

Ferramenta de Auxílio ao Guia de Apoio à Escolha de Técnicas de Priorização de Requisitos (G-4REPrioritization)

Title: Support Tool for the Backing Guide to Choose Techniques for Requirements Prioritization (G-4REPrioritization)

Danilo Santana Vieira

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
dsv@ecomp.poli.br

Maria Lencastre

Escola Politécnica de Pernambuco
Universidade de Pernambuco
50.720-001 - Recife, Brasil
mlpm@ecomp.poli.br

Resumo *A demanda por softwares para a execução de tarefas e resolução de problemas diários vêm crescendo a cada dia. Cada vez mais empresas requerem o desenvolvimento de softwares, com um crescente nível de complexidade, para serem facilitadores em seus processos, assim como na tomada de decisão. Neste cenário, torna-se indispensável aos desenvolvedores de softwares a aplicação da priorização de requisitos durante o processo de desenvolvimento de software, buscando entregar o máximo valor comercial de forma mais ágil ao cliente, dentro das restrições impostas no projeto. Este trabalho implementa uma ferramenta para a plataforma web, como um auxílio à aplicação do guia, chamado G-4REPrioritization, de apoio à escolha de técnicas de priorização, que se propõe a ajudar engenheiros de requisitos a aplicarem a priorização em seus projetos.*

Palavras-Chave: *Priorização de Requisitos, Técnicas de Priorização, G-4REPrioritization*

Abstract *The demand for softwares for tasks execution and daily problems resolution is growing every day. More and more companies require the development of software, with a growing level of complexity, to be facilitators in their processes, as well as in decision making. In this scenario, it becomes indispensable for software developers to apply the requirements prioritization during the software development process, seeking to deliver the maximum commercial value in a more agile way to the client, within the constraints imposed in the project. This work implements a tool for the web platform, as an aid to the application of the guide, called G-4REPrioritization, in support of the choice of prioritization techniques, which aims to help requirements engineers apply the prioritization in their projects.*

Keywords: *Requirements Prioritization, Prioritization Techniques, G-4REPrioritization*

1 Introdução

Nos últimos tempos, as ferramentas de software vêm sendo um importante apoio a diversas atividades nas empresas; usuários de áreas distintas buscam auxílio à resolução de problemas ou suporte à realização de suas tarefas diárias, incluindo a tomada de decisão.

Por outro lado, sistemas cada vez mais complexos e o envolvimento de uma grande quantidade de requisitos tornaram-se um desafio para os desenvolvedores, que buscam a entrega de produtos que atendam, satisfatoriamente, às necessidades de seus usuários.

Um dos maiores riscos enfrentados pelas empresas que desenvolvem *software* está relacionado ao não atendimento das necessidades e expectativas dos usuários; isto pode ocasionar danos na reputação da empresa, perda de pedidos e, conseqüentemente, a redução dos seus lucros [1]. O investimento em um projeto de *software* é justificado pelas necessidades dos usuários, portanto, faz sentido ter como o foco a garantia que o produto atende às funcionalidades mais críticas e de maior valor para o usuário o mais cedo possível. A priorização de requisitos torna-se, assim, fundamental no processo de desenvolvimento de *software*.

Apesar da grande importância da priorização de requisitos, a identificação da estratégia adequada para a realização desta etapa é complexa. Geralmente, o cliente tem certa dificuldade em definir quais requisitos possuem maior importância, principalmente quando existem diversos *stakeholders* (partes interessadas), com diferentes expectativas para o projeto. Frequentemente todos consideram os requisitos como de alta prioridade, pensando que se não os considerarem assim jamais conseguirão vê-los implementados. Neste cenário, existe a necessidade da aplicação de técnicas de priorização que auxiliem na classificação dos requisitos de acordo com suas prioridades, ajudando a entregar o máximo valor comercial o mais rápido possível dentro das restrições do projeto.

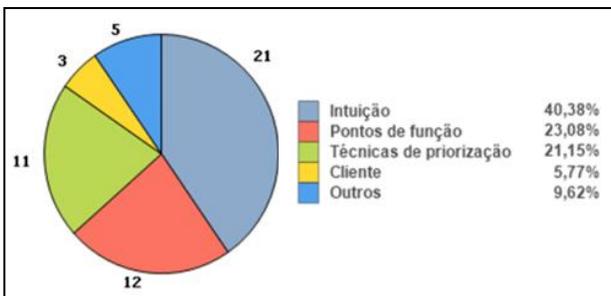


Figura 1: Forma de priorizar requisitos nas empresas do Recife.

Mesmo sendo uma etapa essencial no processo de desenvolvimento de *software*, é possível constatar que utilização de técnicas de priorização ainda é pouco comum

nas empresas de *software*. Em uma pesquisa realizada em 2015, com participação de 38 empresas na cidade do Recife [2], foi observado que aproximadamente 21% dos entrevistados aplicam alguma técnica de priorização e a grande maioria utiliza apenas a intuição para realizar a priorização de requisitos. As formas de priorizar requisitos utilizadas pelas empresas pesquisadas são exibidas no gráfico da Figura 1.

Na literatura é possível encontrar mais de 49 técnicas de priorização que podem ser aplicadas, no entanto, existem poucas orientações que facilitem a escolha da técnica que será utilizada no projeto. Para auxiliar nesta importante tarefa, existe um guia, chamado G-4REPriorization, de apoio à escolha de técnicas de priorização, montado a partir de critérios de caracterização encontrados na literatura [2]. No entanto, ainda não há uma ferramenta que facilite o estudo e a aplicação desse guia de apoio à priorização de requisitos.

O objetivo geral deste trabalho é a implementação de uma ferramenta de suporte ao G-4REPriorization, que apoie o engenheiro de requisitos na etapa de priorização de requisitos, durante o processo de desenvolvimento de *software*.

O restante do artigo está organizado da seguinte forma. Inicialmente, na seção 2, é apresentada uma visão geral da priorização e o guia a ser implementado – o G-4REpriorization; em seguida, na seção 3, faz-se o detalhamento da ferramenta propriamente dita. Por fim, na seção 4, são apresentadas as considerações finais e trabalhos futuros.

2 Fundamentação Teórica

O processo de desenvolvimento de um *software* abrange um conjunto de fases e atividades que têm como principal objetivo a entrega de um sistema que supra a necessidade do cliente, dentro de orçamento e prazos anteriormente definidos. Em um cenário onde a crescente competitividade leva empresas a adotar metodologias de desenvolvimento cada vez mais ágeis para garantir a vantagem competitiva e aumentar a satisfação de clientes, a priorização de requisitos se mostra como uma atividade de extrema importância no processo de desenvolvimento de sistemas.

Nesta seção são detalhados os principais conceitos da priorização de requisitos, o processo preparatório para a priorização de requisitos, e quatro técnicas de priorização implementadas na ferramenta de suporte ao guia.

2.1 Visão Geral da Priorização

Os requisitos são a base do trabalho de desenvolvimento de *software*, pois são propriedades necessárias com a qual o sistema deve estar de acordo para que ele

tenha valor e utilidade para um cliente. Se o conjunto incorreto dos requisitos for implementado e os usuários resistirem ao uso do produto, não importa o quão sólido o produto é ou quão completamente ele foi testado [3]. A maioria dos projetos de *software* têm mais requisitos do que pode ser realizado dentro das restrições de tempo e custo, sendo a priorização uma importante ajuda na identificação dos requisitos mais importantes deste conjunto.

Young [4] relata que nem todos os requisitos são de igual importância para os clientes e usuários do sistema planejado, sendo importante a priorização de todos os requisitos devido a falta de tempo ou dinheiro suficientes para implementar tudo o que deseja-se ter no sistema desenvolvido. De acordo com o autor, identificar os requisitos reais requer um processo de requisitos interativo e iterativo, apoiado por práticas, processos, mecanismos, métodos, técnicas e ferramentas eficazes. Ruhe [5] simplifica essa afirmação da seguinte forma: “O desafio é selecionar os requisitos ‘certos’ de um determinado superconjunto dos requisitos candidatos para que os diferentes interesses fundamentais, restrições técnicas e preferências dos atores críticos sejam cumpridos e o valor comercial global do produto seja maximizado”.

Diversos critérios podem ser levados em conta quando da priorização de requisitos, entre eles tem-se:

- **Importância:** Avalia a importância do desenvolvimento do requisito, levando em consideração aspectos como urgência ou necessidade estratégica do requisito para a empresa.
- **Custo:** Avalia o investimento financeiro necessário em termos de orçamento para desenvolver o requisito. Depende muito da complexidade do requisito, a quantidade de documentação necessária e possibilidade de reúso de código.
- **Duração:** Avalia a duração, o prazo esperado para a implementação do requisito, considerando diversos aspectos, como outras atividades sendo executadas em paralelo.
- **Risco:** Avalia e considera os riscos envolvidos no projeto na realização do requisito, como, por exemplo, o risco de não atender a necessidade do cliente, ou a baixa performance do *software* após o desenvolvimento do requisito.

O processo de engenharia de requisitos, por sua vez, contempla a execução de 4 atividades base [6]:

- **Elicitação:** Atividade onde é realizado o levantamento dos requisitos. Identificação das necessidades dos usuários e os objetivos que o sistema que será desen-

volvido deve atingir. Pode-se aplicar diversas técnicas, como, por exemplo, entrevistas com usuários importantes, aplicação de questionários, prototipagem, entre outras.

- **Documentação:** Ocorre a especificação dos requisitos, onde os requisitos levantados são descritos e documentados. Pode-se usar modelos conceituais para a representação dos requisitos.
- **Negociação:** Nesta atividade ocorre a resolução dos conflitos que podem existir, como, por exemplo, necessidades de usuários conflitantes, ou uma quantidade de requisitos superior ao que se pode desenvolver com os recursos limitados. A negociação e resolução de conflitos ajuda a garantir que os critérios de qualidade definidos sejam atingidos.
- **Gerenciamento:** Os requisitos do sistema estão em constante evolução, sendo necessário o controle de mudanças, mantendo os requisitos estruturados, sendo um processo contínuo durante todo o projeto.

A importância da priorização pode ser observada em cada uma das cinco atividades [2]:

- **Elicitação:** as prioridades podem determinar, por exemplo, quais requisitos devem ser elaborados primeiro ou quais as fontes de requisitos (por exemplo, *stakeholders*) devem ser considerados primeiro;
- **Documentação:** a priorização pode ser usada, por exemplo, para determinar uma ordenação em que os requisitos devem ser documentados para satisfazer a regras de documentação e formatos pré-definidos;
- **Negociação:** requisitos conflituosos podem ser priorizados, por exemplo, no que se refere à sua influência sobre o êxito do projeto. A priorização pode ser utilizada inicialmente para resolver os conflitos para os requisitos mais importantes;
- **Validação:** a priorização pode ser utilizada, por exemplo, para determinar a ordem em que: os requisitos devem ser validados, para determinar diferentes formas de validação, ou para definir a ordem pela qual os defeitos detectados devem ser resolvidos;
- **Gerenciamento:** a priorização pode ser utilizada, por exemplo, para definir as solicitações de mudanças que devem ser processadas em primeiro lugar, de modo que as mais urgentes serão atendidas em primeiro lugar.

2.2 Processo de Preparação para a Priorização de Requisitos

Para realizar a priorização de requisitos no processo do desenvolvimento de *software*, é importante a execução de 4 atividades preparatórias:

- Selecionar os *Stakeholders*: identificar grupo de pessoas que têm interesse/impacto na realização do projeto. Os *stakeholders* são pessoas que possuem o conhecimento dos processos da empresa e o contexto no qual o sistema será utilizado. A boa seleção dos *stakeholders* é importante para garantir o retorno de resultados corretos no processo de priorização.
- Selecionar os Requisitos: Escolher os requisitos que participarão do processo de priorização. É importante a escolha de requisitos com o mesmo nível de abstração, sendo relevante iniciar a priorização pelos requisitos com nível mais alto de abstração, para em seguida priorizar os níveis mais baixos.
- Definir o(s) critério(s) da priorização: A escolha do critério a ser considerado quando da aplicação de uma técnica de priorização é fundamental. Dependendo da técnica pode ser necessária a escolha de um único critério ou de múltiplos critérios.
- Selecionar a Técnica de Priorização: A escolha da técnica depende de vários fatos, como os critérios escolhidos para a priorização, número de requisitos que serão priorizados, quais *stakeholders* serão envolvidos no processo. Nesta atividade a técnica mais adequada é escolhida para a execução da priorização.

2.3 Técnicas de Priorização de Requisitos

Nesta subseção são detalhadas quatro das principais técnicas de priorização de requisitos, que estão entre as mais usadas [2]. São elas: *Hundred Dolar* [8], MoSCoW [9], Matriz de Wieggers [9] e *Analytic Hierarchy Process* [10].

2.3.1 Hundred Dolar (\$100)

Esta técnica propõe dinamizar a priorização de requisitos juntamente com os *stakeholders*, mensurando a prioridade dos requisitos através da atribuição de um valor financeiro. Cada *stakeholder* participante do processo recebe a quantia imaginária de 100 dólares para ponderar e alocar valores nos requisitos candidatos de acordo com a prioridade de cada requisito. Os requisitos que o *stakeholder* acredita ter maior prioridade receberiam, assim, um maior valor que os requisitos de menor prioridade. Terminada a distribuição dos 100 dólares

pelos *stakeholders* entre todos os requisitos candidatos, soma-se os valores atribuídos aos requisitos para saber a prioridade dos requisitos, para então montar um *ranking* dos requisitos por nível de importância.

2.3.2 MoSCoW

Nesta técnica de priorização os *stakeholders* definem o grau de importância sobre a entrega de cada requisito. Embora muitos dos requisitos sejam considerados importantes, é essencial destacar aqueles que agregam maior valor ao sistema, permitindo focar o trabalho de forma mais eficiente nos requisitos de maior prioridade. A técnica MoSCoW requer a classificação dos requisitos em uma das quatro categorias:

- *Must*: Requisitos imprescindíveis para o escopo do projeto. Críticos para a geração de valor para a empresa, ou seja, de alta prioridade, cujo o desenvolvimento é necessário para se obter êxito.
- *Should*: Requisitos classificados como importantes e que devem ser considerados ao máximo quanto a inclusão na solução final, mas que podem não ser necessários no momento e o não desenvolvimento não impactaria negativamente no sucesso do projeto.
- *Could*: Requisitos desejáveis, mas não necessários, podendo ser atendidos apenas se os recursos disponíveis permitirem.
- *Want*: Requisitos desejados pelos *stakeholders*, mas que não serão desenvolvidos no momento, podendo ser considerados para uma versão futura.

2.3.3 Matriz de Wieggers

Esta técnica propõe uma forma sistemática de analisar os requisitos considerando quatro critérios: benefício, penalidade, custo e risco. Primeiramente são ponderados os valores de peso para os critérios estabelecidos; no padrão inicial todos os critérios têm o mesmo peso. Após serem definidos os requisitos participantes da priorização, é realizada a estimativa dos valores de benefícios e penalidades relativas utilizando a escala de valores de 1 a 9, para em seguida realizar o cálculo dos valores totais, demonstrado na Equação 1, além dos valores percentuais para cada requisito.

$$\text{Valor Total} = (\text{Benefício} \times \text{PesoBenefício}) + (\text{Penalidade} \times \text{PesoPenalidade})$$

Equação 1: Cálculo do valor total considerando benefício e penalidade dos requisitos.

Logo após, são estimados os valores de custo relativo e risco relativo, sendo calculados os valores percentuais para, por fim, ser aplicado o cálculo das prioridades para

os valores da tabela referentes a cada requisito, como é demonstrado na Equação 2.

$$Prioridade = \frac{Valor\%}{(Custo\% \times PesoCusto) + (Risco\% \times PesoRisco)}$$

Equação 2: Calculo da prioridade do requisito.

A estimativa é baseada em várias variáveis retornando resultados informativos e claros. No entanto, os resultados dependem da capacidade das pessoas de estimar o valor, custo e risco.

2.3.4 Analytic Hierarchy Process (AHP)

É uma técnica criada para auxiliar a tomada de decisões complexas, baseando-se na matemática e psicologia, e foi adaptada para a realização da priorização de requisitos. A técnica consiste em realizar a comparação de todos os possíveis pares de requisitos hierarquicamente organizados, com o objetivo de poder determinar o nível de prioridade de cada requisito, utilizando uma escala de 1 a 9, onde o 1 representa que o requisito tem igual importância e o 9 que é mais importante.

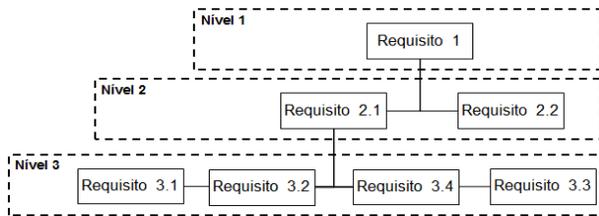


Figura 2: Requisitos organizados em níveis.

Inicialmente os requisitos são colocados em níveis hierárquicos, dispostos em uma árvore, de acordo com critérios normalmente selecionados pelos *stakeholders*. A Figura 2 apresenta um exemplo dos requisitos organizados em níveis, no formato requerido pela AHP.

Após a separação dos níveis, são selecionados os requisitos do mesmo nível hierárquico que serão priorizados, sendo organizados em uma tabela para a realização da comparação entre os pares, dispostos em linhas e colunas, através da escala de pontuação com importâncias classificadas pelos *stakeholders*, onde os valores das colunas são somados, exibindo os valores totais. Em seguida é realizada a normalização da matriz, realizando a divisão de cada valor da tabela pelo valor total da coluna no qual está contido. No último passo, é realizado o cálculo da prioridade, obtendo o valor percentual através da média aritmética dos valores de cada linha, referentes aos requisitos.

2.4 G-4REPrioritization

O Guia de Priorização de Requisitos (G-4REPrioritization) [2] é um roteiro de apoio à escolha de técnicas de priorização. Através da análise do perfil do escopo do projeto e critérios definidos, o objetivo do G-4REPrioritization é o de fornecer orientação ao engenheiro de requisitos quanto à seleção da técnica de priorização de requisitos mais apropriada para o escopo. O guia considera a realização de um conjunto de passos após a concretização da elicitação e negociação dos requisitos.

A metodologia aplicada para o desenvolvimento do guia consistiu na seleção de 13 técnicas dentre as 49 técnicas de priorização encontradas na literatura, além da extração de 9 características da literatura para caracterizar técnicas de priorização, entre elas complexidade, critérios e popularidade da técnica. Em seguida, foi realizada a caracterização de cada uma das 13 técnicas, de acordo com as características definidas e, após a categorização, foram definidos filtros que podem ser preenchidos opcionalmente no escopo do projeto, baseados nas características gerais das técnicas consideradas. Por fim, foi realizada a estruturação do processo do guia. A metodologia de criação do G-4REPrioritization é exibida na Figura 3.

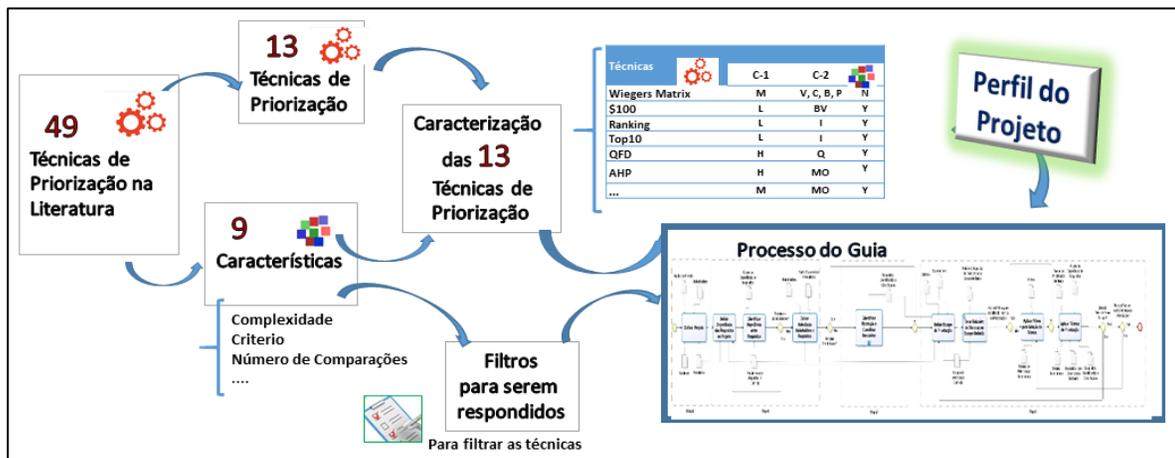


Figura 3: Metodologia Usada para Desenvolver o Guia.

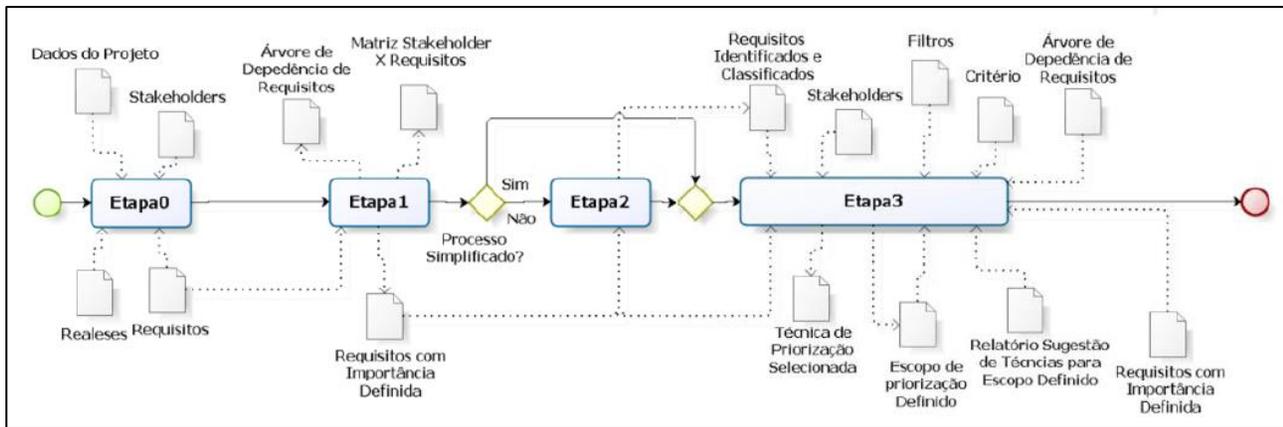


Figura 4: Modelo exibindo as etapas do Guia, bem como as suas entradas.

As entradas para o G-4REPrioritization incluem um conjunto de informações sobre o projeto que será analisado. São elas:

- Detalhamento do projeto: Nome, requisitos e *stakeholders*.
- Detalhamento dos requisitos: Dependência entre requisitos, abstração e classificação dos requisitos, relevância entre *stakeholders* e requisitos.
- Definição do escopo: Critérios que serão utilizados, requisitos e *stakeholders* que participarão do processo de priorização, além da seleção de filtros para auxílio na escolha de técnicas.

Conforme apresentado na Figura 4, o G-4REPrioritization propõe 4 etapas principais para a escolha da técnica de priorização, descritas a seguir.

Etapa0: Definir Projeto

Nesta etapa deve-se considerar a definição de um projeto de *software* que está em desenvolvimento, para o qual já estão definidos os *stakeholders* e uma lista de requisitos, levantados e negociados, que ainda não foram priorizados.

Etapa1: Definir importância, Dependência entre Requisitos e a sua relevância para os *stakeholders*

Nesta etapa é feita a: Classificação da importância dos requisitos, podendo ser obrigatório, interessante e opcional; Identificação dos requisitos que possuem dependência com outros requisitos; Definição das importâncias dos *stakeholders* para os requisitos do projeto.

Etapa2: Identificar Abstração e Classificar Requisitos

Esta etapa é opcional, nela é realizada a identificação do nível de abstração dos requisitos, a classificação entre

funcionais e não funcionais, assim como a identificação de requisitos não legais.

Etapa3: Definir Escopo de Priorização e Aplicar Técnica de Priorização

Esta etapa é dividida em 4 atividades:

E3.1 Definir escopo de Priorização: Deve-se escolher o escopo a ser priorizado, tendo como entrada os critérios, os *stakeholders* e os requisitos que serão priorizados.

E3.2 Gerar Relatório de Técnicas: Atividade automática de geração de um relatório demonstrando o comportamento das 13 técnicas no escopo definido.

E3.3 Aplicar Filtros para Seleção da Técnica: O guia inclui diretrizes para a análise de 13 técnicas de priorização. Através de perguntas, as técnicas são pontuadas, para, após a realização do questionário, retornar a técnica mais adequada.

E3.4 Aplicar Técnica de Priorização: Inicia o processo de priorização utilizando a técnica escolhida, retornando os requisitos priorizados.

3 Ferramenta G-4REPrioritization

Para apoiar a utilização do G-4REPrioritization, foi implementada uma ferramenta para auxiliar o engenheiro de requisitos que necessita aplicar um processo de priorização de requisitos em seu projeto. O sistema contempla a realização dos passos descritos no guia, assim como realiza a comunicação com API (*Application Programming Interface*) externa, através de requisições utilizando o formato JSON (*JavaScript Object Notation*), que realiza a execução de 4 diferentes técnicas (\$100, MoSCoW, AHP e Matriz de Wieggers) que retornam os requisitos do escopo priorizados [7].

A ferramenta foi desenvolvida para a plataforma *web*, utilizando a linguagem de programação PHP, por se tratar

de uma linguagem com um modelo de desenvolvimento simples e de fácil portabilidade e manutenção, sendo bastante difundida e indicada para aplicações *web*. Para auxiliar o desenvolvimento com PHP, foi escolhido o *framework Codeigniter*, por se tratar de um poderoso *framework* criado para o auxílio de desenvolvedores que necessitam de um conjunto de ferramentas simples para criar aplicações para *web*, permitindo o desenvolvimento de projetos com mais agilidade do que seria possível realizar ao escrever o código a partir do zero, fornecendo um rico conjunto de bibliotecas para tarefas que normalmente são necessárias. A utilização do *framework* permite uma melhor padronização do código, possibilita a reutilização de código, além de ser um ótimo apoio para a manipulação de bancos de dados, agilizando, assim, o desenvolvimento do sistema.

A utilização do *framework Codeigniter* possibilita a utilização do padrão de arquitetura MVC (*Model-view-controller*), no qual a aplicação é dividida em três camadas: *Model*, *View* e *Controller*. A camada *Model* é a parte da aplicação onde é implementada a lógica do negócio, assim como a comunicação e manipulação dos dados. A *View* é a camada onde é implementada a interface de apresentação, onde os dados são exibidos para o usuário. A camada *Controller* faz o gerenciamento das requisições entre as camadas *View* e *Model*, isolando a lógica da aplicação da camada de interface do usuário.

O modelo de arquitetura utilizada no projeto pode ser visto na Figura 5, que demonstra a requisição de um usuário à classe do *controller*, responsável por fazer a comunicação com a camada *model*, além de retornar a *view* para exibição no monitor do usuário.

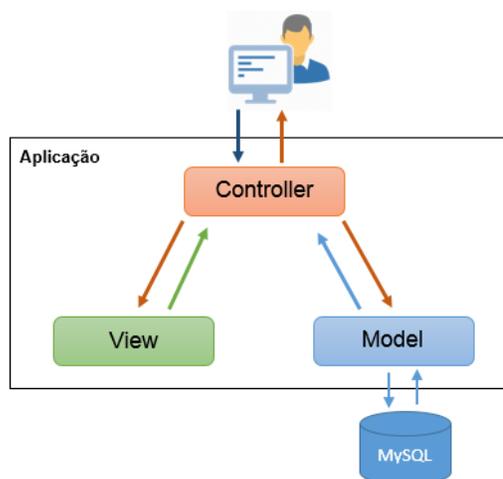


Figura 5: Padrão de arquitetura MVC na aplicação.

No desenvolvimento da ferramenta, pacotes são utilizados para a separação das camadas. No pacote de *controllers* estão contidas todas as classes de controle, as quais o usuário tem acesso às funcionalidades do sistema

através de requisição a cada controlador específico. O Codeigniter fornece um sistema prático no qual os URL das aplicações são roteados a controladores e métodos desses controladores. O framework faz com que o primeiro elemento após o endereço do domínio da aplicação seja a classe controladora. Em seguida ao controlador, o próximo elemento identificado é o método que está dentro do controlador como pode ser visto no URL a seguir: <http://dominioaplicacao.com/controlador/metodo>.

No pacote de *models* são encontradas as classes de modelo para cada entidade do sistema, onde estão contidas as funções que tornam possíveis a recuperação, inserção e atualização das informações no banco de dados. No terceiro pacote de *views* estão contidos sub-pacotes divididos por funcionalidade, onde estão contidos todos os arquivos de telas que são exibidas aos usuários do sistema.

O banco de dados escolhido foi o MySQL, por ser um banco de dados de fácil integração com o PHP, que apresenta um bom desempenho e estabilidade, fácil manipulação, além de ser de livre utilização e de grande aceitação em diversos servidores.

Para auxiliar a parte visual do sistema foi selecionado o *framework Bootstrap*, que torna o desenvolvimento *front-end web* mais fácil, pois inclui padrões de *design* prontos, além de adaptar o projeto *web* para ser bem visualizado em diversos dispositivos móveis, tais como celulares e *tablets*.

3.1 Funcionalidades

A partir das definições do guia para priorização de requisitos, para realizar todas as etapas, as principais funcionalidades da ferramenta foram divididas em 3 grupos dentro do sistema:

- **Projetos:** Cadastro e manutenção do projeto e suas características. Dentro da visão do projeto pode-se realizar o cadastro e manutenção dos requisitos relacionados ao projeto, assim como a associação de *stakeholders*.
- **Stakeholders:** Cadastro e manutenção de um repositório de *stakeholders* que são criados independentes dos projetos, podendo ser associado a diversos projetos.
- **Escopos:** Cadastro, definição e manutenção do escopo para a priorização, envolvendo um projeto, selecionando requisitos e *stakeholders* que participarão da priorização. Dentro da área de escopos são preenchidas as entradas dos *stakeholders* para a técnica de priorização selecionada, assim como o envio das entradas para a aplicação da técnica escolhida.

A Figura 6 apresenta as principais funcionalidades do sistema através do diagrama de caso de uso, sendo representado como ator o engenheiro de requisitos que pode manter projetos, *stakeholders* e escopos.

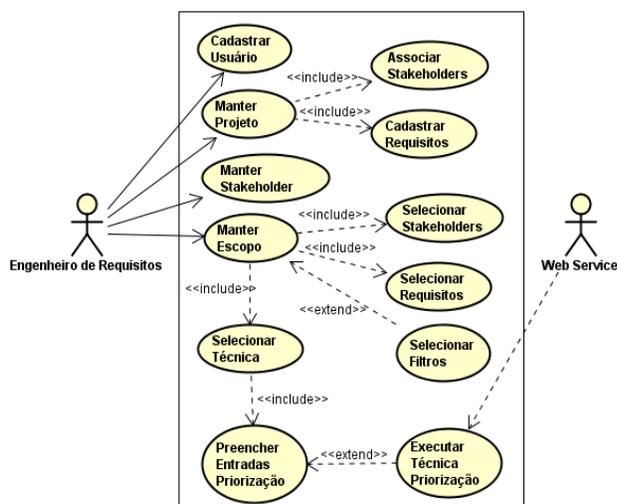


Figura 6: Diagrama de caso de uso.

3.1.1 Projetos

O sistema permite o cadastro e a manutenção dos projetos. Ao acessar a área referente a Projetos, o usuário pode consultar projetos já criados para visualização ou edição, assim como incluir um novo projeto, ver Figura 7. Ao selecionar a opção de incluir um novo projeto, o usuário é direcionado para um formulário onde é pedido o nome do projeto que deseja cadastrar.

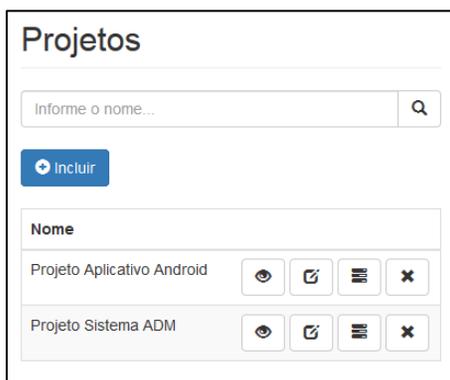


Figura 7: Primeira tela da área de Projetos.

Após a criação do projeto, o usuário é direcionado para a área de manutenção, onde o usuário pode realizar a associação de *stakeholders*, selecionando um previamente cadastrado no sistema e informando o peso do *stakeholder* para o projeto, ver Figura 8. Os *stakeholders* associados ao projeto são, então, exibidos em uma lista.

Também na tela de manutenção do projeto, é exibida a opção de cadastrar novos requisitos, através do botão

Novo Requisito exibido na Figura 8.

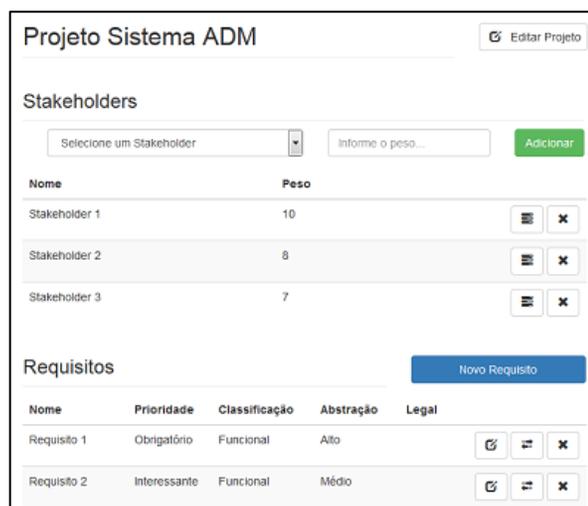


Figura 8: Tela de manutenção de Projetos.

No cadastro de um novo requisito é pedido obrigatoriamente o nome do requisito, ver Figura 9. Neste passo as informações sobre prioridade, classificação, nível de abstração e se o requisito é legal, são de preenchimento opcional, ou seja, o usuário poderá preencher essas informações num momento seguinte do processo de preparação para a priorização dos requisitos.



Figura 9: Tela de inclusão de novo requisito.

Com os requisitos já incluídos ao projeto, o sistema permite também a identificação da dependência entre os requisitos. Por questão de espaço essa tela não foi aqui incluída.

3.1.2 Stakeholders

Na área de *Stakeholders*, o usuário pode manter um repositório de pessoas importantes para os projetos. O *stakeholder* é cadastrado sendo informado seu nome e função na estrutura organizacional na empresa. Caso a função desejada não seja encontrada na lista disponível, é possível realizar a inclusão de um novo tipo de função.

Após o cadastro, os *stakeholders* podem ser associados a um ou mais projetos. A Figura 10 exibe a tela principal da área de manutenção de *stakeholders*.

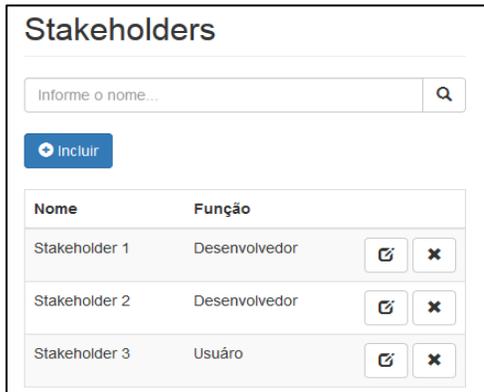


Figura 10: Tela de manutenção de *Stakeholders*.

3.1.3 Escopos

Com a criação de projetos, o usuário pode, na área de Escopos, realizar o cadastro de um escopo para um determinado projeto. Na Figura 11 é exibida a tela principal da área de Escopos, onde é possível consultar um escopo pelo nome para visualização e manutenção, assim como realizar a criação de um novo escopo, informando o projeto que terá seus requisitos priorizados. A Figura 12 exibe a tela de cadastro de escopo.

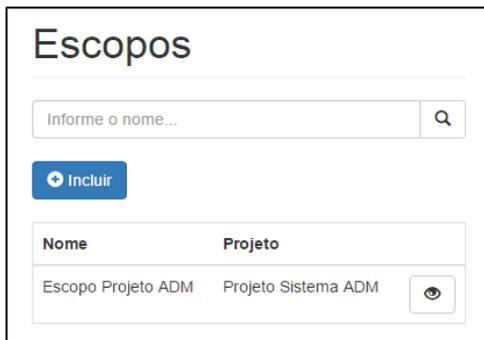


Figura 11: Primeira tela da área de Escopos.



Figura 12: Tela de inclusão de Escopo.

Após a criação do escopo, o usuário é direcionado pa-

ra a tela de definição do escopo, que é exibida na Figura 13. Nesta tela o usuário pode, então, ter acesso à quatro funcionalidades relacionadas ao escopo, separadas em abas, sendo elas, a seleção dos requisitos do projeto que serão priorizados, a seleção dos *stakeholders* que participarão do processo de priorização, a seleção de filtros para o auxílio de escolha de técnicas e, por fim, o preenchimento das entradas relacionadas à técnica escolhida e execução da priorização.

Na aba Requisitos, serão exibidos todos os requisitos do projeto, sendo obrigatoriamente filtrados pelo nível de abstração, de acordo com a definição do guia de que os requisitos priorizados num escopo não devem ser de níveis de abstração diferentes. Outros filtros como prioridade, classificação e requisitos legais são de preenchimento opcional. O usuário pode selecionar os requisitos que participarão da priorização e salvar a sua escolha.



Figura 13: Tela de definição de escopo na aba de seleção de requisitos.

Na aba seguinte, serão exibidos todos os *stakeholders* que foram associados ao projeto, para a seleção dos que participarão do processo de priorização do escopo. Esta tela é demonstrada na Figura 14.



Figura 14: Definição de escopo na aba de seleção de *stakeholders*.

O guia implementado, fornece um conjunto de filtros com o objetivo de auxiliar o engenheiro de requisitos na escolha da melhor técnica para o escopo definido, ver Figura 15.

Técnica	F1	F2	F3	F4	F5	F6	F7
\$100	✓	✓	✓	✓	✓	✓	✓

Figura 15: Aba de filtros para escolha de técnicas.

Além de escolher os critérios que irá utilizar na priorização, o usuário tem acesso a utilização de filtros opcionais para informar o nível de complexidade esperado para a técnica, se pretende utilizar uma técnica popular e se deseja aplica uma técnica automática. Também poderá informar se deseja considerar repriorização de requisitos, influência dos *stakeholders*, assim como a dependência entre os requisitos, no processo de priorização. O preenchimento e aplicação do formulário de filtros, trará como retorno, a sugestão de uma ou mais técnicas que se adequam ao escopo definido.

Tendo a tabela de retorno com as técnicas mais adequadas de acordo com os filtros selecionados, o usuário pode escolher uma das técnicas retornadas para a execução, clicando no botão Selecionar Técnica, sendo direcionado à aba de Priorização, onde deve preencher os campos de entrada para a aplicação da técnica.

Após a seleção dos requisitos que serão priorizados, dos *stakeholders* que participarão da priorização e a escolha da técnica que será aplicada, a aba da Priorização será habilitada e será exibido o formulário de entradas para a técnica que será executada. A Figura 16 mostra, como exemplo, o formulário referente a técnica *Hundred Dolar* (\$100), onde são exibidos os *stakeholders* participantes da priorização, bem como os campos onde são informa-

dos os valores que cada *stakeholders* alocou nos requisitos participantes. Após o preenchimento das entradas de todos os *stakeholders*, o botão Executar Priorização é habilitado, permitindo que a técnica seja executada através do envio dos valores das entradas dos *stakeholders* através do formato JSON, que possui um padrão definido para cada técnica, para a API externa que aplica a técnica selecionada e retorna uma lista no formato JSON com os requisitos ordenados de forma crescente com base na prioridade [7]. O resultado da aplicação da técnica é então exibido como um relatório, que também pode ser acessado e impresso através do botão Relatório de Priorização, que pode ser visto na Figura 16.

Figura 16: Preenchimento das entradas e execução da priorização.

3.2 Modelo de Dados

O modelo conceitual de dados entidade-relacionamento, usado como base no desenvolvimento da ferramenta de apoio ao G-4REPrioritization é demonstrado na Figura 17.

O diagrama descreve as entidades envolvidas no sistema, identificando suas características e como elas se relacionam. A entidade Usuário tem, em suas características, os atributos id, nome, e-mail e senha, sendo o id a chave primária desta entidade. A relação da entidade Usuário com as entidades Projeto, *Stakeholder* e Escopo é de 1 para n, ou seja, 1 usuário pode fazer o registro de n Projetos, *Stakeholders* e Escopos. A entidade Escopo pode estar relacionada a um único objeto da entidade *Tecnica_priorizacao*, através da seleção de sua chave primária. O *Stakeholder* deve possuir uma função, através do relacionamento com a entidade *Funcao*, assim como pode estar associado a Projetos e Requisitos, além de ser selecionado em um Escopo. A entidade *Priorizacao* registra a execução de uma determinada técnica de priorização a um escopo definido, armazenando a data da execução.

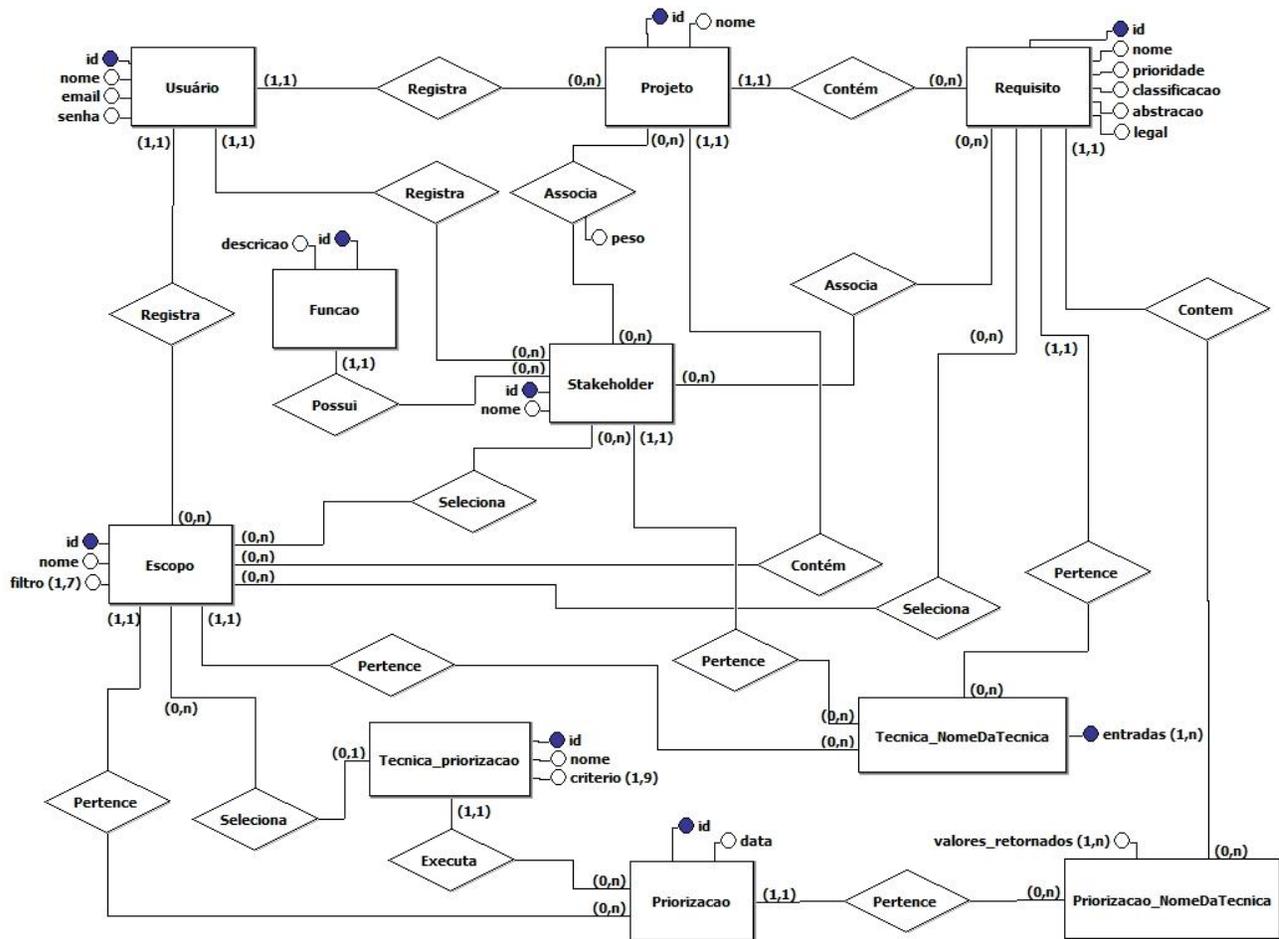


Figura 17: Modelo conceitual Entidade-Relacionamento da ferramenta.

Dois entidades genéricas são exibidas no modelo conceitual, sendo elas *Tecnica_NomeDaTecnica* e *Priorizacao_NomeDaTecnica*. Para cada técnica implementada no sistema, foi criada uma entidade para guardar as entradas dos *stakeholders* para cada requisito pertencente a um escopo. No exemplo da técnica MoSCoW, a entidade criada foi chamada *Tecnica_moscow*. A segunda entidade genérica, exibida no modelo como *Priorizacao_NomeDaTecnica*, representa as entidades criadas para cada técnica implementada, com o objetivo de guardar os valores retornados, para cada requisito, após a execução da técnica de priorização. Também segundo o exemplo da técnica MoSCoW, a entidade que armazena os resultados foi chamada *Priorizacao_moscow*.

4 Conclusões e Trabalhos Futuros

O objetivo deste trabalho foi desenvolver uma ferramenta *web* de auxílio à aplicação do guia de apoio à escolha de técnicas de priorização de requisitos (G-4REPrioritization), fornecendo uma aplicação de fácil acesso e utilização para usuários que tenham interesse em

utilizar o guia para fins acadêmicos ou profissionais.

A ferramenta funciona como um estímulo para uma maior utilização e importância da etapa de priorização de requisitos no processo de desenvolvimento de *softwares*, resultando em produtos que atendam cada vez mais as necessidades e expectativas dos clientes.

Como trabalhos futuros estão: a validação da ferramenta através de experimentos; a implementação de mais técnicas de priorização; a aplicação do sistema para a utilização do guia em casos reais em empresas ou projetos acadêmicos, avaliando a usabilidade e coletando sugestões de melhorias, bem como possibilitar a seleção e execução das demais técnicas de priorização de requisitos selecionadas pelo G-4REPrioritization. Além disso, mostra-se relevante, o desenvolvimento de uma ferramenta integrada para a interação dos stakeholders participantes do escopo, na aplicação das técnicas de priorização.

Atualmente a ferramenta está disponível para utilização através do endereço www.g-4reprioritization.pe.hu. No desenvolvimento do projeto não foi utilizado um

sistema para controle de versões na nuvem, sendo realizado em um servidor local privado. No entanto, há planejamento de distribuição livre do código da ferramenta em uma plataforma de hospedagem com controle de versão e colaboração online.

Referências

- [1] J. Karlsson, K. Ryan, Supporting the selection of Software Requirements, 8th International Workshop on Software Specification and Design (IWSSD '96), páginas p. 146-149, 1996.
- [2] J. M. S. Júnior, G-4REPrioritization: Um Guia para Apoio à Escolha de Técnicas de Priorização de Requisitos, Dissertação de Mestrado, Universidade de Pernambuco, Outubro, 2015.
- [3] P. Berander, Prioritization of Stakeholder Needs in Software Engineering - Understanding and Evaluation, Licentiate Series No. 2004:12, Department of Systems and Software Engineering, Blekinge Institute of Technology, 2004.
- [4] R. R. Young, The Requirements Engineering Handbook, Artech House, Massachusetts, 2004.
- [5] Ruhe, G., Eberlein, A., e Pfahl, D.: 'Quantitative WinWin - A New Method for Decision Support in Requirements Negotiation', Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02), páginas 159-166, 2002.
- [6] K. Pohl, Requirements Engineering: Fundamentals, Principles and Technologies, Springer, 2010.
- [7] D. B. O. Diniz, Ferramenta de Suporte à execução de Técnicas de Priorização de Requisitos, Trabalho de Conclusão de Curso, Universidade de Pernambuco, Junho, 2017.
- [8] A. M. Davis, Just Enough Requirements Management: Where Software Development Meets Marketing, Dorset House Publishing, 2005
- [9] K. Wiegers, J. Beatty, Software Requirements, 3 ed., Redmond, Washington, 2013.
- [10] A. Aurum, C. Wohlin, Engineering and Managing Software Requirements, 3 ed., Springer, Berlin, 2005.

Autorização de publicação de PFC

Eu, **DANILO SANTANA VIEIRA** autor do projeto de final de curso intitulado:
**FERRAMENTA DE AUXÍLIO AO GUIA DE APOIO À ESCOLHA DE TÉCNICAS DE
PRIORIZAÇÃO DE REQUISITOS**

(G-4REPRIORITIZATION); autorizo a publicação de seu conteúdo na internet nos
portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.



DANILO SANTANA VIEIRA



Orientador(a) Maria Lencastre Pinheiro de Menezes Cruz

coorientador(a)



Professor de TCC Sérgio Campello Oliveira

17/07/2017
Data

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 7 de Julho de 2017, às 9:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **DANILO SANTANA VIEIRA**, orientado pelo professor **Maria Lencastre Pinheiro de Menezes Cruz**, sob título **FERRAMENTA DE AUXÍLIO AO GUIA DE APOIO À ESCOLHA DE TÉCNICAS DE PRIORIZAÇÃO DE REQUISITOS**

(G-4REPRIORITIZATION), a banca composta pelos professores:

Joabe Bezerra Jesus Junior

Maria Lencastre Pinheiro de Menezes Cruz

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 8,5 (oito e meio)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

JOABE BEZERRA JESUS JUNIOR

MARIA LENCASTRE PINHEIRO DE MENEZES CRUZ