações nas Atividades relacionadas ao Ciclo de Vida utilizado no projeto. Porém, estas alterações somente poderão ser utilizadas por ele mesmo dentro de sua equipe. Desta forma, o Ciclo de Vida será transparente a todos os Gerentes do projeto, porém, cada um poderá personalizá-lo conforme a necessidade.

a.1) Saída: Diagrama de Sequência sobre Cadastro de Processos

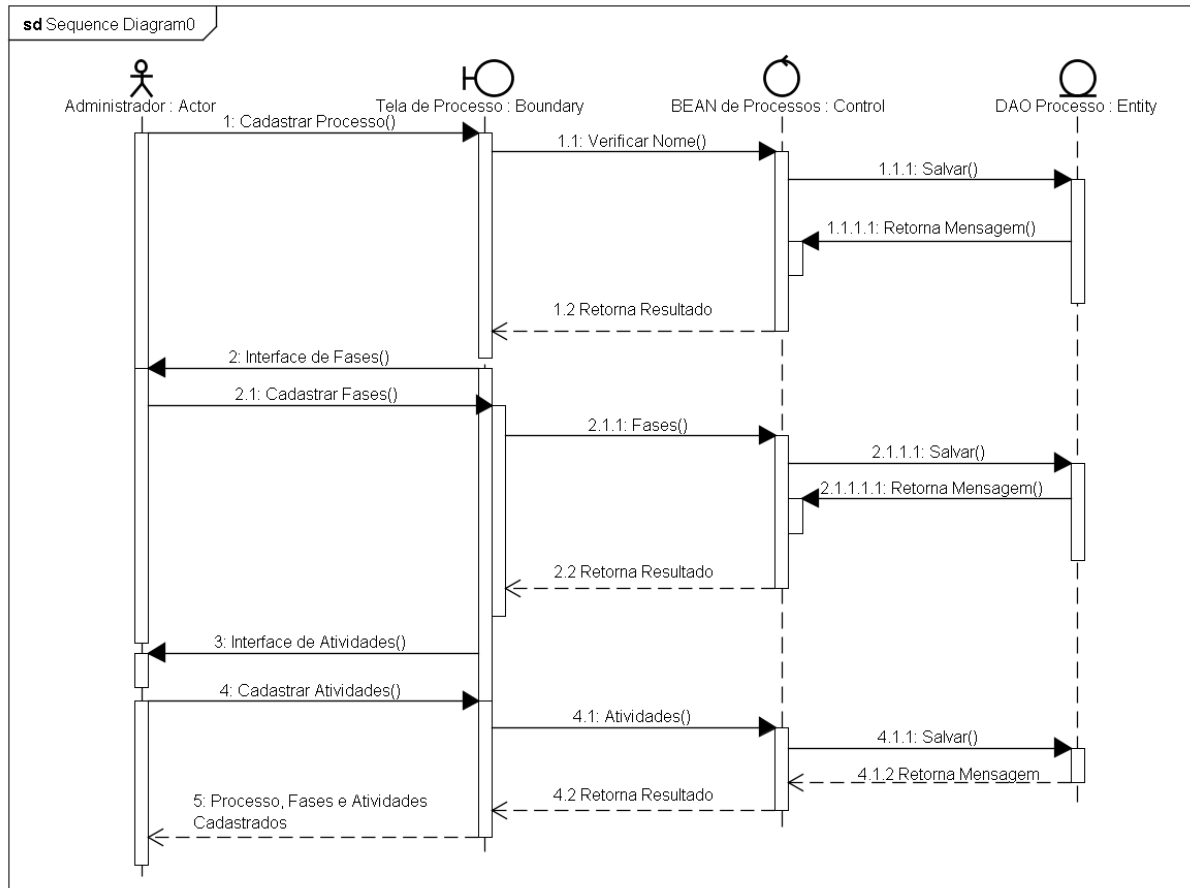


Figura 12: Diagrama de Sequência - Cadastro de Processos de Desenvolvimento.

A Figura 12 apresentou o Diagrama de Sequência sobre o Cadastro de um Processo de Desenvolvimento no Sistema. Inicialmente o Administrador Cadastra um Processo com nome, descrição e total de Fases, ocorre a verificação para avaliar se o nome é disponível ou se o Administrador já possui um Processo semelhante cadastrado. Em caso de validação positiva, o processo é salvo, retornando uma mensagem de sucesso e abrindo uma interface para o cadastro de fases deste processo. Caso contrário é emitida uma mensagem para o usuário utilizar um novo nome.

Após a validação e cadastro do Processo, é liberado a interface para o cadastro de fases. Nesta interface o Administrador deve cadastrar o nome e a descrição de cada fase do Processo.

Com isso ocorre a persistência dos dados, e o retorno de mensagem para o usuário cadastrar as atividades referentes a cada fase do processo.

Com a interface de atividades, o Administrador deve adicionar o nome, a sua descrição e fazer referência as atividades antecessoras e sucessoras. Ao fim do processo é retornado uma mensagem e o registro de um processo contendo fases e atividades.

a.2) Saída: Diagrama de Sequência - Cadastrar Atividade por Colaborador.

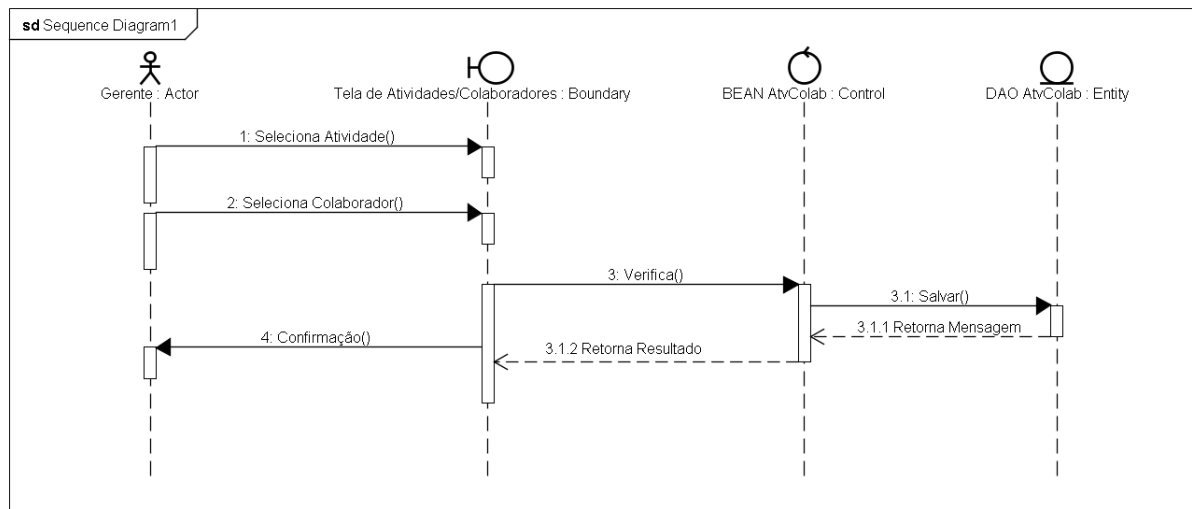


Figura 13: Diagrama de Sequência - Cadastro de Processos de Desenvolvimento.

A Figura 13 apresentou o Diagrama de Sequência referente ao cadastro de uma atividade para um determinado colaborador. Inicialmente, o Gerente seleciona uma atividade e o respectivo colaborador responsável pela sua execução. O sistema verifica se o colaborador já possui tal atividade alocada no projeto. Caso possua, retorna a mensagem que o cadastro não foi efetuado. Caso contrário, cadastra a atividade para o colaborador.

4.1.5 QUINTA SPRINT

A Quinta *Sprint* foi responsável por finalizar as interfaces implementadas do sistema. Teve início no dia 29 de Setembro de 2014 e término em 28 de Outubro de 2014.

Na primeira reunião foi apresentado as interfaces responsáveis pelo cadastro de informação do usuário que possui permissão de Gerência no sistema. Desta forma foi concluído as interfaces de cadastro sobre equipes, atividades e atividades por membro. Também ocorreu a apresentação da interface responsável pelo *log* de usuário, *upload* e *download* de artefato. Com isso, foi delineado que na próxima reunião ocorreria a otimização da interface para realizar a construção dos relatórios do sistema.

Na segunda reunião desta *Sprint* foram apresentados a conclusão de todos os Casos de Uso referente ao cadastro de informações no sistema e geração de relatórios. O próximo passo a ser desenvolvido no sistema era a respeito da opção de alteração e remoção de informações, e com isso, realizar o desenvolvimento de um manual de utilização do sistema. Como tarefas para a próxima reunião foi definido realizar o desenvolvimento das opções de Alterar e Excluir e a construção do Mapa do Site.

a) **Entrada:** Interfaces de Cadastro do Sistema

a.1) Saída: Interface de *log* de usuário, *upload* e *download* de artefato.

A funcionalidade de *log* do usuário é responsável por registrar o tempo gasto, a atividade, identificar o colaborador e verificar se gerou algum artefato. Com isso, todas as interfaces de cadastro foram concluídas, restando a realização de relatórios, alteração e remoção.

a.2) Saída: Diagrama de Sequência sobre Registro de *Log*

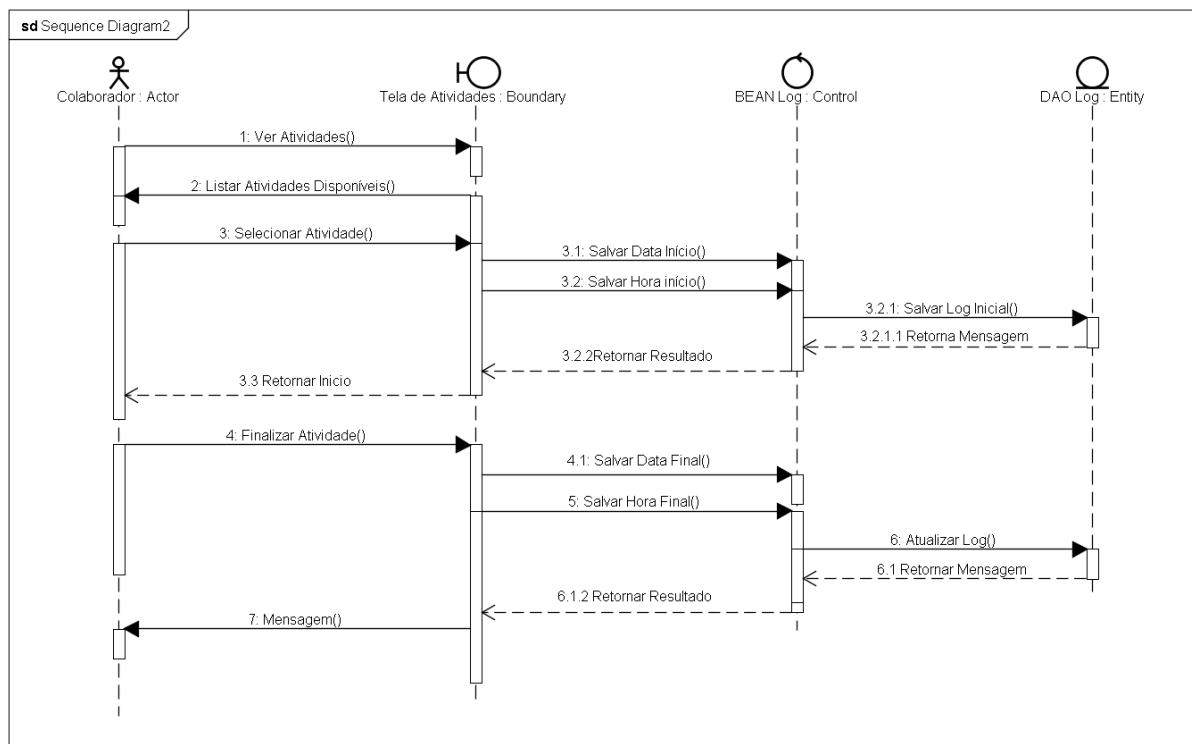


Figura 14: Diagrama de Sequência - Registro de *Log*.

A Figura 14 apresentou o Diagrama de Sequência referente ao registro de *Log* no sistema. Inicialmente, o Colaborador faz a requisição para verificar quais atividades podem ser desenvolvidas, e o sistema apresenta uma lista contendo a atividade e sua descrição. O Colaborador pode então selecionar a atividade escolhida, e o sistema automaticamente registra a data e hora de início da atividade. É retornado para o usuário a informação sobre a hora de

início.

Ao fim da execução, o Colaborador deve selecionar a opção de finalizar atividade. O sistema então registra automaticamente a hora e data de término, e ao fim retorna o resultado referente ao registro da atividade.

a.3) Saída: Construção de Relatórios e Interfaces de Alteração e Remoção de Registros do Sistema.

Estas interfaces são mais complexas que as anteriores, sobretudo, por realizarem mudanças significativas na base de dados do sistema. Com isso, inicialmente foi procurado realizar a construção de relatórios, utilizando os *log's* individuais dos colaboradores. As interfaces referente a alteração e remoção de registro representaram a última etapa de codificação do sistema. Atendendo assim, todas as funcionalidades que foram especificadas no Caso de Uso Geral deste documento. No Apêndice C deste documento, encontram-se as principais telas codificadas no sistema.

4.1.6 SEXTA SPRINT

A Sexta *Sprint* foi responsável pela documentação do produto final deste projeto, ocorreu durante os dias 28 de Outubro à 20 de Novembro de 2014.

Na primeira reunião foi apresentado a concepção inicial da monografia do projeto. Inicialmente este documento contava com as seções referentes ao Contexto Geral deste trabalho, a Fundamentação teórica sobre temas e trabalhos relacionados ao escopo e o relato do processo de desenvolvimento. Por orientação do *Product owner* do projeto foi indicado que prosseguisse o desenvolvimento deste documento. Apresentando todas as *Sprints* do projeto.

Na segunda reunião ocorreu a apresentação da monografia, apresentando os tópicos congruentes deste trabalho. Foram apresentados o relato do processo de desenvolvimento aplicado, o conjunto de resultados apresentados e a relação sobre trabalhos futuros. Durante a realização desta *Sprint* ocorreu o teste do sistema, aonde a solução desenvolvida foi preliminarmente validada. Durante a realização desta *Sprint* foram gerados os seguintes artefatos.

a) **Entrada:** Codificação de classes, Diagramas da solução, Interfaces produzidas.

a.1) Saída: Validação da solução.

A validação desta solução, bem como a sua configuração está apresentada no capítulo 5 deste trabalho, denominado Resultados e Discussões.

a.2) Saída: Escrita da Monografia

5 RESULTADOS E DISCUSSÕES

Esta seção apresenta a validação do sistema, a partir dos testes realizados na solução desenvolvida. Bem como os métodos e processos empregados para a sua realização, demonstrando ao fim os resultados gerados.

5.1 VALIDAÇÃO

A validação apresenta o contexto ao qual a solução desenvolvida foi submetida a fim de avaliar a sua qualidade final. Verificando se as metas estipuladas para resolver os desafios referentes a produção em ambientes distribuídos foram alcançadas.

Ocorreu um teste preliminar como forma de colher amostras sobre o comportamento da solução desenvolvida em um ambiente simulado de produção. De acordo com Neves (2009) o teste de sistema verifica se os elementos foram integrados corretamente e realizam as funcionalidades atribuídas a eles.

A validação desta solução ocorreu a partir da realização de um teste entre uma equipe que simulou características de ambientes DDS. De acordo com (LOPES; AUDY, 2003) características comuns de equipes distribuídas são: (1) diferentes níveis de conhecimento, confiança, cultura, etc; (2) dispersão física e (3) dispersão temporal.

A equipe que validou esta solução consistia entre oito colaboradores com diferentes níveis de conhecimento relacionado a programação. Os colaboradores são discentes de três cursos de engenharia e dois da tecnologia em análise da Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio (UTFPR-CP). Sendo eles três alunos da Engenharia Elétrica, um aluno da Engenharia de Controle e Automação, três alunos da Engenharia da Computação e dois alunos de Análise e Desenvolvimento de Sistemas.

A diferença entre cursos procurou aumentar a similaridade com ambientes distribuídos de produção, de acordo com Ferreira et al 2013 um projeto executado entre alunos de cursos distintos pode ser classificado como um cenário distribuído de nível dois, sendo que a classificação

ocorre entre três cenários possíveis.

Como foi relatado, a equipe de colaboradores também possuía disparidade conceitual, os integrantes estão em diferentes semestres do curso, sendo que haviam quatro integrantes que estavam no 6º período, dois no 5º período, dois integrantes estavam no 2º e outro no 1º. A Tabela 5 apresenta a configuração da equipe:

Tabela 5: Configuração da Equipe Responsável pela Validação

Colaborador	Semestre	Curso	Função
Colab1	6º	Análise e Des. Sistemas	Administrador
Colab2	6º	Análise e Des. Sistemas	Gerente
Colab3	6º	Engenharia Elétrica	Colaborador
Colab4	6º	Engenharia Elétrica	Colaborador
Colab5	5º	Engenharia Elétrica	Colaborador
Colab6	2º	Engenharia da Computação	Colaborador
Colab7	2º	Engenharia da Computação	Colaborador
Colab8	1º	Engenharia da Computação	Colaborador
Colab9	5º	Engenharia Cont. e Automação	Colaborador

A Tabela 5 apresentou a configuração dos colaboradores cadastrados na equipe. Devido a disparidade conceitual, foi definido que a atividade a ser desenvolvida seria uma a qual todos possuísem conhecimento, que pudesse ser avaliada e que gerasse algum artefato ao fim de seu desenvolvimento. Quando toda a equipe enviasse os artefatos produzidos, o Gerente verificava os aspectos da produção e a qualidade do artefato. Foi definido que os colaboradores deveriam desenvolver um algoritmo sobre pesquisa ou ordenação de vetores, preferencialmente utilizando a linguagem C. Esta linguagem foi definida como preferencial, pois todos os integrantes já a haviam utilizado durante o curso. Devido ao fato de ser disciplina comum da grade geral dos cursos de todos os integrantes.

Tal atividade foi cadastrada na solução desenvolvida, junto com a relação de colaboradores da equipe. Então a atividade foi atribuída a todos os colaboradores da equipe. Cada colaborador deveria desenvolvê-la a partir das especificações, a solução registrava o tempo gasto e ao fim deste processo, quando todos os colaboradores enviassem os artefatos, ocorria a verificação de qualidade e de produção.

5.2 RESULTADOS

A fase de validação ocorreu durante dois dias distintos de trabalho. Em virtude da dispersão temporal que cada colaborador da equipe possuía. Foi registrado o tempo próximo a quatro horas trabalhadas para a execução da atividade. A solução desenvolvida armazenou

os registros referentes ao tempo e os artefatos produzidos de todos os integrantes. Foi verificado que todos os colaboradores conseguiram gerar o artefato, entretanto, dois artefatos não se enquadraram nas especificações registradas.

Essa margem representa que dois colaboradores não desenvolveram a atividade de maneira satisfatória, apresentando as seguintes não conformidades:

- 1 artefato não atendeu a capacidade preestabelecida do vetor.
- 1 artefato não declarou as bibliotecas necessárias

Se faz necessário destacar que todos os artefatos gerados poderiam ser utilizados, entretanto estes listados continham o risco de apresentar falhas ou avisos de verificação se executados em ambientes não configurados. Por isso, os colaboradores que tiveram o artefato aprovado, receberam a modificação de status da sua atividade para concluído, enquanto os colaboradores responsáveis pelos artefatos com não conformidades permaneceram com o status da sua atividade como em desenvolvimento.

Ao fim dessa verificação, foi solicitado que os colaboradores responsáveis pelos artefatos reprovados adaptassem o código de acordo com as especificações da atividade. Os integrantes responsáveis modificaram a sua codificação, adequando-se a especificação pré-definida e novamente registraram o tempo gasto neste retrabalho e ao fim enviaram o artefato gerado.

Os novos artefatos foram aprovados, e as não conformidades encontradas durante a fase de verificação foram solucionadas.

5.3 DISCUSSÕES

O teste apresentado procurou verificar se a solução desenvolvida poderia mensurar a produção de uma equipe, verificando também a qualidade dos artefatos gerados durante o desenvolvimento de uma atividade.

Foi constatado que a solução conseguiu auxiliar esta mensuração, apresentando dados consistentes referentes a produção individual de todos os colaboradores. Sendo possível distinguir qual colaborador foi responsável por elaborar um artefato, e se este apresentou adequação a descrição ou falha. Também foi possível verificar o tempo utilizado para a realização de uma atividade, verificando qual horário de início e término, além dos dias trabalhados para a realização da atividade.

A partir do fluxo de trabalho apresentado na Figura 7, foi constatado que a validação preliminar efetuada conseguiu atingir os seguintes objetivos:

O Administrador do projeto conseguiu cadastrar o processo de desenvolvimento, autorizar o início/fim do projeto e conceder a Gerência de uma equipe à um Colaborador.

O Gerente conseguiu realizar a coordenação de sua equipe, acompanhando o status de desenvolvimento das atividades, qualidade do produto final e produção de sua equipe. Entretanto, a instanciação do processo e sua customização não foram englobadas dentro desta validação. Pois, a mesma atividade foi realizada por todos os membros, sendo que esta já havia sido cadastrada anteriormente pelo Administrador do projeto.

Ao Colaborador, conseguiu-se realizar a solicitação de gerência, iniciar/finalizar a execução de uma atividade e envio de um Artefato.

Esta validação apresentou um ciclo em que pode ser verificada a qualidade de um artefato gerado através da ótica de um gerente. Listando as não conformidades que foram encontradas e os colaboradores responsáveis por elas.

Contudo, o ambiente ao qual a verificação ocorreu procurar aproximar da realidade DDS, apresentando colaboradores com conceitos em diferentes níveis e até mesmo variados horários disponíveis para a execução de uma tarefa. Entretanto a realidade DDS pode apresentar mais disparidades, sobretudo com fronteiras geográficas maiores entre os colaboradores ou equipes. Devido a essa questão, a solução desenvolvida vem sendo revalidada com o desenvolvimento de um artigo referente a este trabalho e o seu produto final.

Este artigo apresentará a solução desenvolvida e será gerado um novo teste, revalidado no primeiro semestre de 2015 a partir da construção de um novo ambiente simulado de produção. Este ambiente será construído a partir de algumas características mais elaboradas referente a produção DDS, e vai procurar aumentar a qualidade do produto final deste trabalho.

6 CONCLUSÃO

O desenvolvimento deste trabalho apresentou uma solução de software para auxiliar a gestão de produção em projetos de desenvolvimento distribuído. Para alcançar esse objetivo, foram estudados pontos referentes aos processos de produção e o ambiente de desenvolvimento distribuído de software para acrescentar maior qualidade no trabalho final.

Foi verificado que os sistemas atuais disponíveis para auxiliarem o gerenciamento de produção apresentam lacunas sobre a coordenação de equipes, controle de *Log's* individuais de trabalho e utilização referente ao ciclo de vida. Estes sistemas foram concebidos em grande parte para gerenciar a produção tradicional de software, aonde não existe uma grande distância física e temporal entre colaboradores ou equipes que desenvolvem um mesmo projeto ou parte dele. Com isso, a eficiência, o planejamento e o gerenciamento de produção em projetos DDS podem ser subutilizados.

A solução final desenvolvida procurou adequar-se a realidade de ambientes distribuídos, utilizando diferentes atores com permissões e deveres. Além disso, permitiu o controle e personalização de um processo de desenvolvimento, com as suas referentes fases e atividades. Destacando a possibilidade em realizar de maneira simples a mensuração sobre dados relevantes entre um projeto distribuído, referente a suas equipe e os seus colaboradores.

O produto final apresentado mostrou-se ser eficiente para auxiliar o gerenciamento de produção distribuída. Porém, deve ser levado em consideração o fato que a validação efetuada pode ter limitado os resultados mais concisos. Por isso, esse fator vem sendo avaliado a fim de apresentar mais uma etapa de validação referente à solução aqui apresentada.

No desenvolvimento distribuído de software, muitos problemas localmente sutis ganham proporções mais acentuadas em função da distância física. O desenvolvimento desta solução procurou minimizar os desafios referentes ao gerenciamento de produção, a coordenação de atividades, a identificação de papéis, a organização de equipes e a utilização de processo de desenvolvimento em projetos que utilizam o ambiente DDS.

O solução final apresentada também ofereceu a possibilidade de verificar quais equi-

pes/colaboradores estão sobrecarregados. Permitindo delegar atividades para setores mais ociosos ou fragmentar o escopo do trabalho a fim de ocorrer a divisão entre outras equipes. Desta forma, o produto final deste trabalho procurou auxiliar o gerenciamento de produção em ambientes distribuídos, oferecendo maior qualidade durante a aplicação do DDS.

6.1 TRABALHOS FUTUROS

Com relação aos trabalhos futuros referentes ao desenvolvimento deste projeto, vem sendo adotado as seguintes sugestões.

Primeiramente, é destacando o desenvolvimento de um artigo referente à experiência gerada por este trabalho. Este artigo apresentará as características da produção distribuída, demonstrando ao fim uma proposta de solução arquitetônica utilizada no desenvolvimento desta solução. Para isto, o produto final deste projeto provavelmente será revalidado, procurando apresentar resultados mais precisos sobre o gerenciamento de produção DDS.

Apesar do objetivo final deste trabalho tratar-se unicamente sobre a produção em ambientes distribuídos, pretende-se que futuramente a solução apresentada possa ser aprimorada. Para isso, pontos significantes em projetos DDS deverão ser alinhados. Uma contribuição futura seria oferecer ao produto final mecanismos de comunicação entre colaboradores de um projeto. A fim de minimizar ainda mais os desafios de gerenciamento de produção de software em ambientes distribuídos.

REFERÊNCIAS

- ALENCAR, D. C. P. et al. Proposta de Protótipo de Monitor de Temperatura Corporal Utilizando Software Embarcado e Tempo Real Baseado em Computação Ubíqua. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO DE ENGENHARIA, Gramado, RS, Brasil. **Proceedings...** [S.l.]: Cobenge, 2013. p. 1–12.
- ANTUNES, J. **Sistemas De Produção: Conceitos E Práticas Para Projetos E Gestão Da Produção Enxuta**. Porto Alegre: Bookman, 2008.
- AUDY, J. L.; PRIKLADINICKI, R. **Desenvolvimento Distribuído de Software**. Porto Alegre: Elsevier, 2007.
- BOCK, H. **Java Persistence API. The Definitive Guide To Netbeans Platform 7**. New York: Apress, 2011.
- BOOCH, G. **The Unified Modeling Language Reference Manual**. Los Angeles: The. Pearson Higher Education, 2004.
- CARVALHO, B. V.; MELLO, C. H. P. Revisão, Análise e Classificação da Literatura Sobre o Método de Desenvolvimento de Produtos Ágil Scrum. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAC OES INTERNACIONAIS, São Paulo, SP, Brasil. **Anais...** [S.l.]: SIMPOI, 2009. p. 1–16.
- CARVALHO, M.; PALADINE, E. **Gestão da Qualidade**. Rio de Janeiro: Editora Campus No Prelo, 2005.
- CUNHA, G. D. **Um Panorama da Engenharia de Produção**. 2002. Disponível em: <www.abepro.org.br>. Acesso em: 10 mai. 2014.
- GOULART, C. **Proposta de um Modelo de Referência para Planejamento e Controle da Produção em Empresas Virtuais**. Dissertação (Mestrado em Engenharia de Produção) — Universidade de São Paulo, São Carlos, Brasil, 2000.
- GROUP, S. **The Chaos Manifest**. 2014. Disponível em: <<http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>>. Acesso em: Acesso em 14 de abril de 2014.
- HASSAN, S. S.; ISSAC, R. K. **An Integrated Approach of MAS CommonKADS, Model View Controller and Web Application Optimization Strategies for Web-Based Expert System Development**. 2011. Disponível em: <<http://www.sciencedirect.com/science/journal/09574174/38/1>>. Acesso em: 06 de junho de 2014.
- ILHAM, A. A.; MURAKAMI, K. Evaluation and optimization of java object ordering schemes. In: INTERNATIONAL CONFERENCE ELECTRICAL ENGINEERING AND INFORMATICS, Bandung. **Proceedings...** [S.l.]: IEEE, 2011. p. 253–262.

L'ERARIO, A. **M3DS: Um Modelo de Dinâmica de Desenvolvimento Distribuído de Software**. Dissertação (Doutorado em Engenharia de Produção) – Escola Politécnica de São Paulo, Universidade de São Paulo, São Paulo,

LOPES, L. T.; AUDY, J. L. N. Em Busca de um Modelo de Referência para Engenharia de Requisitos em Ambientes de Desenvolvimento Distribuído de Software. In: INTERNATIONAL WORKSHOP ON REQUIREMENTS ENGINEERING, Piracicaba, SP, Brasil. **Anais...** [S.l.]: IWRE, 2003. p. 1–8.

LUDER, A.; ROSENDAHL, R.; SCHIMDT, N. Validation of Behavior Specifications of Production Systems Within Different Phases of the Engineering Process. In: CONFERENCE EMERGING TECHNOLOGIES E FACTORY AUTOMATION, Cagliari, Italy. **Proceedings...** [S.l.]: IEEE, 2013. p. 1–8.

MARCKZAK, S. Towards a Communication Maturity Model for Distributed Software Development. In: GLOBAL SOFTWARE ENGINEERING WORKSHOPS, Bary. **Proceedings...** [S.l.]: IEEE, 2013. p. 81–83.

MIGUEL, P. A. C. **Metodologia De Pesquisa Em Engenharia De Produção E Gestão De Operações**. Rio de Janeiro: Elsevier, 2012.

MUSCIANO, C.; KENNEDY, B. **HTML and XHTML: The Definitive Guide**. Sebastopol: O'Reilly Media, 1997.

NEVES, V. d. O. **Teste de Integração Contextual de Programas Orientados a Objetos e a Aspectos: critérios e automação**. Dissertação (Mestrado em Ciências Matemática e de Computação) — Universidade de São Paulo, São Carlos, Brasil, 2009.

OLIVEIRA, J. P. N. d. et al. Desafios no Gerenciamento de Conflitos em Projetos de Desenvolvimento Distribuído de Softwares. In: WORKSHOP DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE, Brasília, Brazil. **Anais...** [S.l.]: CBSoft, 2013. p. 101–108.

OLIVEIRA, O. J. **Gestão Da Qualidade-Tópicos Avançados**. São Paulo: Cengage Learning Editores, 2003.

PALADINI, E. P. **Gestão Da Qualidade No Processo: A Qualidade Na Produção De Bens E Serviços**. São Paulo: Campus, 1995.

PEREIRA, P.; PAULA, T. ao; MARÇAL, A. S. Entendendo Scrum para Gerenciar Projetos de Forma Ágil. In: INTERNATIONAL ENTERPRISE DISTRIBUTED OBJECT COMPUTING CONFERENCE WORKSHOPS, Curitiba, PR, Brasil. **Anais...** [S.l.]: Mundo PM-Bok, 2007. p. 64–71.

PRIKLADNICKI, R.; ROCHA, R.; SILVA, F. Q. B. Desafios e Boas Práticas para o Gerenciamento de Projetos no Desenvolvimento Distribuído de Software. In: WORKSHOP DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE, Salvador, BA, Brazil. **Anais...** [S.l.]: CBSoft, 2010. p. 66–69.

ROBINSON M, K. **Offshore Outsourcing: Business Modelos,ROI and Best Pratices**. EUA: Mivar Press, 2009.

SHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. New Jersey: Prentice Hall, 2002.

SILVA, D. M. D.; MACHADO, G. B. M. Guilherme b. m. Serviço para Execução de Processamento em Lotes (batch) com Agendamento e Gerenciamento pela Web. In: CONGRESSO SUL BRASILEIRO DE COMPUTAÇÃO, Criciã^oma, SC, Brazil. **Proceedings...** [S.l.]: Sul-comp, 2010. p. 1–10.

STONEBRAKER, M.; ROWE, L. The design of postgres. In: INTERNACIONAL CONFERENCE ON MANAGEMENT OF DATA, Washington, EUA, States United. **Anais...** [S.l.]: ACM, 1986. p. 340–355.

MUNDRA, A. et al. Practical Scrum–Scrum Team: Way to Produce Successful and Quality Software. In: INTERNATIONAL CONFERENCE COMPUTATIONAL SCIENCE AND ITS APPLICATIONS, Ho Chi Minh. **Proceedings...** [S.l.]: IEEE, 2013. p. 119–123.

VASCONCELLOS, M. A. S.; GARCIA, M. E. **Fundamentos De Economia**. São Paulo: Saraiva, 2004.

ZIMMERMANN, T. et al. Improving bug tracking systems. In: **Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on**. [S.l.: s.n.], 2009. p. 247–250.

APÊNDICE A - DOCUMENTO DE REQUISITOS

Introdução

Este documento contempla as especificações de funcionalidades da solução a ser desenvolvida, a fim de oferecer uma ferramenta para auxiliar o gerenciamento de produção em ambientes DDS. Desta forma serão implementados os seguintes conjuntos de requisitos.

Requisitos Funcionais

Esta seção descreve de maneira sumarizada as principais funcionalidades que o sistema irá realizar. Desta forma, foram definidos os seguintes requisitos funcionais.

Tabela 6: Requisitos Funcionais

ID	Nome	Descrição
RF01	Cadastro de Usuario	Opção de um usuário externo cadastrar-se no sistema utilizando nome, login, senha e função em um projeto.
RF02	Seleção de Função	Permitir ao usuário escolher entre a opção de colaborador ou administrador de um projeto.
RF03	Autenticação de usuário	Realizar autenticação de usuários no sistema a partir da utilização de um login e senha.
RF04	Autenticar Gerente	Oferecer aos Administradores opções de autenticação de Gerentes em seus projetos.
RF05	Ciclo de Vida	Permitir a utilização de um ciclo de vida, com um conjunto de fases e atividades.
RF05.1	Fases	Capacidade de cadastrar todas as fases de um ciclo de vida.
RF05.2	Atividades	Capacidade de cadastrar as atividades referentes a cada fase, e suas atividades antecessoras e sucessoras.
RF06	Status do Projeto	Definir o status atual do projeto [Desenvolvimento/Concluído/Cancelado].
RF07	Gerência de Equipe	Possibilidade de gerenciar colaboradores e atividades alocados nas equipes de um projeto DDS.
RF08	Gerência de Atividades	Possibilitar que as atividades cadastradas no ciclo de vida possam ser delegadas aos colaboradores alocados nas equipes.
RF09	Gerência de Log's	Capacidade em verificar dia, hora e colaborador responsável pela execução de atividades.
RF10	Envio de Artefatos	Permitir que ao finalizar uma atividade, possa ser enviado o artefato que foi produzido.

Requisitos Não Funcionais

O desenvolvimento do sistema levará em consideração aspectos limitadores referentes a utilização das tecnologias empregadas e ao obrigatoriedade em estabelecer a seção de um usuário para conseguir mensurar dados técnicos referentes a sua produção:

Tabela 7: Requisitos Não Funcionais

ID	Nome	Descrição
NF01	Timeout de sessão	A sessão de cada usuário do sistema dependerá diretamente sobre a implementação de Log's individuais.
NF02	Compatibilidade de navegadores	O sistema final pode apresentar erros de exibição em navegadores antigos ou desatualizados que não oferecem suporte aos padrões de programação adotados.
NF03	Linguagem e Padrão Adotado	O sistema final será desenvolvido baseando na tecnologia Java web, utilizando JSF, JPA e Frameworks de apoio..

Escopo Negativo

O desenvolvimento da solução final não contemplará características referentes a comunicação de um projeto DDS, desta forma não serão oferecidos mecanismos para troca de mensagem entre usuários.

APÊNDICE B – ATAS DAS REUNIÕES

Tabela 8: Ata da Reunião 1

Data	Pauta
28/07/2014	Organização do ambiente de versionamento
Pauta	
Foi definido e organizado o ambiente para o versionamento do projeto e da monografia. Optou-se pelo ambiente Source Forge, aonde já existia um projeto inicial cadastrado que constava com arquivos gerados pela IDE de programação. Desta maneira foi definida a estrutura de pastas deste ambiente que suportará a organização sobre a documentação do projeto, as páginas e <i>scripts</i> relacionados ao projeto e os artefatos desenvolvidos.	
Tarefas para a próxima reunião	
Modelo objeto-relacional, Execução do projeto e Correção da proposta.	

Tabela 9: Ata da Reunião 2

Data	Pauta
04/08/2014	Definições de Regras de Negócios
Pauta	
Ocorreu a definição sobre as regras de permissões entre Administrador e Gerente do projeto. Optou-se pelo Administrador realizar a gerencia sobre um projeto e a definição de um processo (ciclo de vida), derivando ao Gerente a tarefa de especificação desse processo, bem como as suas atividades. Também foi definido que um Usuário (membro/gerente) poderá estar em diversos projetos simultaneamente.	
Tarefas para a próxima reunião	
Refinação do CRUD de <i>Log's</i> individuais	

Tabela 10: Ata da Reunião 3

Data	Pauta
26/08/2014	Revisão de alguns CRUD?s implementados
	<p>Pauta</p> <p>Foi apresentado de uma forma geral a implementação do projeto (CRUD?s). As observações feitas pelo professor dizem respeito ao refinamento da regra de negócios sobre o cadastramento e utilização de um processo de desenvolvimento em um projeto. Desta forma foi definido que um Administrador (Gerente de Processo) poderá cadastrar um novo processo, este deverá obrigatoriamente possuir um nome, uma descrição e o total de fases. Também será necessário que o Administrador cadastre cada fase, definindo as atividades desenvolvidas, bem como as antecessoras e sucessoras. Com isso ele pode criar um projeto e selecionar os Gerentes de Projeto. Ao Gerente de Projeto caberá a responsabilidade de instanciar o processo, ele pode optar por remodelar o processo [modificando fases/atividades/sucessoras/antecessoras] e também definir qual membro da sua equipe deverá desenvolver determinada atividade. Foi aconselhado a implementação de um painel, estilo Kanban, aonde esteja disponível ao membro da equipe a atividade que foi liberada para ele desenvolver e a atividade sucessora já desenvolvida e/ou seu status de desenvolvimento. Também foi feita a sugestão de o Gerente de Projeto possuir a permissão de visualização do Kanban de sua equipe. Podendo definir se os membros de sua equipe apenas poderão visualizar os status da sua atividade, ou atividades de toda a equipe. Recomendação 1: Se o Gerente de Projeto instanciar uma nova versão de um Processo de Desenvolvimento, somente ele (Gerente) poderá visualizar e utilizar. Recomendação 2: O Administrador poderá visualizar e utilizar todas as versões de todos os Administradores e instâncias dos Gerentes.</p> <p>Tarefas para a próxima reunião</p> <p>Implementar a nova regra de negócios definida pelo orientador</p>

Tabela 11: Ata da Reunião 4

Data	Pauta
08/09/2014	Apresentação das correções da última reunião
	<p>Pauta</p> <p>Foi apresentada a correção sobre os CRUD?s de cadastro de processo, projeto, solicitação de gerência e gerente. O orientador falou ser interessante implementar a interface do gerente com um combo inicial carregando os projetos ao qual ele está ligado. Também, que carregue uma tabela geral sobre os projetos ao qual ele participa como gerente e possua alguma pendência, por exemplo, cadastre uma equipe; esta tarefa está sem atribuição a um membro, etc.. Também é interessante implementar uma tabela que ele possa estimar o total de horas gastas em uma determinada atividade, à exemplo de grandes empresas. O orientador disse que o teste do sistema final poderá ocorrer dentro do experimento de oficina de integração, ou sugeriu que ocorra na turma de Calouros do curso de Análise.</p> <p>Tarefas para a próxima reunião</p> <p>Implementar as interfaces do Gerente e seus recursos</p>

Tabela 12: Ata da Reunião 5

Data	Pauta
22/09/2014	Apresentação das interfaces do Gerente
Pauta	
<p>Foi apresentado as interfaces responsáveis pelo cadastro de informação do usuário Gerente. Desta forma foi concluído as interfaces de cadastro sobre equipes, atividades, atividades por membro. Também ocorreu a apresentação da interface responsável pelo log de usuário, upload de artefato e download de artefato. Por último foi apresentado o novo estilo do projeto, com base nas modificações realizadas no CSS do sistema.</p>	
Tarefas para a próxima reunião	
Otimizar a interface para realizar a construção dos relatórios	

Tabela 13: Ata da Reunião 6

Data	Pauta
07/10/2014	Apresentação do Projeto.
Pauta	
<p>Foram concluídos todos os Casos de Uso referente ao Cadastro de informações no sistema. A partir de agora ficou estabelecido que ocorrerá o desenvolvimento sobre a realização de opção de editar e excluir. Foi recomendado procurar uma maneira de deixar a aplicação online para testes iniciais, desta forma será pesquisada a forma de como proceder com este passo. Também foi recomendado o desenvolvimento de um passo-a-passo para a utilização do sistema. Visto que se o projeto ficar online, deverá existir um documento que possa nortear possíveis usuários.</p>	
Tarefas para a próxima reunião	
Desenvolvimento das opções de Alterar e Excluir. Pesquisar métodos e processos para deixar a aplicação online. Manual de utilização para usuários.	

Tabela 14: Ata da Reunião 7

Data	Pauta
27/10/2014	Escrita da Monografia.
Pauta	
<p>Foi definido a estrutura que o documento de monografia possuirá, entre eles deu-se uma ênfase no desenvolvimento do capítulo referente a aplicação do processo de desenvolvimento. De forma a justificar a utilização do <i>Scrum</i></p>	
Tarefas para a próxima reunião	
Apresentar tópicos iniciais da monografia	

Tabela 15: Ata da Reunião 8

Data	Pauta
17/11/2014	Apresentação e correção da Monografia.
	Pauta Foi apresentado o documento final do projeto para o orientador. Ocorreu algumas breves considerações referente a estrutura que foi adotada. Por último procurou-se adequar o documento nas especificações do orientador
	Tarefas para a próxima reunião Não possui.

APÊNDICE C – INTERFACES DO SISTEMA

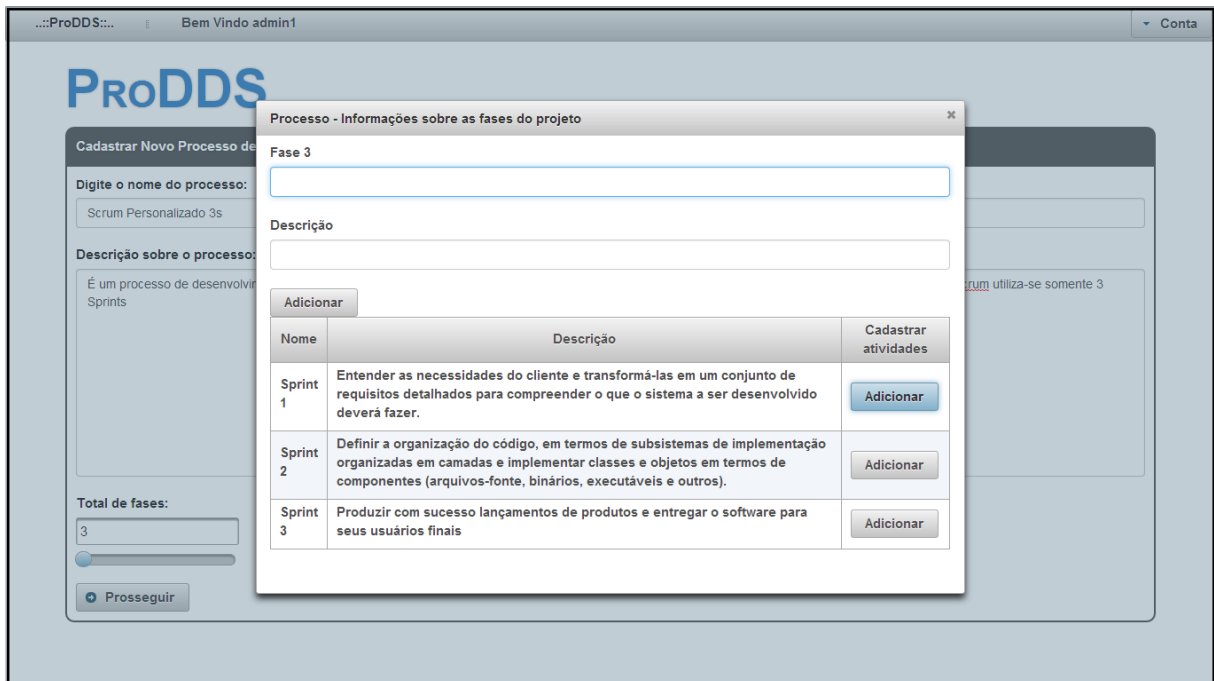


Figura 15: Interface de Cadastro de Ciclo de Vida.

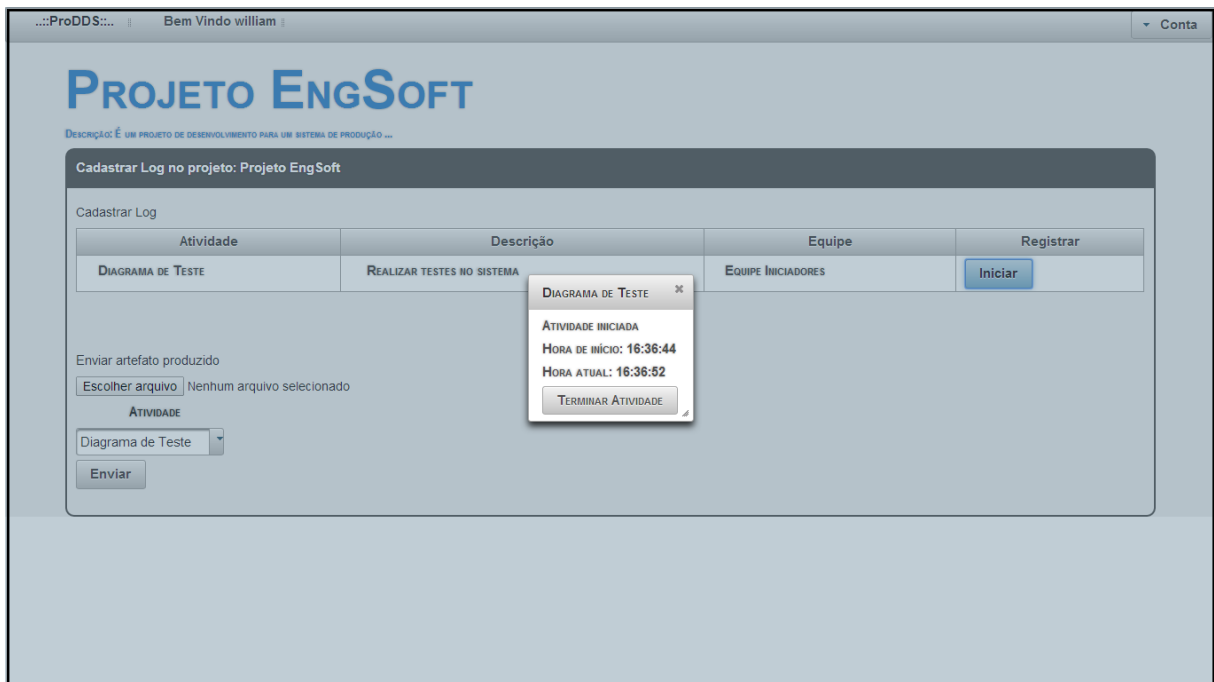


Figura 16: Interface de Log's.

...ProDDS... Bem Vindo gerente1 Conta

ProDDS

DESCRIÇÃO: PROJETO PARA O DESENVOLVIMENTO DE UM GESTOR DE PRODUÇÃO EM AMBIENTES DE DESENVOLVIMENTO DISTRIBUÍDO

Relatórios - Gerente

Pesquisar: Usuário

Log's individuais dos membros da sua equipe

Colaborador	Atividade	Hora de início	Hora de término	Data de início	Data de término
william	Definição do tema	16:13:11	16:13:11	21/09/2014	21/09/2014
william	Definição do tema	16:23:02	16:24:13	21/09/2014	21/09/2014
william	Definição do tema	19:13:27	19:13:43	26/09/2014	26/09/2014
Gerente 2	Diagrama de testes	18:12:34	18:13:13	27/09/2014	27/09/2014
william	Definição do tema	20:06:51	20:06:51	08/10/2014	08/10/2014

localhost:8084/TCCBackup/Relatorios/relatorioGer.xhtml

Figura 17: Interface de Relatório.