

**FACULDADES INTEGRADAS DE TAQUARA
CURSO DE SISTEMAS DE INFORMAÇÃO**

**TOOGER: FERRAMENTA DE GERENCIAMENTO DE PROJETOS ATRAVÉS DO
USO DE MATRIZ GUT**

PATRICK FERNANDO SOUZA DA SILVA

Taquara

2018

FACULDADES INTEGRADAS DE TAQUARA
CURSO DE SISTEMAS DE INFORMAÇÃO

TOOGER: FERRAMENTA DE GERENCIAMENTO DE PROJETOS ATRAVÉS DO
USO DE MATRIZ GUT

Trabalho de conclusão apresentado ao Curso de Sistemas de Informação das Faculdades Integradas de Taquara, como requisito parcial para obtenção do grau de bacharel sob orientação do Profº. M.e. Leonardo Ribeiro Machado.

Taquara
2018

TOOGER: FERRAMENTA DE GERENCIAMENTO DE PROJETOS ATRAVÉS DO USO DE MATRIZ GUT

Patrick Fernando Souza da Silva¹

Leonardo Ribeiro Machado²

RESUMO

O presente artigo tem como objetivo documentar a construção da ferramenta Tooger e demonstrar os resultados e os benefícios obtidos através da sua utilização. A ferramenta foi desenvolvida para a visualização e o gerenciamento de projetos, utilizando a matriz GUT para priorizá-los. O estudo demonstra o desenvolvimento da ferramenta, abordando todas as fases percorridas, assim como as tecnologias e metodologia utilizadas nesse processo. Além disso, este artigo aborda conteúdos sobre gerenciamento de projetos e as dificuldades para gerenciá-los em um contexto que envolve o portfólio em que estão inseridos.

Palavras-chave: cronograma, gerenciamento, projetos, portfólios, matriz GUT.

ABSTRACT

The purpose of this paper is to document the construction of the Tooger tool and to demonstrate the results and benefits obtained through its use. The tool was developed for project visualization and management, using the GUT matrix to prioritize them. The study demonstrates the development of the tool, addressing all the stages covered, as well as the technologies and methodology used in this process. In addition, this article discusses content about project management and the difficulties to manage them in a context that involves the portfolio in which they are inserted.

Keywords: *schedule, management, projects, portfolios, GUT matrix.*

¹ Acadêmico do curso de Sistemas de Informação das Faculdades Integradas de Taquara – FACCAT. E-mail: patricksilva.sw@gmail.com

² Professor do curso de Sistemas de Informação das Faculdades Integradas de Taquara – FACCAT. Mestre em computação aplicada. E-mail: leonardomachado@faccat.br

1 INTRODUÇÃO

A execução de um projeto necessita de seu gerenciamento, esta tarefa é complexa, pois envolve um conjunto de técnicas, ferramentas, habilidades e tomada de decisão. Além de todas as constantes mudanças ao longo do desenvolvimento do projeto a fim de atender os requisitos, que também dificultam seu gerenciamento. Se juntar todo esse contexto que comporta um único projeto e aplicar a um conjunto de projetos, logo se tem um conjunto de problemas, que deve ser gerenciado (CIERCO et al., 2015).

Portfólio é a definição para o contexto onde se tem um conjunto de projetos, portfólio retrata um cenário comum em empresas de desenvolvimento de software, onde se tem inúmeros projetos a serem gerenciados. O objetivo de um portfólio é facilitar a administração, permitindo aplicar técnicas ao conjunto de projetos que auxiliarão na tomada de decisão, principalmente a técnica que consiste em priorizar projetos (CIERCO et al., 2015).

Este trabalho documenta o desenvolvimento da ferramenta Tooger, a qual permite ao usuário gerenciar seus projetos, visualizar cronogramas de projetos priorizados e registrar tarefas que fazem parte dos projetos. A visualização do cronograma de projetos utiliza a técnica Matriz GUT, utilizada para priorizar os projetos a serem desenvolvidos. A utilização da matriz GUT permite ordenar os projetos de acordo com um cálculo que considera: a gravidade (G), a urgência (U) e tendência (T).

As ferramentas de gestão de projetos existentes possuem uma série de funcionalidades para garantir o processo de execução dos projetos. Baseiam-se nas tarefas de cada projeto e se destacam por vincular tarefas às outras e também indicar tarefas que devem ser concluídas antes de novas tarefas serem iniciadas. Apesar das funcionalidades oferecidas, essas ferramentas não permitem a visualização de um conjunto de projetos priorizados, disponibilizando uma visão de todos os projetos organizados em um cronograma seguindo algum tipo de prioridade, ou alguma premissa estabelecida em tendência de mercado. A ferramenta desenvolvida supre esta lacuna, permitindo ao usuário a priorização dos projetos de seu portfólio através do uso da matriz GUT.

A seção dois traz um referencial teórico a respeito dos assuntos relacionados à pesquisa. A seção três aborda as pesquisas relacionadas ao assunto deste trabalho. A seção quatro retrata a análise, planejamento e projeto do sistema desenvolvido. A seção cinco explica o desenvolvimento da ferramenta, técnicas, tecnologias utilizadas e os testes realizados. A seção seis aborda os resultados que o software oferece ao usuário.

2 REFERENCIAL TEÓRICO

2.1 Gerenciamento de Projetos

O avanço tecnológico está constantemente inovando produtos. Estes produtos surgem através de um projeto elaborado, que é formalmente a sequência da ideia principal. O projeto pode ser um produto qualquer e como um todo é produzido por uma organização. Essa empresa, criadora do produto, estabelece todas as etapas do projeto definindo como serão construídas (VARGAS, 2016).

Nas empresas de desenvolvimento de software, gerenciar inúmeros projetos é uma tarefa com elevada dificuldade. Pois o gerenciamento envolve todas as etapas do projeto. Essas etapas são fornecidas por uma equipe, implicam em custo e tempo para serem desenvolvidas. Além disso, desenvolver projetos exige tomadas de decisão para atribuir prioridades. Portanto, o gerenciamento de projetos é um conjunto de ferramentas que permite à empresa distribuir e organizar todos os projetos e suas equipes envolvidas (RICARDI, 2013).

2.2 Portfólio de projetos

Portfólios representam um conjunto de projetos, sua gestão adequada pode trazer benefícios para as empresas, diferenciando-as em um mercado competitivo, tanto em nível nacional como internacional. O aspecto que pode fazer a diferença é o crescimento das empresas que monitoram o desenvolvimento de seus produtos por projetos (MORAES, 2015).

Para gerir um portfólio, é necessário o uso de técnicas que identificarão qual projeto deve ser incluído ao conjunto de projetos, ou qual deles deve deixar o grupo, podendo ser aplicado a outro portfólio. A gestão também permite aplicar regras que determinarão uma ordem em que os projetos devem ser executados. Essa priorização é determinada por metas específicas da empresa, visando otimizar o máximo do resultado do conjunto de projetos (SANTOS et al., 2016).

Portanto, o portfólio mantém um conjunto de projetos e também é orientado por um conjunto de técnicas de priorização, que vão facilitar as organizações nas tomadas de decisões, permitindo que os projetos sejam bem executados (ARAÚJO, 2017).

2.3 Priorização de projetos através da matriz GUT

Matriz GUT é uma ferramenta utilizada com o objetivo de priorizar problemas. Este recurso ajuda o gestor quando é necessário priorizar tarefas onde existem inúmeras atividades a serem efetuadas. GUT é uma sigla que significa: gravidade (G), urgência (U) e tendência (T) (DAYCHOUM, 2018).

De acordo com César (2013), a gravidade representa o impacto do problema, caso ele venha a ocorrer. A urgência representa o prazo para se resolver o problema. A tendência representa o potencial de crescimento do problema, as chances de ele se tornar maior. Para se efetuar o cálculo da Matriz GUT, é atribuído um valor de 1 a 5 para cada uma das variáveis da matriz. Após isso, os valores são multiplicados um pelo outro, e o valor final da matriz GUT é obtido.

Uma das principais dificuldades da gestão é a priorização, utilizar algum parâmetro ou um modelo para determinar prioridade é uma boa estratégia. Os projetos não deixam de ser problemas, alguns com prioridades, outros nem tanto. Portanto, a grande vantagem em aplicar o conceito GUT está em determinar valores nas categorias gravidade, urgência e tendência. Sendo assim, com a atribuição dos valores, o método da matriz GUT pode auxiliar na tomada de decisão (CONCEIÇÃO et al., 2015).

2.4 Softwares gerenciadores de projetos

Gerenciar projetos requer planejamento, conjunto de conhecimentos, habilidades, processos, diretrizes e técnicas para distribuir e organizar tarefas relacionadas aos projetos. Por conta destes diferentes itens que contemplam a gerência de projetos, recomenda-se a utilização de um software como auxílio (KERZNER, 2016).

Os softwares utilizados para gerenciar projetos permitem que a solução seja dividida em fases: planejamento, desenvolvimento e encerramento. As fases do projeto são divididas em etapas, portanto um software que gerencia projetos tem como objetivo também gerenciar todas estas etapas. Entende-se que um sistema de gestão de projetos deve fazer uma boa gestão considerando premissas, um exemplo é bloquear o início de novas tarefas que contenham pré-requisitos relacionados a outras tarefas, respeitando um cronograma (MARTINS, 2010).

Ferramentas de gerenciamento de projetos possuem inúmeros recursos, entre eles a visualização de cronogramas, para que as atividades sejam realizadas de acordo com os prazos pré-determinados. Esses cronogramas são feitos de acordo com as etapas. Porém o objetivo dessas

ferramentas não é distribuir vários projetos em um cronograma e nem priorizar os projetos, mas sim dispor as tarefas e delegar de quem é sua responsabilidade (MARTINS, 2010).

2.5 Arquitetura, metodologia e tecnologias

Esta seção traz o referencial teórico acerca da arquitetura, metodologia e tecnologias utilizadas neste projeto de pesquisa.

2.5.1 Arquitetura MVC

MVC é um padrão de arquitetura para desenvolvimento de software, consiste em definir de forma estruturada e organizada as responsabilidades do código da aplicação com as responsabilidades do código da apresentação, onde os objetos da interface vão interagir com os objetos da aplicação. A sigla MVC (*model, view, controller*), significa: modelo, visão e controlador (BEZERRA, 2015).

A camada Modelo (*model*), é composta pelas classes que são responsáveis pela modelagem e definição do software que corresponde ao estado, comportamento e estrutura de dados a serem visualizados pelo usuário. De modo geral é como uma interface onde as demais rotinas da aplicação possam manipular seus dados (BEZERRA, 2015).

A camada Visão (*view*), representa a interface e como vai interagir com o usuário. São as diferentes perspectivas, exemplo: as telas de cadastros, as operações de lançamentos e também os relatórios. Esta camada recebe e envia valores de cada propriedade até chegar na camada modelo (BEZERRA, 2015).

O objeto Controlador (*controller*) representa um formato responsável por intermediar as solicitações da camada view para montar o modelo (*model*). Também é através desta camada que os dados (*model*) são recuperados e a camada visão (*view*) pode ser montada (BEZERRA, 2015).

2.5.2 Visual Studio C# .NET Framework

Visual Studio é uma IDE (*Integrated Development Environment*), ou ambiente de desenvolvimento integrado, mantida pela Microsoft. Permite trabalhar com várias tecnologias, além de combinar várias linguagens em seu ambiente de desenvolvimento, tais como: C#/VB ou C++, Javascript ou Python (MICROSOFT, 2018).

Essa IDE é uma ferramenta muito utilizada, é consistente, segura e de grande potencial para criação de softwares. O Visual Studio é uma ótima ferramenta para desenvolvimento de software, repleto de recursos a serem utilizados para criar softwares. A IDE está disponível para Windows e Mac, além do editor, a IDE permite depurar os códigos permitindo corrigir problemas rapidamente (MICROSOFT, 2018).

2.5.3 Bootstrap

Bootstrap é um *framework* para desenvolvimento HTML, CSS e Javascript. Este conjunto de padrões e especificações para o *front-end* tem a finalidade de ser aplicado em sistemas web, deixando o sistema com interface mais agradável, e eliminando preocupações com o layout ao executar a aplicação em diferentes ambientes (BOOTSTRAP, 2018).

Bootstrap é o *framework* com grande representatividade no seguimento de desenvolvimento de sites responsivos, além de aplicações web e soluções mobile. Este *framework* contém os arquivos de estilo CSS, componentes Javascript necessários para o rápido início no desenvolvimento de uma aplicação web, além de uma documentação simples e confiável (BOOTSTRAP, 2018).

2.5.4 Banco de dados MySQL

MySQL é um sistema gerenciador de banco de dados (SGDB) relacional, robusto, rápido, multitarefa e também multiusuário. Utiliza a linguagem SQL (linguagem de consulta estruturada) como instrução para buscar ou manipular registros armazenados no banco de dados. MySQL é a marca comercial e pertence à empresa Oracle Corporation (ORACLE, 2018).

O software MySQL oferece duas licenças para escolha do usuário, uma licença pode ser comprada como licença padrão da Oracle, e outra é um produto Open Source, sob os termos da Licença pública geral GNU. Seu projeto consiste em operar com sistemas de pequeno, médio e grande porte (ORACLE, 2018).

2.5.5 Método Kanban

Kanban é um método ágil para organizar tarefas, um método bastante eficiente que consiste em listar as tarefas distribuindo-as em quadros que são responsáveis por indicar a fase das tarefas. Permite ter uma visão geral de todas as tarefas a serem desenvolvidas ou as que estão em andamento (MARIOTTI, 2016).

A metodologia de desenvolvimento de software utilizada neste trabalho usa o método de fichas e quadro de atividades do Kanban - junto com o princípio de reduzir o WIP (Work in Progress), por exemplo - como um método para a gestão de projeto de software dentro de uma metodologia ágil de desenvolvimento de software. O Kanban (físico ou digital) é uma ferramenta que se utiliza na técnica de controle do fluxo de produção de um software, como parte do método de gestão de projeto de software adotado dentro de uma metodologia ágil, que pode envolver ainda vários métodos e técnicas para análise, projeto, codificação, testes, implantação e manutenção de software (CASTELLO et al., 2018).

3 TRABALHOS RELACIONADOS

Nesta seção são apresentados os trabalhos relacionados à atual pesquisa, detalhando as características em cada um deles.

3.1.1 Artia – Ferramenta de Gestão de Projetos Online

O software Artia é uma ferramenta para gestão de projetos que possui uma interface *web*, abstraindo o formato de instalação e dependência de sistema operacional específico para ser executado. É uma ferramenta intuitiva e com recursos de fácil utilização, a interface ainda permite visualizar atividades, e também possui um cronograma de Gantt (ARTIA, 2018).

O principal objetivo da ferramenta é gerenciar projetos, atividades em equipes, horas do projeto e entregas. A ideia é que o usuário registre as atividades que pertencem ao projeto, responsabilidades, prazos e também horas por atividade, assim o usuário terá uma visão das tarefas realizadas ou não concluídas, relacionadas aos membros da equipe. O usuário pode observar indicadores que demonstram como está o andamento do projeto, de forma colaborativa, ou seja, todos os membros do projeto podem acessar facilmente seus projetos por um endereço da internet (ARTIA, 2018).

A ferramenta possui um plano básico gratuito para uso individual, os demais planos podem ter mais usuários e são repassados os valores de acordo com a quantidade de funcionalidades. O usuário também pode optar por planos mensais ou anuais (ARTIA, 2018).

Embora a ferramenta apresente diversas funcionalidades bem interessantes, ela não permite a visualização de um cronograma que organiza vários projetos, determinando prioridade de início para cada projeto. Funcionalidade que a ferramenta apresentada neste artigo executa como seu principal objetivo.

3.1.2 Asana – Ferramenta On-line para Gestão de Projetos

Asana é uma ferramenta web de ótima qualidade para gerenciar projetos, sua operação é orientada às tarefas dos projetos e suas equipes. Este formato de operação no sistema é bastante intuitivo e de fácil utilização, proporcionando ao usuário uma experiência de uso onde ele terá uma visão rápida do status de cada projeto (ASANA, 2018).

Como objetivo principal, a ferramenta exibe o trabalho realizado em cada projeto por indivíduo da equipe, oferecendo uma navegabilidade rápida onde é exibido cada estágio das tarefas, disponibilizando também um cronograma. O plano permite visualizar todas as interações do projeto, destacando prazos, assim o cronograma consegue dispor os projetos orientado a um calendário (ASANA, 2018).

O sistema tem um plano simples e gratuito para uso de pequenas equipes que estão começando a interagir e organizar seus projetos. Na sequência, a empresa oferece um plano que é cobrado por usuário. Oferece também um plano para maiores empresas, onde contempla maior controle e suporte (ASANA, 2018).

Como seu principal propósito é gerenciar equipes e seus projetos, a ferramenta não permite visualizar um cronograma do portfólio, determinando prioridade para seus projetos. Diferente da ferramenta Tooger, desenvolvida neste estudo, pois ferramenta Tooger disponibiliza essa função, além de priorizar os projetos, utilizando o conceito de matriz GUT.

3.1.3 Jira – Serviço de Gerenciamento de Trabalho em Grupo

Jira é uma ferramenta bastante popular entre os desenvolvedores. Robusta, concentra-se em utilizar o conceito Kanban para organizar tarefas, possui vários recursos avançados e interações com ferramentas externas (ATLASSIAN, 2018).

A ferramenta tem como objetivo trazer um ambiente completo, com vários recursos, integrações, relatórios. Possui controles que monitoram o tempo e recurso aplicado em cada tarefa. Possui um ambiente bastante profissional diretamente direcionado a gerência de projetos, aplica-se facilmente a grandes equipes de projetos, e traz um comportamento que facilita a visão e acompanhamento do progresso de cada etapa do projeto (ATLASSIAN, 2018).

O serviço não possui nenhum plano gratuito, o controle da licença é feito por número de usuários, o plano básico é para até 10 usuários. Uma equipe de 11 até 100 usuários o valor mensal da licença é multiplicado por usuário. Quando existir equipe com mais de 100 usuários é possível calcular o valor da licença no próprio site (ATLASSIAN, 2018).

A ferramenta é completa, mas não dispõe do recurso para visualização do cronograma de vários projetos priorizados através da matriz GUT. Este é um recurso importante e disponibilizado pela ferramenta Tooger, desenvolvida para essa específica finalidade.

4 METODOLOGIA

4.1 Concepção

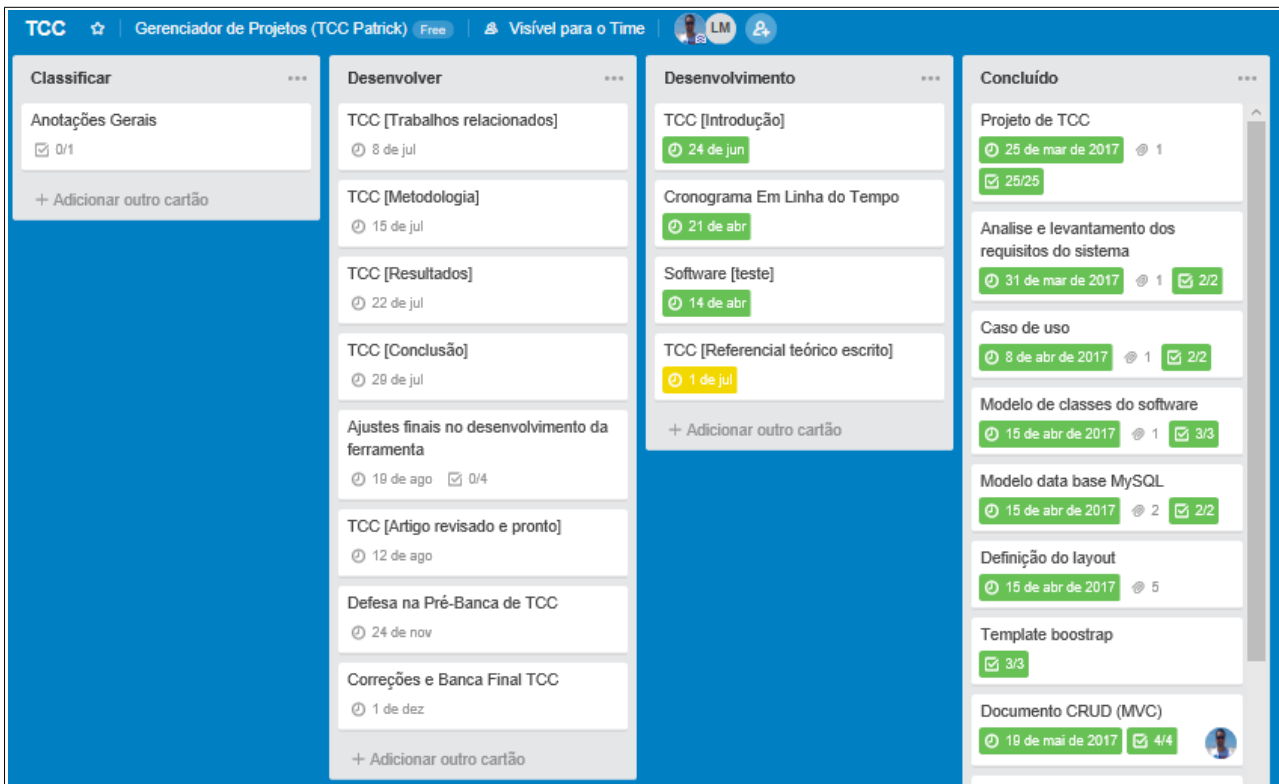
Os assuntos tratados no referencial teórico retratam circunstancialmente as dificuldades em gerir projetos de forma eficiente. A dificuldade aumenta quando é preciso gerenciar um conjunto de projetos, ter uma visão de quais devem ser desenvolvidos primeiro ou projetar um extenso cronograma envolvendo um portfólio de projetos. Observando todo esse cenário, e considerando que muitas empresas utilizam planilhas eletrônicas para controlar os projetos em desenvolvimento, já que essa tarefa é bastante trabalhosa, imagina-se que todo esse trabalho possa ser minimizado, se tornando realmente mais eficiente, onde a relação de tempo e prioridade dos projetos reflitam em um melhor aproveitamento.

O sistema desenvolvido aborda diretamente a visualização de um cronograma de portfólios baseado na prioridade determinada por uma matriz GUT. A ideia é que o usuário possa registrar os projetos aplicando uma pontuação para cada variável entre estas: gravidade(G), urgência(U) e tendência(T). De acordo com o que foi aplicado em cada variável a ferramenta permite visualizar o cronograma, ou seja, o sistema prioriza um conjunto de projetos seguindo a técnica de matriz GUT.

Ainda com a ideia pré-elaborada, foi necessária a construção do planejamento da ferramenta e a validação de alguns conceitos e funcionalidades. Logo identificou-se uma possível complexidade no algoritmo que elabora o cronograma, por isso foram definidos alguns conceitos importantes a serem tratados na ferramenta: equipes, usuários ou membros da equipe, projetos, tarefas, requisições e as definições de prioridade.

Com os conceitos abordados e o planejamento do software a ser iniciado, é necessária a escolha de uma metodologia de desenvolvimento. Para tanto optou-se por utilizar o método Kanban, utilizou-se a ferramenta Trello, que possibilita uma boa organização de tarefas a serem realizadas. A figura 1 abaixo demonstra o quadro Kanban que foi mantido para a construção da ferramenta.

Figura 1 – Distribuição das Tarefas.



Fonte: Autor.

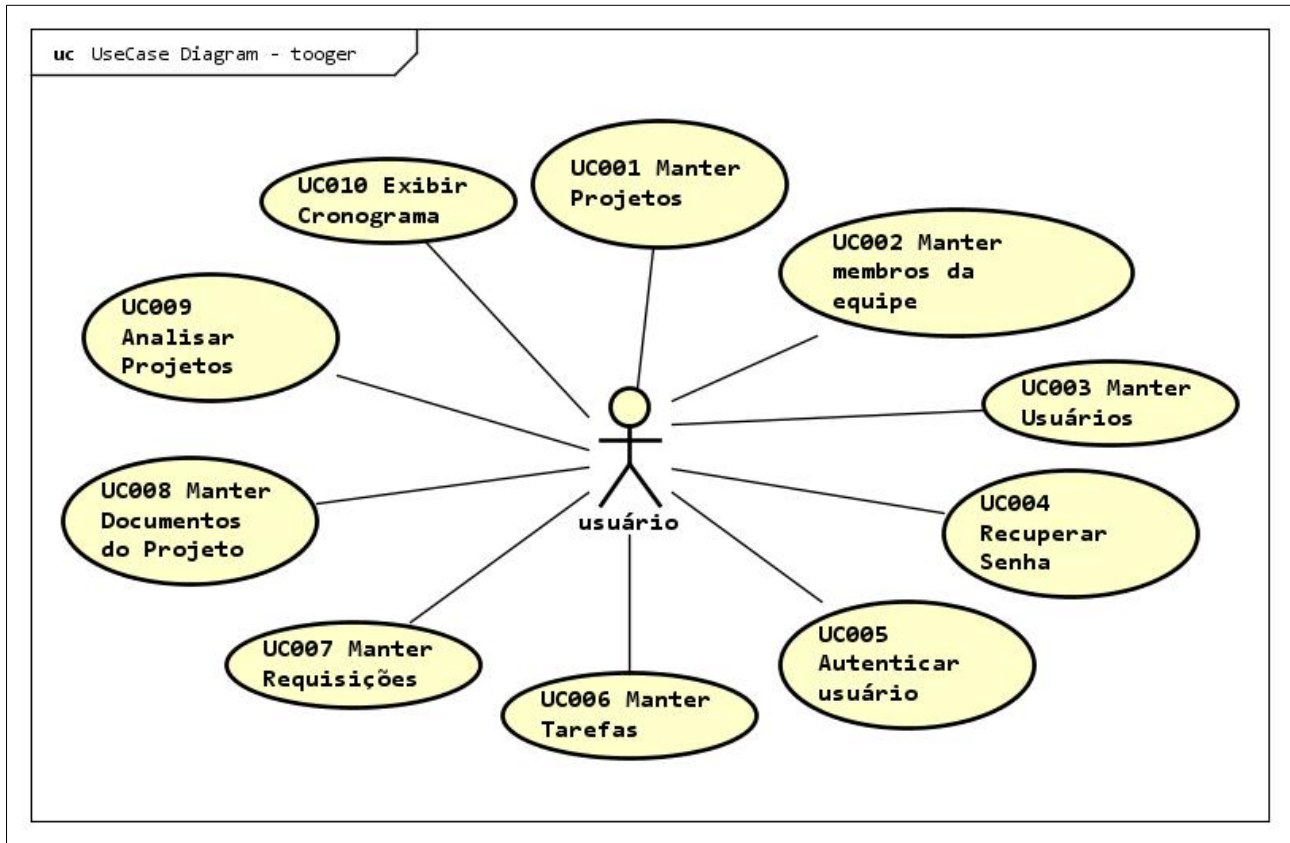
4.2 Elaboração

O desenvolvimento do software iniciou com a análise do sistema. No início, existiu uma preocupação com o algoritmo e com os recursos do sistema, principalmente os recursos que se relacionam, portanto a análise foi indispensável nesse processo, tanto para identificar cada recurso do sistema quanto para identificar posteriores falhas.

A análise foi baseada em histórias de usuários. Foi desenvolvida a documentação da ferramenta, incluindo todos os requisitos do sistema, diagrama de casos de uso e diagrama de domínio do sistema. Essa documentação precisou ser aprovada, já que o modelo da aplicação é destinado a uma empresa de desenvolvimento de software que tem por necessidade priorizar o projeto conforme a ferramenta Tooger proporciona, pois essa empresa monta o cronograma de seus projetos de forma manual.

Partindo dos requisitos do software foi gerado o diagrama de casos de uso essenciais para o desenvolvimento da ferramenta. As histórias de usuário construídas refletem o comportamento esperado pelo sistema. A figura 2 abaixo demonstra o diagrama de casos de uso da ferramenta Tooger, gerado a partir da ferramenta *Astah*.

Figura 2 – Diagrama de casos de uso



Fonte: Autor.

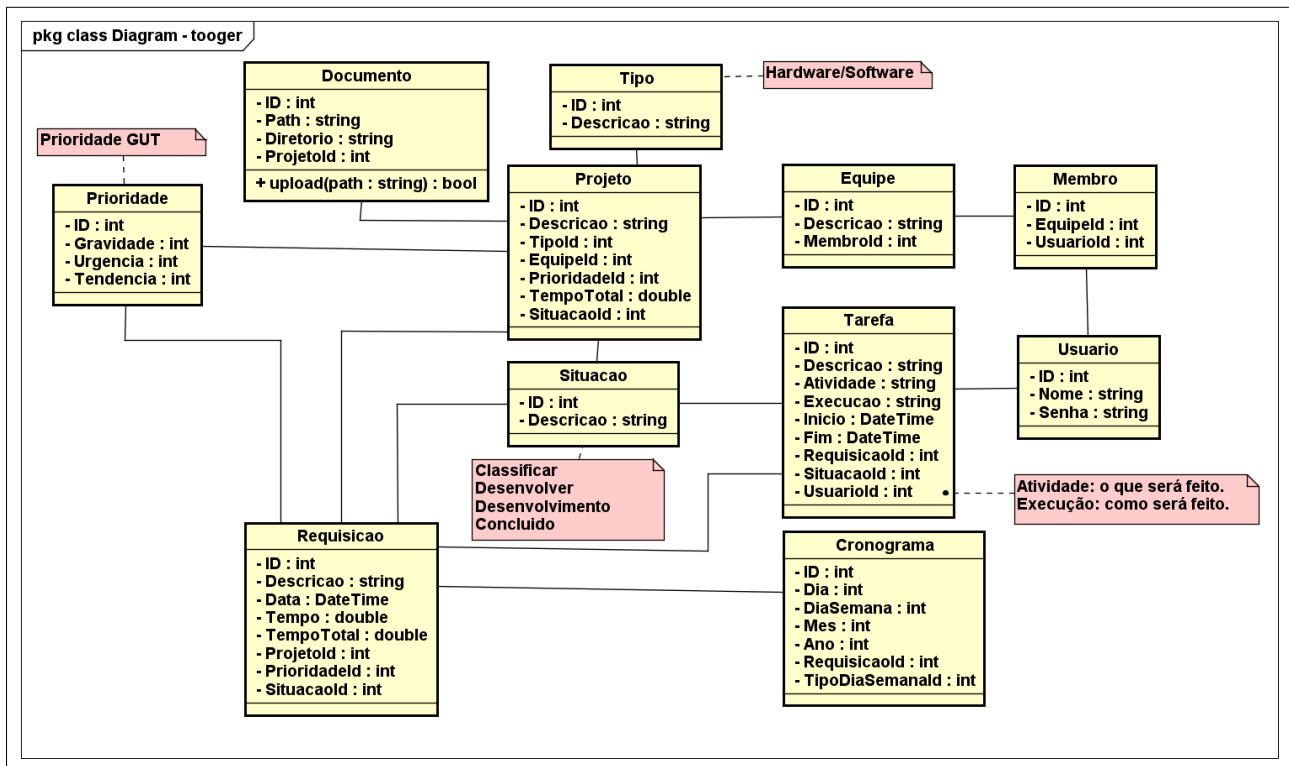
Além dos requisitos funcionais relacionados no quadro acima, durante a fase de análise também foram trabalhados os requisitos não funcionais da ferramenta. Neste âmbito, houve também uma preocupação com a usabilidade oferecida pela ferramenta.

O projeto do sistema considerou a utilização de um layout padrão utilizando um *framework* para o *front-end* da aplicação.

Após a definição do layout a ser utilizado pela ferramenta, bem como todas as funcionalidades que são oferecidas pela mesma, seguiu-se para a elaboração do diagrama de domínio, através de todos os conceitos identificados nos requisitos documentados.

A Figura 3 abaixo demonstra o modelo de domínio da ferramenta Tooger.

Figura 3 – Diagrama de Domínio



Fonte: Autor.

A documentação e os diagramas desenvolvidos auxiliaram durante a fase de construção da ferramenta, pois permitiram a antecipação de problemas que de outra forma só seriam detectados nas fases finais do projeto, ou até mesmo não detectados. Após a fase de elaboração ter sido concluída, o próximo passo foi a construção da ferramenta.

4.3 Construção

Foram determinadas, no projeto de desenvolvimento do sistema, as tecnologias a serem utilizadas na construção. A IDE de desenvolvimento foi o Visual Studio. A linguagem utilizada no desenvolvimento foi C#, por ser uma linguagem orientada a objetos, bastante atual e comum. O processo de codificação se apoiou na documentação elaborada nas fases anteriores.

A implementação seguiu o cronograma de tarefas distribuído no quadro Kanban. As tarefas foram sendo implementadas uma a uma. Eventualmente foram encontradas inconformidades no código, porém as ferramentas e a metodologia utilizadas facilitaram bastante a correção de erros durante este processo.

Durante as etapas do desenvolvimento, mudanças surgiram e foram sendo tratadas de acordo com o método utilizado. A cada alteração necessária, as tarefas eram retornadas, movendo os quadros de modo a serem revisadas ou refeitas, percorrendo novamente as fases anteriores.

5 DESENVOLVIMENTO

5.1 Estrutura do software

O desenvolvimento da ferramenta iniciou com a produção de uma tela básica contendo o *layout* com um protótipo para teste, validando tecnologia, banco de dados, e principais componentes distribuídos na tela. O sistema deveria ser rápido, intuitivo, com um layout seguro, inclusive a execução em diferentes navegadores não deve apresentar distorção. Por isso optou-se pelo front-end *Bootstrap*, facilitando o desenvolvimento e eliminando essas preocupações com o layout. O padrão do *Bootstrap* usado se manteve em todo o projeto, apesar de ter algumas modificações no resultado do *template* final, seguindo o padrão de cores adotadas. Para esse resultado a estrutura dos arquivos foram distribuídas em três pastas:

- a) Pasta “*css*”: onde se encontram todos os estilos da aplicação, os estilos do *front-end* e também os estilos customizados.
- b) Pasta “*js*”: onde se encontram todos arquivos “*javascript*” da aplicação, os arquivos padrões do *template* e também os arquivos customizados.
- c) Pasta “*images*”: pasta que contém todos os arquivos de imagem do *template* e também da aplicação em geral.

Os dados são registrados no banco de dados MySQL. Para esse controle, a aplicação mantém um conector responsável pela execução dos comandos SQL. O conector padrão normalmente é uma DLL, com uma série de comandos para realizar as instruções no banco de dados. Isso induz a criação de uma classe no sistema que vai tratar todas estas questões. O padrão *MVC* da aplicação *Web* criado pela IDE da Microsoft (Visual Studio) já sugere uma classe para esta finalidade, a classe “*ApplicationDbContext*”. Para uma melhor organização estrutural no sistema, partindo do princípio que uma aplicação poderá ter conexão com vários bancos de dados, foi criado um diretório chamado “*Data*”, que contém todas as classes de conexão com os diferentes bancos de dados caso a aplicação precisar.

O desenvolvimento da aplicação, muitas vezes, precisa que algum atributo não previsto na análise, em determinada classe, seja criado. Essa tarefa requer também a criação deste atributo no banco de dados, uma vez que uma propriedade de uma classe também é armazenada no banco de

dados. O uso do método Kanban no desenvolvimento facilitou esse processo. As construções de aplicações mais modernas tratam esse problema, conhecido conceitualmente como "migrations". As *migrations* são ações que facilitam o desenvolvedor, pois garantem que automaticamente todos os atributos criados na classe modelo sejam replicados no banco dados. Esse controle é feito por versões, uma outra facilidade para manter uma melhor gestão do código em desenvolvimento. Essa estrutura é organizada em um diretório chamado de *migrations*, cada arquivo é uma versão da *migration* que está contido dentro deste diretório.

As instruções de programação seguiram o modelo MVC. Para isso, o sistema mantém uma estrutura com as classes nos devidos diretórios: “*Models*”, “*Views*” e “*Controllers*”. As classes que correspondem a cada um destes conceitos serão tratadas na próxima seção, codificação.

5.2 Codificação

O conteúdo do banco de dados contém as estruturas de tabelas que são traduzidas para uma classe de domínio ou modelo no sistema, permitindo o desenvolvimento orientado a objetos. Um exemplo é a tabela “equipe” que é traduzida para a classe “Equipe”, onde todas as colunas da tabela correspondem às propriedades da classe. Os arquivos destas classes estão organizados no diretório “*Models*”.

É necessária uma interface para que o usuário possa trafegar, manipular os dados e visualizar os dados registrados no sistema. Para a interface web, o projeto da ferramenta se preocupou em formato simples para organizar as páginas que serão renderizadas para o usuário. Sendo assim, o diretório *views* contém um diretório para cada objeto da ferramenta. Permite-se, com essa organização, separar adequadamente cada interação responsável por manipular o objeto. A IDE permite programação direta usando C# nestas páginas HTML, conceito conhecido como razor. Outra característica da ferramenta é *minificar* todos os arquivos *css* e *js* no momento da compilação, garantindo maior desempenho. Essa configuração é tratada nos *bundlers* do sistema.

Os dados informados em cada interface com o usuário precisam ser registrados no banco de dados, até agora o sistema contém as estruturas que correspondem às tabelas e também as estruturas que são responsáveis por exibir as informações ao usuário. Porém existe uma série de funcionalidades que devem ser desenvolvidas para garantir que o dado informado interaja com o banco de dados. Essas funcionalidades são organizadas no diretório *Controllers* onde cada classe que corresponde ao banco de dados terá também uma classe *controller*, responsável pelas operações entre a *view* e a *model*.

Neste projeto, apenas um programador foi envolvido, sendo assim o melhor é determinar um roteiro de desenvolvimento. O roteiro teve início na codificação de um objeto que possuía poucos atributos e de fácil implementação. Portanto, a primeira funcionalidade da ferramenta a ser desenvolvida foi o conceito Equipe, e depois foi implementado o conceito mais importante do sistema: o Projeto. Concluída a codificação destes dois conceitos, foi possível desenvolver os demais.

Para o versionamento do código-fonte da ferramenta foi utilizada a ferramenta *TortoiseGit*. A cada código gerado uma versão tinha o *commit* na nuvem sobre o controle da plataforma da Microsoft *Team Foundation Server*, do Visual Studio. Assim era garantido o backup seguro do código em versões.

5.3 Testes

Os testes no sistema garantem maior qualidade, e também a identificação de possíveis falhas no desenvolvimento.

No projeto foram aplicados dois principais testes, o primeiro foi o teste de consistência dos modelos orientados a objetos. Para este teste, são aplicadas revisões técnicas, com a finalidade de validar o projeto. Este teste consiste em inspecionar os modelos do sistema, examinando a consistência dos objetos e suas relações, coesão das classes e identificação dos atributos (PRESSMAN e MAXIN, 2016).

Para aplicar a revisão técnica é necessário se apropriar da documentação da análise do projeto, depois analisar cada arquivo de classe do sistema a fim de identificar as boas práticas de desenvolvimento, relacionando com a documentação. Para as conexões entre objetos, a técnica a ser aplicada é inverter as conexões, desfazendo os processos do sistema a fim de pontuar se o serviço solicitado é razoável. Assim, determina-se a necessidade de haver mais classes ou unificação de responsabilidades. Esta tarefa foi distribuída entre dois profissionais, um deles o próprio desenvolvedor. O teste foi satisfatório.

Como a ferramenta é web e com intenção de possuir rotinas simples proporcionando ao usuário uma boa experiência, são necessárias rotinas de testes que vão explorar falhas no sistema. O teste exploratório foi o próximo teste aplicado. Esse tipo de teste, consiste em utilizar o sistema, tentando fazer a rotina mais próximo possível que o usuário o faria no sistema, também colocar valores indevidos nos campos solicitados a fim de explorar falhas. Para aplicar o teste é necessário que o usuário tenha interação com o sistema. Neste teste foi observado que o sistema deveria

proporcionar uma melhor usabilidade nas funcionalidades oferecidas ao usuário. Todos os ajustes foram feitos para garantir o melhor uso da ferramenta.

6 RESULTADOS

Através desse trabalho foi desenvolvida a ferramenta Tooger, a qual tem a finalidade de demonstrar um cronograma contendo um conjunto de projetos, disponibilizando as informações de cada projeto de forma centralizada, exibindo tarefas, requisições, e a quem pertence a responsabilidade de cada projeto, garantindo um acesso simples e intuitivo, e que o usuário possa acessar na web. A Figura 4 apresenta a tela onde são exibidas as informações do projeto selecionado, demonstrando as principais informações do projeto: horas projetadas, horas executadas, equipe responsável, documentos anexados ao projeto, e também as últimas interações dos usuários nas tarefas.

Figura 4: Tela de visualização do projeto

Fonte: Autor.

O projeto de software mantém inúmeras iterações. Uma iteração pode ser uma fase do projeto de desenvolvimento de software, ou então uma solicitação de melhoria. O software trata isso como: requisição. Para qualquer fase ou solicitação para um projeto é necessário registrar uma requisição, que poderá agrupar inúmeras tarefas. A Figura 5 abaixo ilustra a visualização objetiva do

andamento das requisições e dispõe as tarefas em quadros, destacando em raias como: classificar, o que deve ser desenvolvido (desenvolver), desenvolvimento e o que está concluído.

Figura 5 – Tela de visualização de requisições

Fonte: Autor.

Na tela demonstrada na Figura 5 acima, o sistema permite a edição de uma requisição. Além disso, nessa tela também é possível a criação e manutenção das tarefas, as quais são distribuídas no quadro de tarefas de acordo com a sua fase. O sistema exibe o total de horas projetado para a requisição, bem como o total de horas que já foi executado. A tela demonstrada possui também um quadro de status, onde é demonstrada a previsão de conclusão da requisição e uma barra com o percentual de andamento do projeto no momento.

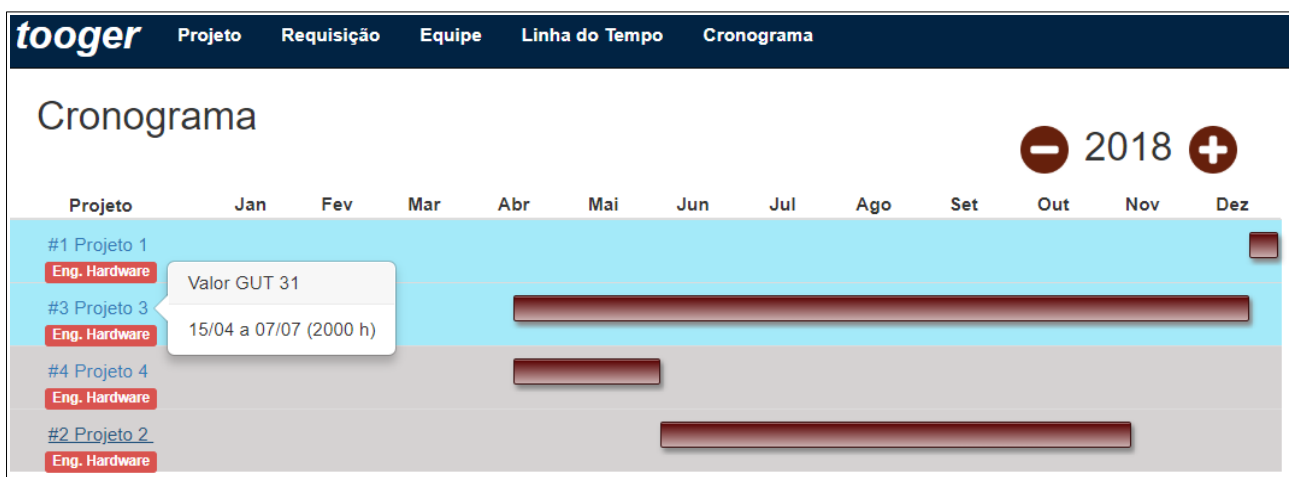
Outro recurso disponibilizado pela ferramenta é a visualização do cronograma de projetos. O modo padrão do sistema é exibir todos os projetos no período anual, organizando em linhas do tempo, priorizando os projetos que têm o valor maior correspondente a GUT, ocupando no gráfico o espaço que o projeto está planejado para ser realizado. O modo calcula automaticamente as datas do projeto, baseado na primeira data do projeto priorizado. Os demais projetos têm a data ajustada automaticamente caso o projeto esteja concorrente para a mesma equipe e mesma linha do tempo.

No momento da criação de um projeto, assim como na edição dos atributos de um projeto, o usuário pode atribuir os valores para Gravidade (G), Urgência (U) e Tendência (T) do projeto. A partir dos valores informados pelo usuário para cada um desses três atributos, o sistema calcula o

seu valor de acordo com a técnica da Matriz GUT. Isso permite que o sistema consiga priorizar os projetos que compartilham a mesma linha do tempo, demonstrando ao usuário a prioridade de cada um dos projetos através da montagem do cronograma. Através desse mecanismo, o sistema proporciona ao usuário a priorização dos projetos dentro de um portfólio de projetos, o qual pode ser representado por uma linha do tempo na ferramenta.

A Figura 6 abaixo demonstra a principal tela do sistema, cronograma de projetos, de acordo com o que foi determinado pela matriz GUT.

Figura 6 – Visualização do cronograma



Fonte: Autor.

O cálculo para atribuir os valores para a matriz GUT considera uma multiplicação dos valores informados em gravidade, urgência e tendência, que foram indicados em cada projeto. Essa multiplicação resulta em um valor único chamado no sistema como valor GUT.

O principal ganho com a ferramenta foi o tempo de geração do cronograma. O modo de geração manual é muito falho e exige muito tempo, pois é necessário calcular e posicionar os projetos devidamente. A ferramenta desenvolvida gera o cronograma rapidamente e proporciona uma visão bastante agradável, simples e objetiva.

A utilização do sistema através de seu objetivo principal, exige concluir alguns cadastros antes de registrar os projetos, requisições e tarefas. Cadastro de equipe: neste cadastro é necessário incluir uma descrição e incluir os membros da equipe. Linha do tempo: deve ser informada uma descrição e uma cor de identificação, assim o cronograma terá um destaque onde cada cor vai representar uma linha do tempo.

7 CONCLUSÃO

O objetivo deste trabalho foi documentar o desenvolvimento de uma ferramenta para gerenciamento de projetos através do uso da matriz GUT. Durante o projeto foi criada uma ferramenta destinada a otimizar a função de gerar um cronograma que prioriza os projetos a serem desenvolvidos. Essa ferramenta, de uso simples e intuitiva, mantém os dados dos projetos de forma organizada, além de conter em seu principal aspecto um algoritmo que utiliza uma técnica de matriz GUT como recurso para priorizar os projetos.

O trabalho demonstrou que gerenciar inúmeros projetos é uma tarefa complexa, pois cada projeto é tratado de forma isolada em um portfólio. O portfólio permite compartilhamento de recursos, sendo assim, determinados projetos podem influenciar os demais, ou comprometer todo o portfólio. No entanto, o uso da ferramenta traz uma perspectiva ampla de todos os projetos, e isso proporciona ao usuário uma forma organizada de gerenciá-los e priorizá-los.

O estudo também demonstrou que o uso de ferramentas de gestão de projetos, em geral, melhora a eficiência das atividades do gestor de projetos, trazendo mais facilidade, menor tempo entre atividades, economia de recursos, além da burocracia do projeto ser tratada no software, facilitando as tarefas dos integrantes da equipe.

Neste artigo, a pesquisa sobre trabalhos relacionados trouxe boas ferramentas para gestão de projetos, e também identificou alguns assuntos para futuras pesquisas. Como trabalhos futuros, o estudo pode ser aprofundado em ferramentas como tomada de decisão. Com este recurso, a ferramenta pode oferecer um melhor suporte ao gestor de projetos. Outro ponto que pode ser explorado como trabalhos futuros é o uso de técnicas de inteligência artificial como recursos extra nos métodos clássicos de priorização. A ferramenta poderia, por exemplo, relacionar projetos cadastrados com dados analíticos de tendência de mercado, priorizando assim projetos que seriam mais relevantes.

REFERÊNCIAS

ARAÚJO, Joana Filipa Alves. *Priorização de Projetos de Tecnologias e Sistemas de Informação em Portfólios*. Universidade do Minho – Departamento de Sistemas de Informação, p. 5-25. 2017.

ARTIA. *Ferramenta de Gestão de Projetos Online*. Disponível em: <<https://artia.com/ferramenta-de-gerenciamento-de-projetos/>>. Acesso em: 05 de Julho de 2018.

ASANA. *A maneira mais fácil de gerenciar projetos e tarefas de equipe*. Disponível em: <<https://asana.com/pt>>. Acesso em: 10 de Julho de 2018.

ATLASSIAN. *A ferramenta número 1 de desenvolvimento de software usada por equipes ágeis* 2018. Disponível em: <<https://br.atlassian.com/software/jira>>. Acesso em: 24 de Setembro de 2018.

BEZERRA, Eduardo. *Princípios de análise e projeto de sistema com UML*. 3 ed. São Paulo: Elsevier Brasil, 2015.

BOOTSTRAP. *Biblioteca de Componentes Front-end mais Popular do Mundo* 2018. Disponível em: <<http://getbootstrap.com/>>. Acesso em: 29 de Julho de 2018.

CASTELLO, Fladhimir; AZAMBUJA, Marcelo; MACHADO, Leonardo; NASCIMENTO, Francisco Assis do. *Método Kanban na Metodologia de Desenvolvimento de Software* 2018. Disponível em: <<https://fit.faccat.br/ead/mod/page/view.php?id=10681>>. Acesso em: 29 de Julho de 2018.

CÉSAR, Francisco I. Giocondo. *Ferramentas Gerenciais da Qualidade*. 1 ed. São Paulo: Biblioteca24horas, 2013.

CIERCO, Agliberto Alves; do VALLE, André Bittencourt; FINOCCHIO JR., José; SOARES, Carlos Alberto Pereira. *Fundamentos do gerenciamento de projetos*. 1. ed. Rio de Janeiro: Editora FGV, 2015.

CONCEIÇÃO, Ciro Mendonça da; GASSENFERTH, Walter; KRAUSE, Walther MACHADO, Maria Augusta Soares; PEREIRA, Silvia. *Gestão de Negócios e Sustentabilidade: Textos selecionados*. Rio de Janeiro: Brasport, 2015.

DAYCHOUM, Merhi. *40+20 Ferramentas e Técnicas de Gerenciamento*. 7 ed. Rio de Janeiro: Brasport, 2018.

KERZNER, Harold. *Gestão de Projetos: As Melhores Práticas*. 3 ed. São Paulo: Bookman Editora, 2016.

MARIOTTI, Flavio S. *Kanban: o ágil adaptativo* 2016. Disponível em: <<http://www.garcia.pro.br/EngenhariadeSW/artigosMA/A6%20-%2045-6-%20Kanbam.pdf>>. Acesso em: 10 de Março de 2017.

MARTINS, José Carlos Cordeiro. *Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML*. 5. ed. Rio de Janeiro: Brasport, 2010.

MICROSOFT, Corporation. *Bem-vindo ao IDE do Visual Studio* 2018. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/ide/visual-studio-ide?view=vs-2017>>. Acesso em: 24 de Setembro de 2018.

MORAES, Antonio Augusto Cerati de. ANÁLISE DA APLICAÇÃO DA GESTÃO DE PROJETOS POR CORRENTE CRÍTICA NOS PROCESSOS DE DESENVOLVIMENTO DE PRODUTOS E NA GESTÃO DE PORTFÓLIO DE UM FABRICANTE DE AERONAVES. Universidade Estadual Paulista “Júlio de Mesquita Filho” Programa de Pós-Graduação em Engenharia de Produção, p. 37-41. 2015.

ORACLE, Corporation e/ou suas afiliadas. *MySQL – Informações Gerais* 2018. Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>>. Acesso em: 24 de Setembro de 2018.

PRESSMAN, Roger S.; MAXIN, Bruce R.. *Engenharia de Software Uma Abordagem Profissional*. 8. ed. São Paulo: Amgh Editora Ltda, 2016.

RICARDI, André. *Fundamentos de gerenciamento de projetos*. 1. ed. Rio de Janeiro: Elsevier Brasil, 2013.

SANTOS, Marcos; SOUZA, Hudson Hubner; DIAS, Fabrício Costa; REIS, Marcone Freitas; SANTOS, Fernanda Mattos Carpinteiro dos. Aplicação do Método AHP na Formação de um Portfólio de Projetos: um Estudo de Caso na Área de TI de uma Empresa Sem Fins Lucrativos no Estado do Rio de Janeiro. *Revista Produção Industrial & Serviços*, v. 3, n. 1, p. 15-27. 2016.

VARGAS, Ricardo Viana. *Gerenciamento de Projetos: Estabelecendo diferenciais competitivos*. 8. ed. Rio de Janeiro: Brasport, 2016.