

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

**CONCEPÇÃO DE LEVANTAMENTO DE REQUISITOS
REFERENTE AO MÓDULO-05 DE GESTÃO DE ALERTAS DO
SISTEMA INDRA**

Aluno: Grazielle De Souza Teixeira

Prontuário: 1320122

São João da Boa Vista – SP

2016

Resumo

Com o intuito de criar um sistema de alertas de enchentes e outros eventos climáticos que possam causar danos e como forma de concluir o Ensino Médio Integrado em Informática, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus São João da Boa Vista, teve-se a ideia da construção do Sistema Indra, por intermédio do professor ministrante da disciplina de Prática de Desenvolvimento de *Software* e com auxílio dos alunos no quarto ano do curso citado. Portanto, serão mostradas as metodologias utilizadas para a confecção do mesmo, bem como a fundamentação teórica que explica o porquê da escolha de sua aplicação, isto como forma de mostrar os levantamentos das principais funcionalidades do sistema focando no módulo-05 de gestão e emissão de alertas.

Sumário

1	Introdução	4
2	Desenvolvimento	7
2.1	Engenharia de <i>Software</i>	7
2.2	Extreme Programming	7
2.3	Etapas dos Desenvolvimento de <i>Software</i>	9
2.4	Levantamentos de Requisitos	10
2.5	Técnicas Para Levantamento de Dados	11
2.5.1	Entrevistas	12
2.5.2	Questionário	12
2.5.3	<i>Workshops</i> com <i>Brainstorm</i>	13
2.5.4	Etnografia	13
2.5.5	Prototipação	13
2.5.6	Revisão de Documentação	13
2.6	Método de Análise Kano	13
2.7	Documento de requisitos	15
3	Metodologia	16
3.1	Módulo 05 – Alertas	16
3.2	Documento de requisitos do módulo de gestão de alertas	17
3.2.1	Requisitos funcionais	18
3.2.2	Priorização dos Requisitos Funcionais	21
3.2.3	Requisitos não funcionais	25
3.2.4	Questionário de Perguntas aos <i>Stakeholders</i>	27
4	Conclusões e Recomendações	28
5	Referências Bibliográficas	29

1 Introdução

Decorridas as chuvas de janeiro de 2016, os eventuais estragos nas cidades do interior paulista, Águas da Prata e São João da Boa Vista, como demonstra a imagem abaixo, e as enchentes que são recorrentes nesta região; por intermédio do professor Breno Lisi Romano, houve a ideia da confecção de um sistema, o qual pudesse auxiliar os moradores que são corriqueiramente afetados, emitindo alertas de futuros eventos climáticos que pudessem comprometer a segurança da população.



Figura1: Rua de São João da Boa Vista (SP), completamente alagada. [1]

Uma vez que a concepção estava feita, os alunos do Instituto Federal De Educação, Ciência e Tecnologia De São Paulo, Campus São João da Boa Vista, cursando o 4º ano do Ensino Médio integrado em informática, ficaram incumbidos de realizar tal proposta, isto, dentro da disciplina de Prática e Desenvolvimento de Sistemas (PDS), que possui o intuito de compilar o conhecimento adquirido durante os quatro anos de ensino em um único sistema computacional, como forma de

angariar notas bimestrais e indiretamente, gerar um dispositivo de segurança para a sociedade como um todo, este ultimo sendo especifico ao sistema que será tratado a seguir.

O sistema, denominado Indra, (no hinduísmo, Deus das tempestades), ‘trata-se de um portal para controlar e visualizar informações de estações meteorológicas’ [2] e emitir alertas, sempre que as informações das Plataformas de Coletas de Dados (PCDs) forem críticas.

Para o desenvolvimento do mesmo, estão dispostos trinta e três (33) alunos, cada qual com uma função, dentre três as possíveis: Analista de sistemas, responsável por analisar, documentar, projetar e testar as aplicações; *software* (em português: administrador do banco de dados), responsável pelo gerenciamento, planejamento, monitoração e instalação do banco de dados; e o Desenvolvedor, o qual faz a parte física do sistema, usando as linguagens específicas, tornam realidade o portal. Os alunos com os papéis escolhidos são divididos em módulos, cada um contendo, dois analistas, dois DBAs e de dois a três desenvolvedores, assim formando cinco módulos distintos.

Desse modo, o Módulo 01- Usuários, como o próprio título diz, é responsável pelo gerenciamento dos Usuários, Perfis, e Recepção dos mesmos. O Módulo 02 – PCDS se responsabiliza pela gestão das PCDs, as estações metrológicas. O Módulo 03 – Medições, é o módulo o qual irá administrar os dados coletados pelo Módulo 02, assim, o Módulo 04 – Relatórios, poderá gerar gráficos, tabelas e relatórios com os dados obtidos pelos outros módulos, e por fim, o Módulo 05- Alertas, que emitirá os dados que foram identificados como fora das medições normais.

Cada um dos módulos possui suas próprias tarefas pré-determinadas e definidas, que são muitas, além das citadas anteriormente, unidas elas dão rumo a um único objetivo, alertar os usuários do sistema, desse modo, o Módulo 05 - Alertas, o qual será o foco desse trabalho, é uma peça mais que fundamental para a fabricação do Indra. Gerenciar, exibir, e enviar por emails os alertas, são algumas das funções do módulo, que estão sendo mostradas de uma forma simplificada na figura a seguir, assim como exibir tabelas e gráficos dos dados dos alertas, mostrar na página inicial do usuário o alerta e dentro de gerenciar; permitir a leitura, a atualização, a criação e a exclusão de alertas que os usuários logados ao sistema poderão emitir, auxiliando na precisão do sistema.

O foco do módulo são os alertas e alguns exemplos que serão emitidos pelo portal, são os alertas de: queimadas, poluição, baixa umidade relativa do ar, qualidade d’água, de enchentes, tempestades, riscos de enchentes e elevação anormal do nível d’água, sendo esses últimos os mais relevantes para a região de enfoque do Indra.

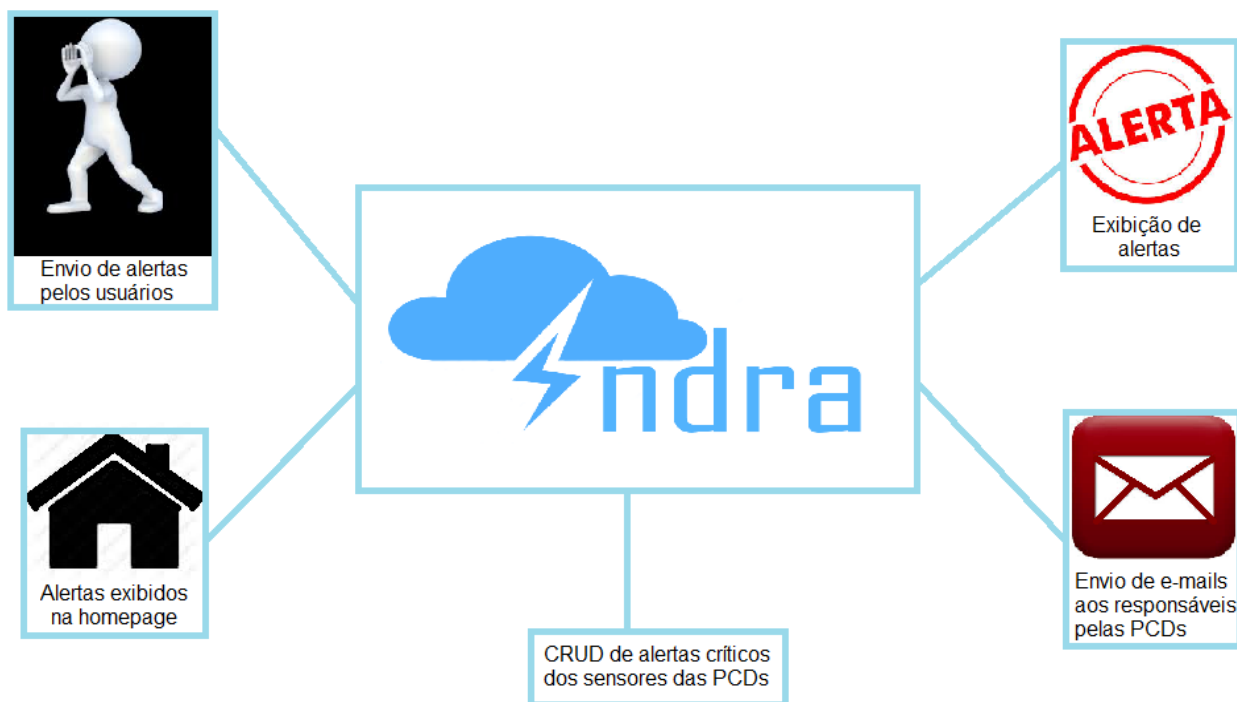


Figura 2: Tarefas de competência do Módulo 05 – Alertas.

De acordo com as tarefas, seria necessário um longo e contínuo processo para a plena realização do sistema, assim, alguns documentos facilitariam o sucesso do módulo, foram criados então: o Documento de Visão, Documento de Levantamento de Requisitos, Documento de Casos de Uso e Documento Casos de Testes. Todos com suas devidas importâncias e funcionalidades, mostrando-se um dos mais necessários o documento de Levantamento de Requisitos, cujo será explanado nessas páginas.

Para um completo entendimento do que se trata o levantamento de requisitos, é essencial ter em mente o conceito de requisitos, que diz respeito à “uma especificação de uma característica ou propriedade que um sistema deve possuir ou fazer, assim como sua restrição de operação” [3], logo o levantamento de requisitos é a concepção de todas as funcionalidades do sistema que suprirão a necessidade do usuário. Os requisitos podem ser funcionais, os quais especificarão o que o sistema será capaz de realizar e os requisitos não funcionais que possuem indicadores qualitativos das funções técnicas do sistema. Assim, o Documento de Levantamento de Requisitos, de uma forma geral deve possuir todas as funcionalidades, bem explicadas, que serão necessárias conter no sistema, também com base nele é que são realizados outros documentos, como o Documento de Casos de Uso e o Documento de Casos de Testes, bem como as aplicabilidades dos protótipos que serão desenvolvidos ao longo do projeto e que darão forma ao Indra.

2 Desenvolvimento

O documento de requisitos é o documento o qual possui uma descrição detalhada de todos os requisitos, funcionais e não funcionais, assim como suas prioridades de acordo com o método de análise Kano, que será tratado mais adiante, bem como o questionário de perguntas feitas aos stakeholders, em português “as partes interessadas”. Por ser o documento base para a construção de outros dois documentos que darão forma e validação ao projeto, o levantamento de requisitos se mostra um tanto significativo para se ter um alcance fiel do intuito do *software* e satisfazer as idealizações descritas no documento de visão que possui as características de alto nível do projeto.

Tal documento, não poder ser confeccionado aleatoriamente, é preciso um estudo prévio, respeitando as fases de desenvolvimento de *software* descritas e formalizadas pela engenharia de sistemas.

2.1 Engenharia de *Software*

Segundo Friedrich Ludwig Bauer, um cientista da computação alemão, idealizador do conceito diz que, "Engenharia de *Software* é o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter *software* de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais" [4], em outras palavras, engenharia de *software* se trata da abrangência do processo de desenvolvimento, manutenção e criação de sistemas, utilizando-se de tecnologias e práticas, aplicadas por meio de um conhecimento específico para se dirigir ao objetivo determinado, permeado de organização, garantido qualidade, agilidade e economia, esta última sendo tanto em trabalho quanto em termos financeiros.

O desencadeamento desse processo envolve recursos computacionais, como por exemplo, linguagens de programação, banco de dados, bibliotecas, plataformas e ferramentas sempre priorizando a qualidade do projeto.

2.2 Extreme Programming

Desenvolvida em 1997, o *Extreme Programming* ou *XP*, conquistou um método de desenvolvimento de *software*, que procura fundamentar as práticas em um conjunto de valores. Essa metodologia é um modo ágil, que não requer profissionais extremamente experientes. É melhor aplicado em pequenas e médias organizações de desenvolvimento, no entanto, a utilização dela em grandes equipes não se mostra desprezível.

Tal método é facilmente ajustável em projetos onde os requisitos não foram bem planejados e requer mudanças constantes, tanto nos requisitos, tanto no projeto como um todo, dado isso é

adotada a estratégia de acompanhamento constante juntamente com a alteração de detalhes que precisam ser melhorados em documentações.

Como a prática é baseada em valores, eis suas explicações e suas influências no desenvolvimento do *software*.

Comunicação, esse valor deixa clara a necessidade de sempre haver a comunicação, entre os clientes com os desenvolvedores e entre eles mesmos, dessa forma há uma real contribuição dos usuários com a formulação do sistema e onde não há a troca de informações o *software* não é bem efetuado. Um tipo de comunicação entre os programadores se dá por meio da programação em pares (*pair programming*), onde a feitura de dá em um único computador, revezado o teclado, ambos juntos, e trabalhando na mesma aplicação, com troca de ideias e se comunicando constantemente, como um trabalho em equipe. A realização de dinâmicas em grupo, rodas de conversa e até mesmo O Jogo do planejamento (*planning poker*) auxilia na comunicação, com esse jogo, acontece a discussão técnica entre as partes que trabalham em prol do projeto. Desse modo, segundo Beck “Os problemas nos projetos invariavelmente recaem sobre alguém não falando com alguém sobre algo importante” [5].

O valor da simplicidade, diz para se tentar produzir primeiramente o simples, com qualidade, com aplicações úteis, necessárias e que funcionem, deixando a complexidade para as próximas etapas do todo, claro se essas não forem de extrema importância. O *feedback*, se mostrando parte da comunicação, é realizada por meio de reuniões e é essencial. Permitindo uma evolução constante do *software* possibilitada pelos testes feitos com a utilização das páginas, é uma forma rápida de recebimento de retornos.

Por falta de iniciativa, ou medo do inesperado, somos presos ao conformismo, com o XP, temos que tomar coragem, sendo outro valor da técnica. Essa nos guia ao paradigma de que é necessário explicar tudo o que está acontecendo ao cliente, bem como acreditar em si próprio, aprender com erros, saber criticar e receber críticas tomando as como algo concreto digno de cuidado e atenção, realizar o que foi planejado e apagar tudo o que não for agregar.

Um valor fundamental na aplicação do *Extreme Programming* é o *Coach* (traduzido do inglês: treinador, técnico, instrutor), é a pessoa que faz com que os valores previamente citados trabalhem e funcionem no dia a dia da organização. Este seria o mais experiente do grupo, o monitor que coordena e garante o trabalho.

2.3 Etapas dos Desenvolvimento de *Software*

Como forma de auxiliar no desenvolvimento de um projeto, existem metodologias organizacionais com etapas bem definidas, que facilitam e padronizam o processo, determinando quem irá fazer o que, quando e como. Sendo as principais etapas a de planejamento, construção, implantação, avaliação e manutenção. Fazendo parte dessas etapas, existem pontos importantes como o levantamento de requisitos, que requer uma boa documentação, para se possuir um acompanhamento do andamento das realizações, principalmente se tratando de um projeto em encargo de uma grande equipe, com diversos profissionais em diferentes funções, esse é um cuidado que foi levado em consideração na idealização do Sistema Indra.

Logo na primeira etapa, a de planejamento, é que encontramos o levantamento dos requisitos bem como a identificação do cliente, levantamento de recursos e custos, avaliação de viabilidade do desenvolvimento e da implantação, contrato com os compromissos explícitos (Termo de abertura) e a elaboração do cronograma inicial. Nesse cronograma devem estar contidas as etapas que serão efetuadas e suas datas de entrega, assim como os recursos utilizamos para a finalização do projeto juntamente com a data de entrega do produto funcionando com a qualidade inicialmente prometida. É também nesse período que é confeccionado o documento de visão e são definidos os macrorrequisitos que serão amadurecidos e darão fundamentação para os requisitos mais específicos, permitindo a existência do documento de casos de uso, que possui descritas todas as funcionalidades que o sistema deve executar, sempre que um cliente clicar em um botão, por exemplo, deve se esperar uma resposta do sistema, e são essas respostas que estarão descritas. Essa etapa de planejamento, também pode ser conhecida por design de *software*.

Na fase de construção é que são realizados os protótipos da aplicação. Utilizando-se as linguagens de programação definidas na primeira etapa são feitos os códigos-fontes, baseados nas descrições presentes na documentação de casos de uso, a segurança do sistema também é considerada. Não menos importante, também há a confecção do banco de dados com todos os seus esquemas e devidas etapas para uma perfeita construção, viabilizando a inserção de dados, por meio das aplicações.

Durante esse período, simultâneos ao desenvolvimento ocorrem os testes nas páginas do sistema, estes são feitos pelos analistas, que previamente desenvolveram o documento de casos de teste, que possui todos os casos de uso mais uma vez explanados com as respostas que supostamente deverão ser recebidas ao se utilizar a aplicação. Esses testes são para verificar se ocorre um correto funcionamento do portal de um modo geral, considerando que possa haver eventuais falhas e sucessos estes são documentados, para que, se ocorrem, os erros possam ser

corrigidos pelo programador, os quais passarão novamente por testes, até que todas as funções estejam agindo como o esperado para que o portal possa ser utilizado.

Os estágios de implantação do sistema junto aos clientes, com treinamentos de funcionários e algumas correções detectadas apenas na utilização, bem como a fase de avaliação do portal pelo usuário e manutenções não fazem jus ao nosso projeto, no entanto, são fases de extrema importância no ciclo de vida de um *software*.

2.4 Levantamentos de Requisitos

Os requisitos são definidos durante a fase de concepção, para se adquirir uma visão geral do sistema, entender o que o cliente deseja, ou o que o cliente acredita que necessita e as regras do negócio, que irão definir ou restringir um aspecto do funcionamento da empresa, esses pontos são essenciais na idealização de um projeto.

Desse modo, ter um levantamento de requisitos bem estruturado e específico se mostra notório no momento de confecção de um sistema bem-sucedido e na fabricação do que o usuário realmente necessita, caso o contrário se torne real, haverá um problema de execução, decorrente da falha de comunicação no levantamento dos requisitos. Muitas das vezes o futuro usuário principal do sistema não sabe com exatidão o que deseja que o sistema realize ou até mesmo sabe, no entanto, não consegue transmitir para o analista, desse modo, os requisitos identificados, devem ser descritos o mais detalhado possível, de modo franco e objetivo, para que todas as partes que estiverem envolvidas no desenvolvimento da aplicação entendam de forma completa, viabilizando a implementação que se mostra útil, de acordo com técnicas específicas, evitando assim, falhas na construção do sistema e eventuais desperdícios de funcionalidades, e para que isso não ocorra, são usadas as técnicas de levantamento de requisitos, que serão demonstradas nas próximas páginas desse trabalho.

A análise dos requisitos é outra etapa que se faz importante após o levantamento, nessa fase, é que o engenheiro de requisitos, o analista e o desenvolvedor de *software*, identificam as necessidades de acordo com os requisitos de um cliente, interpretando com seu olhar profissional os pedidos. Pode ser concretizada uma análise equivocada dos atributos, gerando na hora de entrega um descontentamento por parte do cliente por não ter sido aquele o produto almejado. Essa falha pode se dar também pela quantidade de pessoas envolvidas, com visões discrepantes de uma mesma parcela de um todo e formas distintas de solucionar algo abstrato.

As possíveis falhas na descrição, entendimento e planejamento dos requisitos podem ser exemplificadas na imagem a seguir.

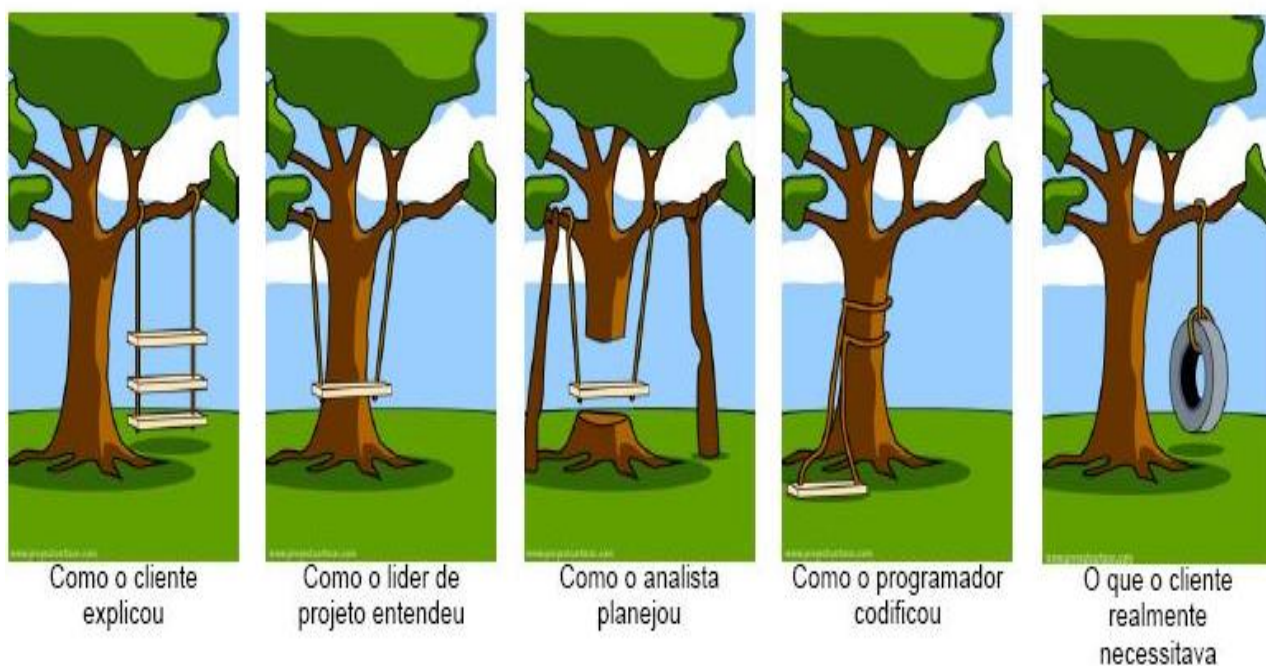


Figura 3: Imagem ilustrativa decorrente falha na comunicação do levantamento de requisitos. [6]

2.5 Técnicas Para Levantamento de Dados

Levando em conta o elevado grau de complexidade e seu devido valor na totalidade do projeto, para se atingir as expectativas tanto na idealização do cliente, quanto nas etapas de desenvolvimento e finalização são utilizadas as técnicas de levantamento de dados que após serem interpretados darão forma aos requisitos do sistema. Logo abaixo se encontra uma tabela com uma breve explicação de cada técnica encontrada, as quais também serão discutidas uma a uma.

Técnicas tradicionais	São aplicadas em várias áreas do conhecimento. Exemplo: questionários, entrevistas, observação, e análise de documentos.
Técnicas de elicitação de grupo	Tem por objetivo compreender melhor o pensamento e comportamento dos grupos e as necessidades dos usuários. Exemplo: <i>brainstorming</i> .

Prototipação	<p>O uso de protótipo auxilia na elicitação e validação dos requisitos de sistema.</p> <p>A prototipação pode ser utilizada para elicitar requisitos quando há um alto grau de incerteza ou quando é necessário um rápido <i>feedback</i> dos usuários.</p>
Técnicas contextuais	<p>Surgiram como uma alternativa para as técnicas tradicionais e cognitivas e inclui técnicas de etnografia e análise social.</p>

Tabela 1. Grupos de técnicas para levantamento de requisitos. [7]

2.5.1 Entrevistas

Essa técnica faz jus á uma entrevista propriamente dita, assim como em uma entrevista de emprego, para saber mais sobre o funcionário, no entanto, nesse caso há a possibilidade de realizar uma reunião entre os que serão os usuários do sistema, ou com o portador do problema, que pretende por intermédio do *software* solucioná-lo, e o analista. São realizadas, aos envolvidos como forma de adquirir informações que fundamentarão as fases de concepção, perguntas objetivas que possuem a intenção de ir direto ao foco e conseguir o maior número de informações sobre as especificidades em um tempo apropriado.

2.5.2 Questionário

O método é realizado para uma pesquisa ou estudo, direcionando o futuro levantamento de requisitos. Existe uma série de perguntas, que podem ser de múltipla escolha, lista de opções ou com respostas livres, escritas de forma clara, objetiva e de fácil entendimento com um foco direto na resposta e nas necessidades dos usuários que os analistas procuram entender. Esse modo possui a vantagem de alcançar um maior público, permitindo a criação de planilhas, gráficos e outros, com os dados angariados, sendo desvantajoso na interpretação e na explanação do mesmo, porque do analista requer uma minuciosidade em tentar entender as respostas, e do usuário que respondeu as perguntas, requer uma tentativa de escrever suas explicações da melhor maneira possível.

2.5.3 Workshops com Brainstorm

Diferente da entrevista, os *Workshops* consistem em reuniões com inúmeras pessoas que são estimuladas a todo o momento a darem ideias de funcionalidades, técnica conhecida por *Brainstorm* (Tempestade de Ideias), ocorrendo uma interação entre todos os elementos presentes com momentos de descontração como forma de dinamizar o trabalho em equipe e ampliar a gama de possibilidades com a interpretação e análise de diferentes pontos de vista.

2.5.4 Etnografia

Funcionando como uma imersão do analista dentro da organização, essa técnica permite a apreensão da cultura e objetivos organizacionais, descobrindo requisitos do sistema dentro da área onde as pessoas trabalham e que realmente vão fazer utilização da aplicação, esses requisitos muitas das vezes não são percebidos com as técnicas previamente expostas, uma vez que o estudo etnográfico é uma análise de componente social e da interação do funcionário com o problema.

2.5.5 Prototipação

Essa etapa diz respeito a uma primeira interação do usuário com a aplicação, quando não ocorre uma boa comunicação entre os stakeholders e o analista, essa técnica entra em ação. São confeccionadas algumas páginas, protótipos, do sistema, para que dessa forma o cliente possa fazer uma utilização de algumas funcionalidades, isto, como forma de validação dos requisitos previamente coletados, servindo também como testes de desempenho e auxiliando na compreensão dos mesmos.

2.5.6 Revisão de Documentação

Outra forma de coletar requisitos é fazendo uma verificação nas documentações previamente escritas. Analisando todos os documentos, contrato inicial com o cliente, por exemplo, pode haver novas percepções, uma análise diferente das especificações e captação de novos requisitos que estão em segundo plano.

2.6 Método de Análise Kano

Para atingir a qualidade necessária, após o levantamento, os requisitos passam por uma análise que faz jus ao modelo de Kano. Que se trata de uma metodologia que capta os requisitos e os classificam em torno de categorias, de acordo com os conceitos de qualidade ao cliente, fornecendo um esquema simples de classificação que distingue os atributos essenciais de acordo com a satisfação e a expectativa dos consumidores.

O modelo classifica os requisitos como unidimensionais, atrativos, obrigatórios, indiferentes, reversos e questionáveis. Os classificados como sendo atrativos, são quando se os atributos forem totalmente realizados, geram uma grande satisfação, no entanto se há uma realização parcial dos mesmos não causam um desagrado generalizado; os unidimensionais geram satisfação quando presentes e em contrapartida quando não estão presentes geram insatisfação; os atributos ditos como necessários, são essenciais ao meio, e quando ausentes produzem grande insatisfação, no mais, sua presença se mostra neutra.

Os indiferentes, o próprio nome caracteriza os atributos que não afetam nem na satisfação muito menos na insatisfação das partes interessadas; e os atributos reversos sem mostram insatisfatórios pelo simples fato de estarem tanto simplificados quanto em demasia. E existem os obrigatórios que são fundamentais para a complexidade trabalhada.

O modelo de Kano nos auxilia a alcançar na medida certa a realização dos requisitos sob a visão crítica do cliente, definindo mais claramente os atributos, bem como o escopo do projeto com o melhor possível em qualidade, desempenho e eficiência. Na figura, temos o eixo da percepção do cliente (*Satisfied, Dissatisfied*) e o eixo da medida em que o atributo deve ser satisfeito (*Need not fulfilled, Need well fulfilled*), bem como as retas de expectativa dos clientes (*Excitement, Indifference, Performance, Basic*).

Breve tradução dos termos presentes na figura:

- *Satisfied* – Satisfeito;
- *Dissatisfied* – Insatisfeito;
- *Need not fulfilled* – Não Precisa ser cumprido completamente;
- *Need well fulfilled* - Precisa ser cumprido completamente;
- *Excitement* – Excitante (Classificação: Atrativo);
- *Indifference* – Indiferente (Classificação: Indiferente);
- *Performance* – Atuantes (Classificação: Obrigatório);
- *Basic* – Básico (Classificação: Necessário).



Figura 4: Gráfico correspondente ao método de análise de Kano [8]

2.7 Documento de requisitos

Levantados os requisitos, se mostra necessário uma documentação dos mesmos como forma de haver um registro. Por parte dos desenvolvedores, o documento serve de base para uma compreensão e planejamento do sistema, bem como para o acompanhamento no processo de desenvolvimento para que todas as funcionalidades sejam confeccionadas com sucesso, sem ser deixada de lado nenhuma. Para os usuário e clientes se faz uso para uma avaliação e consequentes atualizações e alterações. Os analistas e testadores, utilizam a documentação para desenvolver outras etapas do ciclo de vida do *software*, os casos de uso e os casos de teste, assim como os documentos que os registram são desenvolvidos de acordo com o documento de requisitos.

Esse documento deve possuir uma descrição do intuito do sistema, com uma descrição de cada requisito funcional levantado e também dos não funcionais, todos descritos com uma linguagem que permite o fácil entendimento. Encontra-se nesse documento, a priorização dos requisitos de acordo com sua viabilidade e classificação, conforme o modelo Kano de qualidade e um breve questionário realizado aos *stakeholders*.

3 Metodologia

Explanados o longo processo que se deve ser seguido ao realizar a confecção de um *software*, tem-se base como foi o desenvolvimento do Sistema Indra, considerando com mais afinco, o documento de requisitos e todos os seus componentes, analisando métodos de levantamento dos mesmos, bem como a priorização deles. Foi respeitada a maioria das etapas de acordo com o que se era possível aplicar ao ambiente corporativo criado dentro dos laboratórios de informática do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus São João da Boa Vista.

3.1 Módulo 05 – Alertas

Como dito anteriormente, o Projeto Indra, é composto por cinco módulos distintos, cada qual com seus objetivos e macrorrequisitos, referenciados no início desse documento e na imagem a seguir. Dentro do portal, cada módulo foi tido como uma ferramenta que auxilia na construção de todas as partes do sistema, que o usuário tem acesso e os utiliza.

O módulo 05, assim como os outros, além possuir seus próprios objetivos, também possui metas a serem atingidas e funcionalidades do sistema a serem colocas em prática. Tais atividades, só são possíveis com o trabalho dos indivíduos que compõem o subgrupo - alertas, totalizando seis, dois analistas, dois *SOFTWAREs* e dois desenvolvedores.

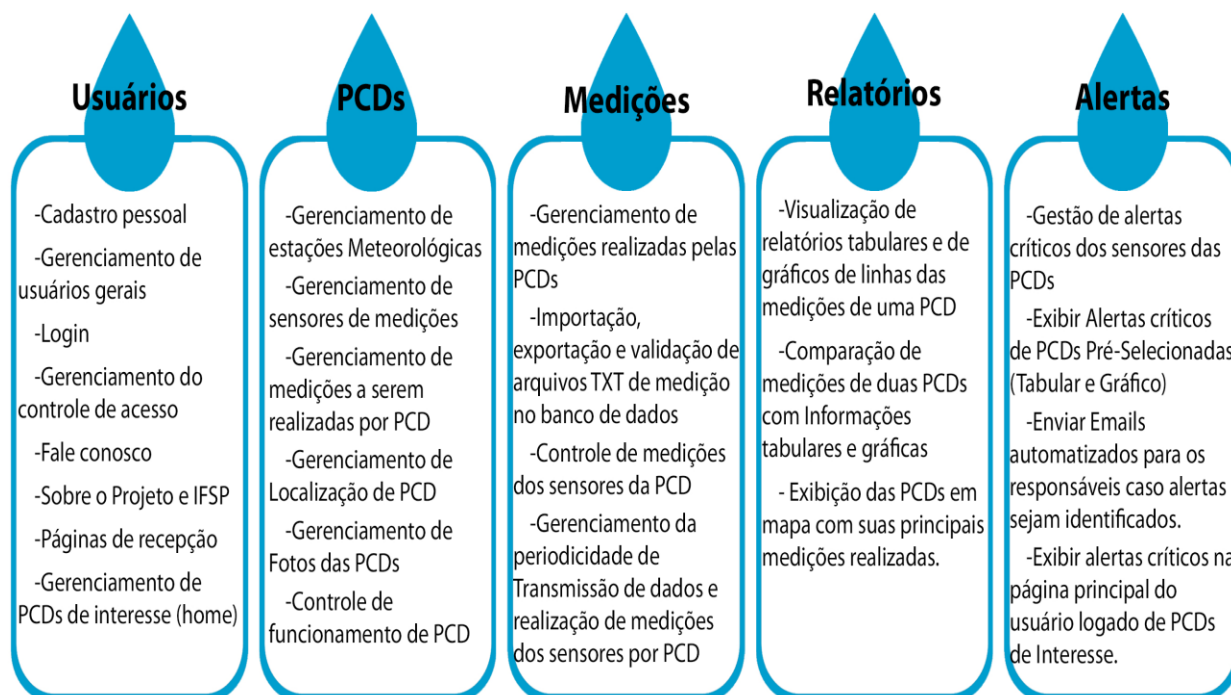


Figura 5: Macrorrequisitos do Sistema Indra (modificado) [9]

Por se tratar de um projeto educacional, não possui um cliente o qual pôde assinar o contrato inicial com os alunos e com o professor, e muito menos auxiliar no levantamento das funcionalidades requeridas, no entanto o pontapé inicial dos requisitos foi sugerido pelo orientador da disciplina. Estes serviram de base para que os estudantes juntos discutissem o que um sistema de alertas de eventos climáticos poderia e deveria possuir para cumprir sua função de nada mais que emitir o alerta propriamente dito.

Baseando-se nos macrorrequisitos e com a ideia da função do sistema, os analistas em sua maior parte, juntamente com os *stakeholders* e os desenvolvedores, levantaram o que seria útil o projeto possuir. Inicialmente foram propostos quatro amplos requisitos, que seriam esmiuçados pelos alunos, e estes são:

- Exibir alertas críticos de PCDs pré-selecionadas pelos usuários logados no sistema, com disposição dos dados destacados no alerta e também de uma forma mais completa em gráficos e/ou tabelas.
- Realizar a gestão dos alertas críticos de PCDs pré-selecionadas pelos usuários logados, mantendo o destaque dos dados obtidos tanto no alerta em si, quanto em gráficos e/ou tabelas. Essa gestão se trata das quatro operações básicas utilizadas em bases de dados, que são criar, consultar, atualizar e excluir dados, em inglês, pegando-se as primeiras letras de cada palavra, tem-se a sigla *CRUD* – *Create, Read, Update e Delete*.
- Enviar alertas aos responsáveis pelas PCDs por meio de e-mails automáticos sempre que os dados coletados forem de risco.
- Exibir na página inicial do usuário logado, quando houver a necessidade, alertas referentes às medições de limite inseridas por ele de acordo com as PCDs de interesse do mesmo.

Dado esses requisitos base, os mesmo foram sendo adaptados e amplificados, bem como outros foram sendo desenvolvidos, para tornar o módulo um pouco mais complexo e aprimorado. As versões modificadas e detalhadas serão expostas nas próximas páginas desse mesmo documento.

3.2 Documento de requisitos do módulo de gestão de alertas

O referido documento deveria em primeira instância, ser editado pela dupla de analista do grupo de gestão de alertas, todavia, como os DBAs estavam sem tarefas específicas, os mesmos foram fundamentais para dar forma ao que seria o fomento dos protótipos e futuras aplicações da quinta ferramenta do sistema.

Assim como todo documento de requisitos, esse não sendo diferente, deve possuir tanto os requisitos funcionais, quanto os não funcionais. A priorização dos funcionais e as perguntas realizadas aos stakeholders.

3.2.1 Requisitos funcionais

Os requisitos funcionais devem ser escritos contento a maior quantidade de detalhes possíveis, uma vez que os primeiros protótipos do sistema seriam desenvolvidos com fundamento nessa documentação. Desse modo alguns pontos específicos, como por exemplo, dados cadastrais das PCDs, tipos de alertas, nomes dos eixos dos gráficos, dados que seriam enviados nos e-mails, previa do preenchimento dos alertas enviados pelos usuários, entre outros, se mostrou necessários uma completa descrição.

Com intuito de possibilitar um entendimento dos requisitos descritos logo a baixo, se mostra necessário uma breve explicação dos usuários que terão contato tanto com a manipulação, quanto com apenas a observação dos dados. Existem quatro tipos de usuários com acesso á aplicação como um todo, o Visitante, que possui acesso apenas á página principal, com a possibilidade da realização de um cadastro e visualização das páginas com informações sobre a Instituição - IFSP e o sobre projeto Indra; Usuário cadastrado, que pode acessar páginas referentes aos alertas emitidos, bem como ter acesso os gráficos e tabelas com as informações de uma PCD e dos alertas, página de comparação entre informações das PCDs, fale conosco e também tem permissão para a criação de comunicados sobre eventos climáticos anormais.

Usuário especialista, indivíduo previamente cadastrado e com aplicação de acesso de hidrólogo/meteorologista (ou instituições meteorológicas), esse dispõe dos mesmos privilégios do usuário cadastrado, com acesso também á possibilidade de envio de resultados de estudos sobre as épocas de maiores ocorrências de alertas, assim como estudos sobre os locais mais afetados, e; Usuários administrativos, que possuem acesso ao banco de dados de comunicados e relatórios, podem bloquear e desbloquear alertas de um usuário, banir e autorizar uso de um usuário ao Indra possuem acesso aos dados de PCDs, podendo realizar CRUD nesses dados e autorizam os usuários serem usuários administrativos. Com acesso a ferramenta de Alertas é apenas três, sendo excluído o visitante que tem acesso limitado apenas nas páginas iniciais do sistema.

Requisitos funcionais do módulo - 05 de gestão de alertas:

Identificador	Descrição do Requisito
RF #01	Esse sistema contém o gerenciamento de alertas, feitos por meio de um usuário administrativo que irá checá-los e validá-los


	<p>como sendo verdadeiros ou falsos, também é de função da administração permitir a criação, leitura, atualização e exclusão destes. Esses alertas críticos são referentes a eventuais chuvas, tempestades, ventanias, alteração do nível dos rios, alagamentos, de acordo com os dados climáticos, tais como, umidade, sensação térmica, pressão, direção dos ventos e outros, que serão coletados pelos sensores das PCDs. O módulo-05 deverá ser capaz de permitir o gerenciamento de alertas das PCDs. Para criação de um alerta, o usuário deverá definir: pcd desejada, um sensor da pcd escolhida, valor mínimo, valor máximo, padrão de cores valor mínimo e padrão de cores valor máximo. A atualização dos alertas permite a alteração de todos os campos definidos em sua criação. A exclusão de alertas críticos é permitida ao usuário que o criou e ao usuário administrador, e uma mensagem de confirmação deve ser exibida ao usuário.</p>
RF #02	<p>Para um alerta mais preciso, é necessário que contenha dados de uma PCD específica, com características necessárias, sendo essas de localização, de sensores pré-selecionados de acordo com o potencial de risco, com informações que possuem uma maior utilidade, como por exemplo: valor máximo e mínimo do nível do rio de um usuário dos alertas em uma comunidade ribeirinha, para que determinados usuários consigam um alerta com maior efetividade de segurança atrelado ao motivo inicial da criação do sistema.</p>
RF #03	<p>Das informações coletadas das PCDs, selecionadas e cadastradas, o sistema ficará responsável pela exibição de alertas críticos, com a emissão de dados relevantes que serão dispostos em formato de gráficos sendo o eixo X horário e o eixo Y nível do alerta desejado, com pesquisa de datas de início e término e legenda.</p>
RF #04	<p>Das informações coletadas das PCDs, selecionadas e cadastradas, o sistema ficará responsável pela exibição de alertas críticos, com a emissão de dados relevantes que serão dispostos em formato de tabelas com dados tais como: pcd desejada, um sensor da PCD escolhida, valor mínimo, valor máximo, padrão de cores valor mínimo e padrão de cores valor máximo, datas de início e término.</p>
RF #05	<p>Caso haja a identificação de sinais críticos, o sistema enviara</p>

	<p>e-mails automáticos, contendo os seguintes dados significativos: qual PCD está com o alerta crítico, sensor, valor mínimo, valor máximo, valor da medição crítica, padrão de cores e localidade da Plataforma de Coletas de Dados informando aos usuários administrativos responsáveis e usuários cadastrados no sistema.</p>
RF #06	<p>O sistema deverá exibir os alertas críticos na página do usuário já mostrando a PCD de interesse, por meio de gráficos sendo o eixo X horário e o eixo Y nível do alerta desejado, com pesquisa de datas de início e término e legendas de acordo com os dados escolhidos pelo usuário, que estão indicados nos eixos dos gráficos.</p>
RF #07	<p>O sistema deverá exibir os alertas críticos na página do usuário já mostrando a PCD de interesse, por meio de uma tabela com dados tais como: pcd, sensor da PCD escolhida, valor mínimo, valor máximo, padrão de cores valor mínimo e padrão de cores valor máximo e uma imagem ilustrativa.</p>
RF #08	<p>As pessoas que são cadastradas podem ajudar o sistema dando alertas próximos a sua localidade de acordo com a ocorrência do momento, explicando o que está acontecendo, data, hora e uma foto mostrando a situação. Assim que possível essas informações serão validados pelo usuário administrativo, de acordo com os dados coletados disponíveis das PCDs e validando-os.</p>
RF #09	<p>Após ser enviado o comunicado passara por uma verificação de linguagem ofensiva e uma relação de comunicados em determinada área e dia. O comunicado ficara visível a todos os usuários logados na página de comunicados.</p> <p>Esse comunicado será verificado por um administrador antes de ser disponibilizado pela população de modo geral, o comunicado será enviado a tabela de comunicados e estará por padrão como não visualizado até que algum administrador aprove ou não o comunicado.</p>

3.2.2 Priorização dos Requisitos Funcionais

Para a priorização foi utilizado o modelo Kano. Como o projeto se trata de um feito acadêmico, o método foi realizado pelos alunos de cada módulo, que são os incumbidos de tornar a portal realidade, os requisitos foram analisados um a um por cada integrante do módulo de alertas, com o auxílio da tabela de priorização desenvolvida pelo Instituto Tecnológico de Aeronáutica (ITA), elaborada pelo grupo de Engenharia de Sistema da instituição.

Os requisitos foram classificados de acordo com as possibilidades dentre: Gostaria, Esperado, Não importa, Convivo com isso e Não gostaria, sendo respondido com o pensamento do impacto que a funcionalidade teria no meio o qual se encontra, se ele estivesse ou não presente. Depois de respondido a própria tabela o classificava de acordo com as possibilidades desenvolvidas por Kano. Essa tabela de priorização se encontra exemplificada na figura abaixo.



Instituto Tecnológico de Aeronáutica - ITA
Grupo de Pesquisa em Engenharia de Software - GPES
Elaborado pelo Grupo de Engenharia de Sistemas do SdS ICA-MMH
04/09/2008

Análise Kano para Priorização dos Requisitos pelo Cliente

Informações Gerais	
Possíveis respostas para:	
Como você se sente, se o requisito estiver presente?	
Como você se sente, se o requisito não estiver presente?	
Gostaria	Eu gostaria, mas não é imprescindível.
Esperado	Deve ser feito. É uma necessidade básica.
Não importa	Neutro. É indiferente. Não importa.
Convivo com isso	Não gostaria, mas é possível conviver com isso.
Não Gostaria	É inaceitável.

Conta Carácter	Identificador do Requisito	Requisito do Cliente	Como você se sente, se o requisito estiver presente	Como você se sente, se o requisito não estiver presente	Avaliação
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito
			Esperado	Não Gostaria	Deve ser feito

Figura 6: Tabela de auxílio para a priorização dos requisitos. [9]

Cada aluno, respondeu as mesmas perguntas, lendo os mesmos requisitos levantados. Disso gerou-se uma tabela específica individual, mostradas logo abaixo, juntamente com a tabela gerada da média ponderada das expectativas. Os atributos que foram classificados como “Deve ser feito”

seriam tidos como o foco e o qual requereria um maior comprometimento do conjunto. Todos os requisitos deveriam ser construídos, independentemente da sua classificação, isso respeitando as especificidades da disciplina e descrições do termo de abertura, assinado pelos trinta e três alunos comprometendo a entrega de pelo menos 90% do projeto como finalizado.

Expectativas do DBA 1 sobre os requisitos funcionais:

Identificador Requisito Funcional	Resposta Positiva	Resposta Negativa	Priorização Final
RF #01	Esperado	Não Gostaria	Deve ser feito
RF #02	Esperado	Não Gostaria	Deve ser feito
RF #03	Gostaria	Não Gostaria	Performance
RF #04	Esperado	Não Gostaria	Deve ser feito
RF #05	Esperado	Não Gostaria	Deve ser feito
RF #06	Esperado	Não Gostaria	Deve ser feito
RF #07	Esperado	Não Gostaria	Deve ser feito
RF #08	Esperado	Não Gostaria	Deve ser feito
RF #09	Esperado	Não Gostaria	Deve ser feito
RF #10	Esperado	Não Gostaria	Deve ser feito

Expectativas do DBA 2 sobre os requisitos funcionais:

Identificador Requisito Funcional	Resposta Positiva	Resposta Negativa	Priorização Final
RF #01	Esperado	Não Gostaria	Deve ser feito
RF #02	Gostaria	Não Gostaria	Performance
RF #03	Gostaria	Não Gostaria	Performance
RF #04	Esperado	Não Gostaria	Deve ser feito
RF #05	Gostaria	Não Gostaria	Performance
RF #06	Esperado	Não Gostaria	Deve ser feito
RF #07	Esperado	Não Gostaria	Deve ser feito
RF #08	Esperado	Não Gostaria	Deve ser feito

RF #09	Não Importa	Convivo com isso	Indiferente
RF #10	Não Importa	Convivo com isso	Indiferente

Expectativas do Analista 1 sobre os requisitos funcionais:

Identificador Requisito Funcional	Resposta Positiva	Resposta Negativa	Priorização Final
RF #01	Esperado	Não Gostaria	Deve ser feito
RF #02	Esperado	Não Gostaria	Deve ser feito
RF #03	Gostaria	Não Importa	Atrativo
RF #04	Gostaria	Não Gostaria	Performance
RF #05	Gostaria	Não Gostaria	Performance
RF #06	Gostaria	Não Gostaria	Performance
RF #07	Gostaria	Não Gostaria	Performance
RF #08	Gostaria	Não Gostaria	Performance
RF #09	Gostaria	Convivo com isso	Atrativo
RF #10	Gostaria	Convivo com isso	Atrativo

Expectativas do Analista 2 sobre os requisitos funcionais:

Identificador Requisito Funcional	Resposta Positiva	Resposta Negativa	Priorização Final
RF #01	Esperado	Não Gostaria	Deve ser feito
RF #02	Esperado	Não Gostaria	Deve ser feito
RF #03	Esperado	Não Gostaria	Deve ser feito
RF #04	Esperado	Não Gostaria	Deve ser feito
RF #05	Esperado	Não Gostaria	Deve ser feito
RF #06	Esperado	Não Gostaria	Deve ser feito
RF #07	Esperado	Não Gostaria	Deve ser feito
RF #08	Esperado	Não Gostaria	Deve ser feito
RF #09	Esperado	Não Gostaria	Deve ser feito
RF #10	Esperado	Não Gostaria	Deve ser feito

Expectativas do Desenvolvedor 1 sobre os requisitos funcionais:

Identificador Requisito Funcional	Resposta Positiva	Resposta Negativa	Priorização Final
RF #01	Esperado	Não Gostaria	Deve ser feito
RF #02	Gostaria	Convivo com Isso	Atrativo
RF #03	Esperado	Não Gostaria	Deve ser feito
RF #04	Gostaria	Não Importa	Atrativo
RF #05	Gostaria	Não Importa	Atrativo
RF #06	Esperado	Não Gostaria	Deve ser feito
RF #07	Não Importa	Não Importa	Indiferente
RF #08	Não Importa	Não Importa	Indiferente
RF #09	Gostaria	Não Importa	Atrativo
RF #10	Gostaria	Não Importa	Atrativo

Expectativas do Desenvolvedor 2 sobre os requisitos funcionais:

Identificador Requisito Funcional	Resposta Positiva	Resposta Negativa	Priorização Final
RF #01	Esperado	Não Gostaria	Deve ser feito
RF #02	Esperado	Não Gostaria	Deve ser feito
RF #03	Esperado	Não Gostaria	Deve ser feito
RF #04	Esperado	Não Gostaria	Deve ser feito
RF #05	Esperado	Não Gostaria	Deve ser feito
RF #06	Esperado	Não Gostaria	Deve ser feito
RF #07	Esperado	Não Gostaria	Deve ser feito
RF #08	Esperado	Não Gostaria	Deve ser feito
RF #09	Gostaria	Gostaria	Questionável (inválido)
RF #10	Gostaria	Gostaria	Questionável (inválido)

Média ponderada da expectativa do módulo de alertas sobre os requisitos funcionais:

Identificador Requisito Funcional	Priorização Final
RF #01	Deve ser feito
RF #02	Deve ser feito
RF #03	Deve ser feito
RF #04	Deve ser feito
RF #05	Deve ser feito
RF #06	Deve ser feito
RF #07	Deve ser feito
RF #08	Atrativo
RF #09	Atrativo

3.2.3 Requisitos não funcionais

Do estágio onde os requisitos funcionais, já estavam levantados, interpretados e explicados detalhadamente, inicializou a fase onde deveria ser pensando como essas funcionalidades seriam realizadas, os requisitos não funcionais. Devido ao fato de que cada módulo iria trabalhar separadamente, e no final apenas, é iriam integrar as partes, foram realizadas reuniões para se decidirem sobre alguns métodos de desenvolvimento padronizado, estes faziam jus ao que já foi aprendido nos anos anteriores de ensino.

Por se tratar de uma aplicação para web, alguns artifícios não puderam ser deixados de lado, como por exemplo, a linguagem de programação que define o esqueleto do sistema, o *HTML*, bem como o *CSS*, que permite a criação das características físicas das páginas, entre outros que estarão na tabela abaixo (requisitos comuns á todos os módulos de RN #01 até RN #05).

.Identificador	Descrição do Requisito
RN #01	Processo de <i>Software</i> : É necessário que o Processo de <i>Software</i> a ser utilizado seja o <i>Extreme Programming (XP)</i> + <i>Kanbam</i> + <i>RedMine</i> .
RN #02	Linguagem <i>PHP</i> , por se tratar de um sistema para web, de forma dinâmica, com a possibilidade de interação do usuário com a introdução de dados e links, é um requisito de extrema necessidade

	para a conclusão do projeto.
RN #03	<i>HTML</i> e <i>CSS</i> que juntos definem a estrutura do sistema, as características e efeitos visuais que o sistema pode efetuar, como: estilo de texto, uso e transições de imagens, entre outras funcionalidades.
RN #04	<i>Java Script</i> é a linguagem de programação que se baseia em scripts.
RN #05	O <i>MySQL</i> é um sistema gerenciador de banco de dados relacional de código aberto. O serviço utiliza a linguagem <i>SQL</i> (<i>Structure Query Language</i> – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.
RN #06	Pchart – é uma biblioteca de gráficos desenvolvida em <i>PHP</i> , que permite a criação de gráficos: de pontos, linhas, curvas de Bezier, etapas, parcelas, de pizza e outros.
RN #07	O sistema tem que proporcionar uma página fácil de ser compreendida pelo usuário, sendo também uma página agradável de mexer e observar.
RN #08	A data final para entrega deste projeto é 11 de novembro sendo entregue no mínimo 90% do sistema até o prazo estipulado.

Diante experiência do professor orientador, foi adquirida no processo de *software* a metodologia de *Extreme Programming*, ou apenas XP (RN #01). Também o auxílio de ferramentas de organização, tanto pessoal quanto do grupo em si, para que houvesse a comunicação entre os integrantes e um acompanhamento do fluxo de como as atividades estava: atrasada, adiantadas, no tempo.

O *Kanban*, próprio da instituição, nada mais é que um quadro virtual com cartões, nomeados com as atividades que deveriam ser feitas, com data de entrega, tempo de horas gastos e as a ser trabalhado, o nome do integrante responsável pela mesma e indicações sobre o andamento de uma determinada tarefa, por exemplo, “para executar”, “em andamento” ou “finalizado”. O *RedMine* possui a mesma função de organização que o *Kanban* possui, no entanto, este último, se baseia na web e no gerenciamento de bugs, permitindo a visualização de calendários com as atividades e gráficos de Gantt, representando visualmente o andamento do projeto e suas datas de entrega (RN #01).

O requisito não funcional 06 (RN #06), diz respeito á uma tarefa quase que exclusiva desse módulo, estando presente apenas nas ferramentas que teriam que fazer a demonstração de gráficos, nesse caso, o gráfico com os dados de alertas que já foram emitidos pela PCD de interesse do usuário cadastrado e logado ao Sistema Indra.

Um cuidado levado em consideração durante os levantamentos dos requisitos não funcionais, referente á qualidade do projeto, foi à idealização de uma interface de fácil uso, com objetivos claros e funções bem definidas. Por se tratar de uma aplicação de utilidade pública, tivemos a noção de que haveria usuários de todas as idades e tipos de instrução, desde a senhora que aprendeu a mexer apenas no Indra, até o programador do site da prefeitura que queria saber sobre o tempo e possíveis eventos climáticos anormais, por exemplo.

A data de entrega, estipulada nas primeiras horas de desenvolvimento do projeto, também esta especificada no documento assim como a porcentagem mínima de conclusão de entrega.

3.2.4 Questionário de Perguntas aos *Stakeholders*

Este questionário trata-se de perguntas que deveriam ser realizadas aos *Stakeholders*, no entanto, serviram apenas como reflexão aos analistas que deveriam pensar e projetar como seriam as funcionalidades do projeto.

1. Quais são os dados climáticos necessários em uma emissão de alerta?
R.:
2. Quais dados os usuários interessados em alertas precisam preencher, para que assim haja uma emissão de alertas mais específicos e precisos?
R.:
3. Conteúdo dos emails enviados aos responsáveis pelas PCDs e usuários cadastrados no sistema?
R.:
4. Tipos de alertas e como serão dispostos os gráficos?
R.:
5. Onde e como serão emitidos os alertas?
R.:

4 Conclusões e Recomendações

Sobre o levantamento de requisitos referente ao módulo-05 de gestão de alertas do sistema Indra, concluí-se que não se trata de uma simples concepção sem um estudo prévio e detalhado, o trabalho envolvido nessa atividade faz parte de um complexo processo, com metodologias aplicadas desde a ideia inicial do *software* até a conclusão do mesmo, respeitando a engenharia de *software* e suas especificidades. Esses pontos foram todos de uma forma simples, porém, completas explicadas nas páginas desse trabalho final de curso.

Pelo fato de os requisitos terem sido desenvolvidos pelos integrantes do módulo, alcançar o objetivo deste se concretizou naturalmente, foi necessário um levantamento teórico fomentando as técnicas empregadas na metodologia de trabalho, na mesma amplitude, em que foram formulados os requisitos, tendo por base os macrorrequisitos inicialmente propostos, os integrantes por meio de discussões e pesquisas os fizeram.

Positivamente adotamos o paradigma de discorrer sobre assuntos que faziam parte do sistema e onde houve um compromisso para a idealização do mesmo, como nesse caso do documento de requisitos, no entanto, analisando por de trás da problemática percebemos que o levantamento poderia ter sido descrito mais claramente, utilizando como auxílio uma das técnicas de levantamento de atributos, descritas e agora compreendidas fielmente.

“Sempre é preciso saber quando uma etapa chega ao final. Se insistirmos em permanecer nela mais do que o tempo necessário, perdemos a alegria e o sentido das outras etapas que precisamos viver.” [10].

Perante o Sistema Indra, a contribuição dos trinta e três alunos se encerra ao final do ano de 2016, porém, por se tratar de uma utilidade pública, o mesmo clama por uma continuidade e verdadeira implantação na sociedade, sendo expandidas em toda a cidade de São João da Boa Vista e região, e talvez até em todo o Brasil, já que o problema, digamos, solucionado pelo Indra atinge inúmeras habitações. Assim, uma efetivação e descobertas de novas funcionalidades que o sistema deve e pode realizar se mostra viável na continuação deste que foi o pontapé de um sistema de alertas contra enchentes.

5 Referências Bibliográficas

- [1] Temporal deixa mais de 40 famílias desalojadas em Águas da Prata, SP G1 São Carlos e Araraquara 04/01/2016 - Disponível em: <<http://g1.globo.com/sp/sao-carlos-regiao/noticia/2016/01/chuva-deixa-40-familias-desalojadas-em-aguas-da-prata-sp-diz-prefeitura.html>> - Acessado em: 26/08/2016.
- [2] Termo de abertura
- [3] A importância do levantamento de requisitos no sucesso dos projetos de *software* – linhadecódigo - Márcio Andrade Silva – Disponível em: <<http://www.linhadecodigo.com.br/artigo/1685/a-importancia-do-levantam.aspx>> - Acessado em: 28/08/2016.
- [4] Princípios da Engenharia de *Software* – DEVMEDIA- Higor Medeiros – Disponível em: <<http://www.devmedia.com.br/principios-da-engenharia-de-software/29630>> - Acessado em: 25/09/2016
- [5] Introdução ao Extreme Programming (XP) – DEVMEDIA - Higor Medeiros - Disponível em: < <http://www.devmedia.com.br/introducao-ao-extreme-programming-xp/29249>> - Acessado em: 27/10/2016
- [6] Levantamento de Requisitos – É mesmo necessário? - Wagner Gomes's Weblog – Wagner Gomes - 29/05/2008 - Disponível em: <<https://wagnergomes.wordpress.com/2008/05/29/levantamento-de-requisitos-e-mesmo-necessario/>> - Acessado em: 14/10/2016
- [7] Engenharia de *Software* 2 - Técnicas para levantamento de Requisitos - DEVMEDIA - Janaína Bedani - Disponível em: <<http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/9151>> - Acessado em: 24/10/2016
- [8] REQUISITOS – MODELO DE KANO – Revista MundoPM - Mario H. Trentim- 24/05/2012 - Disponível em: <<http://blog.mundopm.com.br/2012/05/24/requisitos-%E2%80%93-modelo-de-kano/>> - Acessado em: 26/10/2016
- [9] Portal Acadêmico - Prática de Desenvolvimento de Sistema (PDS) - Breno Lisi Romano - Disponível em: <<https://sites.google.com/site/blromano/disciplinas/pds2014>> - Acessado em: 25/10/2016
- [10] Gloria Hurtado – Pensador - Gloria Hurtado - Disponível em: < https://pensador.uol.com.br/autor/gloria_hurtado/> - Acessado em: 27/10/2016