

**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 e Prof. Luiz Angelo Valota Francisco

**A IMPORTANCIA DO PLANEJAMENTO PARA O PROJETO
GERAÇÕES**

Aluno: Heloisa Domingos Bonaretto

Prontuário: 1620088

São João da Boa Vista – SP

2019

Resumo

A importância do Projeto Gerações para a comunidade local, em especial para as Instituições de Longa Permanência é notável. Tendo sempre um planejamento eficiente por trás de um projeto de sucesso, objetivou-se elaborar um documento que mostrasse a importância desse planejamento no processo de desenvolvimento do projeto. Para cumprir esse objetivo, o documento será dividido em três etapas principais, sendo essas a apresentação do cronograma geral e as ferramentas utilizadas para o controle de seu progresso; a explicação do que foi desenvolvido por bimestre, a correlação entre os processos e a importância desse planejamento durante essa etapa; e a apresentação do planejamento do Módulo 08 (responsável pelo controle administrativo) e o desenvolvimento das Iterações baseado nele. Ao final, serão apresentados os resultados obtidos, os pontos positivos e negativos que puderam ser observados durante seu desenvolvimento, além de melhorias sugeridas para projetos futuros.

Palavras-chave: Planejamento; Casos de Uso; Iterações; Instituição de Longa Permanência; Software.

ÍNDICE DE ILUSTRAÇÕES

Figura 1: Módulos e seus requisitos	8
Figura 2: Demais módulos e seus requisitos	8
Figura 3: Requisito de Usuário e Requisito de Sistema	10
Figura 4: Exemplo de leitores	11
Figura 5: Processo de levantamento e análise de requisitos.....	12
Figura 6: Estrutura básica de uma página HTML	14
Figura 7: Exemplo de PHP inserido no HTML.....	15
Figura 8: Exemplo CSS.....	15
Figura 9: Exemplo de CSS externo	16
Figura 10: Interface do NetBeans IDE 8.2	16
Figura 11: Aparência de um Kanban.....	17
Figura 12: Exemplo de Banco de Dados (MYSQL)	18
Figura 13: Ciclo de desenvolvimento e teste	19
Figura 14: Parte do Cronograma disponibilizado	20
Figura 15: Visão geral - Kanban Módulo 08.....	21
Figura 16: Visão geral - Redmine Módulo 08.....	21
Figura 17: Cronograma 1º Bimestre.....	22
Figura 18: Logo escolhida.....	23
Figura 19: Template escolhido	23
Figura 20: Macro requisitos Módulo 08.....	24
Figura 21: Cronograma 2º Bimestre.....	26
Figura 22: Diagrama Caso de Uso Módulo 08.....	27
Figura 23: Ferramenta de Cálculo de Estimativa	28
Figura 24: Exemplo de Protótipo feito.....	29
Figura 25: Exemplo de integração ao banco - Seleção de setores	33
Figura 26: Iteração 01 - Criação de Tarefas	33
Figura 27: Cronograma 3º Bimestre.....	34
Figura 28: Exemplo de <i>insert</i> - Funcionários.....	35
Figura 29: Iteração 02 - Lista de Tarefas	35
Figura 30: Iteração 02 - Lista de Funcionários	36
Figura 31: Cronograma 4º Bimestre.....	36

ÍNDICE DE TABELAS

Tabela 1: Alguns <i>stakeholders</i> - Módulo 08	24
Tabela 2: Exemplo de requisito funcional Módulo 08	25
Tabela 3: Requisitos não funcionais Módulo 08	26
Tabela 4: Exemplo de Caso de Uso - Excluir Tarefa.....	28
Tabela 5: Tabela Priorização Requisitos - Módulo 08.....	30
Tabela 6: Planejamento Modulo 08 - Modificado	31
Tabela 7: Caso de Teste do Caso de Uso - Excluir Tarefa.....	32
Tabela 8: Tabela de Testes Executados.....	32

ÍNDICE DE ABREVIATURAS E SIGLAS

IBGE	Instituto Brasileiro de Geografia e Estatística
ONG	Organização Não Governamental
PDS	Prática de Desenvolvimento de Sistemas
DBA	Desenvolvedor de Banco de Dados
DEV	Desenvolvedor
MHC-PMS	Mental Health Care Patient Management System
MCU	Modelo de Caso de Uso
HTML	Hypertext Markup Language
PHP	Personal Home Page
CSS	Cascading Style Sheets
W3C	World Wide Web Consortium

SUMÁRIO

1	Introdução	7
1.1	Contextualização/Motivação	7
1.2	Objetivo Geral da Pesquisa	9
1.3	Objetivos Específicos	9
1.4	Estrutura do Documento.....	9
2	Desenvolvimento	10
2.1	Levantamento Bibliográfico.....	10
2.1.1	Requisitos	10
2.1.2	Dos Requisitos ao Caso de Uso	12
2.1.3	Prototipação	13
2.1.4	Planejamento	17
2.1.5	Banco de Dados	18
2.1.6	Desenvolvimento Iterativo e Casos de Teste.....	19
2.2	Etapas do Desenvolvimento da Pesquisa	19
2.2.1	Cronograma Geral do Projeto.....	19
2.2.2	Primeiro Bimestre.....	22
2.2.3	Segundo Bimestre.....	26
2.2.4	Terceiro Bimestre	34
2.2.5	Quarto Bimestre.....	36
3	Conclusões e Recomendações	38
4	Referências Bibliográficas	41

1 Introdução

1.1 Contextualização/Motivação

São João da Boa Vista é um município situado no estado de São Paulo que, segundo pesquisa feita pelo IBGE (2018), conta com aproximadamente 90.637 habitantes, entre eles cerca de 12.000 tem mais de 60 anos, segundo dados do IBGE (2010) [1].

Em 2017, após pesquisa feita pela Fundação Getúlio Vargas, foi eleita a melhor cidade com menos de 100 mil habitantes para idosos viverem. A pesquisa levou em conta o acesso aos serviços públicos e privados de saúde, situação financeira, educação, trabalho e cultura [2].

Atualmente, a cidade conta com diversas Instituições de Longa Permanência, sendo essas ONGs ou instituições privadas.

O município também conta com um Instituto Federal de Ciência e Tecnologia, uma instituição federal que oferece ensino profissionalizante, de forma totalmente gratuita. Entre os diversos cursos oferecidos, está presente o Técnico Integrado ao Ensino Médio, sendo este técnico em Eletrônica ou Informática, ambos com duração de quatro anos.

O curso técnico integrado à informática procura qualificar o aluno em diversas áreas dessa tecnologia, tendo como uma das matérias ministradas a Prática e Desenvolvimento de Sistemas (PDS), que visa desenvolver um projeto voltado para a comunidade local, feito inteiramente pelos alunos do último ano do curso, com orientação dos professores responsáveis.

Foi pensando nisso que os professores Breno Lisi Romano e Luiz Angelo Valota Francisco definiram o tema do Projeto Gerações. Neste ano, o projeto tem como objetivo desenvolver um sistema voltado a Instituições de Longa Permanência da cidade, construindo uma plataforma online onde não apenas os funcionários do local terão acesso, mas também a família do idoso. Podendo verificar diversas informações, desde a ficha do idoso, o histórico de alimentação diária do mesmo, as atividades realizadas durante sua estadia no local, até mesmo a histórico de pagamentos da instituição. Porém, cada usuário tem acesso restrito, para manter a segurança e organização da plataforma e de seus utilizadores.

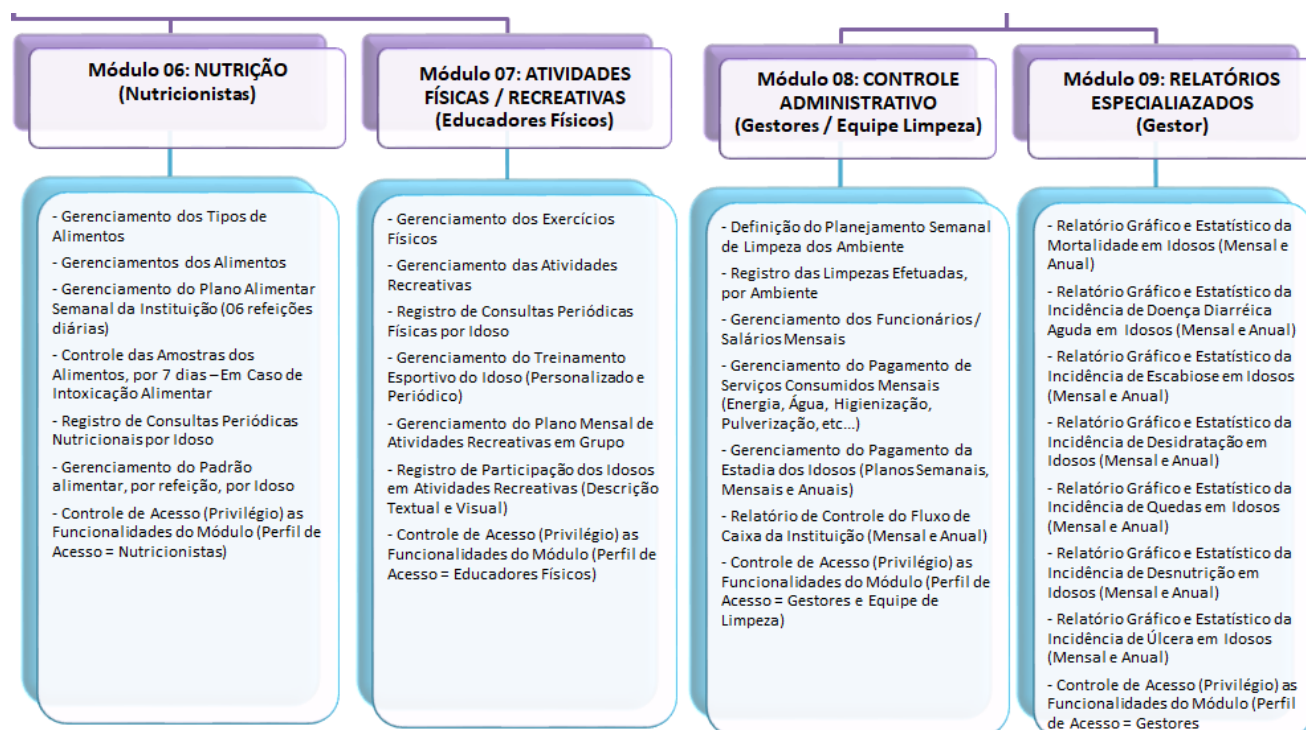
A equipe responsável pelo desenvolvimento é composta por mais de 50 alunos, divididos em nove módulos. Cada pessoa tem uma função, seja essa Analista, Administrador de Banco de Dados (DBA) ou Desenvolvedor (DEV). Cada módulo conta com pelo menos dois alunos de cada função e tem um tema específico, sendo esses e suas especificações:

Figura 1: Módulos e seus requisitos



Fonte: Portal Acadêmico [3]

Figura 2: Demais módulos e seus requisitos



Fonte: Portal Acadêmico [3]

Todos os requisitos aqui listados foram selecionados com base no artigo disponibilizado pelo Ministério da Saúde e a Agência Nacional de Vigilância Sanitária [4], que especifica os critérios mínimos que uma Instituição de Longa Permanência deve ter para oferecer uma boa estadia para seus clientes.

1.2 Objetivo Geral da Pesquisa

Essa pesquisa tem como objetivo mostrar como se deu o desenvolvimento do Projeto Gerações e a importância de seu planejamento.

1.3 Objetivos Específicos

Inicialmente será apresentado o cronograma geral do projeto, feito pelo professor, visando mostrar o que foi feito e as ferramentas utilizadas.

Então será explicado com mais detalhes a importância desse planejamento e todos os processos realizados a partir dele durante o projeto.

Por fim, o foco será no planejamento dos casos de uso do Módulo 08 e como o desenvolvimento se baseou nele.

1.4 Estrutura do Documento

Na introdução é apresentado um pouco sobre a história da cidade, do IFSP e também do projeto. Explicando seu objetivo e como se dá seu desenvolvimento.

No segundo capítulo será apresentado um pouco sobre o cronograma geral do projeto, os documentos criados, as ferramentas utilizadas e o desenvolvimento feito com base no planejamento realizado pelo Módulo 08.

A pesquisa se concluirá com um levantamento de dados dos resultados obtidos durante o projeto, visando dar enfoque na influência que o planejamento teve sobre esses resultados, o que poderia ter acontecido caso ele não tivesse sido feito e futuras melhorias sugeridas.

2 Desenvolvimento

2.1 Levantamento Bibliográfico

2.1.1 Requisitos

Requisitos são descrições do que o sistema deve fazer, o que ele deve oferecer e as restrições de seu funcionamento. Eles mostram as necessidades apresentadas pelos clientes e a finalidade que o *software* terá. Deve-se haver uma análise detalhada dos mesmos, juntamente com documentações e verificações [5].

Em alguns casos o requisito é uma declaração mais abstrata, abrangendo de forma mais ampla as necessidades [5], ficando conhecido como macro requisito. Em outros, é uma definição mais detalhada. Geralmente, o cliente tem uma ideia mais abstrata do que ele necessita, detalhando então brevemente as funções que o sistema deve ter (Requisitos de Usuário), é por isso que diferentes profissionais podem mostrar diferentes maneiras de se atender às necessidades do cliente. Após a contratação, é necessário que esse detalhamento seja mais específico (Requisitos de Sistema), e ambos contratados e contratante devem trabalhar em conjunto para que não se tenha problema de comunicação e entendimento, acarretando em problemas futuros. Na imagem abaixo é possível observar um exemplo desses requisitos em um sistema (MHC-PMS):

Figura 3: Requisito de Usuário e Requisito de Sistema

Definição de requisitos de usuário

1. O MHC-PMS deve gerar relatórios gerenciais mensais que mostrem o custo dos medicamentos prescritos por cada clínica durante aquele mês.

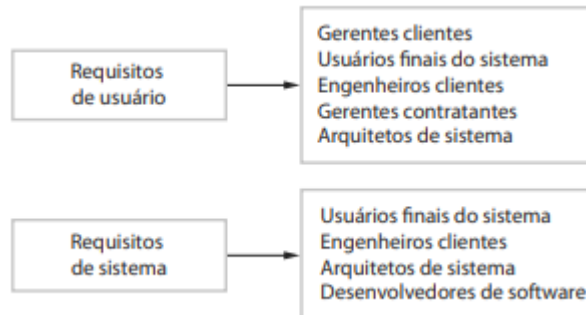
Especificação de requisitos de sistema

- 1.1 No último dia útil de cada mês deve ser gerado um resumo dos medicamentos prescritos, seus custos e as prescrições de cada clínica.
- 1.2 Após 17:30h do último dia útil do mês, o sistema deve gerar automaticamente o relatório para impressão.
- 1.3 Um relatório será criado para cada clínica, listando os nomes dos medicamentos, o número total de prescrições, o número de doses prescritas e o custo total dos medicamentos prescritos.
- 1.4 Se os medicamentos estão disponíveis em diferentes unidades de dosagem (por exemplo, 10 mg, 20 mg), devem ser criados relatórios separados para cada unidade.
- 1.5 O acesso aos relatórios de custos deve ser restrito a usuários autorizados por uma lista de controle de gerenciamento de acesso.

Fonte: Engenharia de Software [5]

Diferentes níveis de requisitos são necessários para que diferentes tipos de leitores possam lê-los. Os leitores de Requisitos de Usuário, por exemplo, não costumam se preocupar com a forma como o sistema será implementado, já os leitores de Requisitos de Sistema precisam saber mais detalhadamente como ele funciona, porque ele estará envolvido com o projeto de forma direta ou porque ele é um dos responsáveis pela criação do projeto [5].

Figura 4: Exemplo de leitores



Fonte: Engenharia de Software [5]

Há também a divisão de requisitos em Requisitos Funcionais e Requisitos Não Funcionais.

Requisitos Funcionais são serviços que o sistema deve fornecer e como ele deve ou não reagir em determinadas situações. Eles dependem de software que será projetado e seus possíveis usuários. Como, por exemplo, para o sistema MHC-PMS, um requisito pode ser o usuário ser capaz de pesquisar listas de agendamento, porém, ao fazer essa especificação, o cliente pode querer dizer que, ao pesquisar o nome do cliente a lista de todas as clínicas será acessada, mas o desenvolvedor pode interpretar isso de outra forma, como, por exemplo, criar uma listagem onde para se fazer uma pesquisa o usuário tenha que selecionar uma clínica específica [5].

Requisitos Não Funcionais são restrições a serviços ou funções do sistema, como, por exemplo, especificações de design, sistema operacional, tempo de resposta, etc. Eles podem parecer simples e alguns podem até mesmo serem alterados sem problema algum para o sistema, no entanto, deixar de atender algum mais específico pode causar a inutilização de todo o sistema. Além disso, um único requisito não funcional pode gerar uma série de requisitos funcionais [5].

Por conseguinte, cotando o grau de complexidade e de importância desses requisitos fica claro a necessidade de se ter um trabalho meticuloso em relação a eles, sabendo que problemas podem surgir após uma análise mais detalhada dos mesmos.

2.1.1.1 Levantamento e Análise de Requisitos

O levantamento de requisitos é uma etapa de extrema importância, que envolve diversos tipos de pessoas (*stakeholders*), sendo esses quaisquer usuários finais que irão interagir com o sistema e/ou serão afetadas por ele e também os engenheiros que o estão desenvolvendo [5].

Figura 5: Processo de levantamento e análise de requisitos



Fonte: Engenharia de Software [5]

O diagrama acima mostra as etapas que devem ser seguidas, sendo elas:

- Descoberta de Requisitos: deve haver uma interação com os *stakeholders* para descobrir quais são os requisitos. Pode-se utilizar entrevistas para o mesmo;
- Classificação e organização de requisitos: os requisitos devem ser agrupados caso tenham correlação entre si. Para isso, pode-se utilizar subsistemas;
- Priorização e negociação de requisitos: quando vários *stakeholders* se envolvem, gera-se um conflito, então é necessário priorizar alguns requisitos;
- Especificação de Requisitos: documentação dos requisitos já encontrados e busca por novos, juntamente com orientação dos *stakeholders* para que todos seus pedidos sejam analisados e verificados (pois algumas solicitações são inviáveis).

2.1.2 Dos Requisitos ao Caso de Uso

O MCU (Modelo de Caso de Uso) representa os possíveis usos de um sistema, estes estão associados a um ou mais requisitos funcionais identificados anteriormente. Nenhum sistema atua isoladamente, ele interage com atores humanos e outros sistemas. Sendo assim, torna-se necessário a compreensão de alguns termos: [6]

Um caso de uso especifica um comportamento de um sistema segundo uma perspectiva externa e é uma descrição de um conjunto de ações realizadas pelo sistema para produzir um

resultado [7]. Um modelo de caso de uso contém vários casos de uso, ele representa todas as possíveis iterações que serão descritas nos requisitos de sistema, quanto mais complexo o sistema, maior a quantidade de casos de uso que será usada [6]. Cada caso de uso é representado por uma elipse.

Ator é algo que interage com o sistema, enviando ou recebendo informações, ele é representado com figuras de “palito”. Uma mesma pessoa pode atuar como dois atores diferentes, como por exemplo, ser funcionário e cliente ao mesmo tempo [5][6].

Um ator pode estar associado a um ou mais casos de uso. Essa associação se dá por meio de relacionamentos. Ele é representado por uma linha e pode ser:

- Relacionamento de Comunicação (o mais utilizado) é quando o ator está associado ao caso de uso e ele interage com o sistema por meio deste [6];
- Relacionamento de Inclusão é utilizado quando alguns comportamentos são similares entre alguns casos de uso, para não ser necessário repetir sua descrição. O caso de uso inclusor é aquele que inclui o comportamento de outro, e é representado por uma dependência (*include*) [6];
- Relacionamento de Extensão indica que um ou mais cenários de um caso de uso pode se incluir em um comportamento de outro. Ele pode permanecer isolado, mas, em certas circunstâncias, seu comportamento é estendido pelo comportamento de outro caso de uso [6]. Como por exemplo a opção de “editar”, que não é necessariamente obrigatória, mas existe. Ele é representado pela dependência *extend*;
- Relacionamento de Generalização permite que um caso de uso ou um ator herde características de outro. E ambos participam dos relacionamentos que o outro participa. Como por exemplo um professor (caso de uso pai) e um aluno (caso de uso herdeiro) que podem pegar emprestado livros da biblioteca, porém, apenas o professor pode requisitar a compra de novos livros, ou seja, este caso de uso é específico para o professor, mas todos os outros também valem para o aluno [6].

2.1.3 Prototipação

A prototipação é a fase onde o desenvolvedor cria um modelo de *software* que posteriormente será avaliado pelo cliente e, então, implementado. Essa abordagem se inicia após a coleta e análise de requisitos, o responsável pelo desenvolvimento se baseia neles e constrói um sistema simples, para que o cliente avalie e veja as mudanças que serão necessárias. Então é

possível que se tenha um remodelamento do projeto, satisfazendo melhor as necessidades do cliente e tendo, então, um melhor resultado [8].

2.1.3.1 HTML

Quando se está navegando na *web*, é possível encontrar um conjunto de imagens, gráficos, textos, etc. Toda página *web* parece ser única, porém, elas têm algo em comum: todas utilizam HTML [9]. Ele é como um bloco de construção e todo o resto do conteúdo atua como um tijolo de suporte. Para que ele seja interpretado pelo navegador, são utilizadas *tags*, que já são pré-definidas, todas elas possuem um padrão de formato, se iniciando com "<" e finalizando com ">" [10].

O HTML 5 é a versão mais atual desse tipo de linguagem, ela surgiu para resolver problemas de compatibilidade da versão anterior (HTML 4) e permite criar aplicações *web* que podem interagir com dados locais e servidores mais facilmente.

Figura 6: Estrutura básica de uma página HTML

```
1 01 <!DOCTYPE html>
2 02 <html>
3 03 <head>
4 04     <meta charset="UTF-8"/>
5 05     <title>Document</title>
6 06 </head>
7 07 <body>
8 08     <!-- Conteúdo -->
9 09 </body>
10 10 </html>
```

Fonte: Devmedia [11]

2.1.3.2 PHP

PHP se originou em 1995, quando Rasmus Lerdof desenvolveu um *script* que possibilitou a ele saber quantos visitantes estavam lendo seu currículo online [12]. Sendo atualmente a versão 6 a mais nova.

Os desenvolvedores de PHP podem utilizar mais de 180 bibliotecas, contendo mais de 1.000 funções. Ela permite se conectar com o Banco de Dados, manipular informações de formulário e muito mais [12], devendo estar entre as *tags* "<?php" e ">?"

Figura 7: Exemplo de PHP inserido no HTML

```
<html>
  <head>
    <title>Teste PHP</title>
  </head>
  <body>
    <?php echo "<p>Olá Mundo</p>"; ?>
  </body>
</html>
```

Fonte: PHP [13]

2.1.3.3 CSS

A World Wide Web Consortium (W3C), uma comissão que define os padrões de programação, define CSS como um mecanismo simples para adicionar estilos (por exemplo: fontes, cores, espaçamentos) à documentos web. Este pode ser inserido junto ao HTML dentro das *tags* `<style>` ou pode ser um documento externo de extensão `.css`.

Assim como o HTML, ele também tem atributos pré-definidos. A sintaxe se dá por um seletor, uma propriedade e um valor.

Figura 8: Exemplo CSS

```
seletor {propriedade: valor;}
```

Por exemplo:

```
p {text-indent:10px;}
```

Fonte: Construindo Sites com CSS (modificado) [14]

Há diversos tipos de seletores: O simples, que não precisa estar necessariamente associado a uma classe; O universal, que abrange todos os elementos do documento e é representado por um `*`; E o de classe ou ID, que se associa a classe ou ID especificado [14].

Figura 9: Exemplo de CSS externo

```
1  h1{
2    font-style: italic;
3  }
4  #fundo{
5    background-color: #a3a9fa;
6  }
7  .paragrafo{
8    color: red;
9  }
10 .paragrafo2{
11   color: #0018ff;
12 }
```

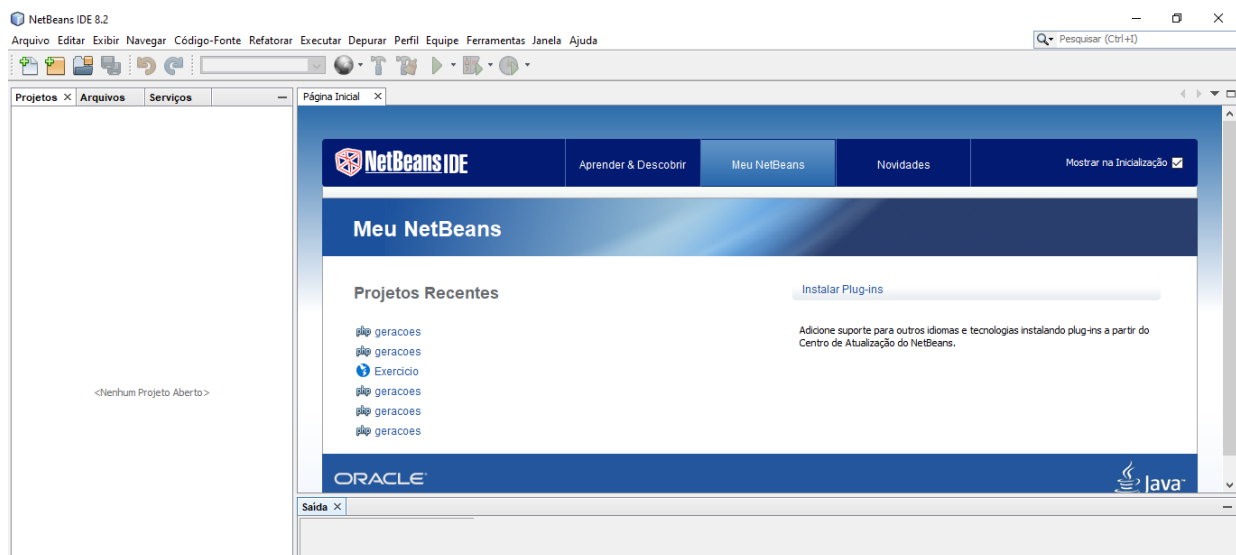
Fonte: Devmedia [15]

2.1.3.4 Netbeans

O NetBeans é um ambiente criado para desenvolvimento, onde se é permitido escrever, compilar e debugar [16], utilizando-se vários tipos de linguagens, entre elas HTML, PHP, CSS, etc.

Sua interface é de fácil entendimento e ela vem evoluindo rapidamente a cada versão [16]. Por isso, essa foi a ferramenta escolhida para ser utilizada no Projeto Gerações, além de que anteriormente aulas teóricas e práticas foram ministradas com o intuito de mostrar como se dá seu funcionamento, facilitando assim o trabalho dos Desenvolvedores, que já estavam aptos para lidar com essa aplicação.

Figura 10: Interface do NetBeans IDE 8.2



Fonte: Autoria Própria

2.1.4 Planejamento

Planejamento é um processo dinâmico e interativo para determinar objetivos, políticas e estratégias. Ele é elaborado por meio de técnicas de análise de ambientes (interno e externo), ameaças e oportunidades. Sua importância é inquestionável por visar o projeto como um todo, desde seu início até mesmo seu futuro, após ser implementado [17]. Ele exige uma visão abrangente, de modo que riscos sejam entendidos e possam ser enfrentados, facilitando o gerenciamento do projeto [18].

Um planejamento é criado para registrar o trabalho a ser feito, quem o fará, o cronograma de desenvolvimento e os produtos que devem ser gerados. Ele deve conter os objetivos do projeto, definir suas restrições (orçamentais, por exemplo), como a equipe desenvolvedora se organizará, quais são os possíveis riscos do projeto e as estratégias para contê-los, quais plataformas serão utilizadas e de que forma, quais resultados são esperados de cada atividade e os prazos para entrega de cada uma delas, além de monitorar todos os relatórios que deverão ser produzidos.

É inevitável que ele se modifique ao decorrer do projeto, por isso, o cronograma deve ser pessimista, já visando que problemas podem ocorrer e acarretar no atraso da entrega. Além disso, deve-se haver também a priorização dos atributos mais necessários, as coisas mais importantes devem ser desenvolvidas primeiramente. Deve-se também analisar o grau de dificuldade de cada uma, buscando sempre inicializar com as tarefas que são base de outras, as mais fáceis, ou as que poderão ser reutilizadas parcialmente em outra ocasião. Pois caso um atraso muito significativo ocorra o cliente deve ser contatado para que haja uma renegociação [5].

Para se controlar o cronograma e as tarefas que devem se realizadas, pode-se ser utilizado algumas plataformas mais amplas como Word e Excel ou até mesmo as específicas para isso, como Kanban e Redmine.

Figura 11: Aparência de um Kanban



Fonte: Método Ágil [19]

É crucial que se verifique sempre o andamento do projeto, visto que o atraso de uma tarefa pode afetar as demais tarefas, pois elas não poderão se iniciar sem que a anterior seja concluída, atrasando assim a equipe responsável.

Outrossim, o planejamento é importante para que não apenas a equipe responsável pelo desenvolvimento esteja controlada, mas também para que o cliente possa acompanhar o que é produzido. Estando previamente consciente do cronograma, dos preços, etc.

2.1.5 Banco de Dados

O banco de dados pode ser considerado como um armário de arquivamento que servirá de repositório para arquivos computadorizados. O sistema de banco de dados é para manutenção de registros, sua finalidade é armazenar informações e permitir que elas sejam buscadas e atualizadas quando necessário. Dados refere-se ao que é armazenado no banco, informações refere-se ao significado desses dados para determinado usuário [20].

2.1.5.1 MYSQL

O MYSQL é um servidor e gerenciador de banco de dados. Entre as vantagens de utiliza-lo estão: a maioria dos sistemas operacionais suporta a execução do mesmo; Ele armazena dados e provê outras funcionalidades como gerenciamento de acesso, integridade de dados, entre outros; Ele disponibiliza vários tipos de tabelas para armazenamento de dados, cada uma com suas próprias características, para qualquer tipo de situação, priorizando velocidade, volume de dados, etc; Além disso é altamente veloz e seguro [21]. Ele também funciona com *tags* pré-definidas. No exemplo abaixo, *drop database* atua como uma “pesquisa” para verificar se o banco já existe ou não, caso não, ele será então criado (*create*) e usado (*use*). Há também a criação de uma tabela (*create table*) e um exemplo de inserção (*insert*) de dados para esta tabela.

Figura 12: Exemplo de Banco de Dados (MYSQL)

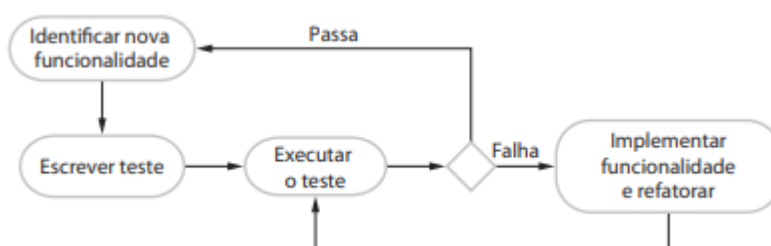
```
3 • DROP DATABASE IF EXISTS geracoes;
4 • CREATE DATABASE geracoes;
5
6 • USE geracoes;
7
8 • CREATE TABLE USUARIOS (
9     USU_RG VARCHAR(9) NOT NULL,
10    USU_DATA_NASCIMENTO DATE NOT NULL
11 );
```

Fonte: Autoria Própria

2.1.6 Desenvolvimento Iterativo e Casos de Teste

O processo iterativo busca fazer progresso através de tentativas sucessivas de refinamento. A equipe de desenvolvimento faz a primeira construção do primeiro caso de uso, ele é testado, seguindo todas as restrições que ele deve ter, se existirem pontos falhos ou incompletos ele deve ser refeito, até que o projeto atinja o resultado esperado [22]. A vantagem de utilizá-lo é que ele permite que as falhas sejam vistas rapidamente. Quanto mais cedo elas forem descobertas mais precisas serão as abordagens para resolvê-las.

Figura 13: Ciclo de desenvolvimento e teste



Fonte: Engenharia de Software [5]

2.2 Etapas do Desenvolvimento da Pesquisa

2.2.1 Cronograma Geral do Projeto

Antes do projeto se iniciar, o Professor Breno Lisi Romano tratou de criar um cronograma, no Excel, que deveria ser seguido durante todo o ano, para que se tivesse mais controle do que, como e quando seria feito; além de controlar futuros atrasos, já se preparando para tomar medidas corretivas, evitando conflitos.

Nele foram utilizadas as cores: verde (para representar itens já concluídos), vermelho (itens pendentes) e amarelo (itens que deveriam ser feitos com urgência). Também já foram previstos atrasos, tendo uma aba com a data ideal para entrega e a data em que a tarefa foi realmente entregue. Além disso, há não apenas o nome como a descrição do item, visando facilitar a compreensão.

Abaixo é possível observar uma parte do cronograma, disponibilizado aos alunos em formato de imagem no Portal Acadêmico [3].

Figura 14: Parte do Cronograma disponibilizado

Status	Data Real de Entrega	Data Inicialmente Prevista	Título da Atividade	Descrição
#01		08-Feb	Estudo de Caso	Decidir qual será o Contexto do Projeto Acadêmico??
#02		08-Feb	Definição da Plataforma de Desenvolvimento	Linguagem de Programação e Banco de Dados
#03		08-Feb	Definição das Equipes	Analistas de Sistemas, Desenvolvedores, DBAs, Testadores e Equipe de Documentação
#04		15-Feb	Definição dos Módulos (Macrorequisitos)	Definição dos Módulos do Estudo de Caso escolhido e os respectivos macro-requisitos a serem contemplados
#05	21-Feb	15-Feb	Definição das Responsabilidades de Gerência	Divisão das responsabilidades de gerência entre os integrantes de cada um dos módulos do projeto.
#06	Pendente	01-Mar	Termo de Abertura do Projeto	Definir o Contrato Inicial do Projeto
#07		15-Mar	Definição do Nome Final do Projeto e do Logo (Identidade Visual)	Deve-se estabelecer um único nome ao projeto com uma data anterior a sugerida e até a data final, já ter definido uma identidade visual do projeto (logo) com base no nome estabelecido.

Fonte: Portal Acadêmico [3]

2.2.1.1 Ferramentas utilizadas

Para controlar o andamento da realização das atividades foram utilizados Kanban e Redmine, ambas plataformas de organização e planejamento, como citado anteriormente.

No Kanban, há a separação por abas. Algumas tarefas que deveriam ser realizadas por todos deveriam ser colocadas em “Projeto Geral”, já as demais, específicas de cada módulo, deveriam ser colocadas na aba direcionada para o módulo em questão. Abaixo é possível observar um pouco das do módulo 08, onde Backlog diz respeito a atividades pendentes, Desenvolvimento contem as tarefas que estão sendo realizadas, Verificação e Validação representa os itens que já foram finalizados e deveriam passar por aprovação do grupo e consequentemente do professor (quando se trata de um documento) ou que deveria ser testado (quando se trata de uma iteração). Caso tudo estivesse funcionando corretamente, o item poderia ser levado para a aba Finalizado, caso não, deveria voltar para a aba Desenvolvimento, e assim por conseguinte, sendo desenvolvida, testada... até que ela fosse aprovada para ser finalizada

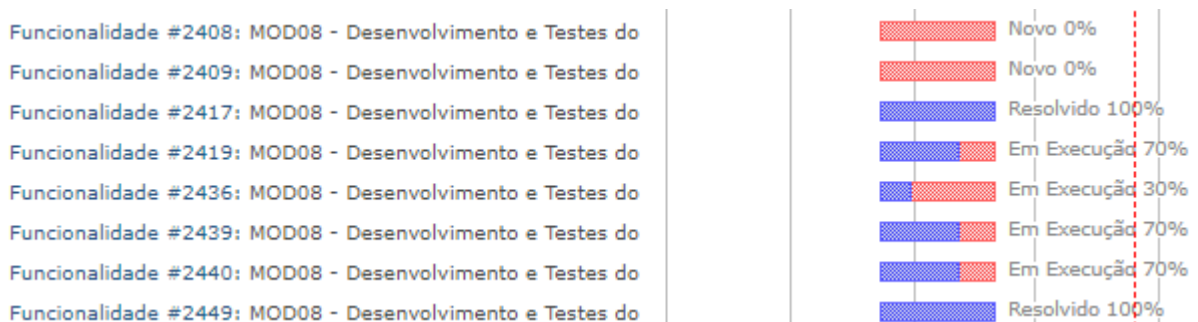
Figura 15: Visão geral - Kanban Módulo 08



Fonte: Autoria Própria

O Redmine, que tem a mesma função, tem um design diferente. Ele deve ser acessado com uma senha específica de cada módulo e é mostrado em forma de gráfico a realização das tarefas. É possível filtrar por dia, mês, ano, apenas tarefas pendentes, apenas tarefas realizadas etc. Ao contrário do Kanban, não é possível verificar em que etapa do desenvolvimento a tarefa está, porém, o responsável deve analisá-la e colocar uma porcentagem relativa de conclusão. Por exemplo, 0% significa que a tarefa ainda não foi iniciada, 90% significa que ela ainda deve passar pelo processo de verificação e validação, 100% significa que ela foi totalmente finalizada, enquanto isso, qualquer outro valor significa que ela está em desenvolvimento.

Figura 16: Visão geral - Redmine Módulo 08



Fonte: Autoria Própria

A utilização dessas duas funções e a responsabilidade de cada integrante ao sempre manter atualizado o andamento das tarefas foi de extrema importância para o controle do projeto. Não apenas para o professor responsável como também para a equipe, que pôde acompanhar as pendências e itens finalizados de seu módulo e também dos colegas.

2.2.2 Primeiro Bimestre

Figura 17: Cronograma 1º Bimestre



Fonte: Autoria Própria

No primeiro bimestre o foco foi apresentar a ideia inicial do projeto e decidir qual linguagem de programação seria utilizada e quais programas seriam usados. Então teve-se a divisão da turma por módulos, cada aluno tendo sua função.

Iniciou-se finalmente o trabalho no projeto, com a criação do contrato inicial (Termo de Abertura), contando informações importantes, tais como as necessidades que o projeto deve atender, as restrições (orçamentarias, de acesso, de qualidade, etc). Entre diversas restrições, decidiu-se que, no mínimo, 80% do projeto deveria ser entregue até 29 de novembro de 2019. Esse contrato é fundamental para que se tenha total transparência com o cliente além de um controle do que é pedido e das restrições impostas, tanto pelo cliente quanto pelo sistema. O mesmo foi assinado por todos os envolvidos no projeto.

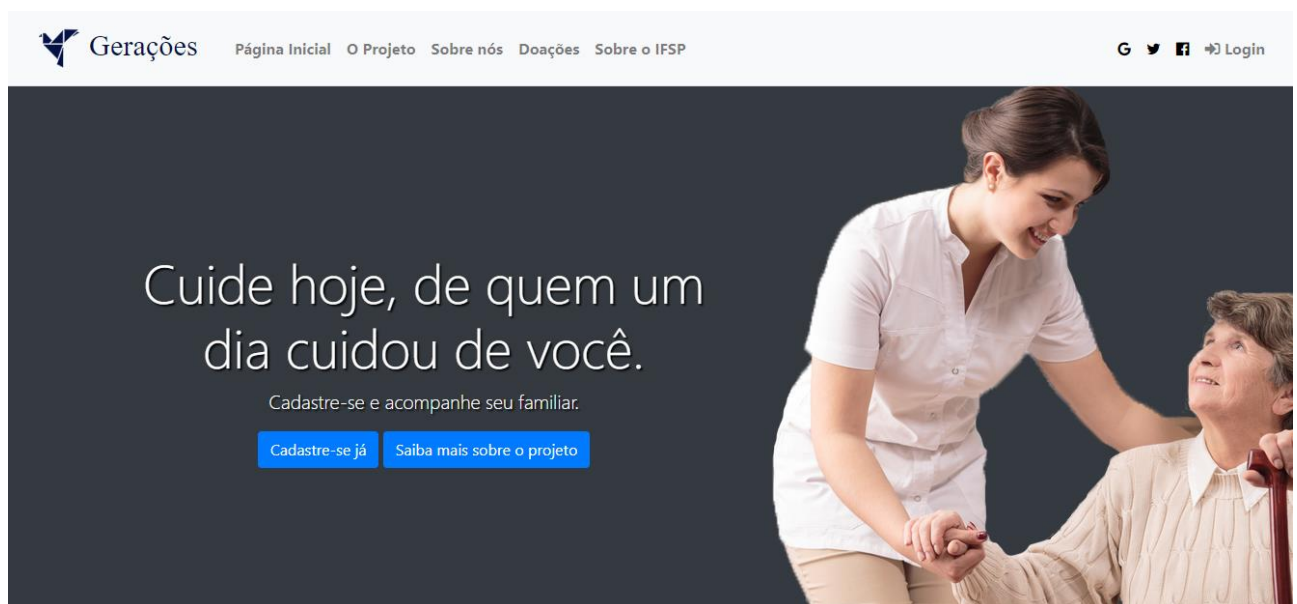
Além disso, foram idealizadas pelos desenvolvedores logos, nomes e templates para o projeto, sendo escolhido ao final apenas um de cada. Para isso, uma votação foi feita com toda a sala.

Figura 18: Logo escolhida



Fonte: Autoria Própria

Figura 19: Template escolhido



Fonte: Autoria Própria

2.2.2.1 Identificação de macro requisitos

Cada um dos módulos tem seus próprios macro requisitos, ou seja, é uma forma abrangente de identificar o que o sistema deve fazer. No caso do módulo 08, focado no controle administrativo e no planejamento da limpeza do local, os macro requisitos encontrados foram:

Figura 20: Macro requisitos Módulo 08



Fonte: Autoria Própria

2.2.2.2 Identificação de stakeholders

Com base no que o sistema deve fazer, é possível então saber quais são os stakeholders, ou seja, as pessoas que interagirão com o sistema, sendo estas usuários finais ou também outros módulos que necessitam de informações do módulo 08 ou vice versa:

Tabela 1: Alguns stakeholders - Módulo 08

Nome	Descrição	Responsabilidades
Módulo de Prontuário dos Idosos	Professor e Alunos do Curso Técnico em Informática Integrado ao Ensino Médio, da matéria PDS: Prática e Desenvolvimento de Sistemas, do IFSP-SBV	Define os objetivos e o escopo do Módulo de Prontuário dos Idosos; permite que o gestor visualize a entrada de novos idosos para gerenciar os pagamentos mensais deles.
Módulo de Acompanhamento dos Familiares	Professor e Alunos do Curso Técnico em Informática Integrado ao Ensino Médio, da matéria PDS: Prática e Desenvolvimento de Sistemas, do IFSP-SBV	Define os objetivos e o escopo do Módulo de Acompanhamento dos Familiares; o módulo de CA permite que os familiares acompanhem o histórico de pagamento da estadia dos idosos.
Gestor	O responsável por administrar funcionários e gastos da instituição	Gerenciamento do salário dos funcionários, pagamentos de mensalidade e controle do fluxo da caixa da Instituição.
Representante da equipe de Limpeza	Funcionário que organiza e coordena as funções dos demais	Definir setores de limpeza para cada funcionário e definir o Planejamento Semanal de Limpeza.
Responsável pelo Idoso	Familiar do idoso	Acompanha o idoso por meio da aplicação web 'Gerações', realiza e acompanha o pagamento semanal/mensal/anual.

Fonte: Autoria Própria

2.2.2.3 Levantamento de requisitos

Pensando nos macro requisitos encontrados, foi levantado então os requisitos funcionais e não funcionais. Para cada um dos requisitos encontrados foi necessária a explicação detalhada das ações que ele deve oferecer, de suas restrições, de todos os campos que devem existir nele e suas características (tais como tamanho máximo de caracteres, se é aceito apenas caracteres do tipo texto ou número, etc). Para melhor observação e exemplificação apenas um dos total de onze requisitos funcionais levantados será apresentado:

Tabela 2: Exemplo de requisito funcional Módulo 08

Identificador	Descrição do Requisito
RF #01	<p>O sistema deve permitir que o representante da equipe de limpeza possa criar (insert), editar (update) e excluir (delete from table) tarefas no planejamento semanal da limpeza. Este usuário também poderá fazer (se estiver logado) uma marcação nas tarefas que já foram realizadas através de uma caixa de seleção que estará localizada logo no início da descrição. Para ele poder cadastrar o planejamento deverá aparecer uma tela com os seguintes campos nesta respectiva ordem:</p> <ul style="list-style-type: none">• Tabela: Planejamento Semanal• Nome da tarefa*: Varchar (50);• Descrição da tarefa*: Varchar (250);• Data de realização: Date-time;• Status da tarefa* (concluída ou não): Boolean• Funcionário responsável pelo setor*: Varchar (100);• Nome do setor*: Varchar (50);• Observações: Varchar (500); <p>Abaixo dos campos de preenchimento deverá ter um botão (button) “Cadastrar” e um “Cancelar”. O botão Cadastrar deverá registrar as informações no banco de dados permitindo que elas sejam acessadas pelo usuário em uma pesquisa. O botão Cancelar deverá fechar a janela.</p> <p>O sistema deverá ter uma página de pesquisa onde será possível pesquisar pelas tarefas do dia de um funcionário específico visualizando isto a partir de uma tabela. Para realizar a pesquisa o usuário precisa preencher os campos “Data de início” (Date-time); “Data de término” (Date-time); “Setor” (button dropdown do bootstrap). Na frente desses campos deve haver um botão “Pesquisar” (select), quando o usuário clicar nele, o sistema deverá apresentar uma tabela contendo 5 colunas com os seguintes dados na respectiva ordem:</p> <ul style="list-style-type: none">• A primeira linha da tabela conterá os nomes dos dados: “código”, “nome da tarefa”, “data da tarefa”, “setor” e “configurações”;• A primeira coluna deverá ser ocupada pelos códigos (autoincrement);• Nome da tarefa• Data de realização;• Nome do setor;• Dos ícones impressos na tela sendo eles para editar e outro para deletar.

Fonte: Autoria Própria

Esses requisitos são específicos de cada módulo, pois cada um tem um objetivo diferente. Já muitos dos requisitos não funcionais foram exatamente os mesmos para quase todos os módulos, pois especificam o tipo de plataforma a ser utilizado, as linguagens, as restrições etc. Por isso, apenas alguns dos requisitos não funcionais pertencentes somente ao módulo 08 serão aqui apresentados:

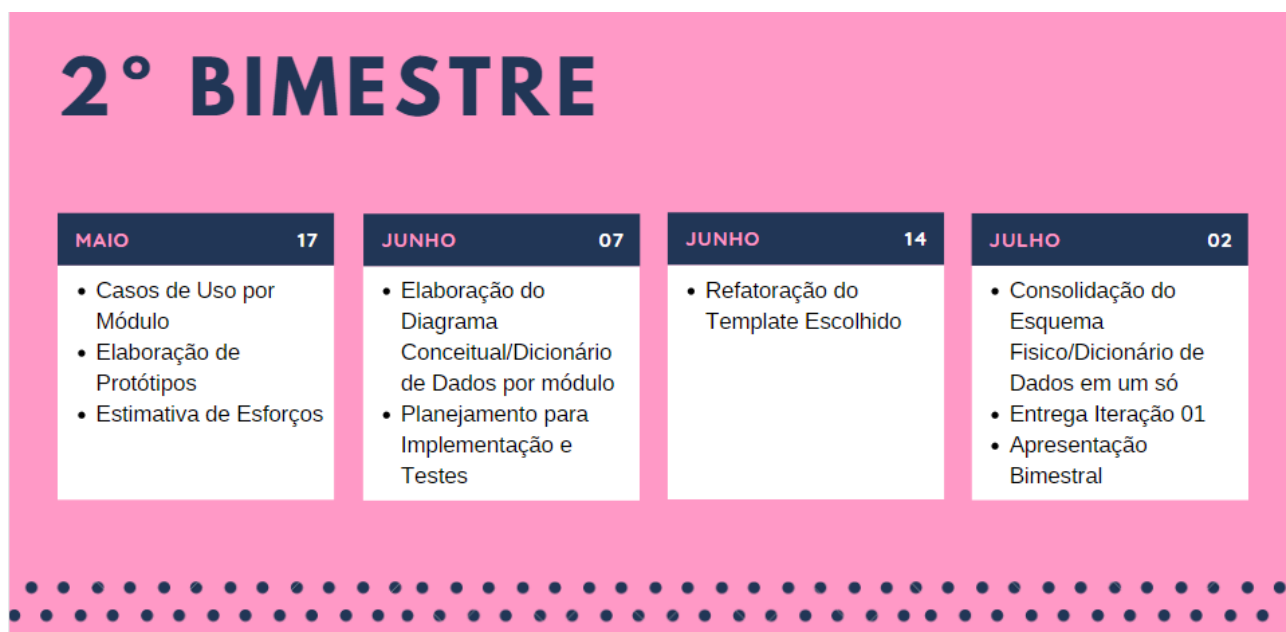
Tabela 3: Requisitos não funcionais Módulo 08

Identificador	Descrição do Requisito
RN #12	É necessário que o sistema faça controle de acesso, restringindo o acesso da equipe da limpeza para somente a aba de Limpeza. O gestor deve ter acesso total ao site.
RN #14	Será necessário que o Sistema exiba um aviso caso o gestor não sinalize o pagamento de alguma despesa. O aviso deve ser construído a partir de um “alert dismissing” do bootstrap. Ele deve ter a cor amarela (warning) e possuir um “x” no canto direito para que o usuário tenha a opção de fechar o alerta.
RN #15	O sistema deverá emitir um aviso indicando que o responsável pelo idoso não efetuou o pagamento da dívida. O aviso deve ser construído a partir de um “alert dismissing” do bootstrap. Ele deve ter a cor amarela (warning) e possuir um “x” no canto direito para que o usuário tenha a opção de fechar o alerta.

Fonte: Autoria Própria

2.2.3 Segundo Bimestre

Figura 21: Cronograma 2º Bimestre

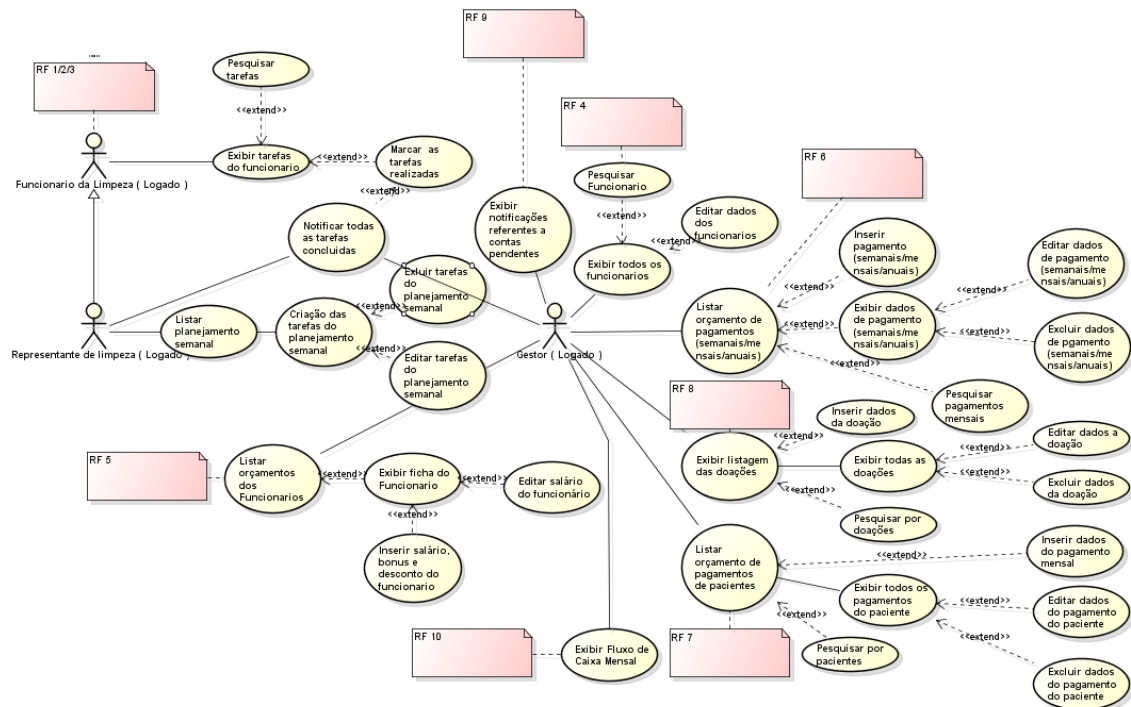


Fonte: Autoria Própria

2.2.3.1 Diagrama de Caso de Uso

Tendo-se os requisitos e os atores envolvidos, foi então elaborado, por cada módulo, um Diagrama de Caso de Uso. Ele especifica as ações realizadas pelo sistema e deve estar vinculado aos atores que interagem com o mesmo, além de ter as ações que se estendem a ele (que podem acontecer mas não são obrigatórias).

Figura 22: Diagrama Caso de Uso Módulo 08

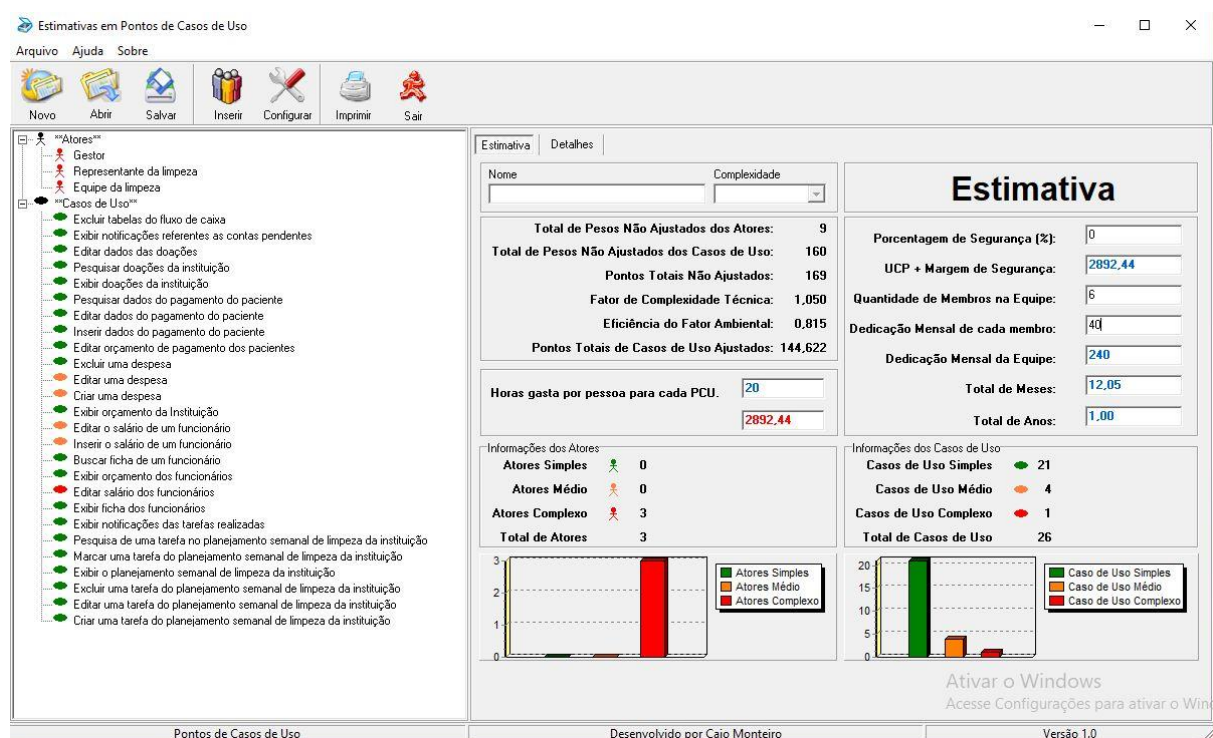


Fonte: Autoria Própria

2.2.3.2 Estimativa de esforços

Depois, utilizando uma ferramenta específica foi feito uma estimativa de custo/tempo para se ter uma base de quanto deveria ser cobrado por cada módulo.

Figura 23: Ferramenta de Cálculo de Estimativa



Fonte: Autoria Própria

O módulo 08, como mostra a imagem, ficou com o preço estipado em cerca de 28.000,00; o projeto como um todo ficou com o custo de aproximadamente 278.000,00.

2.2.3.3 Especificação dos Casos de Uso

Com base nos Casos de Uso gerados foi criado um documento, onde cada caso de uso virou uma tabela, devendo conter uma breve descrição do mesmo, quem seria a pessoa principal a utilizá-lo (no caso, o representante da equipe de limpeza), todas as ações que deveriam ser feitas e quais as os possíveis fluxos alternativos (ações não obrigatórias, mas que podem acontecer, identificadas como *extend* no Diagrama de Caso de Uso).

Tabela 4: Exemplo de Caso de Uso - Excluir Tarefa

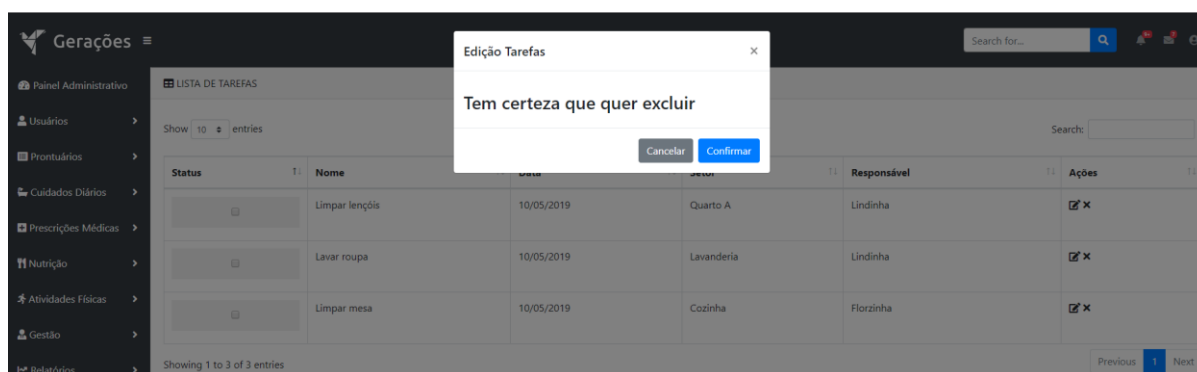
Nome do Caso de Uso: Excluir uma tarefa do planejamento semanal de limpeza da instituição	
Breve Descrição:	Este Caso de Uso deve ocorrer sempre que o usuário clicar na opção de excluir tarefa
Ator Principal:	Representante da equipe de limpeza.
Pré-Condição:	Estar logado no sistema e possuir um perfil de representante da limpeza.

Fluxo Principal	
Ações dos Atores:	Ações do Sistema:
1. Clicar em “Excluir tarefa”.	2. O sistema deverá exibir uma tela de confirmação com as opções de confirmar ou cancelar.
3. Clicar em confirmar.	4. Excluir os dados e atualizar o banco de dados.
5. Fim do Caso de Uso.	
Fluxo Alternativo A: Cancelar exclusão	
3. Na mensagem de confirmação que o sistema mostra na tela, o usuário clica em cancelar a exclusão	4. O sistema fecha a janela e não exclui nenhum dado.
5. Fim do Caso de Uso	

Fonte: Autoria Própria

Baseado nas tabelas criadas, foi desenvolvido um ou mais protótipos para cada uma delas, um para o fluxo principal e outros para cada um dos fluxos alternativos, para se ter uma base do que estava sendo pedido e quais melhorias poderiam ser feitas no desenvolvimento final. Estes foram codificados na plataforma Netbeans, utilizando as linguagens HTML e CSS. No caso acima, o protótipo foi:

Figura 24: Exemplo de Protótipo feito



Fonte: Autoria Própria

Onde mostra que, após as tarefas serem criadas elas deveriam ser listadas e ter a opção de exclusão. Caso selecionada, o usuário deveria realmente confirmar se era essa sua intenção (clicando no botão confirmar) ou não (clicando em cancelar). Não há protótipo de fluxo alternativo pois em caso de cancelamento nada acontece, apenas a tela aberta se fecha, e em caso de confirmação a tarefa é excluída, ambos resultando na volta à tela de listagem de tarefas.

Após todos os protótipos de todos os módulos estarem prontos foi feita uma integração do projeto, por exemplo, antes cada módulo tinha seu próprio documento CSS, porém, após a

integração, todos os códigos foram agrupados em apenas um documento CSS. Uma pasta foi criada para cada módulo, onde suas páginas PHP deveriam ser inseridas etc; Isso foi de extrema importância para que se tivesse um controle sobre o template, evitando que tudo ficasse bagunçado e tivesse código perdido. Além disso, ele já ficou totalmente pronto para receber os códigos de integração com o Banco de Dados que receberia posteriormente.

2.2.3.4 Criação do Planejamento de cada Módulo

Então, os Casos de Uso foram analisados e agrupados, a fim de criar-se um planejamento. Esse planejamento tem grande importância no projeto pois, além de orientar os desenvolvedores, controla quando cada coisa deve ser feita. Cada módulo criou seu próprio planejamento, escolhendo quais itens seriam desenvolvidos primeiro, essa escolha foi feita de forma estratégica a fim de priorizar as tarefas mais importantes primordialmente, para que, em caso de algum atraso, não houvesse muita perda de funcionalidade no sistema.

Essa priorização foi feita utilizando-se o método Kano. Cada integrante deveria analisar cada um dos requisitos funcionais, pensando nas perguntas:

- Como você se sente se o requisito estiver presente?
- Como você se sente se o requisito não estiver presente?

Para isso, foi utilizada uma tabela no Excel, onde as respostas deveriam ser escolhidas entre as opções pré-definidas: Gostaria, Esperado, Não importa, Convivo com isso, Não gostaria.

Ao final, cada integrante teve um resultado final para cada um dos requisitos. Foi então feita uma média, analisando as respostas que mais apareceram. O resultado final do módulo 08 pode ser observado abaixo:

Tabela 5: Tabela Priorização Requisitos - Módulo 08

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

Fonte: Autoria Própria

Os requisitos considerados como deve ser feito são os que devem, obrigatoriamente, serem cumpridos para que o sistema seja funcional. Já os considerados como atrativos são os requisitos aos quais seria interessante se ter, porém sua falta não gera muita insatisfação.

Tabela 6: Planejamento Modulo 08 - Modificado

Iteração	Data Prevista	Casos de Uso	Equipe Responsável	Desenvolvido?	Testado?
#01	28/06	#4.2. Criar tarefas	DEVs	100%	100%
		#4.3. Editar tarefas	DEVs	100%	100%
		#4.4. Excluir tarefas	DEVs	100%	100%
#02	23/08	#4.1. Listar planejamento	Grupo 02	100%	100%
		#4.5. Listar tarefas	Grupo 01	100%	100%
		#4.7. Pesquisa de uma tarefa	Grupo 01	100%	100%
		#4.6. Marcar tarefa realizada	Grupo 01	100%	100%
		#4.11. Exibir orçamento dos funcionários	Grupo 02	100%	100%
		#4.12. Buscar ficha de um funcionário	Grupo 02	100%	100%
		#4.9. Exibir ficha do funcionário	Grupo 02	100%	100%
		#4.13. Inserir o salário, bônus e/ou desconto de um funcionário	Grupo 01	100%	100%
		#4.10. Editar dados dos funcionários	Grupo 02	100%	100%
		#4.14. Editar o salário de um funcionário	Grupo 02	100%	100%

Fonte: Autoria Própria

No planejamento, cada agrupamento foi chamado de Iteração, tendo sua data prevista, os Casos de Uso que ela agrupa, a equipe responsável: um analista, responsável por testar o que foi desenvolvido; um desenvolvedor, para criar as páginas dessa iteração e conectá-la ao banco; e um DBA, este responsável pela criação e manutenção do banco de dados. Apenas 2 das 5 Iterações foram apresentadas aqui, para melhor compreensão, visto que ambas já estão finalizadas.

2.2.3.5 Integração do Banco de Dados e Casos de Teste

Para que os desenvolvedores pudessem trabalhar nesses itens, foi necessário que os DBAs integrassem o modelo físico que tinha sido feito em um só. Foi uma tarefa difícil que necessitou muita atenção do grupo para que tudo funcionasse corretamente. Enquanto isso os analistas criaram

um documento de Caso de Teste, a fim de controlar as iterações feitas. Nele, todos os casos de uso foram colocados juntamente com sua descrição e o resultado esperado. Para cada um foi feito um acompanhamento de teste, ou seja, deveria ser anotado o dia que o item foi testado e seu resultado, caso o resultado for positivo isso significa que o resultado esperado foi executado em sucesso, caso contrário significa que há algum problema em sua execução, sendo então anotado como falha e repassado ao desenvolvedor, que deveria refazer o item. Com base nisso foi calculado uma porcentagem de desenvolvimento e teste de cada Caso de Uso, ambas foram anotadas na planilha do planejamento.

Tabela 7: Caso de Teste do Caso de Uso - Excluir Tarefa

Casos de Teste	Descrição	Resultado Esperado
CT #05	Verifica se o usuário seleciona o ícone de exclusão na coluna de ações do planejamento semanal.	O sistema deverá exibir uma tela de confirmação com as opções de confirmar ou cancelar.
CT #06	Verifica se, na mensagem de confirmação, o usuário confirma a exclusão.	O sistema deverá apagar a tarefa do banco de dados.
CT #07	Verifica se, na mensagem de confirmação, o usuário cancela a exclusão.	O sistema deve abortar a exclusão da tarefa.

Fonte: Autoria Própria

Tabela 8: Tabela de Testes Executados

	02/08/2019	09/08/2019	23/08/2019	06/09/2019	20/09/2019	24/09/2019
CT #01	Sucesso				Falha	Falha
CT #02	Falha		Sucesso			
CT #03	Falha		Sucesso			

Fonte: Autoria Própria

2.2.3.6 Desenvolvimento de Iterações

Iniciou-se então a criação das iterações. Por ainda não ter o banco de dados populado (ou seja, ter informações previamente cadastradas lá) não foi possível fazer sua integração com o Banco, pois era necessário ter informações tais como os nomes dos funcionários da empresa para

que a tarefa fosse cadastrada. Porém, para solucionar o problema temporariamente, os desenvolvedores criaram um banco próprio, provisório, apenas para poder entregar a iteração a tempo e depois modificá-la, quando o banco integrado estivesse pronto.

As iterações são um refinamento dos protótipos anteriormente criados, o código base já estava pronto e as modificações necessárias foram feitas. Para se inserir, editar, excluir e mostrar informações do banco foi necessário utilizar-se da linguagem PHP, inserida junto ao código HTML da própria página que estava em desenvolvimento, como mostra a imagem abaixo:

Figura 25: Exemplo de integração ao banco - Seleção de setores

```
<label for="setor">Setor a ser limpo:</label>
<select id="setor" name="setor" class="form-control" >
    <?php
        foreach ($this->getView()->dados2 as $set) {
            echo '<option value="' . $set->__get('SET_ID') . '>' . $set->__get('SET_NOME') . '</option>';
        }
    ?>
</select>
```

Fonte: Autoria Própria

Como descrito no planejamento a primeira iteração do Módulo 08 se refere a criação, edição e exclusão de tarefas do planejamento semanal da limpeza. A imagem abaixo mostra o campo de criação de tarefa, contendo campos no formato data, texto (onde o usuário pode inserir o que quiser) e *select* (onde o usuário seleciona entre as opções que existem, como no campo “Funcionário responsável”). Esses nomes de funcionários estão sendo puxados do Bando de Dados, pelo PHP, da mesma forma como na imagem acima.

Figura 26: Iteração 01 - Criação de Tarefas

A interface de criação de tarefas apresenta um formulário com os seguintes elementos:

- Menu Lateral:** Usários, Prontuários, Responsável, Cuidados Diários, Prescrições Médicas, Nutrição, Atividades Físicas, Gestão, Relatórios, Enviar Feedback.
- Cabeçalho:** Lista de Tarefas / Criação de Nova Tarefa.
- Ação:** Criar Nova Tarefa.
- Formulário:**
 - Data que deve ser realizada:** 05/02/2000
 - Nome:** Limpar
 - Setor a ser limpo:** PÁTIO
 - Funcionário responsável:** Yuri Ian Benjamin Pires (menu suspenso com opções: Yuri Ian Benjamin Pires, Bryan Leandro Lorenzo Mendes, Silvana Camila Helena Castro, Antônia Manuela Almeida, **Calebe Guilherme Baptista**, Enrico Luiz Ryan Rezende, Antonella Isabelly da Mata)
 - Descrição da tarefa:** [Campo de texto]
 - Observações:** [Campo de texto]

Fonte: Autoria Própria

2.2.4 Terceiro Bimestre

Figura 27: Cronograma 3º Bimestre



Fonte: Autoria Própria

Já no início do novo semestre o planejamento foi revisto para ver quais eram as tarefas pendentes. Tudo que estava em pendencia foi priorizado para ser feito rapidamente e evitar mais atrasos.

2.2.4.1 População do Banco de Dados e Desenvolvimento de Iterações

Os DBAs trabalharam então no dicionário de dados e também na população do banco. Ou seja, eles tiveram que fazer um documento explicando o que cada um dos itens significava e também inserir no banco algumas coisas já especificadas, como por exemplo alguns funcionários, cidades, etc. Tudo isso foi necessário para que o software pudesse ser executado da forma correta, como no módulo 08, onde era necessário ter-se uma lista dos funcionários da instituição, e, para isso, essas informações deveriam já estar no Banco de Dados possibilitando que elas fossem chamadas na página HTML.

Na imagem abaixo foi modificado o primeiro insert para melhor compreensão de como os dados estavam inseridos, os demais continuam da mesma forma original.

Figura 28: Exemplo de *insert* - Funcionários

```
-- TABELA: USUARIOS
INSERT INTO `geracoes`.`usuarios` (`USU_NOME`,`USU_CPF`,`USU_PRONTUARIO`,`USU_FUNCAO`,)
VALUES ('Yuri Ian Benjamin Pires','7546859', '38381342570','Enfermeiro');
INSERT INTO `geracoes`.`usuarios` (`USU_RG`,`USU_ENDERECO`,`USU_DATA_NASCIMENTO`,`USU_CEP`,`USU_BAIRRO`
INSERT INTO `geracoes`.`usuarios` (`USU_RG`,`USU_ENDERECO`,`USU_DATA_NASCIMENTO`,`USU_CEP`,`USU_BAIRRO`
INSERT INTO `geracoes`.`usuarios` (`USU_RG`,`USU_ENDERECO`,`USU_DATA_NASCIMENTO`,`USU_CEP`,`USU_BAIRRO`
INSERT INTO `geracoes`.`usuarios` (`USU_RG`,`USU_ENDERECO`,`USU_DATA_NASCIMENTO`,`USU_CEP`,`USU_BAIRRO`
INSERT INTO `geracoes`.`usuarios` (`USU_RG`,`USU_ENDERECO`,`USU_DATA_NASCIMENTO`,`USU_CEP`,`USU_BAIRRO`
INSERT INTO `geracoes`.`usuarios` (`USU_RG`,`USU_ENDERECO`,`USU_DATA_NASCIMENTO`,`USU_CEP`,`USU_BAIRRO`
```

Fonte: Autoria Própria

Como previsto no planejamento, foi feito a entrega da segunda iteração do Módulo 08, esta deveria possibilitar ao usuário uma lista das tarefas de limpeza do local, sendo possível editá-las ou excluí-las.

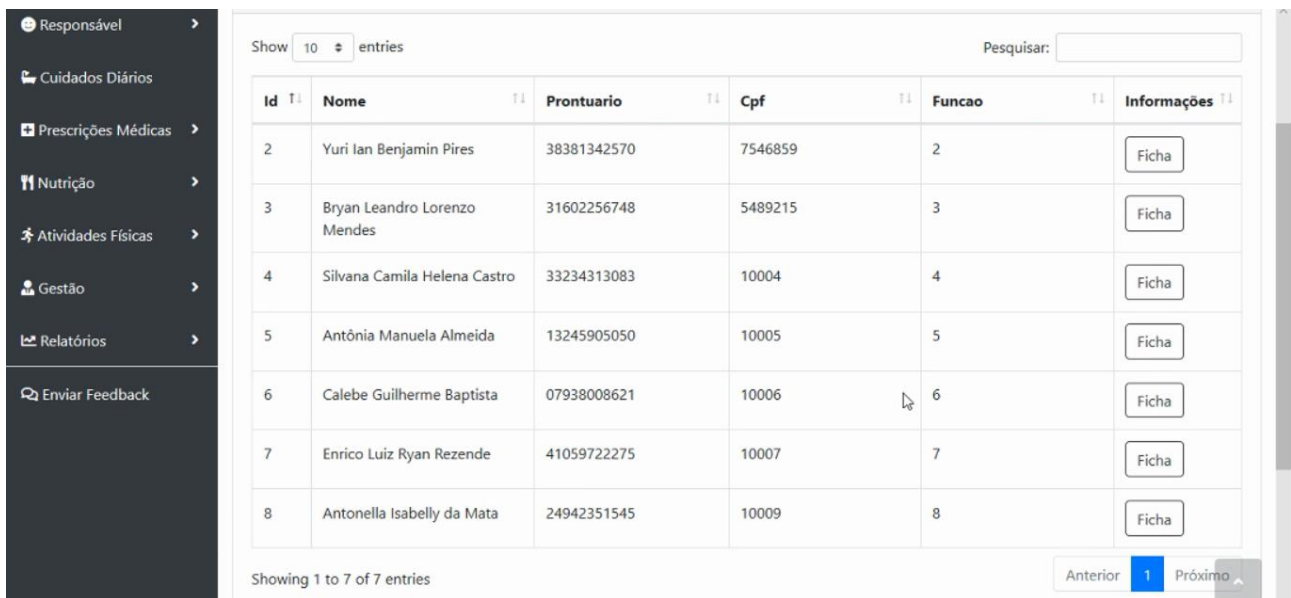
Figura 29: Iteração 02 - Lista de Tarefas

Identificação	Nome	Setor	Responsável	Descrições	Observação	Alterar	Excluir
29	Limpar quatro	5	7	cozinha	banheiro	EDITAR	EXCLUIR
30	limpar janelas	1	2	qwe	ewq	EDITAR	EXCLUIR

Fonte: Autoria Própria

Ele também preve que se tenha uma lista de funcionários, podendo acessar a ficha de cada um, a fim de se obter mais informações daquele funcionário. Foi escolhido mostrar apenas uma parte da Iteração para melhor compreensão. Vale notar que abaixo estão aparecendo as informações que já tinham sido inseridas no banco, como mostra a Figura 28.

Figura 30: Iteração 02 - Lista de Funcionários



Id	Nome	Prontuario	Cpf	Funcao	Informações
2	Yuri Ian Benjamin Pires	38381342570	7546859	2	Ficha
3	Bryan Leandro Lorenzo Mendes	31602256748	5489215	3	Ficha
4	Silvana Camila Helena Castro	33234313083	10004	4	Ficha
5	Antônia Manuela Almeida	13245905050	10005	5	Ficha
6	Calebe Guilherme Baptista	07938008621	10006	6	Ficha
7	Enrico Luiz Ryan Rezende	41059722275	10007	7	Ficha
8	Antonella Isabelly da Mata	24942351545	10009	8	Ficha

Fonte: Autoria Própria

Também foi iniciado o desenvolvimento da iteração 03, responsável pela manutenção do fluxo de caixa da instituição. Porém, por ainda não estar devidamente completa e testada, ela não será apresentada aqui neste documento.

2.2.5 Quarto Bimestre

Figura 31: Cronograma 4º Bimestre



Fonte: Autoria Própria

Para o quarto bimestre foi programado a entrega da Iteração 04. Considerando que cerca de 90% do que era para ser entregue até o bimestre anterior havia sido feito, espera-se que seja entregue mais que 80% do projeto (porcentagem mínima do total finalizado do projeto a qual o grupo ficou comprometido de entregar no Termo de Abertura), representando então um grande sucesso.

Futuramente será feita então uma integração de todos os módulos, um manual do usuário visando explicar como o software funciona de forma simples, um portal de documentações contendo todas as documentações necessárias e um manual de implantação. Tudo isso para auxiliar os futuros profissionais que trabalharão com a manutenção do produto e também seus utilizadores.

3 Conclusões e Recomendações

Esta pesquisa descreveu as etapas de um trabalho extenso que contou com a participação de 58 pessoas para seu desenvolvimento. Este projeto, que se mostrou de extrema importância para a comunidade local e também para as cidades vizinhas, teve como foco a criação de um software para Instituições de Longa Permanência, buscando melhorar a qualidade de vida dos idosos residentes, facilitando o controle dos funcionários sobre cada um deles, verificando as atividades realizadas, a alimentação, as restrições médicas etc; além de auxiliar na aproximação da família com seu ente querido, essa que muitas vezes não sabe como está sendo a vida dentro destas instituições.

Além disso, em qualquer instituição é notória a necessidade de um controle administrativo, este que controle todo o fluxo de caixa, todas as receitas e todas as despesas; que permita o acesso à uma lista com todos os funcionários, suas funções, seu salário etc. Tudo isso buscando da melhor forma auxiliar o gestor do local, facilitando seu trabalho.

Consequentemente, por trás de todo grande projeto bem-sucedido existe um planejamento complexo. Por isso, a pesquisa presente neste documento teve como objetivo abordar o planejamento do projeto e a influência que ele teve durante todo o processo de desenvolvimento. Para isto, ele foi dividido em três objetivos principais.

A primeira etapa se deu com uma breve apresentação do cronograma geral do projeto, este criado pelos professores orientadores do Projeto Gerações, mostrando que, além de auxiliar os envolvidos no projeto, a fim de se ter uma lista de afazeres, ele foi criado para que se tivesse um controle das atividades propostas. Ainda, foram apresentadas as ferramentas utilizadas para o controle dessas atividades, isso se mostrou de grande importância para se ter uma noção do progresso de cada item.

A segunda etapa se deu com o detalhamento do que foi realizado em cada um dos bimestres, tendo-se por base o cronograma geral, buscando mostrar sua importância. Primeiramente houve a divisão dos grupos em módulos, cada um com seus macro requisitos. Essa foi a primeira etapa, possibilitando a identificação e criação de outras informações essenciais, tais como quais os *stakeholders*, quais os requisitos funcionais e não funcionais, o desenvolvimento do Diagrama de Caso de Uso, etc. Ou seja, tudo está interligado, seria impossível iniciar o projeto de qualquer outro ponto, pois isso acarretaria em falhas. Outrossim, também foi mostrado como o projeto sempre foi voltado as necessidades do cliente, buscando saber de suas necessidades e sempre mantendo total transparência com o mesmo.

A terceira etapa se deu com a amostra e explicação do planejamento dos casos de uso do Módulo 08. Após se ter identificado todos os requisitos funcionais, que foram 11 no total, eles foram transformados em casos de uso, totalizando 29, estes foram agrupados de acordo com suas relações, surgindo-se então as Iterações. Para se ter uma data de entrega das mesmas, foi feita uma priorização, utilizando-se o Método Kano, buscando analisar quais os itens mais essenciais para o software, que não poderiam de jeito nenhum não estarem presentes no seu resultado final. Para cada item das iterações um grupo de 3 alunos ficou responsável, para que se tivesse o acompanhamento do desenvolvimento, do Banco de Dados, e dos Casos de Teste. O último, de extrema importância para que se tenha um acompanhamento do que será entregue ao cliente no produto final. Após terminado o desenvolvimento, todos os casos de uso passam por um teste, este baseado no que o sistema deve oferecer e nas possíveis falhas que ele poderia apresentar, caso esteja tudo correto, ele deveria ser considerado como sucesso, caso não, ele seria considerado como falha e deveria ser refeito.

Analisando os resultados, é possível observar um grande sucesso. Até o terceiro bimestre cerca de 80% do Módulo 08 foi desenvolvido e testado, esse número passa para 90% quando se considera o projeto como um todo. Isso se deu não apenas pelo planejamento bem elaborado como também pela responsabilidade de cada um dos integrantes do projeto e seu árduo trabalho em equipe, onde buscaram sempre saber o que, quando e como deveria ser feito, além de auxiliar os colegas que precisavam de ajuda.

É notório também que houveram falhas e atrasos durante o decorrer do projeto, algumas atividades foram mal executadas, como não seguir as definições presentes no documento de Casos de Uso e apresentar Iterações incompletas; produzir incorretamente um documento e ter que refazê-lo; não poder programar pois o Banco de Dados apresentava falhas; fazer a entrega de um item depois do prazo de entrega, etc. Porém, esse documento buscou mostrar o outro lado de um planejamento, não apenas uma lista de afazeres mas também um meio de controle total do projeto, que funciona para auxiliar os envolvidos e também para discipliná-los; além de os deixar cientes de que falhas ocorrem e é necessário ter-se um plano pronto para poder lidar com elas.

Para futuros projetos é válido que se tenha uma mudança no cronograma geral, priorizando algumas atividades que levam mais tempo que outras, dando mais tempo para elas serem realizadas. Além disso, atividades que servirão de base para outras devem ocorrer com antecedência. Houveram muitos erros no Banco de Dados, o que acarretou em atrasos nas Iterações, pois os desenvolvedores foram incapazes de fazer seu trabalho enquanto o erro não fosse solucionado, por isso, seria interessante que o desenvolvimento do banco se iniciasse com mais rapidez. Outrossim, é necessário que haja, por parte de cada Módulo, uma melhor divisão no planejamento de Casos de

Uso, as tarefas devem ser analisadas, deve-se anotar tudo que será necessário se fazer e analisar o grau de dificuldade de cada uma, a fim de priorizar o desenvolvimento das mais necessárias ao sistema, porém, iniciando-se pelas mais fáceis; além de sempre estar adiantado com o trabalho, em caso de conclusão rápida de uma tarefa, o início da próxima é totalmente válido, mesmo que sua data de entrega ainda esteja muito longe. Por final, é interessante que se tenha uma maior interação entre os integrantes, para que todos saibam no que cada um está trabalhando, quais são as dificuldades (se existirem) e qual o andamento do projeto, a fim de se ter uma melhor comunicação e ser possível que o grupo se ajude com mais facilidade.

Em síntese, a pesquisa contida nesse documento foi considerada satisfatória, visto que seu objetivo principal foi alcançado ao longo de seu desenvolvimento. Também, foi de extrema importância a apresentação de pontos negativos e positivos, a fim de se ter resultados cada vez mais satisfatórios no futuro.

4 Referências Bibliográficas

- [1] IBGE. **Brasil em Síntese:** São João da Boa Vista. Disponível em: <<https://cidades.ibge.gov.br/brasil/sp/sao-joao-da-boa-vista/panorama>>. Acesso em 16 ago. 2019
- [2] G1. **Pesquisa aponta São João da Boa Vista como melhor cidade para idosos.** Disponível em: <<http://g1.globo.com/sp/sao-carlos-regiao/noticia/2017/03/pesquisa-aponta-sao-joao-da-boa-vista-como-melhor-cidade-para-idosos.html>>. Acesso em 16 ago. 2019
- [3] LISI ROMANO, Breno. **Portal Acadêmico:** Prática e Desenvolvimento de Sistemas (PDS). Disponível em: <<https://sites.google.com/site/blromano/disciplinas/pds2014>>. Acesso em 24 ago. 2019
- [4] ANVISA. **Resolução de Diretoria Colegiada.** Disponível em: <http://portal.anvisa.gov.br/documents/10181/2718376/RDC_283_2005_COMP.pdf/a38f2055-c23a-4eca-94ed-76fa43acb1df>. Acesso em 24 ago. 2019
- [5] SUMMERVILLE, Ian. **Engenharia de Software.** 9. ed. São Paulo: Person, 2011.
- [6] MESQUITA, Rafael Vargas. **Análise e Projeto de Sistemas.** Disponível em: <<ftp://ftp.ci.ifes.edu.br/informatica/daniel/APS/IFES-APS-Apostila.pdf>>. Acesso em 25 set. 2019
- [7] BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. **UML:** Guia do Usuário. 2. ed. Rio de Janeiro: Campus, 2006.
- [8] Sperandio, Dircelene Jussara; Martinez Évora, Yolanda Dora. **Planejamento da assistência de enfermagem:** proposta de um software-protótipo. Disponível em: <<https://www.redalyc.org/pdf/2814/281421850004.pdf>>. Acesso em: 25 set. 2019
- [9] PRESCOTT, Preston. **HTML5.** Disponível em: <<https://books.google.com.br/books?id=5OyNCgAAQBAJ&printsec=frontcover&dq=html&hl=pt-BR&sa=X&ved=0ahUKewjF5t-Uw-zkAhVYDrkGHZ-4CsQQ6AEIKTAA#v=onepage&q=html&f=false>>. Acesso em 25 set. 2019

- [10] COSTA, Wilker. **O que são tags HTML?**. Disponível em: <<https://www.cursos.wlconsultoria.net/blog/o-que-sao-tags-html/>>. Acesso em 25 set. 2019
- [11] RODRIGUES, Joel. **HTML básico:** códigos HTML. Disponível em: <<https://www.devmedia.com.br/html-basico-codigos-html/16596>>. Acesso em 25 set. 2019
- [12] GILMORE, W. Jason. **Dominando PHP e MySQL:** Do Iniciante ao Profissional. 3. ed. Rio de Janeiro: Alta Books, 2008
- [13] **PHP:** Sua primeira página PHP. Disponível em: <https://www.php.net/manual/pt_BR/tutorial.firstpage.php#example-2>. Acesso em 25 set. 2019
- [14] SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML.** São Paulo: Novatec, 2007
- [15] ARRIGONI, Ricardo. **Código CSS:** entendendo a folha de estilos. Disponível em: <<https://www.devmedia.com.br/codigo-css-entendendo-a-folha-de-estilos/37459>>. Acesso em 25 set. 2019
- [16] GONÇALVES, Edson. **Dominando NetBeans:** Construa aplicativos Java tanto para Desktop, como para Web, trabalhando com uma das ferramentas mais usadas no mundo. Rio de Janeiro: Ciência Moderna, 2006
- [17] REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação.** 3 ed. Rio de Janeiro: Brasport, 2005
- [18] NAKAMURA, Emilio Tissato; DE GEUS, Paulo Lício. **Segurança de Redes em Ambientes Cooperativos.** São Paulo: Novatec, 2007
- [19] BRASILEIRO, Roberto. **O que é Kanban? E porque ele não é apenas um quadro de tarefas?**. Disponível em: <<http://www.metodoagil.com/o-que-e-kanban/>>. Acesso em 25 set. 2019

[20] DATE, C.J. **Introdução a sistemas de banco de dados**. 8. ed. Rio de Janeiro: Campus, 2004

[21] MILANI, André. **MYSQL: Guia do programador**. São Paulo: Novatec, 2006

[22] BERNARDO, Kleber. **Iterativo e incremental: Suas definições**. Disponível em: <<https://www.culturaagil.com.br/iterativo-e-incremental-suas-definicoes/>>. Acesso em 25 set. 2019