

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE
DESENVOLVIMENTO DE SISTEMAS**

RUBEM ALVES CARDOSO

**UMA ANÁLISE DO KANBAN EM CENÁRIOS DE
DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2015

RUBEM ALVES CARDOSO

**UMA ANÁLISE DO KANBAN EM CENÁRIOS DE
DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE**

Trabalho de Conclusão de Curso apresentada ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientador: Prof. Dr. Alexandre L'Erario

CORNÉLIO PROCÓPIO

2015

RESUMO

CARDOSO, Rubem. UMA ANÁLISE DO KANBAN EM CENÁRIOS DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE. 46 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Análise Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

É cada vez maior o número de empresas a adotar o Desenvolvimento Distribuído de Software como uma maneira de reduzir custos, empregar profissionais e abranger maiores mercados. Com isso há a preocupação de que alunos egressos tenham ao menos a preparação para enfrentar tais desafios de mercado. Assim, este estudo propõe analisar o uso do Kanban, que é bastante adotado em outras áreas de produção e ainda relativamente novo no desenvolvimento de software, para auxílio dos cenários de DDS. O método a ser utilizado para validação do estudo, em que grupos de alunos de diferentes períodos trabalham em um ambiente distribuído simulado para desenvolver um projeto em comum foi o experimental. Para uma abordagem inicial é feita a descrição dos conceitos, ferramentas, métodos utilizados e dos participantes. Os dados foram coletados através de questionários e aplicados testes chi-quadrado e binomial para verificar a validade das respostas. Por fim, a análise e apresentação destes dados, concluindo assim, com os resultados dos testes e o que foi observado durante os experimentos.

Palavras-chave: Desenvolvimento Distribuído de Software, Kanban, Graduação

ABSTRACT

CARDOSO, Rubem. AN ANALISYS OF KANBAN ON DISTRIBUTED SOFTWARE DEVELOPMENT ENVIRONMENTS. 46 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Análise Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

A growing number of enterprises are adopting Distributed Software Development as a way to reduce costs, employ professionals and include large markets. There is a concern that graduate students are prepared to face those challenges. Therefore, the goal of the present study is to analyze the usability of Kanban philosophy, which is largely adopted in other production areas and yet still relatively new in software development to help the DDS scenery. The method, in which groups of undergraduates work on a simulated distributed environment in order to develop a common project, was the experimental one. An initial approach was the description of concepts, tools and methods used. After that, an analysis and presentation of collected data, ending with a larger view of what was observed.

Keywords: Distributed Software Development, Global Software Development, Kanban, Graduate

LISTA DE FIGURAS

FIGURA 1	– Quadro Kanban	16
FIGURA 2	– Projeto Logo	23
FIGURA 3	– Análise de perfil dos participantes	29
FIGURA 4	– Projeto Logo	44
FIGURA 5	– Mapa de Atividades 1	45
FIGURA 6	– Mapa de Atividades 2	46

LISTA DE TABELAS

TABELA 1	– Tabela de possíveis respostas para questionário	23
TABELA 2	– Tabela de Métricas para teste Binomial	24
TABELA 3	– Modelo de avaliação do quadro kanban	27
TABELA 4	– Mediana e moda do atributo Necessidade	29
TABELA 5	– Mediana e Moda do atributo Utilidade	30
TABELA 6	– Mediana e Moda do atributo Adaptação	30
TABELA 7	– Dados dispostos de maneira explícita	30
TABELA 8	– Dados dispostos de maneira descritiva	31
TABELA 9	– Matriz chi-quadrado - Necessidade	31
TABELA 10	– Matriz chi-quadrado - Utilidade	32
TABELA 11	– Matriz chi-quadrado - Adaptação	32
TABELA 12	– Valores esperados de cada tabela (N;U;A)	32
TABELA 13	– χ^2 dos atributos de Necessidade	33
TABELA 14	– χ^2 dos atributos de Utilidade	33
TABELA 15	– χ^2 dos atributos de Adaptação	34
TABELA 16	– χ^2 das tabelas 9, 10 e 11	34
TABELA 17	– Práticas de acordo com Grau de Liberdade	35
TABELA 18	– Análise Qualitativa	36
TABELA 19	– Questionário de Perfil do Participante	43

LISTA DE SIGLAS

DDS	Desenvolvimento Distribuído de Software
WIP	Work In Progress
TOC	Theory of Constraints

LISTA DE SÍMBOLOS

χ^2	chi-quadrado
Σ	Símbolo de um somatório ou de variáveis estatísticas
GL	Grau de Liberdade

SUMÁRIO

1	INTRODUÇÃO	9
1.1	PROBLEMATIZAÇÃO	10
1.2	JUSTIFICATIVA	10
1.3	OBJETIVOS	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos Específicos	12
1.4	ORGANIZAÇÃO DO TEXTO	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE	13
2.2	GERENCIAMENTO DE CONFLITOS NO DDS	15
2.3	O ENSINO DE DDS NA GRADUAÇÃO	15
2.4	KANBAN	15
2.5	TRABALHOS RELACIONADOS	18
3	METODOLOGIA DE PESQUISA	20
3.1	CARACTERIZAÇÃO DA PESQUISA	20
3.1.1	Ferramentas	20
3.1.2	Definição do contexto	21
3.1.3	Seleção dos indivíduos	21
3.1.4	Questões	21
3.1.5	Hipótese	22
3.1.6	Questionário de Perfil	22
3.1.7	Descrição da Instrumentação	22
3.1.8	Variáveis	24
3.1.8.1	Variável independente	24
3.1.8.2	Variáveis dependentes	24
3.1.9	Análise Qualitativa	25
3.1.10	Validade	25
3.1.11	Análise dos Dados	26
4	ANÁLISE DOS DADOS	27
4.1	MAPA DAS ATIVIDADES	27
4.2	PROCESSO DA EXPERIMENTAÇÃO	27
4.3	PERFIL DOS PARTICIPANTES	28
4.4	APLICAÇÃO DAS MÉTRICAS ESTATÍSTICAS	29
4.5	ANÁLISE QUANTITATIVA	35
4.6	ANÁLISE QUALITATIVA	36
4.7	RESULTADOS	37
5	CONCLUSÃO	38
	REFERÊNCIAS	40
	Apêndice A – QUESTIONÁRIO DE PERFIL DO PARTICIPANTE	43
	Apêndice B – FLUXO DE EXECUÇÃO DO PROJETO	44
	Apêndice C – MAPA DAS ATIVIDADES DE ANÁLISE DOS DADOS	45

1 INTRODUÇÃO

Desde a antiguidade projetos são executados pelo homem nos mais variados campos de aplicação. Era necessário gerenciar os recursos, tanto materiais quanto operacionais. Muitas das maiores criações da humanidade, como as pirâmides por exemplo, não têm documentação que retrate como suas etapas de desenvolvimento foram percorridas. Para que um projeto fosse entregue em seu tempo estipulado, com a qualidade pretendida e gerando o mínimo retrabalho possível o ser humano veio a perceber que uma boa gestão dos mesmos era importante para que não ultrapassassem tais limites orçamentais e não levassem mais tempo do que o necessário. Após esta constatação, diversas técnicas para uma melhor gestão foram sendo criadas.

Um dos mais importantes nomes, destes que contribuíram com tais técnicas, é Henry Gantt (1861 - 1919), conhecido pelo famoso gráfico de barras (atualmente conhecido como Gráfico de Gantt). Uma vez que estas técnicas de gestão começaram a ser utilizadas, elas também passaram a ser aprimoradas. Com isso, e com o advento da tecnologia, surgiram algumas ferramentas para auxiliar as equipes e gerentes a coordenar e planejar a vida de um projeto. Algumas das mais conhecidas são: *MS Project da Microsoft*, (ferramenta pertencente ao pacote *Office*, que ajuda a criar Gráficos de Gantt), o *WBS Chart Pro* da Critical Tools (ferramenta que ajuda na criação de Estruturas Analíticas de Projetos, ou EAP, que são a divisão do escopo em uma linha do tempo com as tarefas definidas em partes pertencentes a cada uma das fatias do mesmo) e o *Open Project*, uma ferramenta *open source* que também auxilia na criação de Gráficos de Gantt.

A medida que o escopo dos projetos começou a abranger cada vez mais mercados e que as barreiras geográficas passaram a não representar um impedimento para que equipes pudessem trabalhar em um mesmo projeto, surgiu o conceito de Desenvolvimento Distribuído de Software.

O DDS é um cenário onde as equipes de desenvolvimento atuam de forma geograficamente dispersa de forma colaborativa como se estivessem juntos. Apesar de possibilitar grandes vantagens ao desenvolvimento de software o DDS traz consigo algumas adversidades

características dessa dispersão.

Este estudo é baseado na utilização dos conceitos do Kanban (prática de organização visual das tarefas de um fluxo) para auxiliar o processo de gerenciamento de um projeto de Desenvolvimento Distribuído de Software (DDS). A proposta deste trabalho é a inclusão de uma ferramenta que possa facilitar a visualização de tarefas e a comunicação entre as partes interessadas de um projeto, neste caso os alunos de graduação, visando transmitir os conhecimentos para que o mesmo possa ser executado com mais precisão, menos gargalos (ou impedimentos) e visando a garantia de prazos e orçamentos.

1.1 PROBLEMATIZAÇÃO

Alguns dos principais problemas encontrados por gerentes de projetos durante o DDS é mensurar o tempo que a equipe (sua equipe ou uma equipe terceirizada) levará para realizar etapas dos projetos, lidar com o retorno de informações recebido dos membros da equipe, uma vez que estes podem estar em fusos horários diferentes uns dos outros, e avaliar se um membro desta equipe está apto a auxiliar outro em uma tarefa que demande mais tempo.

Para tanto, percebe-se a viabilidade da utilização de uma ferramenta que possa auxiliar no gerenciamento de um projeto distribuído de software e que seja flexível o bastante, utilizando os conceitos do Kanban, para se adaptar aos diversos tipos de métodos adotados pelas mais variadas equipes de desenvolvimento de software. Diante deste contexto questiona-se: o quanto tais métodos podem auxiliar desenvolvedores a garantir uma postura adequada às novas técnicas utilizadas no mercado para se trabalhar em equipes e alcançar a assertividade desejada nos projetos ao longo de sua jornada profissional?

1.2 JUSTIFICATIVA

As proporções do escopo de grandes projetos, pode trazer à tona uma série de adversidades, não enfrentadas por outros de menor escala, uma vez que o gestor do mesmo deve coordenar as equipes de modo que os recursos possam ser realocados de acordo com a possível disponibilidade dos mesmos diante das tarefas terminadas. Portanto, é fortemente recomendado que o gerente de projetos tenha a sua disposição ferramentas que facilitem essa gestão, diminuindo os possíveis lapsos entre a entrega de uma tarefa e o início de outra por seus colaboradores, diminuindo o tempo entre o recebimento de informações pertinentes ao andamento das tarefas e a realocação dos recursos presentes, como citado acima.

O Gerenciamento de projeto de software também não foge à regra do bom planejamento. Sempre há um tempo limite, um valor estipulado e recursos a se gerenciar. No entanto como toda modalidade específica, temos adversidades que representam novos obstáculos para a boa gestão, como por exemplo a realização de tarefas isoladas pelos integrantes de uma equipe ou a paralelização destas tarefas. Também, surgiram ao longo dos anos alguns conceitos de boas práticas no desenvolvimento de software, como por exemplo o *Scrum*, a Programação extrema, o Desenvolvimento de Software Adaptativo, o Desenvolvimento Guiado por Funcionalidades e o Desenvolvimento de Sistemas Dinâmico.

Primeiramente, a equipe de desenvolvimento de software pode fazer parte da sua empresa, de uma empresa terceirizada, pode ser que o projeto tenha sido dividido, parte com sua empresa e parte com a terceirizada. Ainda, como outras adversidades temos, o quadro de desenvolvedores pode não estar alocado no mesmo local físico, podem mesmo nem estar no mesmo país, sofrendo assim a influência de fusos horários diferentes. O DDS é uma modalidade de desenvolvimento de software que vem crescendo atualmente devido ao seu baixo custo. Este caracteriza-se em um projeto onde seus envolvidos estão dispersos localmente, geralmente separados por nacionalidade, cultura e fusos horários distintos, que conseguem realizar um projeto em comum, como se estivessem próximos (CARMEL, 1999), (MARQUARDT; HORVATH, 2001), (KAROLAK, 1999).

Uma outra técnica utilizada para auxiliar no andamento das tarefas realizadas por uma equipe e conseqüentemente no gerenciamento do projeto em questão é o Kanban. O nome Kanban surgiu dos sistemas de cartões usados nas indústrias de produção, que tinham como finalidade o gerenciamento do fluxo de trabalho através da organização de desenvolvimento (ANDERSON, 2010). Com este sinalizador os funcionários tinham como saber se uma tarefa já havia sido iniciada, se estava em andamento, ou até mesmo se já havia sido finalizada para que outra tarefa pudesse ser “puxada”. Este mecanismo de divisão de tarefas evita que uma mesma pessoa realize diversas tarefas como explica David J. Anderson (2010, p35.), criador do conceito por trás do Kanban, “Um interessante efeito colateral de sistemas puxados é que eles limitam o trabalho em andamento (WIP - *Work In Progress*) para certa quantidade acordada, evitando assim que trabalhadores fiquem sobrecarregados.” Como o Kanban não é considerado um método ágil e é considerado pouco prescritivo, pode ser utilizado em conjunto com os métodos ágeis numa tentativa de deixá-los mais visuais e/ou ajudar a melhorar alguns pontos fracos ou que não tenham sido bem absorvidos pela equipe de desenvolvimento. Os problemas encontrados diariamente nos projetos podem comprometer seu andamento, sua entrega e seu orçamento. Diversas ferramentas foram utilizadas ao longo dos anos para tentar reduzir estes reflexos. O que falta, entretanto, são ferramentas que visem auxiliar este processo de DDS,

visto que é um cenário que tem se popularizando recentemente. Existem, contudo, algumas ferramentas já existentes como é o caso das citadas anteriormente que serão discutidas nos trabalhos relacionados.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

O objetido deste estudo é verificar se as práticas oferecidas pelo Kanban podem ou não ser uma ferramenta auxiliar nos cenários de DDS, avaliando seu nível de aceitação pelos desenvolvedores e qualificando os resultados obtidos pelos experimentos.

1.3.2 OBJETIVOS ESPECÍFICOS

Para verificar o uso do Kanban no cenário proposto definiu-se alguns pontos específicos de observação. Através da observação dos experimentos pretende-se acompanhar a performance dos participantes com um fluxo de trabalho ordenado em que não haja sobrecarga, identificar o lapso de tempo causado por fatores adversos entre um *feedback* (retorno de informações) e outro, qualificar o resultado do projeto desenvolvido pelo nível do produto final e verificar se as entregas das etapas estão dentro do escopo e prazo estipulados, e analisar as respostas dos questionários aplicados para qualificar a experiência final dos alunos participantes do experimento.

1.4 ORGANIZAÇÃO DO TEXTO

Esta proposta está dividida em 6 capítulos, organizados da seguinte forma: uma breve introdução abordando os assuntos a serem discutidos, seguido da problematização, justificativa e objetivos do trabalho; no capítulo 2 faz-se uma abordagem mais ampla sobre os principais conceitos que embasam este trabalho, DDS e o Kanban, com suas definições e características; no capítulo 3 a metodologia da pesquisa a ser desenvolvida, as ferramentas e métodos utilizados para alcançar os objetivos propostos, o processo de experimentação, tal como a descrição do experimento a ser realizado; no capítulo 4 serão exibidas as análises dos dados e apuração dos resultados; no capítulo 5 são apresentados os resultados obtidos e por fim, no capítulo 6 são apresentadas a conclusão do trabalho, as considerações finais e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O capítulo a seguir traz os conceitos necessários para um melhor entendimento do tema proposto. Portanto, a seguir, são abordados os temas Desenvolvimento Distribuído de Software e suas modalidades, uma breve constatação do Gerenciamento de conflitos no DDS e o registro de uma aplicação do DDS em cursos de graduação, e por fim, uma sucinta explicação das práticas do Kanban e seus benefícios. Dois conceitos que ganham mercado na indústria de desenvolvimento de software, um caracteriza o ambiente de trabalho e outro oferece métricas de avaliação do esforço de uma equipe, auxiliando a gerência do projeto.

2.1 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

O DDS é uma modalidade de desenvolvimento que vem crescendo atualmente devido ao seu baixo custo. Caracteriza-se em um projeto com os envolvidos dispersos localmente, geralmente separados por nacionalidade, cultura e fusos horários distintos, mas conseguem realizar um projeto em comum, como se estivessem próximos (CARMEL, 1999), (MARQUARDT; HORVATH, 2001), (KAROLAK, 1999), (ROBINSON; KALAKOTA, 2004), (MOCZADLO, 2002), (L'ERARIO et al., 2014).

Essa modalidade de desenvolvimento tem se tornando popular entre as grandes empresas, uma vez que pode trazer benefícios como maior alcance aos clientes, redução de custos, possibilidade de ter profissionais mais qualificados e atender a um âmbito global (CARMEL; TJIA, 2005).

O contexto de desenvolvimento distribuído também tem se mostrado uma boa alternativa para o ensino, com o auxílio e popularização dos cursos e das ferramentas de ensino a distância (*Learning management system*) (EDWARDS et al., 2010). A escolha também acrescenta experiência válida aos alunos em nível de graduação, uma vez que o crescimento desse cenário é visível no mercado de tecnologia (L'ERARIO et al., 2008).

Dentro desta modalidade de desenvolvimento, observam-se quatro níveis distintos: o

offshore, o *onshore*, o *insourcing* e o *outsourcing*.

O desenvolvimento *offshore*, pode ser realizado dentro da empresa contratada, ou através de uma empresa terceirizada, a qual forneceria os serviços de um outro país (*outsourcing*) (ROBINSON; KALAKOTA, 2004). Segundo Moczadlo (2002), as maiores razões para se adotar o desenvolvimento *offshore* é a redução de custos, número flexível de funcionários, melhoria na qualidade e tempo de mercado reduzido.

O desenvolvimento *onshore* é aquele fornecido por uma indústria que se localiza no mesmo país que o cliente. Nesta modalidade, podem ocorrer duas situações. Na primeira, a empresa encontra-se no mesmo país que o cliente, mas é distante fisicamente dele (*offsite*). E na segunda, a empresa contratada utiliza seus recursos para dar suporte as atividades do cliente localmente (*onsite*). No primeiro caso, mantém-se o desenvolvimento perto do cliente e no segundo, dá ao cliente maior controle sobre o projeto (CARMEL; TJIA, 2005), (ROBINSON; KALAKOTA, 2004).

O *insourcing* consiste em departamentos ou subsidiários de uma empresa que possuem seu próprio centro de desenvolvimento de software, o qual se torna provedor de serviço na relação organização-cliente, mas é visto como uma entidade externa, que compete com outros provedores. Alguns motivos para que isto ocorra são maior controle, maior flexibilidade, menores preços a longo prazo, mantendo a cultura interna da empresa (ROBINSON; KALAKOTA, 2004), (HYDER et al., 2004).

O *outsourcing* ou terceirização consiste em um processo no qual uma determinada empresa contrata uma organização externa, que pode estar separada geograficamente da empresa que a solicitou, para delegar as atividades de uma área específica da empresa contratadora, assim, assumindo suas responsabilidades (CHASE et al., 2001), (LANKFORD; PARSA, 1999).

Segundo Prikladnicki e Audy (2006), a engenharia de software tem obtido grandes avanços nas últimas décadas, e muitas melhorias nas ferramentas e métodos utilizados têm permitido que se possa formar equipes de desenvolvimento em grupos distribuídos trabalhando no mesmo projeto, por exemplo as evoluções dos recursos de telecomunicações e das ferramentas de desenvolvimento. E como o mercado vem passando por grandes avanços na economia, com os meios de comunicação mais sofisticados, além de certa pressão por redução de custos, o que têm incentivado maiores investimentos em DDS, muitas organizações começaram a investir neste cenário buscando ter maior qualidade no processo de desenvolvimento, redução os custos e obtenção de recursos em âmbito global.

2.2 GERENCIAMENTO DE CONFLITOS NO DDS

De acordo com o estudo realizado por Junior et al. (2013), o gerenciamento de conflitos está diretamente ligado ao desempenho da equipe, uma vez que bem gerenciados pode-se aumentar as chances de sucesso do projeto. No cenário distribuído, esta gestão é mais complicada, pois fatores como fuso horário e a falta de contato direto entre os membros da equipe podem comprometer as trocas de informação. Após o término de seus experimentos, foi consolidado um modelo de maturidade para auxiliar o desenvolvimento em cenários distribuídos.

2.3 O ENSINO DE DDS NA GRADUAÇÃO

Segundo L'Erário (2012), existem algumas tecnologias com potencial para auxílio no ensino do DDS. Tais tecnologias servem para tentar minimizar os gargalos causados pelas adversidades do cenário em questão. Podem-se encontrar na literatura desde alguns estudos sobre as maneiras mais usadas para esta gestão, como em Santos et al. (2010), onde os autores fazem uma revisão sistemática das ferramentas mais usadas na comunicação e em Souza et al. (2015) onde os autores propõe um modelo para monitorar o processo de desenvolvimento distribuído. Portanto, pode-se afirmar que é possível minimizar o impacto desses gargalos por meio de ferramentas e que as soluções encontradas para tal podem se tornar um diferencial no desenvolvimento distribuído (PRIKLADNICKI, 2011). Considerando tal cenário, onde as organizações cada vez mais se utilizam do DDS, as instituições de ensino se sentem atraídas a oferecer isso aos seus alunos como possível diferencial de mercado (L'ERÁRIO; PESSÔA, 2007).

2.4 KANBAN

Apesar da indústria de software hoje ser uma das mais importantes no Brasil, o produto do desenvolvimento ainda não tem a qualidade esperada, possui custos altos e dificuldades com prazo (GHISI, 2012). Baseado neste realidade surgiu o uso do Kanban no desenvolvimento de software. O Kanban é uma técnica usada para encontrar falhas no sistema de produção, originário da filosofia da linha de produção da empresa Toyota. Foi adaptado por ANDERSON (2010) para ser usado no processo de desenvolvimento de software, visando alcançar os mesmos méritos de não sobrecarga, agilidade e qualidade. Percebeu-se que para uma equipe/empresa de desenvolvimento aceitar novas práticas tem-se certa resistência, uma vez que o processo de criação de software é diferente em cada área de negócio, então David observou que era necessária uma adaptação das técnicas usadas na Toyota e que era importante uma liderança

ativa em cada equipe.

Sendo um conjunto de conceitos que ajudam a visualizar pontos fracos no método usado, na equipe e na gestão do desenvolvimento de software, basicamente o Kanban tem 5 práticas (ANDERSON, 2010):

- Visualização do quadro de tarefas
- Limitação do trabalho em Progresso
- Mensurar e otimizar o fluxo de trabalho
- Tornar explícitas as políticas do processo
- Gerenciar quantitativamente

O quadro de visualizações (conforme figura 1) tem as etapas do desenvolvimento divididas em colunas que deve ter um limite de tarefas em execução para cada integrante da equipe, que só pode “puxar” uma tarefa por vez baseando-se no modelo da TOC (*Theory of Constraints*) e na FDD (*Feature Driven Development*), que por sua vez eram os métodos de trabalho de ANDERSON (2010), que diz, “a habilidade de identificar gargalos em um sistema é o primeiro passo para entender a Teoria das Restrições”. Esta teoria afirma que o ritmo de trabalho é ditado pelos produtos finalizados, diminuindo o número de produtos inacabados em andamento (GHISI, 2012).

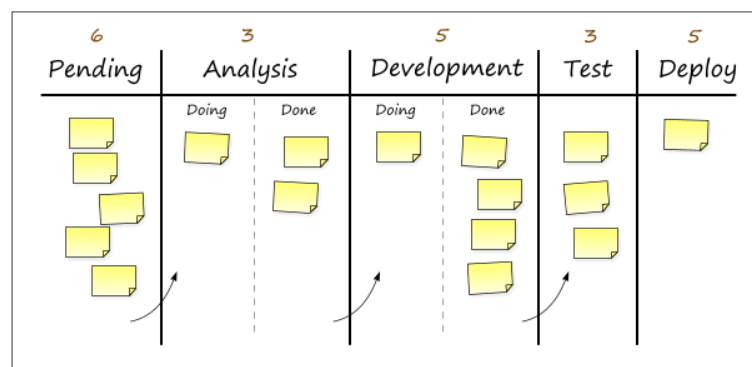


Figura 1: Quadro Kanban

Fonte: [Kanbanblog.com](http://kanbanblog.com) (PETERSON, 2013)

O Kanban tem se tornado popular pela flexibilidade e adaptabilidade que suas técnicas apresentam, pois carrega implícito uma cultura oriunda do Japão chamada *Kaizen*, que quer dizer pequena mudança para melhor. Segundo Mariotti (2012), devido a sua maneira simples de

implementação e sua proposta, encontra-se pouca resistência das equipes ao adotar as práticas do Kanban. Se tais técnicas forem seguidas com disciplina torna-se fácil visualizar onde as equipes encontram problemas e identificar o tempo que cada tarefa leva para ser realizada. No entanto é altamente recomendado que a gestão de projetos tenha maturidade e atenção ao estipular os prazos de entrega de um projeto para que não haja a necessidade de priorizar tarefas durante a execução das mesmas. Essa busca por uma maturidade na produção está diretamente ligada à qualidade e a velocidade no desenvolvimento. Como um desenvolvedor somente realiza uma tarefa por vez, este terá maior foco na qualidade da tarefa realizada e, conseqüentemente, construirá códigos melhor estruturados, o que com o tempo resultará em mais velocidade no desenvolvimento, já que tem-se menos retrabalho nessas tarefas. A limitação do tempo de trabalho em andamento ou WIP, do inglês *Work In Progress* está, por sua vez, ligada ao conhecimento tácito, conforme há menos tempo de espera entre uma tarefa e outra (*lead-time*), perde-se menos do conhecimento adquirido na realização das tarefas, aumentando assim a qualidade dos resultados.

ANDERSON (2010) diz que para ter sucesso ao praticar mudanças no quadro atual de uma empresa, é importante respeitar algumas etapas: focar na qualidade, limitar a quantidade de trabalho em progresso, entregar frequentemente, balancear demanda com capacidade, priorizar tarefas e atacar fontes de variáveis.

Uma das principais características do Kanban é ter um modelo visual onde as tarefas serão disponibilizadas para toda a equipe, seja este físico ou virtual. Os quadros seguem a ideia citada anteriormente, de puxar tarefas e cada coluna do quadro representar uma etapa do processo de desenvolvimento. Por sua vez, cada uma destas etapas tem um número máximo de tarefas, ou seja, tem um limite para que possa-se evitar possíveis sobrecargas na equipe. Essa maneira de limitar as tarefas no quadro faz com que a equipe possa colaborar para etapas mais complexas do projeto.

A utilização do Kanban tem dois principais fatores positivos: os problemas no processo de desenvolvimento tornam-se transparentes e ajuda a identificar possíveis gargalos no mesmo. Ainda, por não ter a complexidade de um método ágil e ser de simples implementação, pode ser utilizado em conjunto com tais métodos para identificar os pontos fracos da equipe e do processo.

2.5 TRABALHOS RELACIONADOS

Em Ikonen et al. (2011), os autores desenvolvem uma pesquisa para avaliar algumas questões sobre o uso do Kanban como método no Desenvolvimento do Software. Dentre as questões propostas constavam os quesitos: resolução de problemas, documentação, comunicação e *feedback*. Para esta pesquisa da *University of Helsinki* na Finlândia, foi organizada uma equipe de 12 estudantes de mestrado e 1 estudante de graduação. Esta equipe foi dividida em um time de desenvolvimento, contendo um líder de equipe, dois *seniors* e cinco *juniors*. O projeto em questão foi desmembrado em tarefas menores pela própria equipe, que por sua vez também foi responsável por se auto organizar e coordenar durante todo o processo de desenvolvimento. Ao final de 7 semanas a equipe foi entrevistada para que os dados do estudo empírico fossem coletados diretamente, além dos dados coletados por históricos e observação durante o experimento. Como conclusão do estudo empírico, dos nove aspectos elicitados, sete tiveram resultado positivo.

No estudo realizado por Ahmad et al. (2014), os autores realizam um *survey* com alunos da Universidade de Oulu na Finlândia, onde utilizam o Kanban como ferramenta de gerenciamento de projetos com o intuito de analisar seus benefícios no ensino do desenvolvimento de software. Após o estudo concluem que o Kanban auxiliou os estudantes a compreender as atividades do projeto, adquirir competências de trabalho em equipe e desenvolver uma visão dividida e efetiva do gerenciamento de tarefas. Também concluíram que é uma boa ferramenta pedagógica, onde com pouco treinamento, os estudantes puderam compreender suas etapas e a visão geral do projeto em desenvolvimento.

Em L'Erario et al. (2012), os autores propõem um modelo para o ensino do DDS em cursos de graduação. De acordo com os mesmos, o *framework* contempla a institucionalização das teorias relativas a prática e como um projeto distribuído pode ser aplicado a graduação. Dois experimentos ocorreram na Universidade Tecnológica Federal do Paraná, tendo como participantes dois cursos da área da computação, sendo um no período integral e o outro no período noturno. Os experimentos têm como premissa simular um ambiente distribuído de desenvolvimento utilizando-se de um sistema de controle de versões e comunicação interna dos participantes para que trabalhem em um mesmo projeto. Com a análise prévia dos dados, que ainda estavam sendo coletados, pôde-se verificar que os grupos que fizeram uso bem elaborado dos controles de versão se saíram melhor ao decorrer do projeto, enquanto alguns grupos tiveram dificuldades no uso das ferramentas. Portanto, ao se trabalhar em um ambiente distribuído, os alunos necessitam de algumas habilidades, técnicas e de gestão, que viabiliza o ensino de tais práticas, concluem os autores.

No estudo de Fagerholm et al. (2013), no intuito de ensinar o DDS em universidades, os autores organizam times de desenvolvimento *off-site* utilizando o conceito de *Software Factory* para efetuar projetos de desenvolvimento entre diferentes instituições parceiras. Neste estudo, são avaliadas as fases de preparação e desenvolvimento de um projeto, sob a perspectiva do aluno, do cliente, dos professores, e das instituições, podendo assim ter uma visão macro da aplicação do DDS nas práticas de ensino e dos cuidados necessários para que os resultados sejam próximos aos esperados. Houve o cuidado de selecionar alunos com habilidades condizentes a cada etapa do projeto, uma vez que, ao relacionarem diferentes instituições, pode não haver um paralelismo entre o nível dos participantes. Esta condição de possível disparidade aproxima o ambiente simulado de um ambiente real encontrado no mercado de trabalho. Ao final do processo, chegaram a resultados encorajadores, onde, diminuindo fardos administrativos dos professores, puderam focar mais em aspectos educacionais do processo como um todo.

Em Ahmad et al. (2013), os autores realizam uma revisão sistemática da literatura existente sobre o Kanban aplicado ao desenvolvimento de software. Foram achados 492 estudos primários e 19 com relevância para o trabalho. Neste estudo os autores percebem que satisfação do cliente, melhor qualidade do software, tempo de entrega, *feedback* mais ágil e diminuição na reclamação sobre erros por parte do cliente são alguns dos benefícios com o uso do Kanban. Porém, ainda há alguma relutância ao se mudar antigas culturas dentro das empresas e é necessária maturidade dos profissionais envolvidos no projeto. Concluem, por fim, que grande parte dos estudos encontrados são de caráter experimental (47%) e a grande maioria envolve projetos piloto ou de pequena escala e que apesar de uma tendência positiva no uso do Kanban nos últimos anos ainda há a necessidade de mais estudos voltados para a análise de sua aplicação no desenvolvimento de software.

Todos os citados envolvem equipes trabalhando de maneira dispersa ou o uso do Kanban no desenvolvimento de software. Após o levantamento da literatura, foi possível perceber que os estudos baseados no DDS já tem uma maior maturidade em relação ao Kanban. No entanto, apesar do Kanban se provar promissor no cenário de desenvolvimento, não foram encontrados relatos de estudos unindo os dois conceitos.

3 METODOLOGIA DE PESQUISA

Neste capítulo apresentam-se os aspectos metodológicos utilizados no estudo, a caracterização da pesquisa, a seleção da população e amostra, a definição das hipóteses, as variáveis, validações e as técnicas utilizadas na coleta e análise dos dados.

3.1 CARACTERIZAÇÃO DA PESQUISA

Para a pesquisa em questão, decidiu-se realizar um experimento visando melhoria para os cenários de DDS. A ideia foi levantar dados estatísticos sobre duas das práticas crescentes, DDS e Kanban como afirmam L’Erario (2013) e Mariotti (2012), no que diz respeito ao quadro do desenvolvimento de software entre as empresas.

O modelo experimental escolhido para o estudo seguiu as métricas sugeridas por Travassos et al. (2002). Ele afirma que para a elaboração de processo experimental é necessário definir um ambiente, configurá-lo e executar o experimento, tendo como sequência a escolha da realização no próprio campo de conhecimento, as definições e caracterizações das entidades envolvidas, definição da amostra, a forma de coleta dos dados e a maneira como estes serão avaliados e validados.

3.1.1 FERRAMENTAS

Para realizar a pesquisa proposta foram utilizadas duas ferramentas auxiliares: *Kanbanize* e Logo. *Kanbanize* apresenta-se como uma plataforma online de gestão de projetos que dispõe de um quadro de Kanban flexível onde pode-se adicionar etapas para o fluxo de trabalho, movimentar e identificar tarefas, além de proporcionar estatísticas sobre as mesmas, como cálculo das métricas, fluxo cumulativo, tempo médio de ciclo, distribuição das atividades criadas versus finalizadas, tempo de resolução do bloco e hierarquia de tarefas (KANBANIZE, 2011). A outra ferramenta, Logo, trata-se de uma linguagem de programação interpretada muito utilizada e aliada tanto ao ensino convencional quanto ao ensino da programação (PAPERT et

al., 1980). Essa linguagem envolve um ambiente com uma tartaruga ao centro, que responde aos comandos do usuário de forma gráfica e visual, estimulando assim o aprendizado.

3.1.2 DEFINIÇÃO DO CONTEXTO

O experimento, configura parte de um processo *off-line*, uma vez que os participantes realizaram a atividade somente durante um certo momento e não no decorrer de um longo período. Foi modelado um cenário proposto para levantamento dos dados de caráter geral e que, apesar da simulação de um ambiente, não há restrições ou impedimentos para que o mesmo seja feito em outros contextos, não se limitando a distribuição das equipes.

O escopo do projeto definido é a modularização de um trabalho feito com a linguagem Logo, a separação dos participantes em equipes heterogêneas e a preparação de um cenário de DDS com níveis *onshore* e *insourcing*. Uma vez preparado o ambiente experimental, cada time de desenvolvimento recebeu uma parte do projeto a ser executado, sendo um dos grupos responsável pela integração de todas as partes.

3.1.3 SELEÇÃO DOS INDIVÍDUOS

Como população do estudo definiu-se por envolver alunos dos cursos de graduação em computação. Para representar o mais próximo de um ambiente de desenvolvimento profissional, optou-se por selecionar alunos de diferentes turmas. Desta forma pode-se criar um ambiente heterogêneo em que suas habilidades não estão niveladas e há menor possibilidade de contato prévio entre os participantes, evitando que os resultados fossem tendenciosos e aproximando ao máximo da realidade proposta. Foi entregue um questionário aos participantes com o objetivo de definir seu perfil e sua experiência com desenvolvimento, com o intuito de garantir que o processo possa ser repetido.

3.1.4 QUESTÕES

As questões propostas a serem respondidas pelo estudo, dadas as métricas estabelecidas e os cenários caracterizados, são:

- Q1 – Das práticas do Kanban apresentadas, existem aquelas que são consideradas úteis pelos desenvolvedores?
- Q2 – Das práticas do Kanban apresentadas, existem aquelas que são consideradas inúteis pelos desenvolvedores?

- Q3 – Das práticas do Kanban apresentadas, existem aquelas que são consideradas úteis, porém necessitam de alguma adaptação?
- Q4 – Das práticas do Kanban apresentadas, existem aquelas que não são consideradas úteis por que necessitam de alguma adaptação?

3.1.5 HIPÓTESE

Uma vez elaboradas as questões a serem respondidas pelo resultado do estudo, surgem duas hipóteses a serem questionadas através destes resultados. A hipótese nula (H_0) afirma negativamente o resultado final e a hipótese alternativa (H_a) configura que o conceito apresentado representa aceitação dos desenvolvedores.

H_0 Não é viável utilizar as técnicas do Kanban para auxiliar os processos de desenvolvimento em cenários distribuídos.

H_a É viável utilizar as técnicas do Kanban para auxiliar os processos de desenvolvimento em cenários distribuídos.

3.1.6 QUESTIONÁRIO DE PERFIL

Dada a repetição do experimento, optou-se pela coleta dos dados que compõem a formação, experiência profissional e acadêmica dos participantes, considerando uma maneira de caracterizá-los e comparar as opiniões de acordo com sua bagagem como desenvolvedores. Portanto, foi elaborado um questionário que segue no apêndice A.

3.1.7 DESCRIÇÃO DA INSTRUMENTAÇÃO

Os experimentos foram conduzidos durante as aulas dos participantes. Foram divididos em dois momentos de contextualização com a apresentação do Logo e do Kanban e em sequência a aplicação do experimento.

O projeto executado foi uma figura (Figura 2) que foi modularizada para que cada um dos grupos fosse responsável por uma das partes e um deles pela integração das outras partes. O fluxo de execução do projeto é apresentado no apêndice B.

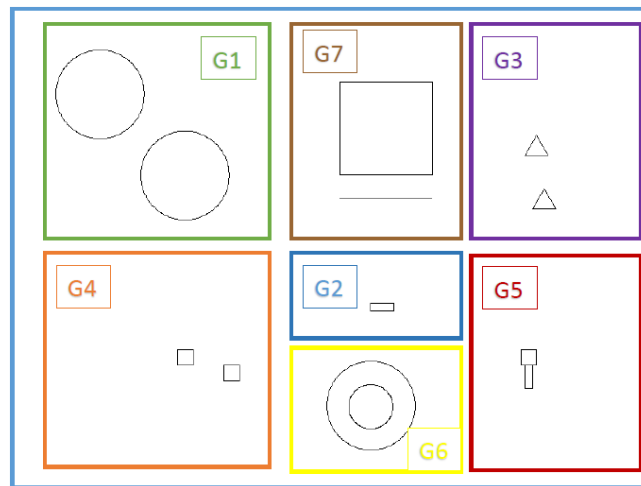


Figura 2: Projeto Logo

Fonte: Autoria própria.

Ao final dos experimentos o questionário avaliando as práticas do Kanban foi entregue aos participantes. Essa avaliação seguirá a opinião dos participantes sobre a necessidade, a utilidade e a adaptação de práticas auxiliares ao DDS. A cada uma delas consideram-se respostas positivas e negativas, que resultam na distribuição das N práticas com valores (N; U; A) (tabela 1), onde:

N - necessidade {0 - não necessária;1 - necessária};

U - Utilidade {0 - não é útil;1 - é útil};

A - Adaptação {0 - precisa de adaptação;1 - não precisa de adaptação}.

Tabela 1: Tabela de possíveis respostas para questionário

(A) Necessidade da Prática	(B) Utilidade da prática	(C) Adaptação do uso da prática
1. É necessário.	1. Não é útil.	1. Necessita de modificações.
2. Não é necessário.	2. Provavelmente é útil, mas não vejo diferença.	2. Não necessita modificações.
	3. É útil e pretendo aplicar.	

Fonte: Modelo adaptado de Travassos et al. (2002)

A tabela 2 mostra as configurações das possíveis respostas dos participantes e a quais questões do estudo elas pretendem responder.

Tabela 2: Tabela de Métricas para teste Binomial

Nº	N	U	A		Questões
1	0	0	0	Não é necessária, não é útil, modificação necessária	Q2
2	0	0	1	Não é necessária, não é útil, modificação não necessária	N/A
3	0	1	0	Não é necessária, é útil, modificação necessária	Q1
4	0	1	1	Não é necessária, é útil, modificação não necessária	Q3
5	1	0	0	É necessária, não é útil, modificação necessária	Q2
6	1	0	1	É necessária, não é útil, modificação não necessária	Q2 Q4
7	1	1	0	É necessária, é útil, modificação necessária	Q1
8	1	1	1	É necessária, é útil, modificação não necessária	Q1 Q3

Fonte: Tentativa de Benoulli

3.1.8 VARIÁVEIS

3.1.8.1 VARIÁVEL INDEPENDENTE

Conjunto de práticas oferecidas pelo Kanban como ferramenta de auxílio do DDS.

3.1.8.2 VARIÁVEIS DEPENDENTES

1. A necessidade de práticas auxiliares para o cenário de DDS.

Pode receber os valores:

Não necessária, quando todas as práticas têm o valor $NUA = \{0, X, X\}$ (métricas 1 a 4);

Necessária, quando todas as práticas têm o valor $NUA = \{1, X, X\}$ (métricas 5 a 8).

Incerteza, quando não se cumprem as condições de “Não necessária” e “Necessária”. O grau de incerteza pode ser avaliado como:

$$100\% - \{1, X, X\} / (\{0, X, X\} + \{1, X, X\}) * 100\% \quad (1)$$

2. A utilidade das práticas do Kanban no cenário de DDS.

Parte útil:

$$\{1, 1, X\} / (\{0, X, X\} + \{1, X, X\}) * 100\% \quad (2)$$

Parte não útil:

$$\{1, 0, X\} / (\{0, X, X\} + \{1, X, X\}) * 100\% \quad (3)$$

3. A adaptação das práticas mostra aquelas que precisam ser adaptadas.

Parte a ser adaptada:

$$\{1, X, 0\} / (\{0, X, X\} + \{1, X, X\}) * 100\% \quad (4)$$

Parte sem necessidade de adaptação:

$$(\{0, X, X\} + \{1, X, X\}) / \{1, X, X\} * 100\% \quad (5)$$

3.1.9 ANÁLISE QUALITATIVA

Para analisar as informações referentes às práticas utilizadas no processo de desenvolvimento, foi aplicada a análise qualitativa, uma vez que o processo foi repetido e comparado. Esta análise apresentará o conjunto de práticas oferecidas pelo Kanban ao longo do experimento e após responderem ao questionário final serão classificadas as que os alunos consideram positivas para o aprendizado e uso do novo conceito. Para tanto, esta análise considerará a prática com valor [1, 1, X] (Métricas 5 a 8), a opção “É útil e pretendo aplicar” e “Provavelmente útil, mas não vejo diferença” como positiva para a aplicação do Kanban no contexto distribuído pela perspectiva dos desenvolvedores participantes.

3.1.10 VALIDADE

1. Validade Interna

Decidiu-se pelos alunos dos cursos de graduação de tecnologia, que costumam desenvolver softwares de cunho acadêmico ao longo do curso, assumindo dessa forma o papel de desenvolvedores em um ambiente simulado de DDS.

Para que houvesse a redução de influências não pertinentes ao estudo, além das já previstas pelo experimento, e não impactar na validade interna do mesmo, foi utilizada uma linguagem de programação acessível aos diferentes níveis de conhecimento dos alunos, independente do estágio em que estivessem como desenvolvedores.

2. Validade de Conclusão

Para avaliar os valores da necessidade, utilidade e adaptação será utilizado o teste binomial de acordo com a amplitude das práticas. A verificação das hipóteses será feita pela demonstração da utilidade ou não da prática na lista que representa as variáveis dependentes. A validação das possíveis melhorias nas práticas do desenvolvimento de software será através da observação de dois experimentos realizados. É importante lembrar que

este estudo não visa solucionar um problema em específico, e sim levantar dados estatísticos da aplicabilidade das práticas mencionadas para embasar novos estudos, com o intuito de minimizar ainda mais os problemas recorrentes do DDS.

3. Validade de Construção

O estudo caracteriza-se pela análise da aplicação das técnicas de organização do Kanban em um ambiente distribuído de desenvolvimento. Devido às constantes situações onde o DDS vem sendo utilizado como meio de redução de custos e expansão da área de atuação de uma empresa e onde as práticas e técnicas do Kanban vem sendo utilizadas mais frequentemente, os alunos receberam um breve treinamento do que são estes dois conceitos e puderam incluir o experimento em suas atividades extracurriculares.

4. Validade externa

Conforme mencionado anteriormente, os participantes podem ser considerados representantes de desenvolvedores. Para a avaliação do nível motivacional e de envolvimento com o projeto, o produto final do desenvolvimento e o questionário entregue são avaliados.

O estudo foi realizado com alunos em graduação, que comumente estão envolvidos no processo de desenvolvimento de software, o que fez com que a apresentação das novas práticas simulando as adversidades de um ambiente DDS não ofereceram um problema temporal, já que os conceitos e os experimentos foram apresentados e executados no momento de suas aulas, não havendo a necessidade dos alunos desenvolverem nenhuma atividade extraordinária.

3.1.11 ANÁLISE DOS DADOS

1. Validação dos dados

A validação dos dados do experimento será baseada na verificação dos formulários. Sabendo assim, se alguma das respostas não foi preenchida ou respondida devidamente, eliminou-se possíveis distorções na análise final dos dados, excluindo tais respostas.

2. Análise e apresentação dos dados

Durante esta etapa serão apresentados os dados de forma descritiva e explícita aplicando o teste *chi-quadrado* (χ^2) nos dados coletados no questionário e *Binomial* que contestará o teste anterior. Foram usados cálculos de Mediana e Moda, uma vez que os valores “Necessidade”, “Utilidade” e “Adaptação” são de escala ordinal. Ainda será apresentada a disposição das respostas em porcentagem, de acordo com sua amplitude.

4 ANÁLISE DOS DADOS

Neste capítulo apresentam-se as análises dos dados e resultados obtidos nos experimentos executados. Os dados são apresentados através de dois diagramas que compõem o mapa das atividades, o processo de experimentação, perfil dos participantes, aplicação das métricas estatísticas, análise quantitativa e análise qualitativa.

4.1 MAPA DAS ATIVIDADES

O mapa que descreve o fluxo das atividades de coleta e análise dos dados encontra-se no apêndice C.

4.2 PROCESSO DA EXPERIMENTAÇÃO

Para garantir a assertividade do experimento, os dois *trials* foram executados antes da coleta principal dos dados, possibilitando definir um modelo de coleta de dados dos participantes e permitindo que a avaliação e análise dos dados seguissem um padrão, como mostra a tabela 3.

Tabela 3: Modelo de avaliação do quadro kanban

Passos	Atividade
1	Identificar participantes
2	Identificar tarefas
3	Identificar tempo do projeto
4	Identificar tempo das tarefas
5	Identificar tempo do fluxo de tarefas
6	Identificar tempo em cada coluna
7	Identificar tempo total em cada coluna
8	Identificar número de transações
9	Identificar número de retrabalhos

Fonte: Autoria própria.

Os *trials* tiveram a aplicação de todos os conceitos explorados do estudo, com exceção da aplicação do questionário, fornecendo assim somente dados empíricos sobre o projeto desenvolvido. Os participantes foram duas turmas de alunos do primeiro semestre, sem muito contato com técnicas de programação, por isso optou-se utilizar a linguagem Logo em um projeto modularizado, como manda o escopo do experimento. Dentro deste cenário os participantes foram divididos em grupos e receberam partes de um projeto que deveria ser integrado após sua execução. Ao longo do projeto cada fração da tarefa atribuída aos grupos foi transformada em um cartão no quadro kanban e gerenciada pelos mesmos.

Em sequência ao molde formulado durante os *trials* foi aplicado o experimento com dois outros grupos de alunos no mesmo perfil acadêmico. Dessa vez foi realizado o questionário previamente estipulado na metodologia do estudo. Uma das aplicações seguiu o mesmo formato dos *trials*, tanto para analisar as estatísticas das tarefas propostas, quanto o próprio escopo de projeto. A outra aplicação teve um exercício real de programação com alunos um pouco mais experientes. Ambos tiveram aplicação do questionário de avaliação.

4.3 PERFIL DOS PARTICIPANTES

O questionário aplicado ao experimento, conforme citado na sessão de metodologias, foi dividido em duas partes visando caracterizar os participantes: perfil e experiência profissional. Desta forma possibilitando a eliciação de métricas sociais. O intuito foi preparar um cenário que pudesse ser utilizado como comparativo para futuros trabalhos.

As estatísticas dos perfis nos sugere um contexto em que é possível classificar as respostas de acordo com o nível de experiência, o nível acadêmico e os tipos de cursos de graduação dos participantes, possibilitando que a repetibilidade do experimento possa variar de acordo com a população escolhida. Desta maneira é possível verificar divergências nas respostas e ainda caracterizá-las de acordo com tais perfis, o que abre um leque comparativo mais abrangente.

Espera-se que, dadas estas coletas, também seja possível verificar se algumas posturas adotadas pelos desenvolvedores tenham relação com seus níveis de maturidade, tanto acadêmica quanto profissionalmente. A figura 3 demonstra alguns indicadores coletados.

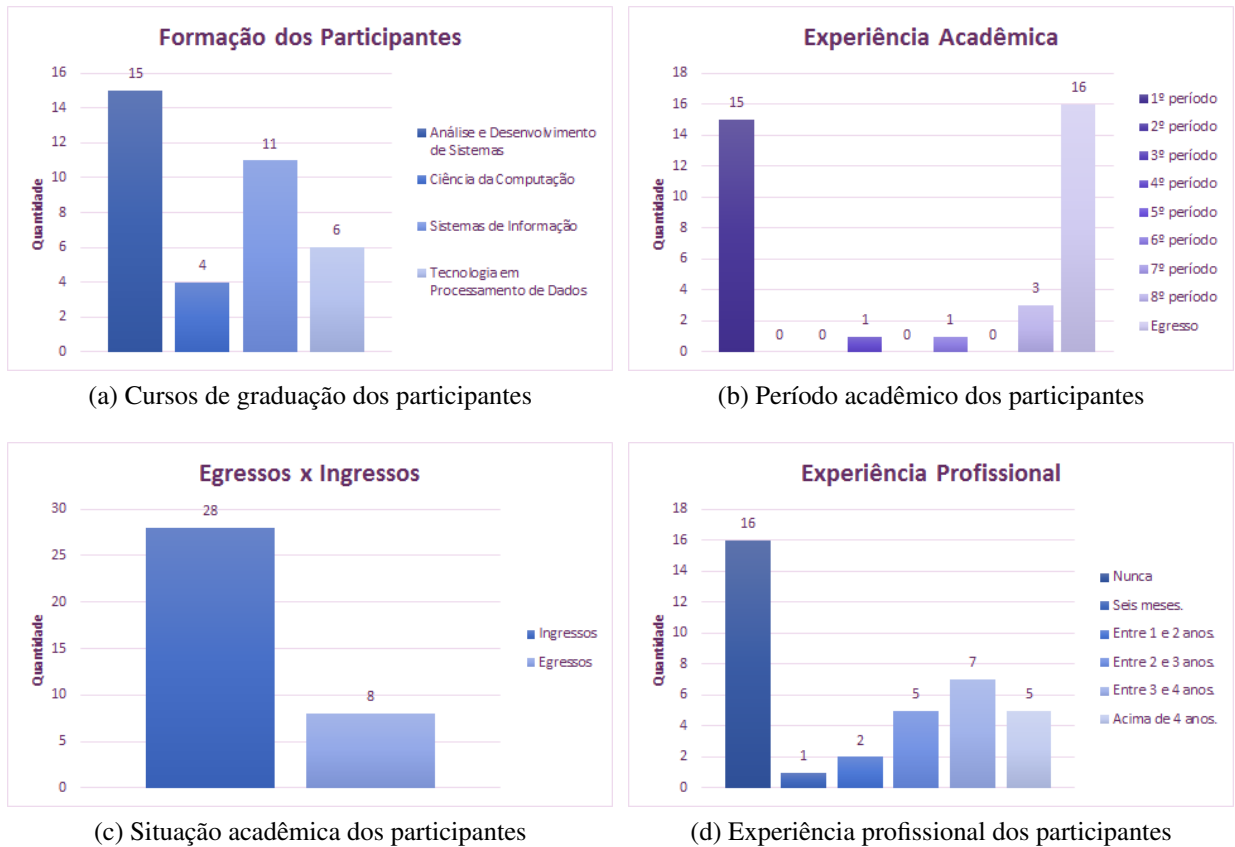


Figura 3: Análise de perfil dos participantes

Fonte: Autoria própria.

4.4 APLICAÇÃO DAS MÉTRICAS ESTATÍSTICAS

Uma vez respondido o questionário foram aplicadas as métricas estatísticas de mediana e moda para avaliar as frequências das respostas de maneira individual e separadas, como mostram as tabelas 4, 5 e 6:

Tabela 4: Mediana e moda do atributo Necessidade

Necessidade	1	2	3	4	5
Mediana	1	1	2	2	1
Moda	1	1	2	2	1

Fonte: Autoria própria.

Tabela 5: Mediana e Moda do atributo Utilidade

Utilidade	1	2	3	4	5
Mediana	3	3	3	3	3
Moda	3	3	3	3	3

Fonte: Autoria própria.

Tabela 6: Mediana e Moda do atributo Adaptação

Adequação	1	2	3	4	5
Mediana	2	2	2	2	2
Moda	2	2	2	2	2

Fonte: Autoria própria.

De acordo com os grupos de respostas recebidas, podemos apresentá-los de maneira explícita (tabela 7) e classificá-las utilizando os resultados de estatística descritiva (tabela 8).

Tabela 7: Dados dispostos de maneira explícita

	Práticas	N		U			A	
		1	2	1	2	3	1	2
1	Visualizar fluxo de trabalho	28	8	0	9	27	3	33
2	Limitar trabalho em andamento	20	16	1	10	25	4	32
3	Medir e otimizar fluxo	15	21	1	9	26	6	30
4	Explicitar políticas	28	8	0	9	27	3	33
5	Gerenciar quantitativamente	20	16	1	13	22	7	29

Fonte: Autoria própria.

Os valores da tabela 8 correspondem a:

- N - necessária:não necessária;
- U - útil:não útil;
- A - com adaptação:sem adaptação;

Tabela 8: Dados dispostos de maneira descritiva

Nº	Práticas	N	U	A
1	Visualizar Fluxo de trabalho	28:8	36:0	33:3
2	Limitar trabalho em andamento	20:16	35:1	32:4
3	Medir e Otimizar Fluxo	15:21	35:1	30:6
4	Explicitar Políticas	28:8	36:0	33:3
5	Gerenciar Quantitativamente	20:16	35:1	29:7

Fonte: Autoria própria.

Com a observação desta distribuição dos dados pudemos verificar que:

- (a) A maioria dos participantes chegou a conclusão de que as práticas apresentadas são necessárias em um projeto de desenvolvimento. A prática 3, que envolve quantificar e qualificar as tarefas teve menor aceitação.
- (b) Na sua grande maioria as práticas são consideradas úteis pelos participantes.
- (c) Os participantes concluíram que não há necessidade de modificarem as práticas para que se adaptem às equipes.

A elicitação dos dados de forma descritiva, permitiu que o teste chi-quadrado fosse aplicado. Primariamente organizou-se os valores de (N;U;A) na matriz conforme pede o método estatístico (LANCASTER, 1969).

Tabela 9: Matriz chi-quadrado - Necessidade

Prática	Positivo	Negativo	Total
Visualizar Fluxo de trabalho	28	8	36
Limitar trabalho em andamento	20	16	36
Medir e Otimizar Fluxo	15	21	36
Explicitar Políticas	28	8	36
Gerenciar Quantitativamente	20	16	36
Total	111	69	180

Fonte: Autoria própria.

Tabela 10: Matriz chi-quadrado - Utilidade

Prática	Positivo	Negativo	Total
Visualizar Fluxo de trabalho	36	0	36
Limitar trabalho em andamento	35	1	36
Medir e Otimizar Fluxo	35	1	36
Explicitar Políticas	36	0	36
Gerenciar Quantitativamente	35	1	36
Total	177	3	180

Fonte: Autoria própria.

Tabela 11: Matriz chi-quadrado - Adaptação

Prática	Positivo	Negativo	Total
Visualizar Fluxo de trabalho	33	3	36
Limitar trabalho em andamento	32	4	36
Medir e Otimizar Fluxo	30	6	36
Explicitar Políticas	33	3	36
Gerenciar Quantitativamente	29	7	36
Total	157	23	180

Fonte: Autoria própria.

Os valores dispostos nas tabelas representam as parcelas de respostas positivas e negativas (+X:-X) de cada participante. Para aplicar a tomada de decisão a estas matrizes se fazem necessários os valores:

- χ^2 = Dispersão total dos dados apresentados na tabela.
- χ_c^2 = Dispersão individual de cada um dos dados da tabela.

Para calcularmos os *esperados* multiplicamos os totais parciais relativos a cada prática e dividimos pelo valor geral, chegando aos elementos da tabela 12.

Tabela 12: Valores esperados de cada tabela (N;U;A)

	Esperado
Necessidade	22,2
Utilidade	35,4
Adaptação	31,4

Fonte: Autoria própria.

Para calcular as discrepâncias entre as proporções observadas, Plackett (1983) sugeriu a seguinte fórmula:

$$\chi^2 = \sum \frac{(O - E)^2}{E} \quad (6)$$

onde:

O: frequência observada para cada classe,

E: frequência esperada para aquela classe.

Portanto, o quadrado da soma (\sum) da diferença entre os valores observados e os valores esperados, divididos pelo valor esperado nos dá o χ^2 de cada prática.

Uma vez calculados os valores esperados, calculamos os valores χ_c^2 de cada uma das práticas das tabelas, considerando as respostas positivas e negativas, conforme as tabelas 13, 14, 15:

Tabela 13: χ^2 dos atributos de Necessidade

Prática	$\chi_c^2 p$	$\chi_c^2 n$
Visualizar fluxo de trabalho	0,186889	1,120222
Limitar trabalho em andamento	0,026889	0,213556
Medir e otimizar fluxo	0,288	0,008
Explicitar políticas	0,186889	1,120222
Gerenciar quantitativamente	0,026889	0,213556

Fonte: Autoria própria.

Tabela 14: χ^2 dos atributos de Utilidade

Prática	$\chi_c^2 p$	$\chi_c^2 n$
Visualizar fluxo de trabalho	0,002	6,962
Limitar trabalho em andamento	0,000889	6,574222
Medir e otimizar fluxo	0,000889	6,574222
Explicitar políticas	0,002	6,962
Gerenciar quantitativamente	0,000889	6,574222

Fonte: Autoria própria.

Tabela 15: χ^2 dos atributos de Adaptação

Prática	$\chi_c^2 p$	$\chi_c^2 n$
Visualizar fluxo de trabalho	0,014222	4,480889
Limitar trabalho em andamento	0,002	4,170889
Medir e otimizar fluxo	0,010889	3,584222
Explicitar políticas	0,014222	4,480889
Gerenciar quantitativamente	0,032	3,307556

Fonte: Autoria própria.

Com os resultados do $\chi_c^2(N;U;A)$ de cada uma das práticas, calculamos o χ^2 , somando todas as parcelas obtidas no cálculo anterior, obtendo os resultados conforme a tabela 16.

Tabela 16: χ^2 das tabelas 9, 10 e 11

	X2
Necessidade	3,391111
Utilidade	33,65333
Adaptação	20,09778

Fonte: Autoria própria.

De posse destes dados, o próximo passo é calcular o *Grau de Liberdade* das tabelas 9, 10 e 11. Obtemos essa grandeza através da seguinte fórmula:

$$GL = (\text{número de linhas} - 1) \times (\text{número de colunas} - 1)$$

Em posse do Grau de Liberdade ($GL = 4$) decidiu-se adotar a factibilidade de 10% na avaliação dos resultados. Este valor, de acordo com a tabela de χ^2 (LANCASTER, 1969) é equivalente a 7,779 que corresponde ao χ_c^2 .

Por fim, verifica-se se os valores χ^2 obtidos de cada um das tabelas é menor do que o χ_c^2 . Ainda, segundo Lancaster (1969):

- Hipótese nula: As frequências observadas não são diferentes das frequências esperadas. Não existe diferença entre as frequências (contagens) dos grupos. Portanto, não há associação entre os grupos.
- Hipótese alternativa: As frequências observadas são diferentes das frequências esperadas, havendo diferença entre as elas. Portanto, há associação entre os grupos.

Chegamos aos resultados finais da análise com o seguinte cenário:

$$\chi_c^2 = 7,779$$

$$\chi_c^2(N) = 3,391111111$$

$$\chi_c^2(U) = 33,65333333$$

$$\chi_c^2(A) = 20,09777778$$

Neste momento compara-se os valores de $\chi_c^2(N)$, $\chi_c^2(U)$ e $\chi_c^2(A)$ com o valor de χ_c^2 . Percebe-se que os valores $\chi_c^2(N)$ são menores que o valor de χ_c^2 . De acordo com a factibilidade adotada, no quesito “Necessidade” as práticas não são necessárias. No quesito “Utilidade” e “Adequação”, os valores são maiores que o valor de χ_c^2 , ou seja, as práticas são consideradas úteis e sem necessidade de adaptação.

De acordo com o grau de liberdade escolhido (GL = 4 = 7,779), podemos considerar que a distribuição das respostas (+20:-16), (+28:-8), (+29:-7), (+30:-6), (+32:-4), (+33:-3), (+35:-1), (+36:-0) recebam os valores nos atributos “Necessidade”, “Utilidade” e “Adequação” = [1]. Para as práticas com distribuições diferentes das mencionadas, podemos considerar os valores de “Necessidade”, “Utilidade” e “Adequação” = [0].

Tabela 17: Práticas de acordo com Grau de Liberdade

	Práticas				
	1	2	3	4	5
Necessidade	1	1	0	1	1
Utilidade	1	1	1	1	1
Adaptação	1	1	1	1	1

Fonte: Autoria própria.

4.5 ANÁLISE QUANTITATIVA

A análise quantitativa se utiliza dos dados encontrados na tabela 17 para calcular as probabilidades encontradas em cada um dos atributos (N;U;A) através das respostas dos participantes. A intenção deste teste é contestar os valores obtidos no teste qui-quadrado com 10% de aceitação.

Para chegar aos valores de incerteza de acordo com a amplitude das respostas, é necessário calcular as variáveis dependentes 1, 2 e 3, conforme os cálculos a seguir:

1. Necessidade:

Não se pode afirmar que são necessárias as observações, uma vez que os valores NUA $\{1, X, X\} = 4 < 5$, também não se pode afirmar que não são necessárias de acordo com a métrica adotada, portanto é preciso calcular o Grau de Incerteza de acordo com a fórmula da variável 1:

$$\text{Grau de Incerteza} = 100\% - (4/5) * 100\% = 20\%$$

2. Utilidade:

Para checar as “úteis” e “não úteis” das práticas apresentadas aos alunos, é preciso verificar o resultado da variável 2:

$$\text{Parte útil: } 4/5 * 100\% = 80\%$$

$$\text{Parte não útil: } 0/5 * 100\% = 0\%$$

3. Adequação:

Para verificar as partes “adequadas” e “não adequadas” temos que calcular o resultado da variável 3:

$$\text{Parte a ser adaptada : } 0/5 * 100\% = 0\%$$

$$\text{Parte sem necessidade de adaptação: } 1/5 * 100\% = 20\%$$

4.6 ANÁLISE QUALITATIVA

Concluindo a avaliação dos dados aplicou-se a métrica adotada para a análise qualitativa de acordo com as respostas dos participantes e obteve-se os valores que configuram o cenário de possível aceitação encontrado levando em consideração as duas respostas positivas do atributo Utilidade. A tabela 18 mostra os resultados obtidos.

Tabela 18: Análise Qualitativa

Práticas	1	2	3	4	5
É útil e pretendo aplicar.	27	25	26	20	22
Provavelmente útil, mas não vejo diferença.	9	10	9	15	13

Fonte: Autoria própria.

4.7 RESULTADOS

Ao final dos quatro experimentos realizados, dois serviram para consolidar o cenário e preparar o processo de avaliação, e os outros dois tiveram a efetiva coleta dos dados através do questionário. Avaliando os experimentos pela ótica estatística concluiu-se que, de acordo com o teste de chi-quadrado e os 10% de factibilidade adotada, as práticas aplicadas aos alunos não são necessárias para o cenário proposto, no entanto os participantes as acharam úteis ao processo como um todo, desacreditando a hipótese nula (respondendo as questões Q1 e Q2). Percebeu-se também, que os participantes não sentem a necessidade de adaptar as práticas apresentadas (questões Q3 e Q4).

As verificações de moda e mediana das práticas mostraram que houve variação nas opiniões dos participantes no que diz respeito aos valores de “Necessidade” dos conceitos do Kanban. No entanto, os valores de “Utilidade” e “Adaptação”, tiveram valores concisos em sua avaliação (respectivamente 3 e 2 da tabela 2), reforçando a confiabilidade do teste chi-quadrado (respondendo as questões Q1).

Nos teste binomiais, comprovou-se que, os índices positivos de necessidade das práticas é maior que o índice de rejeição, o que nos remete ao comparativo do teste de chi-quadrado devido a factibilidade adotada. Apesar de não afirmar a necessidade das práticas ao cenário proposto, as respostas dos participantes tenderam a um resultado positivo. Como os outros dois quesitos tiveram unanimidade de respostas positivas, é possível afirmar que o conceito do Kanban foi bem aceito pela perspectiva dos desenvolvedores.

Do ponto de vista empírico, observou-se quatro fatores ao longo dos experimentos: a necessidade da presença e participação ativa da gerência sobre as equipes, os demasiados lapsos de comunicação, apesar da visualização constante do fluxo de trabalho através do quadro Kanban, a promoção de certa competitividade e algum nível de reatividade entre os times de desenvolvedores, mesmo que este não tenha influenciado negativamente o andamento do processo de desenvolvimento.

5 CONCLUSÃO

O DDS é uma prática que vem ganhando espaço considerável entre as empresas de tecnologia, suportado pelos avanços tecnológicos nas áreas de comunicação e infraestrutura, que hoje permite o contato entre longas distâncias, tornando casual algo que a alguns anos era complexo e dispendioso. Aliado a possibilidade de abranger novos mercados e contratar profissionais mais qualificados a custos mais baixos, consolida-se a necessidade de aprimoramentos nos estudos relacionados a seus aspectos.

O principal agente de impacto nos resultados encontrados foi a comunicação, uma vez que todos os outros pontos que comprometem a qualidade do produto final, estão de alguma forma ligados a ela. Esta desempenha um papel crítico na equação Fracasso X Sucesso do DDS, não somente pelos motivos elicitados nos capítulos anteriores (fuso horário, cultura, idiomas etc.), mas também na regularidade e retorno de informações entre uma equipe e outra, tanto no que diz respeito a comunicação entre equipes e na comunicação das mesmas com o gerente de projetos. Pode-se notar que garantir um modelo disciplinado de comunicação entre os desenvolvedores pode melhorar muito os resultados.

O Kanban, por sua vez, teve uma avaliação positiva pelos desenvolvedores. Sua flexibilidade e simplicidade torna muito grande o grau de aceitabilidade pelas equipes e, não menos importante, sua capacidade de gerar métricas claras permite identificar quais etapas tiveram mais impacto no projeto, etapas estas mostraram o quão oneroso são os atrasos no decorrer de um projeto. No entanto, somente as premissas do Kanban não são suficientes para garantir o sucesso de um projeto distribuído. Há a necessidade de mais estudos voltados para este conjunto de disciplinas, que já se provou valioso em outros ramos de atividade.

O contexto apresentado mostrou um conceito já consolidado em outras áreas de atuação junto a uma modalidade de desenvolvimento que já vem sendo estudada a algum tempo e torna-se mais frequente a cada dia. O foco do projeto era avaliar se este contexto tinha influência positiva no DDS e levantar estatísticas para que novos estudos possam se beneficiar deste esforço.

A aplicação do experimento mostrou que é possível conciliar os dois conceitos se

houver suporte aos complicadores já definidos. Portanto, conclui-se que o Kanban pode sim ser um aliado para o desenvolvimento de software, porém ainda há a necessidade de melhorar a colaboração, a comunicação e a gerência de um projeto para que se possa garantir o sucesso e qualidade esperada.

Como trabalhos futuros, sugere-se consolidar um modelo de utilização do Kanban como ferramenta auxiliar aos cenários de DDS, uma vez que a combinação provou-se útil. Ainda, executar trabalhos similares com um cenário de aplicação mais amplo, envolvendo outras universidades, tanto brasileiras quanto estrangeiras.

REFERÊNCIAS

- AHMAD, M. O.; LIUKKUNEN, K.; MARKKULA, J. Student perceptions and attitudes towards the software factory as a learning environment. In: IEEE. **Global Engineering Education Conference (EDUCON), 2014 IEEE**. [S.l.], 2014. p. 422–428.
- AHMAD, M. O.; MARKKULA, J.; OIVO, M. Kanban in software development: A systematic literature review. In: IEEE. **Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on**. [S.l.], 2013. p. 9–16.
- ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business**. [S.l.]: Editora, 2010.
- CARMEL, E. **Global software teams: collaborating across borders and time zones**. [S.l.]: Prentice Hall PTR, 1999.
- CARMEL, E.; TJIA, P. **Offshoring information technology: sourcing and outsourcing to a global workforce**. [S.l.]: Cambridge University Press, 2005.
- CHASE, R. B.; AQUILANO, N. J.; JACOBS, F. R. **Operations management for competitive advantage**. [S.l.]: Irwin/McGraw-Hill, 2001.
- EDWARDS, R. L.; STEWART, J. K.; FERATI, M. Assessing the effectiveness of distributed pair programming for an online informatics curriculum. **ACM Inroads**, ACM, v. 1, n. 1, p. 48–54, 2010.
- FAGERHOLM, F.; OZA, N.; MUNCH, J. A platform for teaching applied distributed software development: The ongoing journey of the helsinki software factory. In: IEEE. **Collaborative Teaching of Globally Distributed Software Development (CTGDSD), 2013 3rd International Workshop on**. [S.l.], 2013. p. 1–5.
- GHISI, T. Kanban no desenvolvimento de projetos de software. **Engenharia de Software Magazine**, v. 45, n. 4, p. 11–16, 2012.
- HYDER, E. B.; HESTON, K. M.; PAULK, M. C. The esourcing capability model for service providers (escm-sp) v2, part 1: Model overview. **Information Technology Services Qualification Center (ITSQC), Carnegie Mellon University, Technical Report No. CMU-ISRI-04-113**, 2004.
- IKONEN, M. et al. On the impact of kanban on software project work: An empirical case study investigation. In: IEEE. **Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on**. [S.l.], 2011. p. 305–314.
- JUNIOR, I. H. de F.; MOURA, H. P. de; MARCZAK, S. Towards a communication maturity model for distributed software development. In: IEEE. **Global Software Engineering Workshops (ICGSEW), 2013 IEEE 8th International Conference on**. [S.l.], 2013. p. 81–83.

KANBANIZE. **Visualize your activity, gather valuable information about your business and boost your productivity.** 2011. Disponível em: <<https://kanbanize.com/kanban-boards/>>. Acesso em: 15 de Fevereiro de 2015.

KAROLAK, D. W. **Global software development: managing virtual teams and environments.** [S.l.]: IEEE Computer society press, 1999.

LANCASTER, H. O. **Chi-Square Distribution.** [S.l.]: Wiley Online Library, 1969.

LANKFORD, W. M.; PARSA, F. Outsourcing: a primer. **Management Decision**, MCB UP Ltd, v. 37, n. 4, p. 310–316, 1999.

L'ERARIO, A. **M3DS: um modelo de dinâmica de desenvolvimento distribuído de software.** Tese (Doutorado) — Universidade de São Paulo, 2013.

L'ERARIO, A. et al. Um framework para o ensino de desenvolvimento distribuído de software em cursos de graduação. In: **Congresso Brasileiro de Ensino em Engenharia, 2012, Belém.** [S.l.: s.n.], 2012.

L'ERARIO, A. et al. Improved communication in distributed agile software development. In: **Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on.** [S.l.: s.n.], 2014. p. 1–6.

L'ERARIO, A.; SANTOS, L. S.; GONÇALVES, J. A. Proposta de cenários genéricos para definição da comunicação em sítios de desenvolvimento distribuído de software. In: **COBENGE. Anais do XXXVI COBENGE Congresso Brasileiro de Educação em Engenharia, São Paulo.** [S.l.], 2008.

L'ERÁRIO, A.; PESSÔA, M. S. de P. An analysis of the dynamics and properties of the distributed development of software environments: A case study. In: **Software Engineering Research and Practice.** [S.l.: s.n.], 2007. p. 471–477.

L'ERÁRIO, M. O. F. A. Uma proposta de framework para ensino de desenvolvimento distribuído de software. In: **Sicite 2012.** [S.l.: s.n.], 2012.

MARIOTTI, F. S. Kanban: o ágil adaptativo. **Dom Bosco**, 2012.

MARQUARDT, M. J.; HORVATH, L. **Global teams: How top multinationals span boundaries and cultures with high-speed teamwork.** [S.l.]: Nicholas Brealey Publishing, 2001.

MOCZADLO, R. Chancen und risiken des offshore-development—empirische analyse der erfahrungen deutscher unternehmen. 2002.

PAPERT, S.; VALENTE, J. A.; BITELMAN, B. **Logo: computadores e educação.** [S.l.]: Brasiliense, 1980.

PETERSON, D. **What is Kanban?** 2013. Disponível em: <<http://kanbanblog.com/explained/>>. Acesso em: 8 de Agosto de 2015.

PLACKETT, R. L. Karl pearson and the chi-squared test. **International Statistical Review/Revue Internationale de Statistique**, JSTOR, p. 59–72, 1983.

PRIKLADNICKI, R. Can distributed software development help the practitioners to become better software engineers?: insights from academia. In: ACM. **Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development**. [S.l.], 2011. p. 16–19.

PRIKLADNICKI, R.; AUDY, J. L. N. Uma análise comparativa de práticas de desenvolvimento distribuído de software no brasil e no exterior. **XX Simpósio Brasileiro de Engenharia de Software**, p. 255–270, 2006.

ROBINSON, M.; KALAKOTA, R. **Offshore outsourcing: Business models, ROI and best practices**. [S.l.]: Mivar Press, 2004.

SANTOS, A. C. C. dos et al. "dificuldades, fatores e ferramentas no gerenciamento da comunicação em projetos de desenvolvimento distribuído de software: uma revisão sistemática da literatura". **IV WDDS - VII Workshop de Desenvolvimento Distribuído de Software**, 2010.

SOUZA, V. F. de; L'ERARIO, A.; FABRI, J. A. Model for monitoring and control of software production in distributed projects. In: IEEE. **Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on**. [S.l.], 2015. p. 1–6.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. **Introdução à engenharia de software experimental**. [S.l.]: UFRJ, 2002.

APÊNDICE A – QUESTIONÁRIO DE PERFIL DO PARTICIPANTE

Tabela 19: Questionário de Perfil do Participante

Perfil Acadêmico
Qual sua Idade?
Qual seu curso de graduação?
<ul style="list-style-type: none"> · Análise e desenvolvimento de Sistemas · Engenharia da Computação · Engenharia de Software · Sistemas de Informação · Ciência da Computação · Tecnologia em Processamento de Dados
Em qual período você se encontra?
Essa é sua primeira graduação?
<ul style="list-style-type: none"> · Sim · Não
Experiência profissional
Tempo de Experiência:
<ul style="list-style-type: none"> · Nunca trabalhei com desenvolvimento. · Seis meses. · Entre seis meses e 1 ano. · Entre 1 e 2 anos. · Entre 2 e 3 anos. · Entre 3 e 4 anos. · Acima de 4 anos.
Como você classificaria sua experiência com desenvolvimento?
<ul style="list-style-type: none"> · Baixa · Média · Alta

Fonte: Autoria própria.

APÊNDICE B – FLUXO DE EXECUÇÃO DO PROJETO

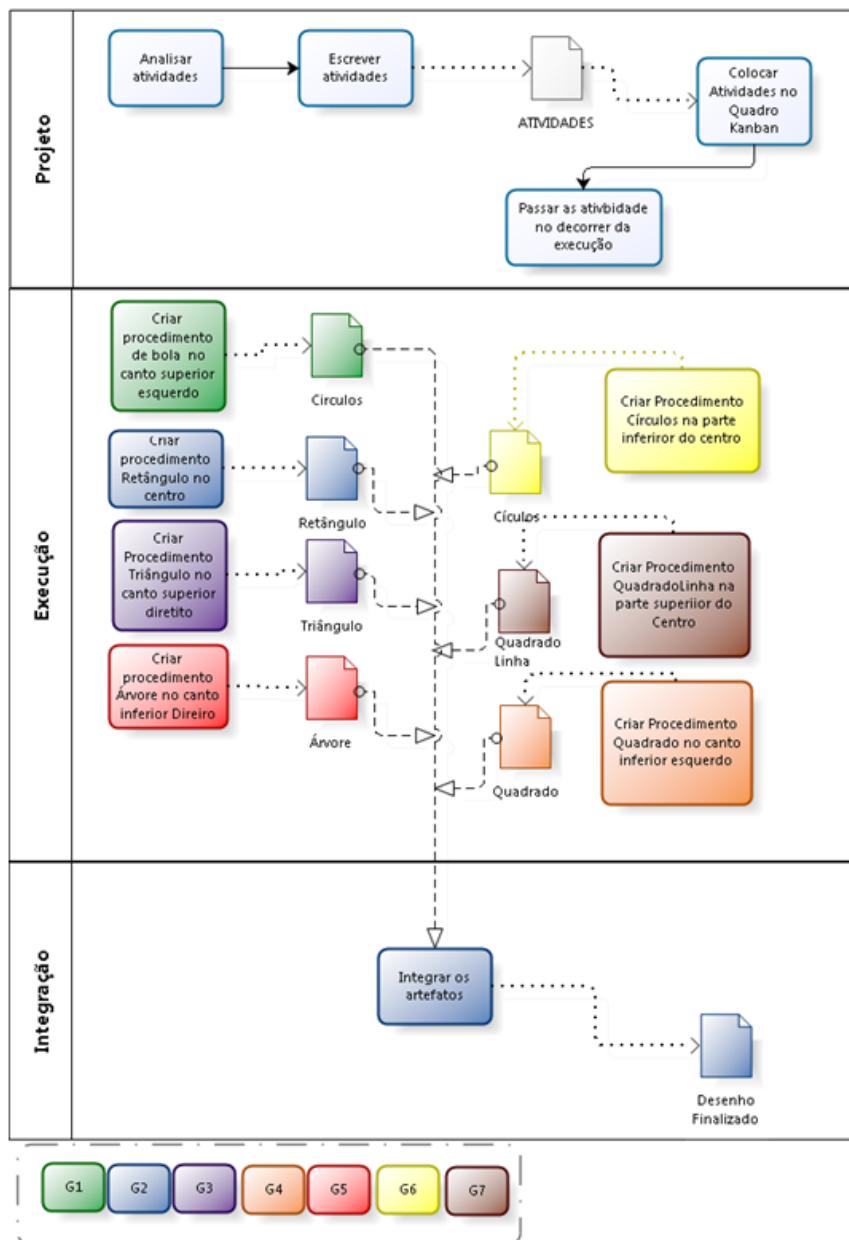


Figura 4: Projeto Logo

Fonte: Autoria própria.

APÊNDICE C – MAPA DAS ATIVIDADES DE ANÁLISE DOS DADOS

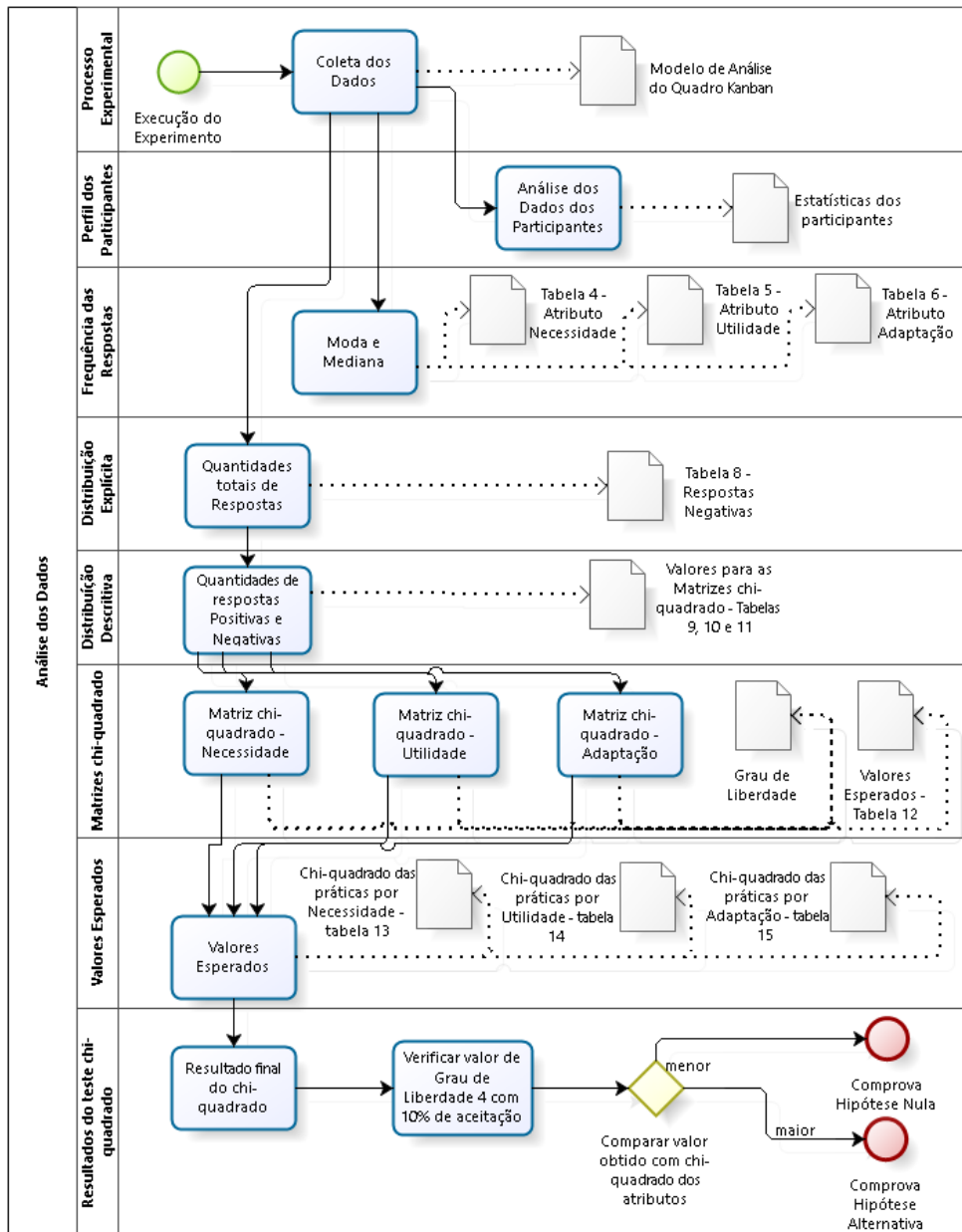


Figura 5: Mapa de Atividades 1

Fonte: Autoria própria.

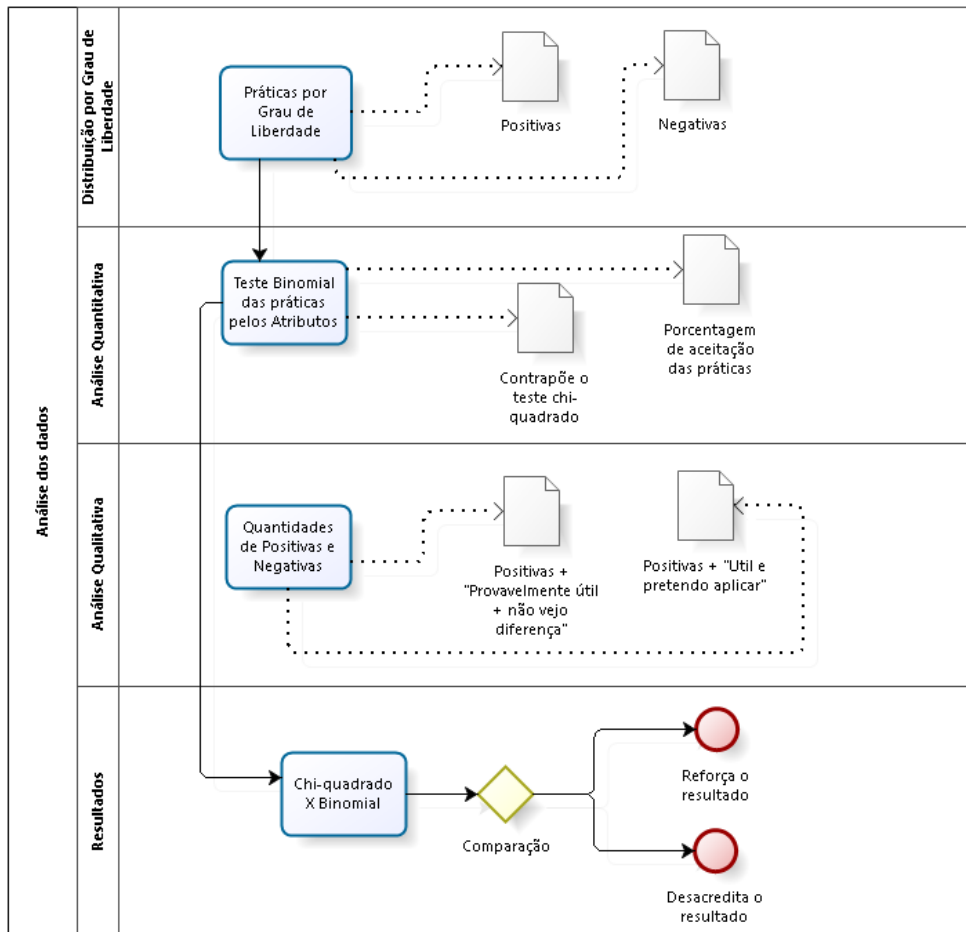


Figura 6: Mapa de Atividades 2

Fonte: Autoria própria.