

**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 IMPORTÂNCIA DO ANALISTA DE SISTEMAS PARA O  
PROJETO GERAÇÕES**

Thalyson da Costa Teixeira

BV1520725

São João da Boa Vista – SP

2019

## Resumo

O *software* Gerações foi criado pelas cerca de 60 pessoas, em conjunto com os professores da matéria de PDS dos 4º anos vespertinos, para que a cidade de São João da Boa Vista se mantivesse no *ranking* 1 como a cidade, contendo entre 50 mil e 100 mil pessoas, que melhor oferece qualidade de vida para idosos. Porém, para explicar a sua montagem, será necessário explicar a utilidade de um analista de sistemas comum e, apesar disso, como ele aplica o UML (*Unified Modeling Language*) e o processo de desenvolvimento de *software* (que abrange o processo de levantamento de requisitos, estudo detalhado e testes de *software*) em *softwares* genéricos em vez de especificamente no *software* Gerações. E assim que essa parte do objetivo geral for explicada, será mostrado, logo em seguida, de uma forma breve, a sua outra parte que é como um analista específico aplica o UML e faz o processo de desenvolvimento (referido como levantamento de requisitos) para o exato *software* Gerações.

**Palavras-chave:** Analista de sistemas, UML, processo de desenvolvimento de *software*, *software* Gerações.

## Lista de Figuras

Figura 1: Divisões do projeto.....	6
Figura 2: Os 14 tipos de diagrama do UML versão 2.4.1 .....	9
Figura 3: Explicação dos elementos de um diagrama de casos de uso .....	10
Figura 4: Explicação dos elementos de um diagrama de casos de uso 2 .....	11
Figura 5: Explicação dos elementos de um diagrama de casos de uso 3 .....	12
Figura 6: Ambiguidades em Análise de Sistemas.....	16
Figura 7: Diagramas de casos de uso do módulo 1 (referentes à tabela 5) .....	57
Figura 8: Diagrama de casos de uso do módulo 2 (referentes à tabela 6).....	58
Figura 9: Diagrama de casos de uso do módulo 3 (referentes à tabela 7).....	59
Figura 10: Diagrama de casos de uso do módulo 4 (referentes à tabela 8).....	60
Figura 11: Diagrama de casos de uso do módulo 5 (referentes à tabela 9).....	61
Figura 12: Diagrama de casos de uso do módulo 6 (referentes à tabela 10).....	62
Figura 13: Diagrama de casos de uso do módulo 7 (referentes à tabela 11).....	63
Figura 14: Diagrama de casos de uso do módulo 8 (referentes à tabela 12).....	64
Figura 15: Diagrama de casos de uso do módulo 9 (referentes à tabela 13).....	65

## Lista de Tabelas

Tabela 1: Amostra de um documento de requisitos .....	14
Tabela 2: Exemplo do requisito $a$ desenvolvido e pronto para testes .....	14
Tabela 3: Acompanhamento da execução dos testes .....	17
Tabela 4: Histórico de execução de testes .....	18
Tabela 5: Documento de requisitos do módulo 1 .....	19
Tabela 6: Documento de requisitos do módulo 2 .....	24
Tabela 7: Documento de requisitos do módulo 3 .....	27
Tabela 8: Documento de requisitos do módulo 4 .....	30
Tabela 9: Documento de requisitos do módulo 5 .....	33
Tabela 10: Documento de requisitos do módulo 6 .....	41
Tabela 11: Documento de requisitos do módulo 7 .....	43
Tabela 12: Documento de requisitos do módulo 8 .....	48
Tabela 13: Documento de requisitos do módulo 9 .....	53

## Sumário

1	Introdução .....	6
1.1	Contextualização .....	6
1.2	Objetivo Geral .....	7
1.3	Objetivos Específicos (refazer) .....	7
1.4	Fazer após o desenvolvimento estiver completo ..... <b>Erro! Indicador não definido.</b>	
2	Desenvolvimento .....	9
2.1	O que são Analistas de Sistemas e o que eles fazem? .....	9
2.1.1	Qual é a utilidade do UML? .....	9
2.1.1.1	A utilidade dos diagramas de casos de uso no diagrama comportamental..	10
2.1.2	Processo de desenvolvimento de <i>software</i> .....	13
2.1.2.1	Processo de levantamento de requisitos .....	13
2.1.2.2	Explicando o Estudo Detalhado .....	15
2.1.2.3	Testes de <i>software</i> .....	17
3	Foco dos Analistas no Projeto Gerações.....	19
3.1	Levantamento de requisitos do projeto Gerações.....	19
3.2	UML e diagramas de caso de uso do projeto Gerações .....	56
4	Conclusão.....	67
5	Referências Bibliográficas .....	69

# 1 Introdução

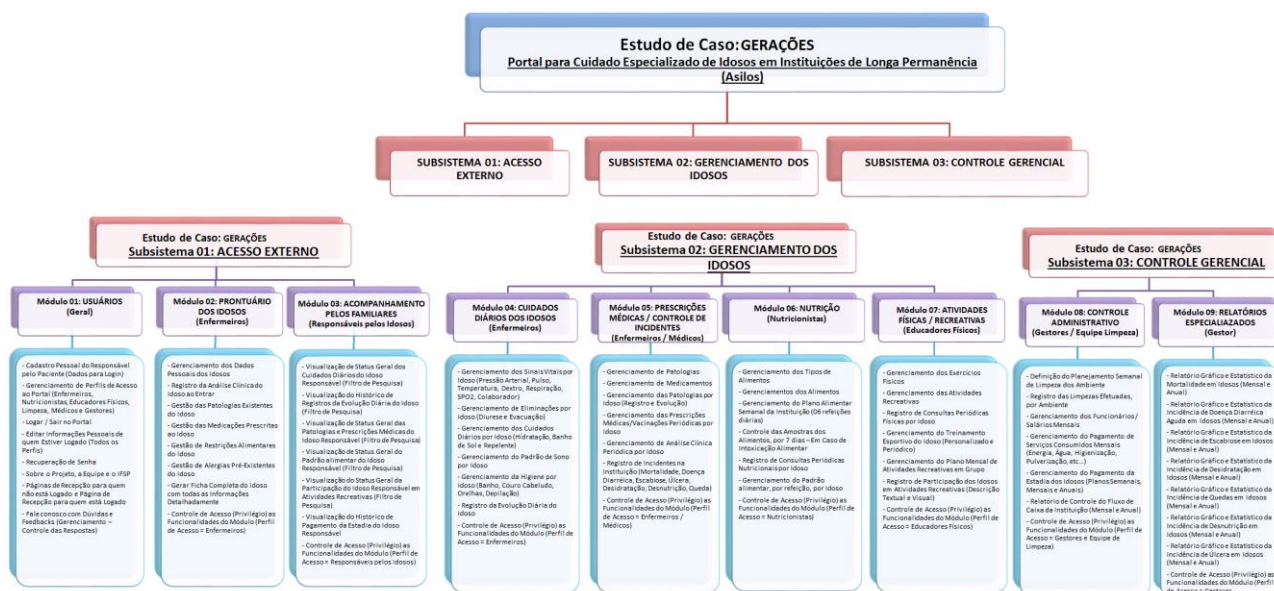
## 1.1 Contextualização

São João da Boa Vista, de acordo com a instituição de ensino superior Fundação Getúlio Vargas, é considerada a cidade contendo entre 50 mil e 100 mil habitantes que melhor oferece qualidade de vida para idosos do Brasil,<sup>[1]</sup> e para manter a sua classificação, os alunos, em conjunto com os professores Breno Romano e Luiz Francisco, do Instituto Federal de Educação, Ciência e Tecnologia local, através da sua disciplina escolar de PDS (Prática de Desenvolvimento de Sistemas) do 4º ano de conclusão do curso Técnico Integrado em Informática ao Ensino Médio, tiveram como finalidade o planejamento e a criação do *software* do Projeto Gerações®, para que este, de uma guisa sucessiva, oferecesse um serviço institucional social de longa permanência para pessoas necessitantes de cuidados especiais.<sup>[2]</sup>

Com a assistência de ambos os professores, o projeto fora melhor organizado pelos alunos concluintes do 4º ano que, por sua vez, definiram-no, juntos, em diversos subsistemas e, então, módulos contendo descrições para os quais cada um foi selecionado, assim formando 9 grupos de cerca de 6 membros cada.

Para mais detalhes e contextualização sobre a organização desse projeto, tem-se a imagem abaixo como fonte de informação<sup>[2]</sup>:

Figura 1: Divisões do projeto



Fonte: <https://sites.google.com/site/blromano/disciplinas/pds2014>

É importante indicar, também, que se tinham 3 distintas funções pelas quais, em média, 2 pessoas de cada módulo foram responsáveis; as funções foram as seguintes: desenvolvedor — aquele que foi responsável pela compilação dos códigos —, DBA — o que administrava o banco de dados —, e, por fim, o analista de sistemas — que foi o encarregado de verificar a funcionalidade dos códigos analisados de uma forma mais aplicada e acurada. Esta última função será o principal assunto deste Trabalho de Conclusão de Curso (TFC), ou seja, o desenvolvedor e o DBA serão mencionados poucas vezes.

## **1.2 Objetivo Geral**

Mostrar, através deste vigente TFC, o quão importante é um Analista de Sistemas para o auxílio e vantagem do desenvolvimento e funcionamento de um projeto qualquer e o Projeto Gerações específico, o qual foi proposto pelo Instituto Federal de São João da Boa Vista em prol da qualidade de vida de idosos.

## **1.3 Objetivos Específicos**

Expor o que é um Analista de Sistemas e como ele pode ter uma grande influência e utilidade;

Explanar o uso do UML e a utilidade de seus diagramas de casos de uso;

Explicar o processo de desenvolvimento de *software* e sua parte de processo de levantamento de requisitos, estudo detalhado e testes de *software*;

Detalhar o levantamento real de requisitos no projeto Gerações bem como suas etapas;

Mostrar o uso real do UML e diagramas de caso de uso no projeto Gerações de uma maneira sucinta.

## **1.4 Estrutura do Documento**

O capítulo 1 se trata da contextualização do projeto Gerações no qual é citado o motivo de ele ter sido criado em 2019 pelos professores e alunos bem como o objetivo geral dele que mostra que o analista de sistemas é importante para o projeto. E além disso, ele, outrossim, se trata dos objetivos específicos os quais mencionam o que será feito.

O capítulo seguinte é sobre um exemplo de um desenvolvimento de um software qualquer no qual abrange as partes em que se explanam o que são analistas bem como o que eles fazem; também abrange a utilidade do UML e seus diagramas de casos de uso e, por fim, o processo de desenvolvimento de software (e seu processo de levantamento de requisitos, estudo detalhado e testes de *software*).

O capítulo 3 é sobre o desenvolvimento do *software* Gerações específico no qual se seguem todos os passos usados no capítulo anterior.



## 2 Desenvolvimento

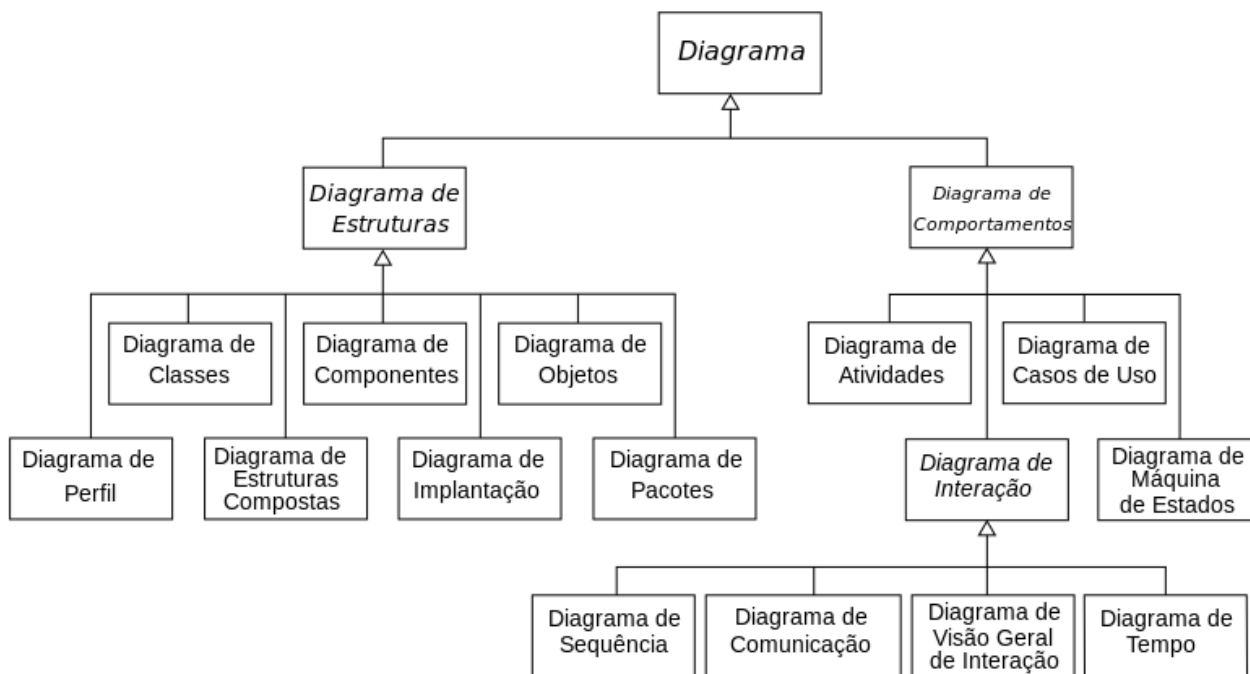
### 2.1 O que são analistas de sistemas e o que eles fazem?

Os analistas de sistemas são os componentes humanos que exercem o seu papel na área de Tecnologia da Informação, conhecida também pela abreviatura TI<sup>[2]</sup>; eles costumam desempenhar seus papéis com os desenvolvedores de sistemas, que são conhecidos como programadores. E, juntos, em equipe, são capazes de incorporar projetos. Entretanto, para isso, é necessário que sigam e apliquem-lhes o UML bem como realizem o processo de desenvolvimento de *software* (ver 2.1.2).

#### 2.1.1 Qual é a utilidade do UML?

O UML (*Unified Modeling Language*, do inglês, Linguagem de modelagem unificada), que é tido como aprovado pelo OMG, é uma linguagem que propõe às equipes da TI um modelo hierárquico como forma de organização contendo 7 diagramas da categoria estrutural e 7 da categoria comportamental como na imagem abaixo:

Figura 2: Os 14 tipos de diagrama do UML versão 2.4.1



Adaptado de: <https://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>

Entretanto, não é necessário que todos os 14 tipos de diagramas sejam usados dependendo da complexidade e da exigência do sistema ou *software* que está sendo construído ou planejado pela equipe. O *software* do projeto Gerações, por exemplo, usa apenas os diagramas de casos de uso, que

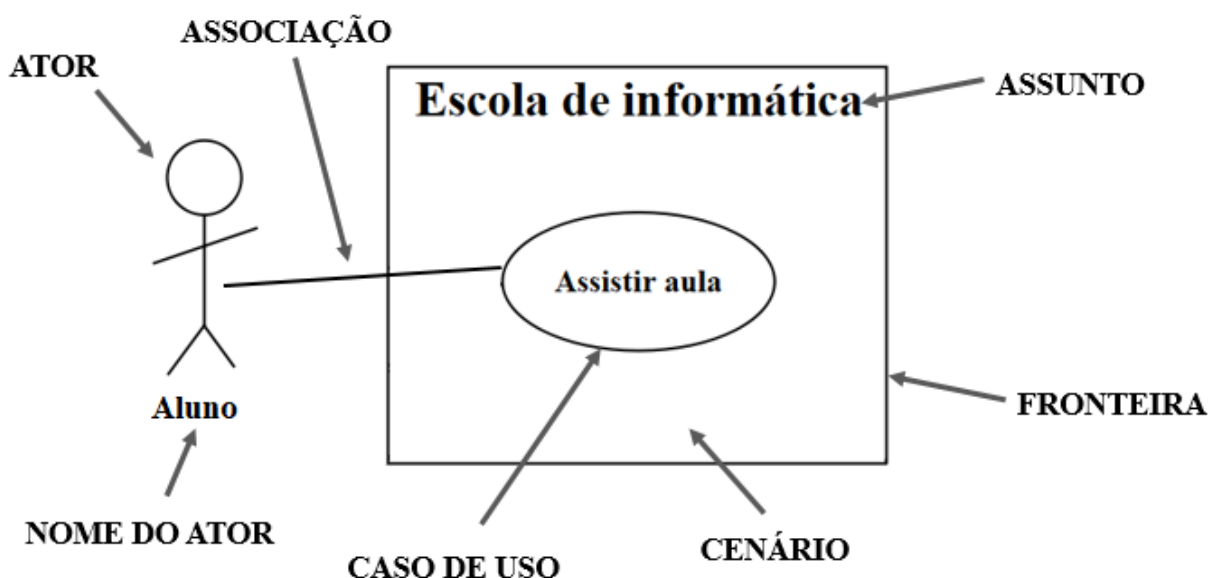
está contido em diagrama de comportamento como na figura 2, página 7; porém, de uma forma competente e eficaz, para que possa evitar problemas de desenvolvimento que, às vezes, são notados tardiamente. Ele, além dos diagramas, propõe 5 tipos de visões interdependentes como a visão de casos de uso, visão de projeto, visão de implementação e visão de processo.

### 2.1.1.1 A utilidade dos diagramas de casos de uso no diagrama comportamental

Por estarem abrangidos no diagrama comportamental, que é chamado também de diagrama de comportamentos, os diagramas de caso de uso, pelo menos, possuem algum tipo de alteração comportamental na classe; contudo, o que os difere de outros diagramas é eles terem, como utilidade e função, o papel de expor a funcionalidade apresentada a um sistema que, por fim, será planejado; ademais, o levantamento dos requisitos do sistema lhe é um importantíssimo instrumento (ver 2.1.2.1), tanto que os casos de uso podem ser montados através dele (ver 3.1 e 3.2). Em relação a isso, o cientista da computação sueco chamado Ivar Hjalmar Jacobson afirma que um caso de uso é, de fato, um documento usado para descrever a ordem de acontecimentos de eventos de um determinado ator que utiliza o sistema escolhido a fim de completar seu processo. Em poucas palavras, um caso de uso, por exemplo, retrata a interação que há entre o próprio sistema e um usuário, sendo este humano, máquina ou algum outro elemento, de um modo mais organizado.<sup>[4][5]</sup>

Para que isso fique mais fácil de ser compreendido, eis a figura a seguir que melhor fornece um exemplo de um diagrama de casos de uso em utilização:

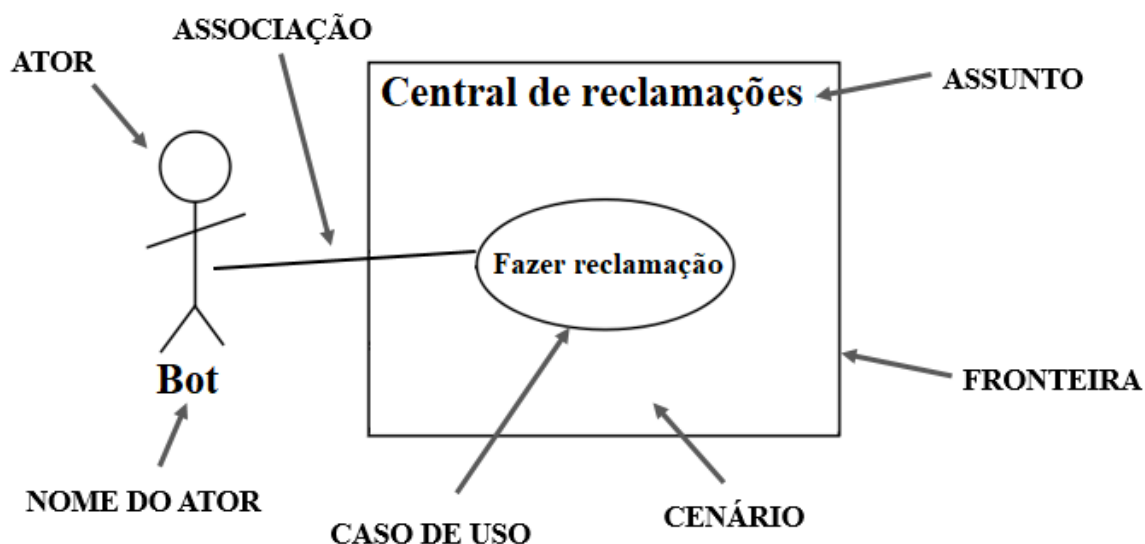
Figura 3: Explicação dos elementos de um diagrama de casos de uso



Fonte: Elaboração própria

Analisando-a mais afundo, percebe-se que um diagrama de casos de uso **comum** como esse deve conter uma fronteira que engloba um cenário — ou um espaço — no qual se podem modelar (ou criar) vários casos de uso, como o “Assistir aula”, e definir um determinado assunto (do sistema), como “Escola de informática”. É importante saber que nem todos os analistas dão importância à fronteira, as deixando, então, de lado. Ademais, aprende-se, outrossim, que deve haver, pelo menos, um ator como na figura que é chamado de “Aluno” — um humano — que está conectado ao caso de uso através do relacionamento de associação; poderia ser uma máquina em vez de um aluno humano como na figura abaixo:

Figura 4: Explicação dos elementos de um diagrama de casos de uso 2



Fonte: Elaboração própria

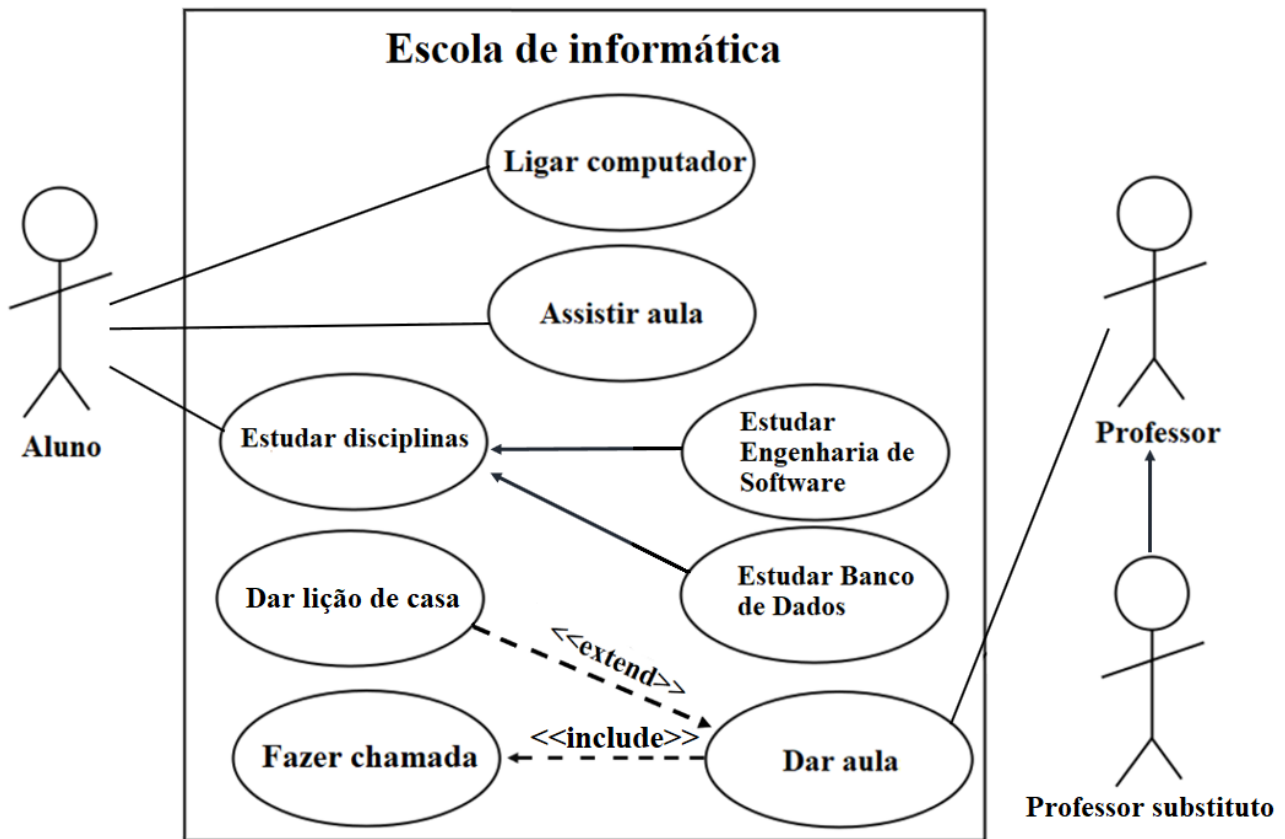
A diferença sua é que o seu assunto passou a ser “Central de reclamações”, o seu ator, uma máquina chamada “Bot”; e, por fim, o seu caso de uso que se tornou “Fazer reclamação”, ou seja, o diagrama da figura 3 simplesmente indica que um aluno, em uma escola de informática, assiste às aulas enquanto o diagrama da figura 4, um robô que faz reclamações em uma central de reclamações.

Para que se possa explorar ainda mais esse assunto, deve-se ter em mente que esse tipo de diagrama, além de conter relacionamento de associação, pode conter, outrossim, outros tipos de relacionamento dos quais: relacionamento de generalização, relacionamento de inclusão (representado por “<<include>>”) e relacionamento de extensão (representado por “<<extend>>”). O relacionamento de generalização é usado para indicar que tal ator filho pode exercer as mesmas atividades que o ator pai, já o de inclusão, para indicar que tal comportamento de caso de uso

sempre inclui o comportamento de um outro caso de uso, e é muitas vezes relacionando com a palavra “obrigação”, dado que se tal comportamento ocorre, o outro deve ocorrer também. Sobrou agora o relacionamento de extensão; ele é usado para indicar que tal comportamento ocorre de uma maneira opcional ou sob determinadas condições ou cenários.

A figura abaixo explica tudo isso de uma forma ainda mais clara e específica:

Figura 5: Explicação dos elementos de um diagrama de casos de uso 3



Fonte: Elaboração própria

Pode-se ver que o assunto do sistema desse diagrama de casos de uso é chamado de “Escola de informática”; nele há três atores dos quais: “Aluno”, “Professor” e “Professor substituto”. Apesar disso, há-lhe, agora, novos elementos como <<extend>> e <<include>> tal como a associação de relacionamento de generalização que existe entre ambos funcionários, sendo o “Professor” o modelo de elemento pai, e o “Professor substituto”, o filho; o mesmo serve para a associação de relacionamento de generalização que está presente entre os casos de uso como o “Estudar disciplinas”, que é o pai de “Estudar Engenharia de Software” e “Estudar Banco de Dados”, sendo estes últimos, então, filhos. Explicando esses casos de uso, têm-se que: o “Aluno” liga o computador, assiste aula e estuda disciplinas das quais: Engenharia de Software e Banco de

Dados; enquanto o “Professor” dá aula e, obrigatoriamente, faz chamada (devido ao <<include>>), e ele, opcionalmente, pode dar lição de casa (devido ao <<extend>>), e o “Professor substituto” pode fazer tudo que esse professor faz. Essas ações se dão em uma escola de informática.<sup>[6][7][8][9]</sup> Diagramas como esse são essenciais para que todos, principalmente os desenvolvedores, possam, de uma guisa mais rápida, entender como um determinado sistema funciona.

### **2.2.1 Processo de desenvolvimento de *software***

O processo de desenvolvimento de *software* tem como virtude catalogar as tarefas, que são realizadas em meio ao tempo da preparação de um sistema de *software*, em apenas tarefas. E sob outra perspectiva, não se há nenhum tipo de processo ideal que se sobressaia muito bem em todos os tipos de ocasiões de desenvolvimento, e isso exige um reconhecimento e conformidade por parte do corpo social responsável por ele, dado que cada parte desse processo tem suas próprias características de realizar as tarefas. Esse processo representa os outros processos dos quais: processo de levantamento de requisitos (que pertence à etapa de compreensão) (ver 2.1.2.1), o estudo detalhado (que pertence à etapa de análise) (ver 2.1.2.2) e testes.

#### **2.2.1.1 Processo de levantamento de requisitos**

Os analistas também têm como suas principais funções e deveres iniciais, a realização do estudo detalhado (ver 2.1.2.2) ao longo da fase de análise sobre os requisitos de um projeto que são, previamente, levantados pelos seus desenvolvedores em conjunto com os clientes, que têm como propósito ter uma percepção igual sobre o problema criado, para que este possa ser resolvido de uma maneira eficiente e fácil. Essa parte de requisitos, anterior à etapa análise, é denominada como levantamento de requisitos ou, às vezes, elicitación de requisitos; ele é pertencente à etapa de compreensão do problema aplicada ao desenvolvimento de *software*. Nessa etapa, em geral, além de exigir que se tenha um diálogo menos técnico e complexo com os clientes pelo bem da comunicação e compreensão deles, a equipe de desenvolvimento, vide os desenvolvedores do projeto, deve criar um documento de requisitos, contendo 3 possíveis tipos de requisitos (requisitos funcionais, requisitos não-funcionais e, por fim, requisitos normativos) pela razão de terem de se preocupar com isso tudo e a organização; essa atitude evita-lhes falhas na hora ou após o desenvolvimento bem como problemas legislativos.

Ademais, é necessário também que os analistas e até mesmo os administradores de banco de dados ajudem a equipe desenvolvedora a estabelecer esses requisitos de uma forma melhor, mais analítica e exata em vez de somente técnica como ela própria faria.<sup>[3][4]</sup>

Tendo tudo isso como informação, a tabela a seguir representa uma amostra de um documento de requisitos no qual requisitos, que foram levantados e incorporados, descrevem o que o sistema deve fazer:

Tabela 1: Amostra de um documento de requisitos

Identificadores	Exemplo de uso de requisitos*
a	O sistema deve permitir que cada professor realize o lançamento de notas das turmas nas quais lecionou
b	O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas em um determinado período
c	A página de <i>login</i> do diretor pode ficar aberta no máximo 1 hora
d	A página de advertência do diretor deve estar disponível durante 99% do tempo” e nela deve conter: <ul style="list-style-type: none"> <li>• Campos para preencher o nome dos envolvidos;</li> <li>• Formulário para selecionar o motivo da advertência;</li> <li>• Botão “confirmar.</li> </ul>
e	O sistema deve fazer rollback caso informações inseridas nele sejam alteradas ou deletadas
f	O sistema, de um modo confidencial, deve armazenar as informações dos clientes respeitando as leis

Fonte: Elaboração própria

(\*Nota: Nessa tabela, há noção de que os requisitos (identificadores) *a* e *b* são funcionais porque, simplesmente, definem a funcionalidade do sistema enquanto os requisitos *c*, *d* e *e* são requisitos não-funcionais, dado que são usados para declarar qualidades que o sistema deve oferecer em torno de suas funcionalidades. No caso do *g*, se trata de um requisito normativo pelo motivo de ser usado para impor restrições e regras no sistema; há analistas que o considera como requisito funcional ou não-funcional em vez de normativo. E devido a isso, é possível identificar quais requisitos são funcionais ou não.)

Com base no requisito *a* da tabela 1, pode-se, outrossim, montar um documento de requisitos, como abaixo, mais completo e complexo, o qual é usado durante o teste de *software* (ver 2.1.2.3):

Tabela 2: Exemplo do requisito *a* desenvolvido e pronto para testes

Casos de Teste	Descrição do caso de teste	Resultado esperado
CT a1	Verifica se o usuário (professor) realiza, sob um sistema, o lançamento de notas das turmas nas	O sistema verifica se o usuário realizou o lançamento de notas e então mostra uma mensagem como:

	quais lecionou	“Lançamento de notas das turmas pronto”; caso contrário, aparecerá outra mensagem como “Lançamento de notas das turmas não realizado”
CT a2	Verifica se o usuário o realiza o lançamento de notas corretamente definindo-as como números arredondados	O sistema verifica se o usuário arredondou as notas, caso contrário, o sistema as arredondará automaticamente (os números têm de ser arredondados para maior)

Fonte: elaboração própria

O requisito *a* referente à tabela 1 que foi então desenvolvido e separado em 2 fluxos (*a1* e *a2*) na tabela 2, trata-se, agora, de o professor realizar, sob um sistema, o lançamento de notas das turmas nas quais lecionou, e esse sistema é o encarregado de verificar se ele realmente realizou o lançamento de notas exibindo a mensagem como “lançamento de notas das turmas pronto” ou “lançamento de notas das turmas não realizado”. Apesar disso, ele verifica também se o professor realizou o lançamento das notas corretamente, definindo-as como números arredondados; caso ele não o tenha feito, o sistema automaticamente os arredonda para números inteiros maiores próximos. E pode-se, do mesmo modo, fazer isso para os requisitos *b*, *c*, *d*, *e* e *f*; entretanto, apenas o *a* foi usado como um exemplo.

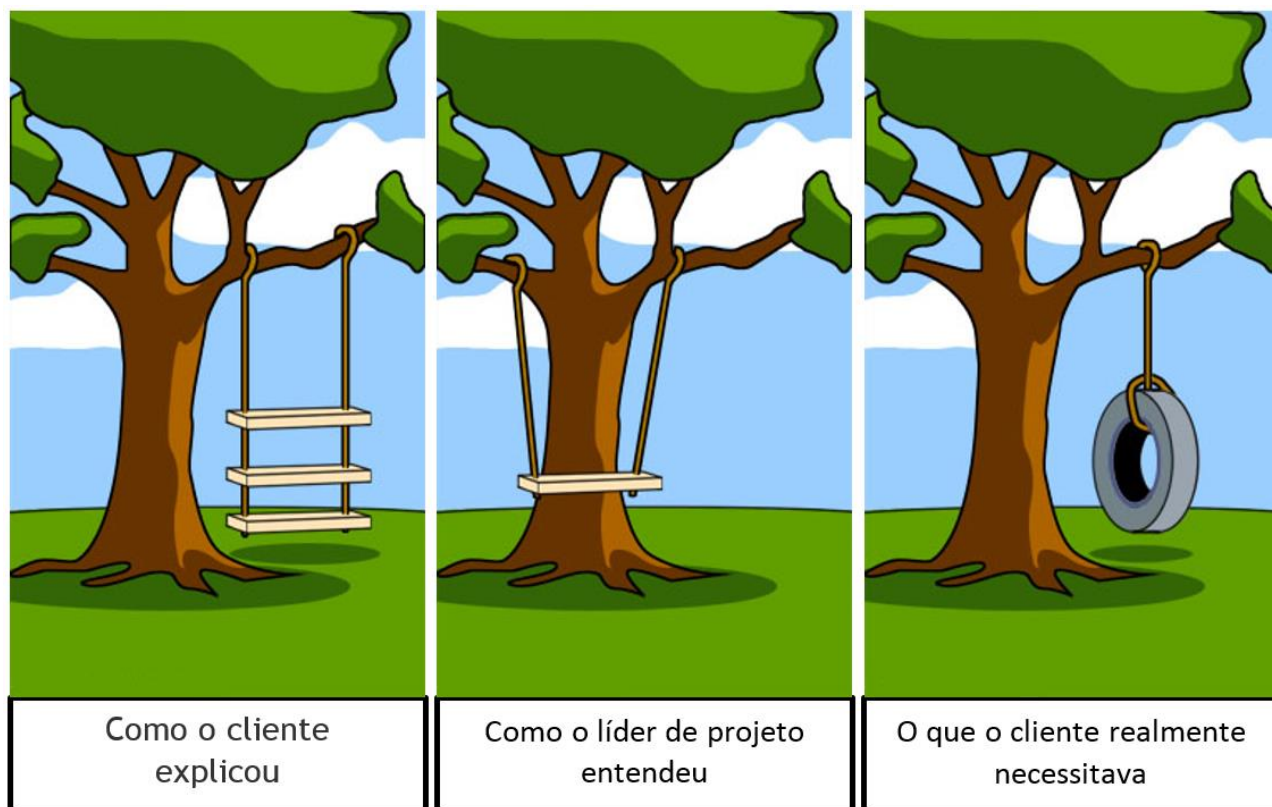
Assim que os membros supostamente, terminam o real documento de requisitos, os analistas já podem montar um estudo detalhado.

### 2.2.1.2 Explicando o estudo detalhado

O estudo detalhado, por ser parte da etapa de análise, que é abrangida pela segunda etapa, deve ser feito após o levantamento dos requisitos e a criação do documento de requisitos (ver 2.1.2.1); ele é usado para construir uma estratégia e um modelo do *software* que solucione o problema criado com base nos requisitos já levantados; não importa o jeito como o modelo será realizado, mas compreender o que sistema apresentado fará depois. Nessa etapa, há tanto a validação quanto a verificação desse modelo; ambas são ferramentas abstratas. A primeira é usada para saber se ele está sendo feito corretamente e está livre de ambiguidades tais como contradições e tem fundamento lógico, ou seja, ela é usada para apresentar o modelo ao cliente. E caso este tenha suas necessidades atendidas, o modelo é validado uma vez que os analistas foram capazes de entendê-lo com a ajuda de sua equipe de desenvolvimento. Entretanto, a segunda, que, na verdade, não se trata de uma atividade de análise, é usada para analisar a construção dos modelos: se eles estão dentro das conformidades dos requisitos e se estão acurados ao modelo também.

Tem-se a seguinte imagem abaixo para uma melhor compreensão sobre as ambiguidades em projetos:

Figura 6: Ambiguidades em Análise de Sistemas



Fonte: <http://projectcartoon.com/>

Na primeira parte da imagem acima, pode-se ver a forma a qual o cliente explicou à equipe de tal projeto; entretanto, o líder dela acabou entendendo isso de uma forma diferente como se pode ver na imagem do meio, já na da direita, está o que o cliente realmente queria. E devido a isso, a ambiguidade não deve ser admissível para os projetos, e caso um de seus requisitos tenha sido levantado incorretamente e, então, reconhecido muito tempo depois, é possível que a equipe inteira tenha complicações ao ponto de ter de gastar mais recursos, como tempo e até mesmo dinheiro, para corrigir esses erros resultantes de enganos e diferentes perspectivas. Se isso consome mais tempo, logo pode causar atraso na entrega do projeto inteiro além de afetar o próprio estudo detalhado que, por consequência, deverá ser refeito com base, logicamente, nos requisitos corrigidos, que têm de ser não-ambíguos. Os requisitos da tabela 1 (e até mesmo tabela 2), páginas 12 e 13, por exemplo, não podem apresentar esses problemas como esse apresentado na figura 6.<sup>[3]</sup>

Assim que o estudo é feito corretamente, o analista pode começar a realizar testes de *software* com base nos requisitos.



### 2.2.1.3 Testes de *software*

Os testes de *software* servem para verificar as funcionalidades e qualidade do sistema, ou seja, para averiguar se nele há algo apresentando *bug*, se algo lhe está incompleto, se o elemento humano não seguiu os requisitos que foram montados no processo de levantamento de requisitos para que pudesse montá-lo e, por fim, se algum tipo de campo não condiz com o esperado por exemplo.

Apesar disso, os testes de *software* podem ser separados em vários tipos além dos quais como:

- Teste de regressão (usado para verificar se tal versão do *software* apresenta ou voltou a apresentar problemas após a modificação de certa funcionalidade sua);
- Teste de unidade (usado para analisar pequenas unidades isoladas do *software*);
- Teste de integração (utilizado para saber se as pequenas unidades, ao serem integradas, geram problemas de funcionalidade);
- Teste usabilidade (que serve para saber se o componente humano é capaz de usar o *software* sem apresentar dificuldades).<sup>[10]</sup>

Se o analista detectar algum problema durante qualquer tipo de teste como esses, ele deverá conversar com os desenvolvedores responsáveis e até mesmo o resto da equipe a fim de que estes arrumem os problemas como os já citados *bugs*, por exemplo, e ele também deverá, como responsabilidade, montar um relatório no qual se especificam os requisitos que apresentam ou não erros de funcionalidades.

Portanto, recorrendo à tabela 2, páginas 12 e 13, pode-se observar que o requisito *a*, que foi aperfeiçoado e separado em *a1* e *a2*, já pode ser testado; porém, supõe-se que *a2* esteja correto e *a1* não e que os testes sobre eles, então, obterão os seguintes resultados como na tabela abaixo:

Tabela 3: Acompanhamento da execução dos testes

	02/08/2019	09/08/2019	06/09/2019
CT* a1	Sucesso	Sucesso	Falha
CT a2	Sucesso	Sucesso	Sucesso

Fonte: Elaboração própria

(Nota: CT = caso de teste)

Repara-se, de acordo com essa tabela, que os casos de teste foram executados nos dias 02/08, 09/08 e 06/09 no ano de 2019 e que o requisito *a1* acabou apresentando falhas enquanto o *a2* está em perfeitas condições desde a primeira data apresentada. E ademais, pode-se perceber que o

teste de regressão, inevitavelmente, acabou sendo utilizado no *al*, já que este funcionou somente entre os dias 02/08 e 09/08 e voltou a apresentar problemas no dia 06/09; isso se deve, provavelmente, pelo motivo de alguém o ter modificado.

O analista, apesar de ter de acompanhar a execução dos testes, tem de especificar detalhadamente os motivos que fizeram os requisitos, como o *al*, apresentarem problemas de funcionalidade em um outro documento ou tabela como a seguir:

Tabela 4: Histórico de execução de testes

<b>Casos de teste</b>	<b>Data da execução</b>	<b>Responsável(is) pela atividade de teste</b>	<b>Status do teste</b>	<b>Motivo(s) da falha</b>	<b>Responsável(is) pela correção da falha</b>	<b>Data prevista para novo teste</b>
<b>CT a1</b>	06/09/2019	Analista	Falha	O sistema não está mostrando a mensagem “Lançamento de notas das turmas pronto” nem “Lançamento de notas das turmas não realizado”	Desenvolvedores	07/09/2019

Fonte: Elaboração própria

E assim que tudo estiver finalizado bem como o processo de desenvolvimento de *software* (composto por processo de levantamento de requisitos, estudo detalhado e o próprio teste de *software*) e o seu sistema estiver totalmente programado e funcionando sem nenhuma falha, a equipe será capaz de satisfazer o cliente e, então, poder vender-lhe o *software* completo com um simples manual de instrução para informar como se usa o produto final. A equipe pode tanto definir o preço pessoalmente com cliente ou definir o preço através de aplicativos.

### 3 Foco dos Analistas no projeto Gerações

Tendo contextualizado e explicado de um modo geral e eficiente o que é um analista de sistemas (ver 2.1) bem como a utilidade do UML (que refere aos diagramas de casos uso) (ver. 2.1.1) e outrossim o processo de desenvolvimento de software (que abrange processo de levantamento de requisitos, estudo detalhado e testes de software) (ver. 2.1.1), é possível, agora, expor direta e resumidamente a importância dos analistas do projeto Gerações e os quão necessários eles foram, através da mesma lógica e didática usadas no capítulo 2 inteiro.

Em poucas palavras, o capítulo 2 serviu para explicar o que um analista faz nos projetos mais genéricos enquanto este — o capítulo 3 —, para especificar o que analista fez no exato e particular projeto Gerações.

#### 3.1 Levantamento de requisitos do projeto Gerações

O levantamento de requisitos é a parte inicial do projeto (ver 2.1.2.1), nele os analistas juntos com os desenvolvedores e membros da equipe conversam com os clientes, para anotar todos os requisitos necessários em um documento de requisitos com uma gramática informal e simples. No projeto Gerações, no qual isso foi aplicado, há 9 módulos, logo foram montados 9 documentos de requisitos em tabela dos quais:

Tabela 5: Documento de requisitos do módulo 1

Identificador	Descrição do Requisito
RF #01	<p>No mínimo, um gerente deverá se cadastrar no sistema, diretamente no Banco de Dados, com os seguintes dados: Nome Completo, RG, CPF, Prontuário, Código de Função, nível de gerencia, Senha e Confirmação de Senha, E-Mail para confirmação. Os campos para cadastro de gerentes devem seguir as seguintes restrições:</p> <ul style="list-style-type: none"><li>• Nome Completo: limite máximo de caracteres 100, campo obrigatório;</li><li>• RG: obrigatoriamente 9 caracteres, campo obrigatório, pontuação automática e exemplificação;</li><li>• CPF: 11 números, campo obrigatório, pontuação automática e exemplificação. Deve possuir alguma validação e uma confirmação de que não existem dois cadastros iguais;</li><li>• Prontuário: Numeração obrigatória gerada pelo sistema;</li><li>• Senha: limite máximo de 64 caracteres e mínimo de 8 caracteres, campo obrigatório;</li><li>• Confirmação de Senha: Deverá ser idêntica a senha;</li><li>• E-Mail: limite máximo de caracteres 255, campo obrigatório. Deve possuir alguma validação;</li><li>• Nível de gerência: poderá ser 1 (gerente administrativo) ou 2</li></ul>

	<p>(gerente superior);</p> <ul style="list-style-type: none"> <li>• Data de Nascimento: formato padrão de data, seguindo a escrita dd/mm/aaaa, campo obrigatório;</li> <li>• Celular: Máximo de 13 números, não obrigatório;</li> <li>• Telefone: Máximo de 10 números, não obrigatório;</li> <li>• Endereço: Máximo de 100 caracteres, obrigatório;</li> <li>• CEP da residência: Obrigatoriamente o de 8 caracteres e deve possuir validação;</li> <li>• Logradouro: Máximo de 5 números, obrigatório;</li> <li>• Complemento: Máximo de 50 caracteres, não obrigatório;</li> <li>• Cidade: deverá ser automaticamente relacionada do CEP;</li> <li>• Estado: deverá ser automaticamente relacionada do CEP.</li> </ul>
RF #02	<p>Os códigos de funções utilizados no cadastro de funcionários, responsável pelo idoso e da gerência devem seguir a seguinte padronização a fim de limitar o acesso dos usuários ao sistema, oferecendo privilégios; são eles:</p> <ul style="list-style-type: none"> <li>• Responsável pelo paciente;</li> <li>• Enfermeiros;</li> <li>• Nutricionistas;</li> <li>• Educadores Físicos;</li> <li>• Setor de Limpeza;</li> <li>• Médicos;</li> <li>• Gestores;</li> <li>• Doadores;</li> <li>• Fisioterapeuta.</li> </ul> <p>Obs.: o gerente deverá realizar tais operações de atribuição somente durante o CADASTRO.</p>
RF #03	<p>Este sistema deverá ser capaz de cadastrar em uma base de dados as informações sobre o usuário. (Tanto o responsável quanto os funcionários). Os campos gerais para cadastro devem seguir as seguintes restrições:</p> <ul style="list-style-type: none"> <li>• Nome Completo: limite máximo de caracteres 128, campo obrigatório;</li> <li>• RG: obrigatoriamente 9 caracteres, campo obrigatório, pontuação automática e exemplificação;</li> <li>• CPF: obrigatoriamente 11 números, campo obrigatório, pontuação automática e exemplificação. Deve possuir alguma validação e uma confirmação de que não existem dois cadastros iguais;</li> <li>• Data de Nascimento: formato padrão de data, seguindo a escrita dd/mm/aaaa, campo obrigatório;</li> <li>• Senha de Acesso (limite máximo de 16 caracteres e mínimo de 8 caracteres, campo obrigatório);</li> <li>• Confirmação de Senha (deverá ser idêntica a senha);</li> <li>• E-Mail (limite máximo de caracteres 255, campo não obrigatório. Deve possuir alguma validação);</li> <li>• Celular: Máximo de 13 números, não obrigatório;</li> <li>• Telefone: Máximo de 10 números, não obrigatório;</li> <li>• Endereço: Máximo de 100 caracteres, obrigatório;</li> <li>• CEP da residência: Obrigatoriamente o de 8 caracteres e deve possuir validação;</li> <li>• Logradouro: Máximo de 5 números, obrigatório;</li> </ul>

	<ul style="list-style-type: none"> <li>• Complemento: Máximo de 50 caracteres, não obrigatório;</li> <li>• Cidade: deverá ser automaticamente relacionada do CEP;</li> <li>• Estado: deverá ser automaticamente relacionada do CEP;</li> <li>• Função Exercida (determinará o código de função. Ao selecionar o cadastro, o usuário escolhe qual será seu perfil dentre as opções: responsável pelo paciente, Enfermeiro, Nutricionista, Educador Físico, Setor de Limpeza, Médico, Gestor, Fisioterapeuta ou Doador. Selecionando uma das opções, o sistema deverá mostrar os respectivos dados para o cadastro de cada perfil.)</li> <li>• Código do Registro Profissional (Ex.: CRM; Médico; CREA; Educador Físico; etc.) (obrigatório caso seja um enfermeiro, médico, nutricionista, fisioterapeuta ou educador físico);</li> <li>• Formação acadêmica (obrigatório caso seja um enfermeiro, médico, nutricionista, fisioterapeuta ou educador físico).</li> </ul> <p>Durante o cadastro, o usuário deverá informar um dos três atributos obrigatoriamente: E-Mail, Celular ou Telefone de contato. Todo usuário pode se auto cadastrar no sistema, pois seu código de função será nulo até que o gerente de nível 2 autentique e valide seu acesso por meio da sua própria página de recepção.</p>
RF #04	<p>Na página de recepção dos gerentes de nível 2 (superiores), uma aba deverá apresentar todos os usuários em aguardo para autenticação e ativação de suas contas. Uma caixa de entrada deverá listar todos os usuários que solicitaram algum tipo de cadastro, para, a partir daí, aprovar ou não seu acesso ao sistema. Se aprovado, o código de função é atribuído a ele automaticamente de acordo com sua função escolhida, caso contrário se o acesso for negado os dados são simplesmente deletados do banco de dados. O atributo que ativa a conta é do tipo boolean e não tem acesso direto pelos usuários.</p>
RF #05	<p>Em perfis de responsáveis, tanto os mesmos quanto os gerentes (1 e 2) podem editar as seguintes informações:</p> <ul style="list-style-type: none"> <li>• E-Mail, desde que seja verificado a confirmação da conta;</li> <li>• Senha;</li> <li>• Celular;</li> <li>• Telefone;</li> <li>• Endereço;</li> <li>• CEP da residência;</li> <li>• Logradouro;</li> <li>• Complemento;</li> <li>• Cidade, desde que o CEP esteja validado;</li> <li>• Estado, desde que o CEP esteja validado.</li> </ul>
RF #06	<p>Em perfis de funcionários, tanto os mesmos quanto os gerentes podem editar as seguintes informações:</p> <ul style="list-style-type: none"> <li>• E-Mail, desde que seja verificada a confirmação da conta;</li> <li>• Senha;</li> <li>• Celular;</li> <li>• Telefone;</li> <li>• Endereço;</li> <li>• CEP da residência;</li> </ul>

	<ul style="list-style-type: none"> <li>• Logradouro;</li> <li>• Complemento;</li> <li>• Cidade, desde que o CEP esteja validado;</li> <li>• Estado, desde que o CEP esteja validado.</li> </ul>
RF #07	<p>Em banco de dados, <b>TODOS OS USUÁRIOS E GESTORES</b> deverão ter os seguintes atributos gerais:</p> <ul style="list-style-type: none"> <li>• Nome Completo;</li> <li>• RG;</li> <li>• CPF;</li> <li>• Data de Nascimento;</li> <li>• Senha;</li> <li>• <i>E-mail</i> (usado para recuperação de senha);</li> <li>• Celular (usado para recuperação de senha);</li> <li>• Telefone (usado para recuperação de senha);</li> <li>• Endereço;</li> <li>• CEP;</li> <li>• Logradouro;</li> <li>• Complemento;</li> <li>• Cidade;</li> <li>• Estado.</li> </ul> <p>Os seguintes campos abaixo são específicos (alguns não aparecem explicitamente para cadastro):</p> <ul style="list-style-type: none"> <li>• Prontuário (gerado automaticamente no sistema);</li> <li>• Código de Função (padronizado pelo sistema, determinado pelo gerente de nível 2);</li> <li>• Conta Ativa (determinado pelo gerente nível 2).</li> </ul> <p>Os seguintes atributos abaixo são obrigatórios apenas para alguns cargos:</p> <ul style="list-style-type: none"> <li>• Código do Registro Profissional (Ex.: CRM; Médico; CREA; educador físico; etc.) (obrigatório caso seja um enfermeiro, médico, nutricionista, fisioterapeuta ou educador físico);</li> <li>• Formação acadêmica (para nutricionista, médico, educador físico e enfermeiro);</li> <li>• Tipo de Gestor (pode ser 1 ou 2 e é determinado apenas por um gestor superior).</li> </ul>
RF #08	<p>O sistema deveser possuir um cadastro para um usuário doador solicitando os mesmos dados de um usuário normal exceto o RG de um parente que está na casa de permanência, assim o usuário poderá acessar o setor de doação e ver o que pode estar em falta na instituição.</p>
RF #09	<p>Na tela de <i>login</i>, caso o usuário tenha esquecido a senha dele, ele pode selecionar a opção para recuperar sua senha. Neste momento, será solicitado o <i>e-mail</i> de cadastro do usuário. Ao confirmar o <i>e-mail</i>, uma nova senha é gerada automaticamente e enviada ao e-mail de cadastro do usuário informado. O usuário pode escolher trocá-la após logar, tendo que, obrigatoriamente, informar a senha gerada automaticamente, a nova senha e uma confirmação da nova senha. A nova senha deve ter o limite máximo de 16 caracteres e mínimo de 8 caracteres.</p>

RF #10	<p>A primeira página de acesso dos visitantes (a <i>home</i> do sistema), deverá apresentar a instituição. A página deverá possuir uma barra de navegação com as opções: <i>Home</i>, o projeto, equipe de desenvolvimento, sobre o IFSP e <i>entrar</i>, onde <i>Home</i> representa a página inicial (que deve apresentar a instituição em si; atrativo para os visitantes).</p> <p>O projeto = descrever o projeto do sistema em si;  A equipe de desenvolvimento = descrever sobre a turma de PDS;  Sobre o IFSP = descrever sobre o Instituto Federal e sua colaboração com a instituição;  Entrar = redirecionar o usuário para a página de <i>login</i>;  Além disso, esta página deverá conter informações de contato da instituição; telefones, tal qual as redes sociais e o e-mail institucional da mesma.</p>
RF #11	<p>Ao realizar o <i>login</i>, o usuário é redirecionado para a página principal de acesso, esta página varia conforme o tipo de usuário (se for responsável, se for enfermeiro, médico etc..), pelo qual:</p> <ul style="list-style-type: none"> <li>• Responsáveis: dados necessitados e que abrangem o módulo 3;</li> <li>• Enfermeiros: dados necessitados e que abrangem os módulos 2,4 e 5;</li> <li>• Gestores: dados necessitados e que abrangem os módulos 8 e 9;</li> <li>• Educadores Físicos e Fisioterapeutas: dados necessitados e que abrangem o módulo 7;</li> <li>• Médicos: dados necessitados e que abrangem o módulo 5;</li> <li>• Nutricionistas: dados necessitados e que abrangem o módulo 6;</li> <li>• Equipe de Limpeza: dados necessitados e que abrangem o módulo 8;</li> <li>• Doadores: mostrar apenas os produtos que estão em falta para que o usuário doador leve os materiais necessários a instituição;</li> </ul>
RF #12	<p>Na página de recepção, se o usuário desejar entrar no sistema, ele deverá informar os dois campos a seguir:</p> <ul style="list-style-type: none"> <li>• Prontuário;</li> <li>• Senha de acesso;</li> <li>• Após logado, o usuário pode realizar a operação de <i>logout</i>.</li> </ul>
RF #13	<p>O modal de feedback deve aparecer na página inicial do responsável logado. Ele poderá enviar mensagens, sem editá-las ou excluí-las. Sendo requerido apenas os dados:</p> <ul style="list-style-type: none"> <li>• Mensagem de feedback: até 4096 caracteres, obrigatória para o envio.</li> </ul> <p>Os feedbacks devem registrar a hora de envio e se ele foi visualizado pelo gestor de nível 2, sendo estas variáveis compartilhadas entre ambos.</p>
RF #14	<p>Os gerentes de nível 1 devem possuir na sua página principal de acesso, a opção para responder aos feedbacks solicitados pelos outros usuários. Os feedbacks devem solicitar:</p> <ul style="list-style-type: none"> <li>• A mensagem de resposta: até 4096 caracteres, obrigatória para o envio;</li> <li>• Data e hora da resposta: registrado automaticamente pelo sistema.</li> </ul> <p>Se foi visualizado, booleano que verifica se o usuário abriu a página que mostra a mensagem de feedback.</p>
RN #15	A base da programação do sistema será feita em linguagens e marcação (HTML,CSS) e de script (php, JavaScript).
RN #16	O framework Bootstrap deverá ser usado.
RN #17	O tempo de resposta do sistema deverá ser de, no máximo, 10 segundos.

RN #18	Os requisitos físicos do sistema deverão ser: <ul style="list-style-type: none"> <li>• Sistema Operacional: Windows 7 ou superior, com 64-bit;</li> <li>• Processador: Intel Core i3 6ª Geração ou superior;</li> <li>• Memória RAM: 4GB.</li> <li>• Memória de Disco (<i>Desktop</i> ou Servidor Interno/Externo): 512MB</li> </ul>
RN #19	Este sistema deverá ser capaz de propiciar uma interface agradável ao usuário, com opções simples, e autoexplicativas.

Fonte: elaboração própria

Tabela 6: Documento de requisitos do módulo 2

Identificador	Descrição do Requisito
RF #01	O sistema deve gerar prontuários automaticamente gerando uma sequência numérica pré-definida, começando pelo número 01 para o cadastro do idoso. O prontuário deve aparecer em cima da tela de cadastro, se cancelarem o cadastro, esse prontuário não será atribuído a nenhum idoso.
RF #02	O Sistema deve cadastrar o idoso, com as seguintes informações: <ul style="list-style-type: none"> <li>• Nome Completo (varchar 255) – Obrigatório;</li> <li>• RG (varchar 9) – Obrigatório;</li> <li>• CPF (char 12 XXX.XXX.XXX-XX) – Obrigatório – deve ser validado;</li> <li>• Sexo (M ou F) – Obrigatório;</li> <li>• Data de nascimento (DD/MM/AAAA) – Obrigatório;</li> <li>• Estado do idoso (Ativo/Inativo), inicialmente ativo – Obrigatório;</li> <li>• Fralda ou forrinho (S/N) – Obrigatório;</li> <li>• Código do responsável (O código deve ser pesquisado a partir do nome do responsável e/ou CPF, se o responsável for cadastrado, será selecionado e será designado ao idoso em questão. Se não houver responsável cadastrado (Botão “Não tenho responsável” (direcionar para a funcionalidade do Módulo 01), o site deverá ir para a página de cadastro de responsável) –obrigatório, não pode ser alterado;</li> <li>• Para atualizar será necessário fazer uma pesquisa antes procurando pelo nome ou CPF do idoso e irá aparecer uma tabela com o nome ou parte do nome dos idosos e nessa tabela conterà o nome, o CPF, o prontuário do idoso, e no final de cada linha de informações conterà um botão de editar onde se clicado levará para página de cadastro do idoso com todas as informações dele contida, onde poderá ser editado todos os campos, com exceção do prontuário;</li> <li>• O cadastro irá funcionar da seguinte forma: o funcionário pesquisa pelo nome ou CPF do idoso, caso não existir cadastro irá aparecer uma mensagem de cadastro inexistente, onde o funcionário irá cadastrar o idoso com todas as informações pedidas, e no final irá ter um botão de salvar, se os campo CPF estiver errado vai aparecer um erro para corrigir, pois esse campo deve ser validado, se tudo estiver certo após salvar, vai aparecer a mesma tela com todas os campos preenchidos, onde a pessoa que está cadastrando poderá modificar algum campo caso esteja errado, após isso terá o botão confirmar onde o cadastro vai ser concluído;</li> </ul>



	<ul style="list-style-type: none"> <li>• Após o cadastro ser concluído voltará para a tela inicial de pesquisa, e o funcionário deverá pesquisar pelo nome, cpf ou prontuário do idoso para conferir se realmente foi cadastrado, se sim, aparecerá o nome do idoso na tabela;</li> <li>• Todas as vezes que forem realizadas pesquisas de idosos, selecionando o nome ou CPF do mesmo, uma tabela irá listar todos os idosos que respeitem os parâmetros de pesquisa, possuindo os seguintes atributos: Prontuário, Nome e CPF. Na frente de cada um dos idosos, será mostrado opções para continuar o seu cadastro de entrada na Instituição de Longa Permanência. As opções que serão apresentadas são: <ul style="list-style-type: none"> <li>• Registro de Análise Clínica: Inserir, Editar e Visualizar (RF#04);</li> <li>• Doenças: Inserir, Editar e Visualizar (RF#05);</li> <li>• Medicações: Inserir, Editar e Visualizar (RF#06);</li> <li>• Restrições Alimentares: Inserir, Editar e Visualizar (RF#07);</li> <li>• Alergias: Inserir, Editar e Visualizar (RF#08).</li> </ul> </li> </ul>
RF #03	<p>O Sistema deve mostrar a listagem de todos os idosos, a partir do nome ou CPF digitado, só o funcionário responsável pelo cadastro terá acesso a essa listagem. Também deverá ter uma lista completa de todos os idosos cadastrados na instituição, aparecendo ao lado do nome o seu código e se está ativo ou não. Essa lista será outra escolha que terá na página inicial na opção de meu idoso.</p> <p>Essa opção terá 3 escolhas sendo elas: pesquisa e cadastro de idosos, ficha completa dos idosos, lista de idosos cadastrado sendo essa última um <i>select</i> da tabela idoso.</p>
RF #04	<p>O sistema deve registrar a análise clínica do idoso ao entrar na casa. Esse registro deve conter:</p> <ul style="list-style-type: none"> <li>• Alergias (S/N);</li> <li>• Tipo sanguíneo (Lista exibindo os seguintes tipos: A+, A-, B+, B-, AB+, AB-, O+ e O-);</li> <li>• Situação atual idoso (Saudável/doenças leves/Enfermo);</li> <li>• Altura e Peso (XXX centímetros, XXX Quilos);</li> <li>• Após concluído o cadastro do idoso, ao final de cada linha na listagem terá outra tabela com 3 botões(registrar a análise clínica, editar e visualizar) se clicar no botão de registro vai aparecer uma página com todas os campos citados acima, preenchendo todos, no botão de editar irá aparecer a mesma página com os campos já preenchidos podendo editar quaisquer campos, no botão de visualizar só será mostrado a análise clínica do idoso em PDF.</li> </ul>
RF #05	<p>O sistema deve registrar todas as doenças pré-existentes do idoso e atualiza-las de acordo com a situação clínica do idoso.</p> <p>Em outra tabela terá outro botão para registrar as doenças existentes onde abrirá uma página com algumas possíveis doenças pré-cadastradas no banco de dados (como diabetes, anemia, pressão baixa, entre outras), com a opção de escolher sim ou não para aquele problema, e se caso o idoso tiver uma doença não presente na página, haverá um campo para digitar quais</p>

	<p>doenças o idoso tem, e terá um botão de editar e visualizar em PDF as doenças do idoso.</p> <p>Tendo este registro exibido a todos os interessados por cada área da instituição.</p>
RF #06	<p>O sistema deve registrar as medicações prescritas dos idosos especificando quais medicamentos e sua quantidade diária.</p> <p>Ao final de cada linha na listagem terá outra tabela de medicações do lado da análise clínica com 3 botões (registrar medicações, editar e visualizar) se clicar no botão de registro vai aparecer uma página com a opção de escolher se sim para quem tem medicações e não para quem não usa nada, se sim vai aparecer um campo medicações onde será descrito que tipo de medicação, para que finalidade ele toma o medicamento e outro para especificar a quantidade diária. Vale destacar que essas 3 informações devem ser inseridas para cada uma das medicações tomadas pelo idoso. No botão de editar irá aparecer a mesma página com os campos já preenchidos podendo editar quaisquer campos, no botão de visualizar só será mostrado as medicações, finalidade e quantidade.</p> <p>Tendo este registro exibido a todos os interessados por cada área da instituição.</p>
RF #07	<p>O sistema deve registrar as restrições alimentares do idoso especificando sua dieta específica para suas necessidades nutricionais.</p> <p>Da mesma forma que medicações e análise clínica haverá outra tabela de restrições alimentares com os mesmos botões, onde no botão de registrar haverá uma tabela com possíveis restrições alimentares demarcando com “x” para as devidas restrições. Essas restrições alimentares serão apresentadas a partir de uma lista pré-cadastrada no Banco de Dados. No botão de editar irá aparecer a mesma página com os campos já preenchidos podendo editar quaisquer campos, no botão de visualizar só será mostrado as restrições alimentares.</p> <p>Tendo este registro exibido a todos aqueles que interessarem.</p>
RF #08	<p>O sistema deve registrar os tipos de alergias que o idoso possui, especificando o tipo de medicamento correto para cada caso.</p> <p>Em uma tabela como os demais, terá os mesmos botões que os anteriores, e quando registrar deverá ser especificado quais tipos de alergias têm, explicando seus cuidados diários e definindo os medicamentos corretos para ela. Podem ser criadas quantas alergias forem necessárias, mas estes 3 campos devem ser sempre informados para cada um. No botão de editar irá aparecer a mesma página com os campos já preenchidos podendo editar quaisquer campos, no botão de visualizar só será mostrado as alergias, seus cuidados diários e as respectivas medicações.</p> <p>Tendo este registro exibido a todos os aqueles que interessarem.</p>
RF #09	<p>O sistema deve gerar uma ficha completa do idoso em PDF, sendo detalhados todos os dados registrados anteriormente, sendo eles a análise clínica do idoso, as doenças pré-existent, as medicações, as alergias, e as restrições alimentares, dando acesso aos responsáveis de cada área médica.</p> <p>Ao entrar na opção de ficha completa do idoso, no menu “Meu Idoso”, aparecerá uma área de pesquisa pelo nome ou prontuário do idoso, o funcionário ao pesquisar e selecionar o nome estará logado e se não estiver, quando selecionar para gerar a ficha será levado para a tela de <i>login</i>, depois de autenticado voltará à página de pesquisa onde terá um botão de gerar ficha de determinado idoso, onde será gerada as informações detalhadas da</p>

	<p>situação do mesmo.</p> <p>A ficha deverá ser apresentada da seguinte forma:</p> <ul style="list-style-type: none"> <li>• Dados gerais;</li> <li>• Nome completo;</li> <li>• Prontuário;</li> <li>• CPF;</li> <li>• RG;</li> <li>• Data de nascimento;</li> <li>• Sexo;</li> <li>• Status;</li> <li>• Responsável;</li> <li>• Usa fralda/forrinho.</li> </ul> <p>Análise Clínica:</p> <ul style="list-style-type: none"> <li>• Alergias;</li> <li>• Tipo sanguíneo;</li> <li>• Situação atual idoso;</li> <li>• Altura e peso;</li> <li>• Doenças pré-Existentes;</li> <li>• Lista com os Nomes da Doenças.</li> </ul> <p>Medicações:</p> <ul style="list-style-type: none"> <li>• Nome do Medicamento;</li> <li>• Finalidade;</li> <li>• Quantidade.</li> </ul> <p>Restrições Alimentares:</p> <ul style="list-style-type: none"> <li>• Lista com os Nomes das Restrições;</li> <li>• Alergias;</li> <li>• Nome da Alergia;</li> <li>• Cuidados Diários;</li> <li>• Medicação.</li> </ul>
RN #10	O Sistema deve ser entregue em no mínimo 80% de conclusão até o dia 29 de Novembro de 2019.
RN #11	O Sistema deve ter um tempo de resposta de no máximo 3 segundos.
RN #12	O sistema deve gerar relatórios mensais e integrá-los ao modulo 9. Quero saber qual é a API de PDF que será utilizada?
RN #13	O sistema deve ser compatível com o Google Chrome ou Firefox.
RN#14	O sistema deve dar acesso às funcionalidades desse modulo somente aos enfermeiros.
RN#15	<p>API utilizada para gerar PDF:</p> <ul style="list-style-type: none"> <li>• Pdflayer (sujeito à mudanças).</li> </ul>

Fonte: elaboração própria

Tabela 7: Documento de requisitos do módulo 3

Identificador	Descrição do Requisito
---------------	------------------------

RF #01	Assim que o usuário fizer <i>login</i> com usuário, senha e a atribuição (responsável, médico, gestor, etc) no sistema, o sistema deverá exibir um menu. Esse menu terá como opções “Home”, “Finanças”, “Categorias” e “Sair”. Abaixo desse menu terá suas informações pessoais (nome, idade, data de nascimento, endereço, telefone, e-mail, cidade, RG e CPF) com uma foto e um rodapé.
RF #02	<p>O sistema deverá exibir os dados financeiros quando o responsável selecionar a opção “Finanças” no menu superior. As informações serão exibidas em uma tabela, onde cada uma das colunas apresenta:</p> <ul style="list-style-type: none"> <li>• Data de pagamento (DD/MM/AAAA);</li> <li>• O Período referente ao Pagamento (Data de Início - DD/MM/AAAA e Data de Fim - DD/MM/AAAA);</li> <li>• O Valor do pagamento (DECIMAL (10,2));</li> <li>• A Forma de pagamento (cartão de débito, dinheiro ou cheque);</li> <li>• O próximo pagamento a ser realizado (que será calculado a partir de 30 dias do dia do último pagamento, ou seja, a mensalidade dá direito a 30 dias de permanência na instituição);</li> <li>• Ex: O último pagamento foi dia 10 de abril de 2019, então o próximo pagamento será no dia 10 de maio de 2019. (Próximo pagamento = Dia do último pagamento + 30 dias);</li> </ul> <p>Cada tabela será referente à um idoso, caso ele tenha mais de 01.</p>
RF #03	<p>Ao selecionar a opção “Categorias”, o sistema deverá exibir os dados da respectiva categoria, onde devem aparecer os seguintes campos:</p> <ul style="list-style-type: none"> <li>• Dados dos cuidados diários dos idosos (em um período de 01 mês): <ul style="list-style-type: none"> <li>• Sinais Vitais: Especificado no RF #06;</li> <li>• Eliminação: Especificado no RF #07;</li> <li>• Hidratação: Um campo mostrando a medida da quantidade de água consumida;</li> <li>• Banho de sol: Uma tabela contendo duas informações, a primeira em quanto tempo foi o banho de sol e a segunda em qual parte do dia foi realizado;</li> <li>• Repelente: Uma barra mostrando se o idoso fez uso do repelente ou não;</li> <li>• Padrão de sono: Uma barra mostrando se o padrão de sono do idoso está Bom, Regular ou Ruim;</li> <li>• Higiene: Nova tabela, contendo campos como: Banho (se tomou ou não), Couro Cabeludo e Orelhas (se a higienização foi ou não feita) e Depilação (se ocorreu ou não);</li> <li>• Evolução diária: Especificado no RF #08;</li> <li>• Status Geral das Patologias e Prescrições Médicas: Especificado no RF #05;</li> <li>• Status Geral do Padrão Alimentar: Especificado no RF #09;</li> <li>• Status de Participações em Atividades Recreativas: Especificado no RF #10;</li> <li>• Ou seja, todos esses dados serão mostrados em um período de 01 mês. Além disso, o sistema deverá ser organizado em Categorias (os dados dos cuidados diários dos idosos, prescrições médicas e patologias, alimentação e a participação do idoso em atividades físicas e recreativas), das quais o responsável pode selecionar as que ele deseja.</li> </ul> </li> </ul>

RF #04	Ao selecionar uma categoria, o sistema deverá exibir dois inputs para que o usuário defina o intervalo de tempo dos dados a serem exibidos (DD/MM/AAAA), além de um terceiro input onde o responsável pesquisará o nome de um dos idosos sob sua responsabilidade.
RF #05	O sistema deve exibir uma tabela contendo data de incidência da patologia, grau (terminal, inicial, avançado, intermediário) e observações iniciais; outra tabela contendo as prescrições médicas com: medicamento (nome), posologia (dosagem), validade da prescrição e observações; e uma última tabela contendo as vacinações periódicas por idoso, com dose (ml), data da vacina e observações.
RF #06	Dentro da categoria de sinais vitais, terá uma lista deles em uma tabela, com Data/Hora que foi medido, média diária da Pressão arterial, do Pulso, da Temperatura, do Dextro, da Respiração, do SPO2 em um período de 01 mês e o nome do colaborador que mediu.
RF #07	Na categoria eliminação deverá ter uma tabela das eliminações com os seguintes campos: Data/Hora que foram efetuadas, média diária da Diurese e da Evacuação em um período de 01 mês e o nome do colaborador que mediu.
RF #08	Todos os cuidados diários do idoso deverão ser armazenados assim que o colaborador responsável os registrar. O registro da evolução diária do idoso é desenvolvido pela análise do enfermeiro (se por exemplo a alimentação está melhor e de que forma isso é perceptível, se ele comeu mais ou se aceitou melhor um alimento, etc.). Aparecerão também nesta tabela as observações do enfermeiro em relação ao idoso.
RF #09	Em Padrão Alimentar o sistema deverá exibir uma tabela contendo um campo com a aceitação do idoso em relação a um alimento específico (pouca aceitação, média aceitação, boa aceitação ou não aceitou) e um campo de observações do enfermeiro em relação aos alimentos e ao padrão alimentar.
RF #10	Nas atividades físicas e recreativas o sistema deverá exibir uma tabela contendo todas as atividades realizadas pelo idoso no mês, a data (DD/MM/AAAA) e a hora (XX:XX) na qual fora realizada.
RF #11	Quando o responsável selecionar um ano específico para ver o desenvolvimento do idoso no período, será a média geral de cada categoria durante aquele ano (sinais vitais, eliminação, hidratação, banho de sol, repelente, padrão de sono, higiene, patologias, medicamentos, padrão alimentar e a participação em atividades recreativas) que será exibida. Será exibido em formato de tabela (uma para cada categoria) e a opção de ver a evolução com gráficos.
RF #12	O sistema deverá disponibilizar um botão de alteração na página do responsável caso ele queira modificar campos como nome, e-mail, telefone, forma de pagamento, foto e cidade.
RF #13	O sistema deverá exibir o histórico de pagamento da estadia do idoso. Será uma tabela com todas as datas de pagamento (DD/MM/AAAA) e o valor pago em cada uma (DECIMAL (10,2)).
RN #14	Processo de Software: É necessário que o Processo de Software a ser utilizado seja o Redmine + Kanban.
RN #15	É necessário que se adote a linguagem PHP, pois é uma linguagem de programação eficiente e versátil.
RN #16	Design Pattern MVC: Será utilizado o design pattern MVC. Esse padrão de projeto será adotado pois facilita a reutilização do código e facilita também a

	forma com que a implementação será gerenciada, permitindo que cada camada possa ser construída de forma quase que completamente independente. Não será utilizada nenhuma ferramenta de geração de código.
RN #17	Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line: É necessário disponibilizar, através da Web, a documentação do desenvolvimento do Protótipo do de Projeto de SGI, para propiciar a troca de informações e experiências entre todos os grupos participantes da disciplina.
RN #18	Este sistema deverá ser capaz de propiciar uma interface agradável ao usuário, com opções simples, e autoexplicativas.
RN #19	O sistema deve garantir que o único que terá permissão para acessar essa página é o responsável pelo idoso.
RN #20	O menu de opções ("Home", "Finanças", "Meu idoso", "Sair") deve ser horizontal e estar posicionado na parte superior da tela.
RN #21	O sistema deverá ser executado nos navegadores: Google Chrome, Mozilla Firefox, e Internet Explorer.

Fonte: elaboração própria

Tabela 8: Documento de requisitos do módulo 4

<b>Identificador</b>	<b>Descrição do Requisito</b>
RF #01	O sistema deverá exigir que somente os enfermeiros e o administrador do sistema tenham permissão para acessar a seção de cuidados diários do prontuário do idoso por meio de um controle de acesso privilegiado, através de: Prontuário; Senha.
RF #02	O sistema deverá possuir uma listagem de todos os idosos da instituição com nome completo (varchar(100)) ou prontuário (int auto_increment) e foto. Acima da listagem deverá aparecer um campo de pesquisa feita através do nome ou parte do nome. À direita de cada nome deverá conter 4 botões (Sinais Vitais, Eliminações, Cuidados Diários, Higiene).
RF #03	O botão de Sinais Vitais deverá permitir o acesso à ficha de sinais vitais. Essa ficha irá conter o prontuário (integer auto incremento), nome completo (varchar(100)), data de nascimento (date DD/MM/AAAA) e data de ingresso (date DD/MM/AAAA). Deverá possuir uma lista com as prescrições médicas (quantidade de vezes a ser realizado certo procedimento, medir a pressão, o pulso, a temperatura, o dextro, a respiração e o SPO2) ou recomendações, identificando o médico pelo nome (varchar (100)). A ficha deverá possuir também uma espécie de histórico com as medições que já foram realizadas no dia ou em outra data que pode ser selecionada. O enfermeiro poderá modificar o dado que está no histórico desde que seja o dia que ele as inseriu. Abaixo da ficha deverá haver um botão (+) que abrirá uma seção com uma tabela de datas à serem selecionadas para visualização dos dados de outro dia. Ela deverá possuir um botão de inserção de dados que abrirá uma tela, onde poderá ser selecionado qual sinal vital ele gostaria de preencher como: <ul style="list-style-type: none"> <li>• Pressão (integer) que possuirá dois campos para serem preenchidos;</li> <li>• Pulso (integer) com um campo ser preenchido;</li> <li>• Temperatura (integer);</li> </ul>

	<ul style="list-style-type: none"> <li>• Dextro (integer) com um campo;</li> <li>• Respiração (integer) com um campo;</li> <li>• SPO2 (integer) com um campo.</li> </ul> <p>Essa seleção será realizada por um menu dropdown, todos os sinais vitais deverão ter uma opção para adicionar mais marcações caso essa seja a prescrição médica todas elas devem marcar automaticamente a data e a hora que foi inserida.</p> <p>Todos os campos são de preenchimento obrigatório, mas não necessariamente devem ser preenchidos ao mesmo tempo.</p> <p>O sistema deverá ser capaz de permitir o acesso ativo às informações das evoluções diárias no prontuário de determinado idoso para que os cuidados sempre evoluam junto à dele. Para o acompanhamento deverá ter um campo de observações não obrigatório para que o enfermeiro insira as evoluções e/ou avaliações sobre o idoso.</p> <p>Vale destacar que os dados podem ser alterados ou excluídos, de acordo com a necessidade dos enfermeiros.</p>
RF #04	<p>O botão de Eliminações deverá permitir o acesso à ficha de eliminação. Essa ficha irá conter o prontuário (integer auto incremento), nome completo (varchar(100)), data de nascimento (date DD/MM/AAAA) e data de ingresso (date DD/MM/AAAA).</p> <p>A ficha deverá possuir também uma espécie de histórico com as eliminações que já ocorreram no dia ou em outra data que pode ser selecionada. O enfermeiro poderá modificar o dado que está no histórico desde que seja o dia que ele as inseriu.</p> <p>Abaixo da ficha deverá haver um botão (+) que abrirá uma seção com uma tabela de datas à serem selecionadas para visualização dos dados de outro dia.</p> <p>Ela deverá possuir um botão de inserção de dados que abrirá uma tela, onde poderá ser selecionado qual tipo de eliminação o enfermeiro gostaria de preencher (diurese ou evacuação), essa seleção será realizada por um menu dropdown, todas as eliminações deverão ter uma opção para adicionar mais marcações caso essa seja necessária. O sistema deverá possuir um campo de preenchimento obrigatório para ser colocado qual período do dia aconteceu a eliminação (manhã, tarde, noite) e o horário desta.</p> <p>O momento da inserção deve ser a registrado automaticamente em formato HH:MM e DD/MM/AAAA.</p> <p>Todos os campos são de preenchimento obrigatório, mas não necessariamente devem ser preenchidos ao mesmo tempo.</p> <p>O sistema deverá ser capaz de permitir o acesso ativo às informações das evoluções diárias no prontuário de determinado idoso para que os cuidados sempre evoluam junto à dele. Para o acompanhamento deverá ter um campo de observações não obrigatório para que o enfermeiro insira as evoluções e/ou avaliações sobre o idoso.</p> <p>Vale destacar que os dados podem ser alterados ou excluídos, de acordo com a necessidade dos enfermeiros.</p>
RF #05	<p>O botão de Cuidados Diários deverá permitir o acesso à ficha dos cuidados diários. Essa ficha irá conter o prontuário (integer auto incremento), nome completo (varchar(100)), data de nascimento (date DD/MM/AAAA) e data de ingresso (date DD/MM/AAAA). A ficha deverá possuir também uma espécie de histórico com aos cuidados que já foram realizados no dia ou em outra data que pode ser selecionada. O enfermeiro poderá modificar o dado</p>

	<p>que está no histórico desde que seja o dia que ele as inseriu.</p> <p>Abaixo da ficha deverá haver um botão (+) que abrirá uma seção com uma tabela de datas à serem selecionadas para visualização dos dados de outro dia.</p> <p>Ela deverá possuir um botão de inserção de dados que abrirá uma tela, onde poderá ser selecionado qual cuidado o enfermeiro gostaria de preencher como:</p> <ul style="list-style-type: none"> <li>• Hidratação (integer) que será medida na quantidade de água consumida</li> <li>• Banho de sol (terá dois campos, o primeiro em quanto tempo foi o banho de sol e o segundo em qual parte do dia foi realizado o banho)</li> <li>• Repelente (sim/não)</li> <li>• Padrão de sono (bom/regular/ruim)</li> </ul> <p>Essa seleção será realizada por um menu dropdown, todos os sinais vitais deverão ter uma opção para adicionar mais marcações caso essa seja a prescrição médica todas elas devem marcar automaticamente a data e a hora que foi inserida.</p> <p>Todos os campos são de preenchimento obrigatório, mas não necessariamente devem ser preenchidos ao mesmo tempo.</p> <p>O sistema deverá ser capaz de permitir o acesso ativo às informações das evoluções diárias no prontuário de determinado idoso para que os cuidados sempre evoluam junto à dele. Para o acompanhamento deverá ter um campo de observações não obrigatório para que o enfermeiro insira as evoluções e/ou avaliações sobre o idoso.</p> <p>Vale destacar que os dados podem ser alterados ou excluídos, de acordo com a necessidade dos enfermeiros.</p>
RF #06	<p>O botão de Higiene deverá permitir o acesso à ficha de higiene diária. Essa ficha irá conter o prontuário (integer auto incremento), nome completo (varchar(100)), data de nascimento (date DD/MM/AAAA) e data de ingresso (date DD/MM/AAAA). A ficha deverá possuir também uma espécie de histórico com aos cuidados de higiene que já foram realizados no dia ou em outra data que poderá ser selecionada. O enfermeiro poderá modificar o dado que está no histórico desde que seja o dia que ele as inseriu.</p> <p>Abaixo da ficha deverá haver um botão (+) que abrirá uma seção com uma tabela de datas à serem selecionadas para visualização dos dados de outro dia.</p> <p>Ela deverá possuir um botão de inserção de dados que abrirá uma tela, onde poderá ser selecionado qual cuidado higiênico o enfermeiro gostaria de preencher como:</p> <ul style="list-style-type: none"> <li>• Banho: com campo de menu (sim/não);</li> <li>• Higienização do couro cabeludo: com campo de menu (sim/não);</li> <li>• Higienização das orelhas: com campo de menu (sim/não);</li> <li>• Depilação: com campo de menu (sim/não).</li> </ul> <p>Essa seleção será realizada por um menu dropdown, todos os sinais cuidados deverão ter uma opção para adicionar mais marcações caso necessário todas elas devem marcar automaticamente a data e a hora que foi inserida.</p> <p>Todos os campos são de preenchimento obrigatório, mas não necessariamente devem ser preenchidos ao mesmo tempo.</p> <p>O sistema deverá ser capaz de permitir o acesso ativo às informações das evoluções diárias no prontuário de determinado idoso para que os cuidados</p>



	sempre evoluam junto à dele. Para o acompanhamento deverá ter um campo de observações não obrigatório para que o enfermeiro insira as evoluções e/ou avaliações sobre o idoso. Vale destacar que os dados podem ser alterados ou excluídos, de acordo com a necessidade dos enfermeiros.
RF #07	O sistema deve exibir um alerta todas as vezes que o enfermeiro clicar em confirmar alguma alteração ou excluir algum dado, para diminuir a possibilidade de falhas.
RN #08	Linguagem PHP e HTML: É necessário que se adote as linguagens PHP e HTML, pois são linguagens eficientes para o desenvolvimento do sistema desejado.
RN #09	Este sistema deverá ser capaz de propiciar uma interface prática ao usuário, com opções e uso simples, e autoexplicativas.
RN #10	Design Pattern MVC: Será utilizado o design pattern MVC. Esse padrão de projeto será adotado pois facilita a reutilização do código e também a forma com que a implementação será gerenciada, permitindo que cada camada possa ser construída de forma quase que completamente independente. Não será utilizada nenhuma ferramenta de geração de código.
RN #11	Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line: É necessário disponibilizar, através da Web, a documentação do desenvolvimento do Protótipo do de Projeto de SGI, para propiciar a troca de informações e experiências entre todos os grupos participantes da disciplina.
RN #12	É necessário que na listagem de idosos apareça para os enfermeiros responsáveis pelos cuidados diários as respectivas fotos dos pacientes cadastrados, para facilidade de reconhecimento no momento da atualização do prontuário.

Fonte: elaboração própria

Tabela 9: Documento de requisitos do módulo 5

Identificador	Descrição do Requisito
RF #01	O sistema deverá permitir o cadastro de patologias, sendo necessário preencher os seguintes campos: o nome, sintomas, tipo (categoria que a patologia se encontra, como doença renal, cardíaca, crônica, etc.), tratamento e prevenção, sendo todos de preenchimento obrigatórios; <ul style="list-style-type: none"> <li>• Nome – varchar (100);</li> <li>• Sintomas – varchar (500);</li> <li>• Tipo – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Tratamento – varchar (500);</li> <li>• Prevenção – varchar (500).</li> </ul>
RF #02	O sistema deverá permitir o gerenciamento das patologias, através de opções que permitam a atualização (imprimindo na tela os dados previamente inseridos no cadastro e permitindo sua edição), para isso o sistema deve efetuar uma pesquisa pelo nome da patologia (sendo que o texto inserido pode estar em qualquer parte do nome completo), a listagem será organizada pela ordem alfabética dos nomes das patologias sendo expresso na tela em formato de tabela dados referentes aquele nome que foi inserido (os dados serão nome e tipo), após isso o usuário selecionará a patologia desejada, imprimindo na tela todos os dados que foram cadastrados (o nome, sintomas, tipo, tratamento e prevenção), permitindo a atualização de todos esses campos. Além da opção de atualização o sistema

	deverá permitir a exclusão da patologia, sendo o mesmo processo da atualização, com pesquisa da patologia por nome, na listagem das patologias, ao lado do botão de atualização deve haver o de exclusão, ao clicar nele deve se pedir a confirmação da exclusão, caso haja a confirmação é feita a exclusão da doença por completo de seus dados da base de dados.
RF #03	<p>O sistema deverá permitir o cadastro dos medicamentos (qualquer substância que é aplicada para o tratamento de uma patologia) sendo necessário preencher os seguintes campos: o nome comercial, nome químico, laboratório, via (forma como o medicamento é aplicado), tipo (genérico, similar, original, manipulado), tarja (amarela, vermelha e preta) e função (antibiótico, antialérgico, antidepressivo, anti-inflamatório, analgésico etc.), sendo todos de preenchimento obrigatórios:</p> <ul style="list-style-type: none"> <li>• Nome comercial – varchar (100);</li> <li>• Nome químico – varchar (100);</li> <li>• Laboratório – varchar (50);</li> <li>• Via – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Tipo – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Tarja – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Função – Lista pré-cadastrada a ser selecionada via Combobox.</li> </ul>
RF #04	<p>O sistema deverá permitir o gerenciamento dos medicamentos, através de opções que permitam a atualização (imprimindo na tela os dados previamente inseridos no cadastro e permitindo sua edição), para isso o sistema deve efetuar uma pesquisa pelo nome do medicamento (sendo que esse nome pode ser tanto o comercial como o químico, podendo o texto inserido pode estar em qualquer parte do nome completo), a listagem será organizada pela ordem alfabética dos nomes dos medicamentos sendo expresso na tela em formato de tabela dados referentes aquele nome que foi inserido (os dados serão nome comercial, nome químico, tipo e via), após isso o usuário selecionará o medicamento desejado, imprimindo na tela todos os dados que foram cadastrados (nome comercial, nome químico, laboratório, tipo, tarja e função). Além da opção de atualização o sistema deverá permitir a exclusão do medicamento, sendo o mesmo processo da atualização, com pesquisa do medicamento por nome, na listagem dos medicamentos, ao lado do botão de atualização deve haver o de exclusão, ao clicar nele deve se pedir a confirmação da exclusão, caso haja a confirmação é feita a exclusão do medicamento por completo de seus dados da base de dados.</p>
RF#05	<p>O sistema deverá permitir o cadastro das vacinas (qualquer substância que é aplicada para a prevenção de uma doença), sendo necessário preencher os seguintes campos: Nome, nome químico, laboratório, composição (elementos químicos presentes na fórmula), função (para que ela serve), observações (quais pessoas não podem tomar, quais são as possíveis reações), periodicidade (de quanto em quanto tempo a vacina precisa ser tomada), sendo todos de preenchimento obrigatório:</p> <ul style="list-style-type: none"> <li>• Nome comercial – varchar (100);</li> <li>• Nome químico – varchar (100);</li> <li>• Laboratório – varchar (50);</li> <li>• Composição – varchar (100);</li> <li>• Função – Lista pré-cadastrada a ser selecionada via Combobox.</li> <li>• Obs.: varchar (500)</li> </ul> <p>Periodicidade – no qual, haverá dois campos, sendo um campo numérico de preenchimento manual e outro com uma lista pré-cadastrada a ser selecionada</p>

	via Combobox com opções de dias, semanas, meses e anos
RF#06	<p>O sistema deverá permitir o gerenciamento das vacinas, através de opções que permitam a atualização (imprimindo na tela os dados previamente inseridos no cadastro e permitindo sua edição), para isso o sistema deve efetuar uma pesquisa pelo nome da vacina (sendo que pode ser o nome ou nome químico, podendo o texto inserido pode estar em qualquer parte do nome completo), a listagem será organizada pela ordem alfabética dos nomes das vacina sendo expresso na tela em formato de tabela dados referentes aquele nome que foi inserido (os dados serão nome, nome químico, composição e função), após isso o usuário selecionará a vacina desejada, imprimindo na tela todos os dados que foram cadastrados (nome, nome químico, laboratório, composição, função, observações e periodicidade). Além da opção de atualização o sistema deverá permitir a exclusão da vacina, sendo o mesmo processo da atualização, com pesquisa da vacina por nome, na listagem das vacinas, ao lado do botão de atualização deve haver o de exclusão, ao clicar nele deve se pedir a confirmação da exclusão, caso haja a confirmação é feita a exclusão da vacina por completo de seus dados da base de dados.</p>
RF#07	<p>O sistema deverá permitir o gerenciamento das vacinas por idoso, fazendo suas respectivas associações, para isso o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (sendo que o texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. O Sistema deve possibilitar então que o médico selecione o idoso entre os resultados encontrados, carregando o prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura), bem como as vacinas que já estão associadas a ele. A listagem das vacinas será em formato de histórico, no qual, apresentará a data, dose, nome da vacina e observações (se houve reação ou efeito colateral), a listagem será organizada pela data mais recente que foram aplicadas, em formato de tabela para melhor distinguir tais vacinas. Após isso, o sistema deverá possibilitar a inserção de uma nova vacina para o idoso, através, primeiramente, da pesquisa por nome (sendo que o texto inserido pode estar em qualquer parte do nome completo) e seleção da vacina, previamente cadastrada no RF #05. Com o idoso e a vacina selecionados, o sistema deverá registrar a data/hora da nova vacina atribuída ao idoso, que deverá ser a data/hora do dispositivo do usuário no momento em que ela ocorrer. Além disso deverá ser preenchido o campo de dose, sendo de preenchimento obrigatório, o campo de observação será de preenchimento opcional:</p> <ul style="list-style-type: none"> <li>• Horário e data da vacina – datetime;</li> <li>• Dose (ml) da vacina – float;</li> <li>• Nome da vacina – varchar (50);</li> <li>• Observações – varchar (500).</li> </ul>
RF #08	<p>O sistema deverá permitir o gerenciamento das vacinas por idoso, permitindo o registro de novas observações a ele associadas (conforme descrição do RF#07). Para isso, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (sendo que o texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. O Sistema deve possibilitar então que o médico selecione o idoso entre os resultados encontrados, carregando o prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos), bem como as vacinas que já estão associadas a ele. Após isso, o sistema deverá possibilitar a alteração dos dados relativos aos registros, imprimindo na tela e</p>

	<p>permitindo a atualização de todos os campos (data, dose, nome da vacina e observações) e registrando a data/hora da atualização, que deverá ser a data/hora do dispositivo do usuário no momento em que ela ocorrer. Para registrar o histórico, o sistema deve replicar, quando a primeiro registro da vacina foi realizado, os dados cadastrados no RF#07 para outra tabela na base de dados, que armazenará toda atualização realizada pelo usuário, criando um novo registro. Além disso, o sistema deverá permitir a inserção de observações para cada atualização, sendo que “observações” é de preenchimento opcional:</p> <ul style="list-style-type: none"> <li>• Horário e data da vacina – datetime;</li> <li>• Dose (ml) da vacina – float;</li> <li>• Nome da vacina – varchar (50);</li> <li>• Observação – varchar (500).</li> </ul>
RF #09	<p>O sistema deverá permitir o gerenciamento das patologias por idoso, permitindo o registro das patologias por idoso (isto é, associar determinada patologia previamente cadastrada, conforme o RF#01, com um idoso cadastrado no sistema pelo módulo 02). Para isso, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (sendo que o texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. O sistema deve possibilitar então que o médico selecione o idoso entre os resultados encontrados, carregando o prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos) bem como as patologias que já estão associadas a ele. A listagem das patologias será em formato de histórico, no qual, apresentará o nome, data de incidência da patologia, grau e observações iniciais, a listagem será organizada pela data mais recente que foram associadas, em formato de tabela para melhor distinguir tais patologias. Após isso, o sistema deverá possibilitar a inserção de uma nova patologia para o idoso, através, primeiramente, da pesquisa com nome e seleção da patologia, previamente cadastrada no RF#01. Com o idoso e a patologia selecionados o sistema deverá registrar a data/hora da inserção, que deverá ser a data/hora do dispositivo do usuário no momento em que ela ocorrer. Além disso, deve-se preencher os seguintes campos, que especificam melhor a condição do idoso: Grau (terminal, inicial, avançado, intermediária), status (se o paciente está fazendo uso do medicamento ou não) e observações iniciais, sendo todos de preenchimento obrigatórios:</p> <ul style="list-style-type: none"> <li>• Nome – varchar (100);</li> <li>• Horário e data de incidência da patologia;</li> <li>• Grau – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Observações Iniciais – varchar (500).</li> </ul>
RF #10	<p>O sistema deverá permitir o gerenciamento das patologias por idoso, permitindo o registro das patologias por idoso (isto é, associar determinada patologia previamente cadastrada, conforme o RF#01, com um idoso cadastrado no sistema pelo módulo 02). Para isso, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (sendo que o texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. O sistema deve possibilitar então que o médico selecione o idoso entre os resultados encontrados, carregando o prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos) bem como as patologias que já estão associadas a ele. A listagem das patologias será em formato de histórico, no qual, apresentará o</p>

	<p>nome, data de incidência da patologia, grau e observações iniciais, a listagem será organizada pela data mais recente que foram associadas, em formato de tabela para melhor distinguir tais patologias. Após isso, o sistema deverá possibilitar a seleção de uma dessas patologias, imprimindo na tela dados acerca da patologia e do idoso (nome do idoso, idade do idoso, sexo do idoso, peso do idoso, nome da patologia, data de incidência, grau e observações iniciais). Nesta mesma tela, o sistema exibir, em formato de tabela, o registro da evolução da patologia do idoso (data de atualização e observações realizadas), e permitir a inserção de novos dados relativos à evolução da patologia do idoso (isto é, uma “atualização” de seu estado), imprimindo na tela e permitindo a atualização do campo “Grau” e “Nome”, registrando a data/hora da atualização, que deverá ser a data/hora do dispositivo do usuário no momento em que ela ocorrer. Para registrar o histórico, o sistema deve replicar, quando a primeiro registro de evolução for realizado, os dados cadastrados no RF#07 para outra tabela na base de dados, que armazenará toda atualização realizada pelo usuário, criando um novo registro. Além disso, o sistema deverá permitir a inserção de observações para cada atualização, sendo que “observações” é de preenchimento opcional:</p> <ul style="list-style-type: none"> <li>• Obs.: varchar (500).</li> </ul>
RF #11	<p>O sistema deverá permitir o gerenciamento das Prescrições Médicas do Idoso, possibilitando, primeiramente, o cadastro de tais prescrições (isto é, associar uma prescrição com um idoso específico já cadastrado no sistema). Para isso, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. Após a seleção do idoso dentre uma lista de resultados apresentados, o sistema deve carregar um prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos), bem como uma lista com as prescrições atuais (isto é, prescrições que já haviam sido feitas anteriormente). O Sistema deverá oferecer então a opção “inserir nova”, que ao ser acessada carregará um formulário para a inserção de uma nova prescrição. Toda prescrição poderá ter mais de um medicamento associado ao idoso, essa associação será feita pela inserção de um novo medicamento na “ficha” de prescrição médica do idoso naquela consulta. Esta associação será realizada pela pesquisa do medicamento no sistema, através de dados que podem ser o nome comercial ou o nome químico (texto inserido pode estar em qualquer parte do nome completo), assim, o medicamento será selecionado dentre uma lista de resultados encontrados, organizada pela ordem alfabética dos nomes dos medicamentos sendo expresso na tela em formato de tabela dados referentes aquele medicamento que foi inserido (os dados serão nome comercial, nome químico, tipo e via). Para o cadastramento da prescrição do sistema, deverá ser registrado a data/hora da prescrição médica atribuída ao idoso, e, posteriormente, são necessários os dados: medicamento (nome), posologia, validade da prescrição e observações, sendo todos os campos de preenchimento obrigatório:</p> <ul style="list-style-type: none"> <li>• Horário e data da prescrição médica – datetime;</li> <li>• Medicamento – varchar (100);</li> <li>• Posologia – varchar (200);</li> <li>• Validade da Prescrição – varchar (20);</li> <li>• Observações – varchar (500).</li> </ul>
RF#12	<p>O Sistema deverá permitir o gerenciamento das Prescrições Médicas do idoso, possibilitando a edição de prescrições já realizadas (conforme descrito no</p>

	<p>RF#11). Da mesma maneira que o requisito anterior, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. Após a seleção do idoso dentre uma lista de resultados apresentados, o sistema deve carregar um prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos), bem como uma lista com as prescrições atuais (isto é, prescrições que já haviam sido feitas anteriormente). O Sistema deverá então oferecer ao usuário a opção de editar uma das prescrições médicas já existentes, permitindo a alteração dos dados dos campos posologia, observações e medicamento, permitindo a opção da edição do medicamento associado no RF#11 fornecendo novamente a opção de uma nova pesquisa por medicamento, além disso o sistema deverá registrar a data/hora da edição das prescrições médicas atribuída ao idoso. Sendo todos os campos de preenchimento obrigatório, enquanto os campos medicamento e validade da prescrição permanecerão inalterados:</p> <ul style="list-style-type: none"> <li>• Horário e data da prescrição médica – datetime;</li> <li>• Medicamento – varchar (100);</li> <li>• Posologia – varchar (200);</li> <li>• Observações – varchar (500).</li> </ul>
RF #13	<p>O sistema deverá permitir o gerenciamento das Análises Clínicas Periódica por Idoso. Para isso, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (sendo que o texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. O sistema deve possibilitar então que o médico selecione o idoso entre os resultados encontrados, carregando o prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos) bem como as análises clínicas periódicas que já foram realizadas anteriormente. Para registrar o histórico, o sistema deve replicar, quando a primeiro registro for realizado, os dados cadastrados no RF#12 para outra tabela na base de dados, que armazenará toda atualização realizada pelo usuário, criando um novo registro. Com o idoso selecionado o sistema deverá possibilitar o registro de uma nova análise clínica periódica com a data/hora do novo requerimento dos exames periódicos atribuída ao idoso, e, posteriormente, são necessários os seguintes dados que especificam melhor a condição da consulta do idoso: O tipo (qual o tipo de doença o idoso tem, dentre as possibilidades daquela patologia), a descrição das patologias e observações, sendo todos os campos de preenchimento obrigatório:</p> <ul style="list-style-type: none"> <li>• Horário e data de requerimento da análise clínica periódica – datetime;</li> <li>• Tipo – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Descrição das patologias – varchar (500);</li> <li>• Obs.: varchar (500).</li> </ul>
RF #14	<p>O Sistema deverá permitir o gerenciamento das Análises Clínicas Periódica por Idoso, possibilitando a edição das análises clínicas periódicas já realizadas (conforme descrito no RF#10). Da mesma maneira que o requisito anterior, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. Após a seleção do idoso dentre uma lista de resultados apresentados, o sistema deve carregar um prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos), bem como as análises clínicas periódicas que já estão</p>

	<p>associadas a ele. A listagem das análises clínicas será em formato de histórico, no qual, apresentará todos os campos que foram preenchidos no registro, o tipo (qual o tipo de doença o idoso tem, dentre as possibilidades daquela patologia), a descrição das patologias, as observações e a data de requerimento da análise clínica periódica, a listagem será organizada pela data mais recente que foram associadas. Após isso, o sistema deverá possibilitar a alteração dos dados relativos aos registros, sendo disponível ao lado de cada registro a opção de alteração referente aquele registro, imprimindo na tela e permitindo a atualização dos campos (descrição das patologias e observações) registrando a data/hora da atualização, que deverá ser a data/hora do dispositivo do usuário no momento em que ela ocorrer. Além disso, o sistema deverá permitir a inserção de observações para cada atualização, sendo que “observações” é de preenchimento opcional, enquanto os campos data de requerimento dos exames periódicos e tipo permanecerão inalterados:</p> <ul style="list-style-type: none"> <li>• Horário e data de requerimento da análise clínica periódica – datetime;</li> <li>• Tipo – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Descrição das patologias – varchar (500);</li> <li>• Obs.: varchar (500).</li> </ul>
RF #15	<p>O sistema deverá permitir o gerenciamento do Registro de Incidentes na Instituição. Para isso, o sistema deve efetuar uma pesquisa pelo idoso, através de dados que podem ser o nome (texto inserido pode estar em qualquer parte do nome completo) ou o prontuário / campo de identificação. Após a seleção do idoso dentre uma lista de resultados apresentados, o sistema deve carregar um prontuário com os seus dados (campo identificador, nome completo, sexo, idade, peso, altura, alergias, medicamentos contínuos), posteriormente, deverá ser registrado a data/hora do acontecimento do registro de incidente referente aquele idoso selecionado, imprimindo o formulário a ser preenchido para a conclusão do registro. Serão necessários preencher os seguintes campos para a realização do cadastro: o tipo do incidente (mortalidade, doença, diarreica, escabiose, úlcera, desidratação, desnutrição, queda), número de registros de incidente, descrição sobre o incidente, nome e prontuário do paciente, sendo todos de preenchimentos obrigatórios:</p> <ul style="list-style-type: none"> <li>• Número de registro de incidente – int ;</li> <li>• Tipo do incidente – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Nome do paciente – Lista pré-cadastrada a ser selecionada via Combobox;</li> <li>• Descrição sobre o incidente – varchar (500);</li> <li>• Horário e data do incidente – datetime.</li> </ul>
RF #16	<p>O Sistema deverá permitir o gerenciamento do Registro de Incidentes na Instituição, possibilitando a edição dos incidentes já realizados (conforme descrito no RF#15). O sistema deve efetuar uma pesquisa pelo incidente, através de dados que podem ser o nome (texto inserido pode estar em qualquer parte do nome completo) ou o número de registro do incidente. Após a seleção do incidente dentre uma lista de resultados apresentados, o sistema deve carregar os dados daquele incidente (o nome do idoso que ocorreu aquele incidente, o número de registro, hora e data da ocorrência do incidente). A listagem dos incidentes será em formato de histórico, no qual, apresentará o nome do idoso e a data de ocorrência do incidente, a listagem será organizada pela data mais recente que foram associadas, em formato de tabela para melhor distinguir tais</p>

	<p>incidentes. Após isso, o sistema deverá possibilitar a alteração dos dados relativos aos registros, sendo disponível ao lado de cada registro a opção de alteração referente aquele registro, imprimindo na tela e permitindo a atualização dos campos (tipo de incidente e nome do paciente) registrando a data/hora da atualização, que deverá ser a data/hora do dispositivo do usuário no momento em que ela ocorrer. Além disso, o sistema deverá permitir a inserção de observações para cada atualização, sendo que “observações” é de preenchimento opcional:</p> <ul style="list-style-type: none"><li>• Horário e data do incidente – datetime;</li><li>• Tipo de incidente – Lista pré-cadastrada a ser selecionada via Combobox;</li><li>• Nome do paciente – varchar (500);</li><li>• Observação – varchar (500).</li></ul>																																																			
RF #17	<p>O sistema deverá fazer o controle de acesso, sendo que somente enfermeiros e médicos devem acessar a plataforma. Esse controle será realizado pela análise da tabela onde será atribuído a cada usuário seu respectivo papel dentro do sistema, essa priorização de papéis será por atributos que foram definidos no cadastro de cada usuário (enfermeiro, médico, nutricionista, educador físico, responsável pelo idoso, gestores ou equipe de limpeza). Após essa análise, será feito o controle de acesso, que será através de um prontuário ou senha atribuídos e cadastrados no módulo 1. Após isso, será exibido os requisitos respectivos aos enfermeiros e aos médicos, abaixo está melhor descrito quais os requisitos e funções que cada usuário poderá realizar.</p> <p>No RF#15 deverá ser restrito a inserção de mortalidade somente para o médico</p> <table><tr><td></td><td>Enfermeiro</td><td>Médico</td></tr><tr><td>RF#01</td><td>X</td><td>X</td></tr><tr><td>RF#02</td><td>X</td><td>X</td></tr><tr><td>RF#03</td><td>X</td><td>X</td></tr><tr><td>RF#04</td><td>X</td><td>X</td></tr><tr><td>RF#05</td><td>X</td><td>X</td></tr><tr><td>RF#06</td><td>X</td><td>X</td></tr><tr><td>RF#07</td><td></td><td>X</td></tr><tr><td>RF#08</td><td></td><td>X</td></tr><tr><td>RF#09</td><td></td><td>X</td></tr><tr><td>RF#10</td><td></td><td>X</td></tr><tr><td>RF#11</td><td></td><td>X</td></tr><tr><td>RF#12</td><td></td><td>X</td></tr><tr><td>RF#13</td><td>X</td><td>X</td></tr><tr><td>RF#14</td><td>X</td><td>X</td></tr><tr><td>RF#15</td><td>X</td><td>X</td></tr><tr><td>RF#16</td><td>X</td><td>X</td></tr></table>		Enfermeiro	Médico	RF#01	X	X	RF#02	X	X	RF#03	X	X	RF#04	X	X	RF#05	X	X	RF#06	X	X	RF#07		X	RF#08		X	RF#09		X	RF#10		X	RF#11		X	RF#12		X	RF#13	X	X	RF#14	X	X	RF#15	X	X	RF#16	X	X
	Enfermeiro	Médico																																																		
RF#01	X	X																																																		
RF#02	X	X																																																		
RF#03	X	X																																																		
RF#04	X	X																																																		
RF#05	X	X																																																		
RF#06	X	X																																																		
RF#07		X																																																		
RF#08		X																																																		
RF#09		X																																																		
RF#10		X																																																		
RF#11		X																																																		
RF#12		X																																																		
RF#13	X	X																																																		
RF#14	X	X																																																		
RF#15	X	X																																																		
RF#16	X	X																																																		
RN #18	Todos as vacinas cadastradas devem ser regulamentados pela Agência Nacional de Vigilância Sanitária (Anvisa).																																																			
RN #19	Todos os medicamentos cadastrados devem ser regulamentados pela Agência Nacional de Vigilância Sanitária (Anvisa).																																																			
RN #20	Todos os médicos devem ser validados pelo Conselho Regional de Medicina																																																			



	(CRM).
RN #21	Todos os enfermeiros devem ser validados pelo Conselho Federal de Enfermagem (Cofen).
RN #22	O desenvolvimento do módulo deverá utilizar programas como o MySQL e linguagens como PHP e HTML.
RN #23	O Sistema deve processar todas as operações em um tempo inferior a 5s;
RN #24	O Sistema deve ser compatível para Web.
RN #25	O Sistema deve ser completamente responsivo (rodar tanto em desktops quanto em dispositivos menores como <i>smartphones</i> e <i>tablets</i> ).

Fonte: elaboração própria

Tabela 10: Documento de requisitos do módulo 6

Identificador	Descrição do Requisito
RF #01	<p>O sistema deve permitir que o nutricionista cadastre os tipos de alimentos com os metadados código (auto-incremento, inteiro) e nome (varchar 50). A atualização só pode ser feita apenas se a alteração ocorrer no campo nome. A exclusão só poderá ocorrer se não houver nenhuma dependência entre tipos de alimentos com alimentos. Na listagem dos tipos de alimentos deverá ser gerado uma tabela com o nome do tipo de alimento e nas outras colunas a possibilidade de excluir e editar. Nesta listagem, pode ser feita uma pesquisa pelo nome do tipo do alimento pertencendo a qualquer parte do nome.</p>
RF #02	<p>O sistema deve permitir que o nutricionista cadastre os alimentos com os metadados código (auto-incremento, inteiro), nome (varchar 50), escolha do tipo de alimento em um combobox com os dados já previamente cadastrados (inteiro), caloria (float), quantidade do alimento (float), unidade (combobox listando unidades pré-cadastradas no banco de dados). Esse cadastro só poderá ocorrer se houver algum tipo de alimento relacionado ao alimento. A atualização não poderá ocorrer no código, os demais campos podem ser editados. A exclusão só poderá ocorrer se não houver nenhuma dependência entre alimentos com planos alimentares. Na listagem dos alimentos deverá ser gerado uma tabela com todos os campos com exceção do código e nas outras colunas a possibilidade de excluir e editar. Nesta listagem, pode ser feita uma pesquisa pelo nome do alimento pertencendo a qualquer parte do nome.</p>
RF #03	<p>O sistema deve permitir que o nutricionista monte 6 refeições diárias colocando as datas de início e término do período do plano alimentar semanal. Para o cadastro de cada refeição o nutricionista deverá selecionar os alimentos previamente cadastrados (combobox), a quantidade (float), um campo de observação onde deverá haver alimentos alternativos para idosos que possuam algum tipo de restrição (varchar 255) e um campo com a somatória automática da caloria de todos os alimentos referentes a essa refeição. No cadastro alimentar de um dia irá conter uma tabela com 6 colunas referentes as refeições (café da manhã, lanche da manhã, almoço, lanche da tarde, jantar e ceia). Após a inserção de todos os alimentos, será gerado uma nova coluna com a somatória das calorias de cada refeição referente as calorias consumidas do dia. O cardápio alimentar só será salvo quando o plano alimentar de todos os dias da semana estiver preenchido.</p>

	<p>Na listagem haverá um campo de pesquisa onde será colocado o período (início e término), uma tabela irá ser gerada com o plano referente a cada dia da semana. Em cada tabela haverá 6 colunas que conterà os alimentos, um campo de observação e a somatória das calorias de cada refeição e uma coluna com a somatória das calorias do dia. Em cada tabela haverá um botão de editar vinculado a ela.</p> <p>A atualização só poderá ocorrer antes da refeição ser realizada e deverá ser informada com uma semana de antecedência, não podendo ter alteração no campo do código. Não poderá ser realizado a exclusão dos cardápios.</p>
RF #04	<p>O nutricionista deve agendar uma consulta por semana para cada idoso da instituição, cada consulta deve ser registrada com a data e hora (datetime, dd-mm-aaaa, hh:mm), local (varchar 10), um código de consulta (auto-incremento, inteiro) e o nome do idoso (combobox com os nomes já cadastrados dos pacientes).</p> <p>A atualização não poderá ocorrer no código, os demais campos podem ser editados, a atualização não poderá ser realizada se a consulta já aconteceu.</p> <p>Na listagem das consultas haverá um campo de pesquisa onde o nutricionista poderá pesquisar pelo nome do idoso e todas as consultas referentes a esse idoso aparecerão. Cada agendamento de consulta será exibido em uma tabela com todos os campos (com exceção do código), com a opção de atualizar agendamento e dados da consulta. A consulta não poderá ser excluída.</p>
RF #05	<p>O nutricionista deve preencher uma ficha referente aos dados da consulta que possuirá um campo de texto (varchar 250) para que o nutricionista registre o que aconteceu na consulta (dados/histórico) e um outro campo de texto (varchar 250) para que o idoso de sua opinião sobre a alimentação. A opinião deve ser preenchida pelo nutricionista. Esses dados não poderão ser alterados nem excluídos.</p>
RF #06	<p>O sistema deve manter um controle das amostras dos alimentos durante 7 dias para que em caso de intoxicação as devidas providencias sejam tomadas, fazendo um cadastro contendo as seguintes informações:</p> <ul style="list-style-type: none"> <li>• Nome do alimento (varchar 50);</li> <li>• Data de entrada (datetime, dd-mm-aaaa, hh:mm);</li> <li>• Saída (datetime, dd-mm-aaaa, hh:mm);</li> <li>• Data da Refeição (date);</li> <li>• Qual Refeição do Dia (combobox);</li> <li>• Foto (file).</li> </ul> <p>Na listagem das amostras, deve apresentar todos os campos. Na frente de cada amostra, aparece um botão permitindo a sua alteração. Não pode alterar o código. A edição deve ser permitida. A exclusão não é permitida no prazo de 02 anos desde a data de entrada da amostra. Após 7 dias as amostras serão descartadas, mas os dados devem ser armazenados por 2 anos.</p>
RF #07	<p>O sistema deverá dar acesso aos dados das refeições para os enfermeiros, para que as refeições sejam preparadas. Haverá uma lista com valores pré-estabelecidos que será feita através de um select onde o enfermeiro irá selecionar a opção referente ao status de alimentação do idoso (pouca aceitação, média aceitação, boa aceitação ou não aceitou). Também existirá um campo de observações (varchar 100) para a inserção de dados dos alimentos consumidos e o nome do idoso (combobox com o nome dos idosos já cadastrados). Haverá uma listagem com todos os campos referentes ao padrão alimentar, e com as opções de exclusão e alteração, onde somente os enfermeiros poderão alterar e</p>

	excluir, o nutricionista só poderá visualizar a listagem.
RN #08	Processo de Software: É necessário que o Processo de Software a ser utilizado seja o RUP + Kanban.
RN #09	Tecnologia: PHP, Bootstrap, Javascript e AJAX.
RN #10	Design Pattern MVC: Será utilizado o design pattern MVC. Esse padrão de projeto será adotado pois facilita a reutilização do código e facilita também a forma com que a implementação será gerenciada, permitindo que cada camada possa ser construída de forma quase que completamente independente. Não será utilizada nenhuma ferramenta de geração de código.
RN #11	Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line: É necessário disponibilizar, através da Web, a documentação do desenvolvimento do Protótipo do de Projeto de SGI, para propiciar a troca de informações e experiências entre todos os grupos participantes da disciplina.
RN #12	Este sistema deverá ser capaz de propiciar uma interface agradável ao usuário, com opções simples, e autoexplicativas.
RN #13	O sistema não deve a cada funcionalidade ser executado com tempo superior a 3 segundos por transação.
RN #14	O sistema deve controlar o acesso as funcionalidades do módulo através do <i>login</i> (usuário e senha) que será realizado pelos enfermeiros e nutricionistas.
RN #15	Portabilidade: Edge e Chrome.

Fonte: elaboração própria

Tabela 11: Documento de requisitos do módulo 7

Identificador	Descrição do Requisito
RF #01	<p>O educador físico para ter acesso às informações do idoso, terá que fazer <i>login</i> na aba LOGIN através do prontuário e da senha.</p> <ul style="list-style-type: none"> <li>• Prontuário do educador (char (7))*;</li> <li>• Senha do educador (varchar (20)).</li> </ul> <p>Uma vez acessado, terá acesso à página inicial. Com as seguintes funcionalidades:</p> <ul style="list-style-type: none"> <li>• Lista de idosos (direcionado para outra aba com a opção de busca por idoso);</li> <li>• Atividades Recreativas (direcionado para a aba de Gerenciamento das atividades, do plano mensal (realizado em grupo), e o registro de participação nas atividades (descrição visual e textual));</li> <li>• Atividades Físicas (direcionado para a aba de Gerenciamento dos exercícios, registro das consultas periódicas);</li> <li>• Chat (servindo como um lembrete para o educador físico).</li> </ul> <p>* = Obrigatório</p>
RF #02	<p>Na janela que diz respeito ao chat, que se encontrara possivelmente no canto inferior direito da tela, haverá informações voltadas ao idoso, de alerta informando:</p> <ul style="list-style-type: none"> <li>• Data de atividade recreativa/física*;</li> </ul>

	<ul style="list-style-type: none"> <li>• Nome do idoso*;</li> <li>• Prontuário do idoso*;</li> <li>• Dúvidas.</li> </ul> <p>* = Obrigatório</p>
RF #03	<p>Encaminha para uma aba com uma lista de idosos residentes nessa instituição. No começo da página haverá um campo para a inserção do prontuário do idoso, que quando inserido o sistema irá direcionar para o idoso específico. Pela utilização do slidedown, ao passar o cursor do mouse no nome do idoso devem ser apresentados as seguintes informações:</p> <ul style="list-style-type: none"> <li>• Foto*;</li> <li>• Nome (varchar (100))*;</li> <li>• Idade (char (3))*.</li> </ul> <p>Quando dado um click no nome do idoso, deverá atualizar a página a partir da mesma self, apresentando, além das informações já citadas:</p> <ul style="list-style-type: none"> <li>• CPF (varchar (14))*;</li> <li>• Peso (float (xxx,x kg))*;</li> <li>• Altura (float (x,xx m))*;</li> <li>• Idade (char (3))*;</li> <li>• Estado civil (varchar (15));</li> <li>• Data de entrada na instituição <input type="checkbox"/> date (xxxx/xx/xx)*.</li> </ul> <p>Sendo apresentadas, anotações básicas sobre o idoso (informações pessoais), o usuário do sistema deverá ter acesso a partir dessa aba (aba do idoso) para a janela de atividades recreativas e físicas.</p> <p>* = Obrigatório</p>
RF #04	<p>Na janela que diz respeito às atividades recreativas, que deverá ser dividida em duas partes. De início uma área voltada ao registro de participação nas atividades de recreação, enfim, seguindo para a definição do plano mensal das atividades recreativas.</p> <p>Registro de participação: cabe ao usuário digitalizar informações voltadas a atividade do dia realizada em grupo ou individualmente.</p> <p>Plano mensal: detalha todas as atividades de recreação acontecidas em grupo, programadas ao longo de todo o mês.</p>
RF #05	<p>Após ter sido realizada qualquer atividade recreativa em questão, o usuário deverá fazer uma anotação, no momento da atividade, e repassar para o sistema. Segue abaixo dados que deverão ser preenchidos pelo usuário de forma a concluir o registro de participação dos idosos:</p> <ul style="list-style-type: none"> <li>• Data ((date (XXXX/XX/XX)) *);</li> <li>• Hora (float (xx:xx))*;</li> <li>• Atividade (ex: bingo, dominó entre outras opções que a instituição disponibilizar) (varchar (50))*.</li> </ul> <p>Deverá ser possível também registrar, individualmente, cada um dos idosos que participaram da atividade. Sendo assim, uma lista com o nome de todos os idosos da instituição deverá estar presente nessa área destinada a registros e ao lado, uma de participação.</p> <p>Vale ressaltar, que uma lista de presença encontrada ao lado direito da lista de todos os residentes na instituição em questão obrigatoriamente deverá ser preenchida após todas as atividades ocorridas. A partir de um checkbox o educador (usuário responsável) tenderá a anotar todas as presenças e faltas de cada idoso.*</p> <p>Essa aba poderá contar ainda com um campo destinado para fotos, sendo de</p>

	<p>total responsabilidade do educador registrá-la. Por fim, dois botões ao final desse espaço deverão estar presentes, com o intuito de que fique salvo as alterações feitas segue abaixo botões de ordem obrigatória:</p> <ul style="list-style-type: none"> <li>• Salvar*;</li> <li>• Cancelar*.</li> </ul> <p>* = Obrigatório</p>
RF #06	<p>Para que haja uma melhor definição das atividades realizadas no momento de recreação, uma aba deverá ser reservada. O plano mensal dessas atividades será distribuído em tabela com os campos a seguir:</p> <ul style="list-style-type: none"> <li>• Data da atividade (date (XXXX/XX/XX))*;</li> <li>• Hora (float (xx:xx))*;</li> <li>• Atividade recreativa (ex: bingo, dominó entre outras opções que a instituição disponibilizar que deverão ser registradas em uma janela específica) (varchar (50))*;</li> <li>• Duração (definido em formato de hora) ((varchar (10)))*.</li> </ul> <p>Ao final de todas o registro de dados, o usuário irá disponibilizar de dois botões, um de inserção “inserir” e um para cancelamento “cancelar”. *</p> <p>* = Obrigatório</p>
RF #07	<p>Na janela que diz respeito ao cadastro das atividades físicas, deverá ser apresentada a seguinte tabela em listagem (1,2,3,4,5,6, ...):</p> <p>Nome da atividade*</p> <p>Tipo da atividade*</p> <p>Apresentando também as seguintes funcionalidades:</p> <p>Exclusão → irá gerar um alert com a seguinte mensagem “Você realmente deseja excluir as informações selecionadas?”, com os botões “sim” ou “não”. *</p> <p>Inserção → para inserir uma nova atividade física, deverão ser informados: nome, tipo de atividade e foto, com os botões “inserir” ou “cancelar”. *</p> <p>Atualização → um dado só será atualizado a partir da abertura de um alert, com os dados já cadastrados no momento da inserção, sendo exibido: nome, tipo de atividade e foto. E, logo, abaixo dois botões, um de confirmação “confirmar” e outro de cancelamento “cancelar”. *</p> <p>Além desses tópicos, a tabela deverá ter um campo de inserção de nome, para filtrar a informação que o usuário necessita, facilitando no manuseio do sistema e economizando o tempo do usuário. E juntamente, um campo para inserção de foto, dessa forma, o educador físico deixa mais claro qual a atividade escolhida.</p> <p>*</p> <p>* = Obrigatório</p>
RF #08	<p>Na janela que diz respeito ao cadastro das atividades recreativas, deverá ser apresentada a seguinte tabela em listagem (1,2,3,4,5,6, ...):</p> <p>Nome da atividade*</p> <p>Tipo da atividade*</p> <p>Apresentando também as seguintes funcionalidades:</p> <p>Exclusão → irá gerar um alert com a seguinte mensagem “Você realmente deseja excluir as informações selecionadas?”, com os botões “sim” ou “não”.</p> <p>Inserção → para inserir uma nova atividade recreativa, deverão ser informados: nome, tipo de atividade e foto, com os botões “inserir” ou “cancelar”. *</p> <p>Atualização → um dado só será atualizado a partir da abertura de um alert, com os dados já cadastrados no momento da inserção, sendo exibido: nome, tipo de atividade e foto. E, logo, abaixo dois botões, um de confirmação “confirmar” e outro de cancelamento “cancelar”.</p>

	<p>Além desses tópicos, a tabela deverá ter um campo de inserção de nome, para filtrar a informação que o usuário necessita, facilitando no manuseio do sistema e economizando o tempo do usuário. E juntamente, um campo para inserção de foto, dessa forma, o educador físico deixa mais claro qual a atividade escolhida.</p> <p>* = Obrigatório</p>
RF #09	<p>Todas as funcionalidades descritas no requisito funcional 03 (RF #03), somente e deverão ser modificadas por pessoas específicas, tendo em vista, que o sistema terá que ser controlado para que não haja alguma alteração indevida. Sendo assim, a inserção, a alteração e exclusão de dados das atividades recreativas caberão ao preparador físico ou fisioterapeuta. Segue abaixo, alguns detalhes que o usuário deverá estar atento.</p> <p>Prontuário do educador (char (7))*</p> <p>Somente com prontuário e senha o educador físico ou ainda, fisioterapeuta poderá alterar ou inserir qualquer que seja a informação. Esse passará por uma verificação, sendo liberado ou não o acesso. Se sim, segue normalmente o fluxo do sistema.</p> <p>Login → Página inicial → Atividades físicas/ recreativas</p> <p>Prontuário do idoso (char (7))*</p> <p>Para ser realizada a funcionalidade escolhida pelo usuário (alteração ou inclusão) o mesmo deverá ter disponível no prontuário do idoso, assim todas as suas tarefas recreativas serão selecionadas e a escolhida poderá então ser mexida com a total precisão.</p> <p>Uma vez alterada, qual seja a informação o mesmo não poderá recuperar os dados. Sendo assim, o usuário deverá estar atento ao que mexe e o quanto é valiosa a informação.</p> <p>A cada alteração no banco de dados, o educador deverá acrescentar alguma:</p> <ul style="list-style-type: none"> <li>• Observação (varchar (500))*.</li> </ul> <p>Para descrever detalhes que não puderam ser informados nas categorias anteriores, como por exemplo, “o idoso X reclamou de dor na perna enquanto realizava Y atividade”.</p> <p>* = Obrigatório</p>
RF #10	<p>Na janela que diz respeito à consulta, terá três tópicos, que são:</p> <p>Recente (apresentará informações das consultas recentes, que vão ser as consultas realizadas nos últimos 7 dias, assim o sistema fará essa atualização. Essas informações serão apresentadas em uma tabela com os seguintes dados):</p> <ul style="list-style-type: none"> <li>• Peso (float (xxx,x kg));</li> <li>• Altura (float (x,xx m));</li> <li>• IMC (float (xx,xx kg/m));</li> <li>• Será gerado automaticamente e quando o peso e altura forem inseridos;</li> <li>• Pressão (float (xx,x mmhg));</li> <li>• Envergadura (float (x,xx m));</li> <li>• Observações adicionais (varchar (500)).</li> </ul> <p>Nota: o médico vai inserir as informações que ele obteve do idoso naquela consulta, que será:</p> <ul style="list-style-type: none"> <li>• Peso (float (xxx,x kg));</li> <li>• Altura (float (x,xx m));</li> <li>• IMC (float (xx,xx kg/m)).</li> </ul>

	<p>Será gerado automaticamente quando o peso e altura forem inseridos:</p> <ul style="list-style-type: none"> <li>• Pressão (float (xx,x mmhg));</li> <li>• Envergadura (float (x,xx m));</li> <li>• Observações adicionais (varchar (500)).</li> </ul> <p>Esse campo servirá para que o médico insira informações adicionais, se caso o idoso apresentar algum resultado diferente do que geralmente costuma ter das informações citadas acima, apresentar sintomas anormais, etc.</p> <p>Histórico <input type="checkbox"/><input type="checkbox"/> apresentará as informações de todas as consultas que foram feitas desde que o idoso entrou na instituição. Essas informações serão apresentadas em uma tabela com os seguintes dados:</p> <ul style="list-style-type: none"> <li>• Peso (float (xxx,x kg));</li> <li>• Altura (float (x,xx m));</li> <li>• IMC (float (xx,xx kg/m)).</li> </ul> <p>Será gerado automaticamente quando o peso e altura forem inseridos:</p> <ul style="list-style-type: none"> <li>• Pressão (float (xx,x mmhg));</li> <li>• Envergadura (float (x,xx m));</li> <li>• Observações adicionais (varchar (500)).</li> </ul>
RF #11	<p>As atividades físicas regulares, além de serem registradas pelo educador, precisam constar sua data de início, data prevista de término, e o objetivo para qual está sendo realizada. Além disso, será necessário um select de sim ou não referente à participação do idoso na atividade física.</p> <p>Todas as respostas dos selects serão armazenadas em uma tabela, e ao final do mês, através desses dados, será contabilizada a frequência percentual desse idoso nas atividades físicas da instituição:</p> <ul style="list-style-type: none"> <li>• Nome (varchar (100));</li> <li>• Prontuário (varchar (xxx.xxx.xxx-xx));</li> <li>• Data de início (date (XXXX/XX/XX))*;</li> <li>• Data prevista de término (date (XXXX/XX/XX))*;</li> <li>• Objetivo (varchar (500))*.</li> </ul> <p>O objetivo terá uma barra de progresso, cuja data determinará o quão próximo o objetivo está de ser realizado. Assim, o educador terá uma visão mais ampla de como está a situação de cada idoso.</p> <p>* = Obrigatório</p>
RF #12	<p>Uma janela específica para o treino dos idosos deverá estar presente no sistema, com todas as funcionalidades necessárias para que se tenha um fluxo. Cabe ao educador físico registrar as atividades por idoso, sendo necessário:</p> <ul style="list-style-type: none"> <li>• Prontuário (select)*;</li> <li>• Dia da semana (select)*;</li> <li>• Atividade (select)*;</li> <li>• Repetições (varchar (100))*.</li> </ul> <p>As elaborações dos treinos deverão ser feitas ao longo da consulta periódica destinada a cada idoso. Dessa forma o avaliador, já terá uma noção de qual é a situação do idoso e de qual exercício em específico aquele idoso necessita. Segue abaixo um exemplo de treino, para melhor entendimento:</p> <p>Exemplo de Treino:</p> <p>Idoso: Tio Breno</p> <p>Atividade 01: Caminhada / 02 vezes por semana / 10 minutos por dia</p> <p>Atividade 02: Alongamento do pescoço / 01 vezes por semana / 10 vezes por dia</p>

	Atividade 03: Alongamento do joelho / 03 vezes por semana / 05 vezes por dia * = Obrigatório
RF #13	<p>O sistema deverá controlar os privilégios de acesso às funcionalidades desse módulo, ou seja, o acesso que o educador físico terá não será o mesmo que um usuário comum.</p> <p>Educadores e fisioterapeutas: podem incluir, atualizar, modificar, excluir, selecionais quaisquer dados referentes aos exercícios físicos, atividades recreativas e treinos, tendo total liberdade de modificação com exceção dos dados pessoais do idoso, como nome, CPF e RG.</p> <p>Usuário comum: poderá apenas visualizar e selecionar, e caso tenha alguma dúvida de modificação dos dados, deve entrar em contato com a instituição pelo formulário de contato.</p>
RN #14	Processo de Software: É necessário que o Processo de Software a ser utilizado seja o Redmine, SVN e Kanban.
RN #15	Linguagem PHP: É necessário que se adote a linguagem PHP, pois ela facilita no processo de programação.
RN #16	O sistema deverá encerrar a sessão após 25 minutos sem a ação do usuário.
RN #17	É necessário disponibilizar, através da Web, a documentação do desenvolvimento do Protótipo do Projeto de AFR, para propiciar a troca de informações e experiências entre todos os grupos participantes da disciplina.
RN #18	O sistema deverá ser capaz de propiciar uma interface agradável ao usuário, com opções simples, e autoexplicativas.
RN #19	O sistema deverá gerar relatórios, tais como: das atividades recreativas em grupo, atividades físicas, das consultas periódicas, etc., tudo com base nas informações inseridas pelo usuário.

Fonte: elaboração própria

Tabela 12: Documento de requisitos do módulo 8

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.</p> <p>O planejamento semanal deverá ser criado pelo funcionário da limpeza que tenha sido escolhido como representante pela equipe. Para ele poder cadastrar o planejamento deverá aparecer uma tela com os seguintes campos nesta respectiva ordem:</p> <ul style="list-style-type: none"> <li>• Tabela: Planejamento Semanal</li> <li>• Nome da tarefa*: Varchar (50);</li> <li>• Descrição da tarefa*: Varchar (250);</li> <li>• Data de realização: Date-time;</li> <li>• Status da tarefa* (concluída ou não): Boolean (deverá indicar se está selecionado ou não).</li> <li>• Funcionário responsável pelo setor*: Varchar (100);</li> <li>• Nome do setor*: Varchar (50);</li> </ul>



	<ul style="list-style-type: none"> <li>• Observações: Varchar (500).</li> </ul> <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 acessar essa página o usuário deverá clicar na segunda opção do menu retrátil.</p> <p>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"> <li>• A primeira linha da tabela conterá os nomes dos dados: “código”, “nome da tarefa”, “data da tarefa”, “setor” e “configurações”;</li> <li>• A primeira coluna deverá ser ocupada pelos códigos (autoincrement);</li> <li>• Nome da tarefa;</li> <li>• Data de realização;</li> <li>• Nome do setor.</li> </ul> <p>Dos ícones impressos na tela sendo eles para editar e outro para deletar.</p>
RF #02	<p>Planejamento semanal: O sistema deverá conter uma opção do menu retrátil para a limpeza. Ao clicar nesse item o usuário deve ser direcionado para uma página onde deverá aparecer todos os setores da instituição que serão limpos e permitir que o gestor e os outros funcionários visualizem o planejamento semanal de cada um dos setores que o funcionário da limpeza representante da equipe criou (select). O planejamento será feito a partir de uma tabela, que aparecerá no formato de lista para o usuário. Nessa lista aparecerá o nome do funcionário responsável pela limpeza do setor e a tarefa que deve ser realizada (indicando se devem haver produtos específicos que não podem ser utilizados, se há algum cuidado especial com o setor, o que não deve ser movido, etc):</p> <ul style="list-style-type: none"> <li>• O menu receberá o nome “controle administrativo” deverá aparecer quando o cursor do mouse passar em cima dele;</li> <li>• Na tabela deve aparecer, respectivamente:</li> <li>• Nome do setor*: Varchar (50);</li> <li>• Data de limpeza: Date;</li> <li>• Descrição de como deve ser realizada a limpeza*: Varchar (250);</li> <li>• Funcionário responsável pelo setor: Varchar (100).</li> </ul> <p>O funcionário deve logar no sistema e acessar a janela de limpezas para poder marcar uma tarefa como realizada. O sistema deverá imprimir na tela uma tabela com todas as tarefas que estão destinadas a este funcionário específico. O gestor tem acesso para visualizar todos os planejamentos semanais.</p>
RF #03	<p>O sistema deverá notificar o gestor quando todas as alas forem selecionadas (boolean) para que um novo planejamento seja feito. A mensagem deve aparecer como um pop-up na tela (utilizando o recurso Dismissing), quando ele acessar a aba de planejamento. Para que o gestor e o representante da limpeza saibam que existe um aviso, deverá aparecer uma bolinha de notificação na frente do nome da ala que teve o planejamento finalizado.</p>
RF #04	<p>O sistema deverá exibir a ficha de todos os funcionários previamente cadastrados, decorrente a sua categoria; essa ficha deve aparecer no formato</p>

	<p>de uma tabela que exiba todos os dados descritos abaixo. Deve também permitir que o gestor busque pelas fichas de cada funcionário, pesquisando pelo código de identificação ou nome completo, exibindo também sua função/cargo na instituição e seus respectivos salários. A barra de pesquisa deve estar no cabeçalho da própria tabela:</p> <ul style="list-style-type: none"> <li>• Tabela: Funcionários;</li> <li>• Prontuário: Integer e autoincremento (USADO ID NOS MODELOS MER)</li> <li>• Nome do funcionário*: Varchar (100);</li> <li>• CPF* (XXX.XXX.XXX-XX): Inteiro (11), pontos e hífen automáticos;</li> <li>• Data de Nascimento* (DD/MM/AAAA): Date;</li> <li>• Salário*: Float;</li> <li>• Cargo*: Varchar (80);</li> <li>• Data de entrada na empresa* (DD/MM/AAAA): Date;</li> <li>• Status* (ativo ou inativo): Boolean;</li> <li>• Bônus salarial: Float;</li> <li>• Desconto salarial: Float.</li> </ul> <p>O sistema também deverá permitir que o gestor ordene a busca dos funcionários de forma que os inativos (quem já não trabalha mais na empresa) apareçam primeiro. O usuário também poderá ordenar a busca (select na tabela de Funcionários) por ordem alfabética (A-Z / Z-A).</p>
RF #05	<p>O sistema deverá conter um segundo item de menu que receberá o nome 'Orçamentos'. Ao selecionar esse item, o usuário deverá ser direcionado a uma página exclusiva para o orçamento dos funcionários. O gestor deverá criar (insert), editar (update) e excluir (delete from) os salários dos funcionários. Ele poderá adicionar (update) um bônus aos seus salários (variável presente no RF#04) ou realizar um desconto em caso de falta não justificada. Se ele pesquisar o nome do funcionário por CPF ou Nome Completo, deverá aparecer uma ficha do funcionário com suas informações. A ficha deverá ser em formato de tabela, contendo nome*, CPF*, salário*, data de entrada na empresa*, cargo*, data de nascimento*, status* (se está trabalhando, de férias, de licença ou afastado). Sempre que forem alterados o Status, Salário, o Bônus Salarial ou o Desconto Salarial, deve ser registrado a data/hora destas alterações, pois isto impacta diretamente na geração do fluxo de caixa mensal da instituição. O histórico deve ficar acessível para pesquisa do gestor.</p>
RF #06	<p>O sistema deverá permitir que o gestor possa criar (insert), editar (update), excluir (delete from) e buscar (select) os pagamentos mensais ocorridos na instituição (água, energia, gastos com alimentação, entre outros). Além disso, o gestor deverá adicionar a data de vencimento de cada pagamento, dessa forma ele poderá ser avisado caso não pague a conta dentro do prazo. A mensagem de aviso deverá aparecer como um pop-up e a bolinha de notificação deverá aparecer seguindo o mesmo esquema da notificação da tarefa descrito no RF#03:</p> <ul style="list-style-type: none"> <li>• Tabela: Contas Mensais;</li> <li>• Nome da conta*: Varchar (50);</li> <li>• Descrição: Varchar (100);</li> <li>• Valor*: Float;</li> <li>• Data de vencimento* (DD/MM/AAAA): Date;</li> </ul>

	<ul style="list-style-type: none"> <li>• Status* (pago ou não): Boolean.</li> </ul> <p>O sistema deve permitir que o gestor busque pelos pagamentos dos serviços da instituição. O sistema deve apresentar ao usuário os seguintes campos de preenchimento do filtro de pesquisa: nome do serviço, status (pago/devendo) e um botão ‘Pesquisar’;</p> <p>Ao realizar a pesquisa o sistema deverá imprimir na tela uma tabela contendo 5 colunas, sendo elas: o código (autoincremento), o nome do serviço, a data de vencimento, o status e na última três ícones representando as opções “editar”, “excluir” (neste caso obrigatoriamente o sistema deve pedir uma confirmação) e “registrar” (pedir uma confirmação se o gestor realmente pagou esta dívida e se a confirmação for “Sim” o sistema deverá registrar a data/hora no banco de dados).</p>
RF #07	<p>O gestor deve cadastrar o prontuário do funcionário, o sistema deverá permitir que o gestor crie (insert), edite (update), exclua (delete from) e busque (select na tabela de pagamentos) o pagamento dos pacientes da instituição, pesquisando por prontuário ou nome completo do idoso. Ao pesquisar um paciente, deverá aparecer sua ficha completa, seguindo o padrão de tabelas dos requisitos anteriores, apresentando todas as informações que foram salvas no cadastro, incluindo o seu histórico de pagamento:</p> <ul style="list-style-type: none"> <li>• Tabela: Pagamento de Estadia dos Idosos;</li> <li>• Nome do Idoso*: Varchar (80);</li> <li>• Tipo de plano* (mensal, anual): Varchar (20);</li> <li>• Responsável pelo idoso*;</li> <li>• Tipo de pagamento: Varchar (50) (Ex.: Dinheiro, Depósito, Cheque, etc.);</li> <li>• Valor*: Float.</li> </ul> <p>O sistema deve permitir que o gestor busque pelos pagamentos de estadia dos pacientes da instituição. O sistema deve apresentar ao usuário os seguintes campos de preenchimento do filtro de pesquisa: nome do idoso, tipo de pagamento, o valor, o status (pago/devendo) e um botão ‘Pesquisar’;</p> <p>Ao realizar a pesquisa o sistema deverá imprimir na tela uma tabela contendo 6 colunas, sendo elas: o código (autoincremento), o nome do idoso, o tipo de pagamento, o valor, o status e na última três ícones representando as opções “editar”, “excluir” (neste caso obrigatoriamente o sistema deve pedir uma confirmação) e “registrar” (pedir uma confirmação se o gestor realmente recebeu este valor e se a confirmação for “Sim” o sistema deverá registrar a data/hora no banco de dados).</p> <p>Quando estiver na página do planejamento, o sistema deve permitir que o gestor faça uma busca ordenada (select) por ‘Planos Semanais’, ‘Planos Mensais’ e ‘Planos Anuais’, aparecendo assim os pacientes que se encaixam em cada categoria.</p>
RF #08	<p>O sistema deverá ter um terceiro item de menu que receberá o nome de doações, contendo o registro de todas as doações realizadas na instituição. O registro deverá conter o tipo da doação, sendo eles: doação monetária (float), alimentação ou agasalhos:</p> <ul style="list-style-type: none"> <li>• Tabela: Doações;</li> <li>• Nome da doação*: Varchar (50);</li> <li>• Tipo* (roupa, comida, dinheiro): Varchar (50);</li> <li>• Descrição: Varchar (300);</li> </ul>

	<ul style="list-style-type: none"> <li>• Valor (em caso de doação monetária): Float.</li> </ul> <p>O sistema deve permitir que o gestor busque pelos pagamentos de estadia dos pacientes da instituição. O sistema deve apresentar ao usuário os seguintes campos preenchimento do filtro de pesquisa: nome da doação, tipo da doação e um botão ‘Pesquisar’;</p> <p>Ao realizar a pesquisa o sistema deverá imprimir na tela uma tabela contendo 3 colunas, sendo elas: o código (autoincremento), o nome da doação, o tipo da doação e na última três ícones representando as opções “editar”, “excluir” (neste caso obrigatoriamente o sistema deve pedir uma confirmação) e “registrar” (pedir uma confirmação se a instituição realmente recebeu esta doação e se a confirmação for “Sim” o sistema deverá registrar a data/hora no banco de dados).</p>
RF #09	<p>A quarta opção do menu retrátil deverá abrir uma aba com todas as notificações que o gestor receber, independente do fato dele ter fechado elas, deverão ficar armazenadas na opção ‘Notificações’ do menu retrátil, na frente deverá estar o número de notificações que o gestor possui. No canto superior direito, logo acima das notificações deve ter uma opção de “Limpar” na qual se o gestor clicar deverá excluir todas as notificações desta aba. Se o gestor não quiser excluir tudo, mas apenas algumas específicas ele tem a opção de clicar em um “X” ao lado de cada notificação. Se o gestor limpar todas as notificações o número ao lado da palavra “Notificações” deverá aparecer como “0” (zero).</p> <p>As notificações que serão enviadas ao gestor são referentes a contas pendentes da instituição (sendo elas contas mensais, salários de funcionários e atualização do pagamento dos idosos), conclusão do planejamento semanal.</p>
RF #10	<p>O sistema deverá imprimir na tela duas tabelas relacionadas ao fluxo de caixa. A primeira deverá apresentar todas as receitas da empresa e o valor de todas elas, o gestor NÃO vai digitar esses dados, o sistema irá trazê-los do banco de dados. Ao final da tabela o sistema deve fazer um auto soma de todos os valores e imprimir o valor total de receitas na tela.</p> <p>A segunda tabela segue o mesmo esquema, apresentando todas as despesas da instituição e o valor de todas elas. Ao final da tabela o sistema deve fazer um auto soma e imprimir o valor total de despesas na tela.</p> <p>Os valores das despesas devem estar na cor vermelha e os valores das receitas devem estar na cor verde.</p> <p>Ao final das duas tabelas o sistema deve imprimir na tela o valor total de dinheiro na instituição. Se o valor for positivo a cor deve ser verde, mas se o valor for negativo a cor deve ser vermelha.</p>
RF #11	<p>O sistema deverá conter um menu retrátil, de forma que quando o usuário posicionar o mouse sobre este botão deverá aparecer um menu com 4 opções: ‘Limpeza’, ‘Orçamento’, ‘Doações’ e “Notificações”. Quando um destes itens de menu for selecionado, o usuário será direcionado a uma página específica em cada caso. Não há um lugar específico no qual esse menu deve ser inserido, pois depende do <i>template</i> escolhido.</p>
RN #12	O sistema será programado em PHP e Java Script
RN #13	O <i>template</i> deve ser implementado em HTML+CSS+PHP
RN #14	Utilização do RUP
RN #15	Para cada caso de uso do projeto deve ser elaborado os protótipos de IHMs
RN #16	Utilização do Design Pattern MVC

RN #17	O sistema obrigatoriamente deve ser compatível com os navegadores Google Chrome, Internet Explorer, Firefox.
RN #18	O sistema deve ter um tempo de resposta inferior a 5 segundos.
RN #19	O sistema deve ser responsivo.
RN #20	O sistema deve disponibilizar a documentação do desenvolvimento deste documento para que todos participantes da matéria tenham acesso a ela.
RN #21	80% do sistema deve estar finalizado até 29/11/2019.
RN #22	O sistema deve proporcionar uma interface com opções simples e autoexplicativas ao usuário.
RN #23	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 #24	O sistema precisa garantir que o banco de dados seja atualizado para que possa ser gerado um gráfico geral das limpezas realizadas por semana e das limpezas que foram fixadas.
RN #25	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 #26	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.
RN #27	O sistema deverá permitir que o gestor visualize um relatório com o fluxo de entrada e saída de dinheiro na instituição, especificando quais foram as despesas pagas e não pagas (mensal e anual).

Fonte: elaboração própria

Tabela 13: Documento de requisitos do módulo 9

Identificador	Descrição do Requisito
RF #01	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a mortalidade dos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, as semanas serão representadas pelo eixo x e o número de idosos pelo eixo y. Caso o período corresponda a dois meses ou mais, os meses serão representadas pelo eixo x e o número de idosos pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: linear (padrão), coluna, barra e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #02	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de doença diarreica aguda nos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, o número de idosos será representado pelo eixo x e as semanas pelo eixo y. Caso o período corresponda a dois meses ou mais, o número de idosos será representado pelo eixo x e as semanas pelo eixo y no gráfico anual. O usuário

	pode selecionar o gráfico para ser de um dos seguintes tipos: barra (padrão), coluna, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #03	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de Escabiose nos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, o número de idosos será representado pelo eixo x e as semanas pelo eixo y. Caso o período corresponda a dois meses ou mais, o número de idosos será representado pelo eixo x e as semanas pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: barra (padrão), coluna, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #04	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de desidratação nos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, o número de idosos será representado pelo eixo x e as semanas pelo eixo y. Caso o período corresponda a dois meses ou mais, o número de idosos será representado pelo eixo x e as semanas pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: barra (padrão), coluna, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #05	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de quedas dos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, as semanas serão representadas pelo eixo x e os idosos pelo eixo y. Caso o período corresponda a dois meses ou mais, as semanas serão representadas pelo eixo x e os idosos pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: coluna (padrão), barra, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #06	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de desnutrição em idosos nos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, o número de idosos será representado pelo eixo x e as semanas pelo eixo y. Caso o período corresponda a dois meses ou mais, o número de idosos será representado pelo eixo x e as semanas pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: barra (padrão), coluna, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #07	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de Úlcera em idosos nos idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, o número de idosos será representado pelo eixo x e as semanas pelo eixo y. Caso o período corresponda a dois meses ou mais, o número de idosos será

	representado pelo eixo x e as semanas pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: barra (padrão), coluna, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #08	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a incidência de fuga de idosos, usando como base o registro de incidentes e mortalidade do Módulo 05. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, as semanas serão representadas pelo eixo x e os idosos pelo eixo y. Caso o período corresponda a dois meses ou mais, as semanas serão representadas pelo eixo x e os idosos pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: coluna (padrão), barra, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #09	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre a rotina de limpeza dos cômodos, usando como base o registro das limpezas efetuadas por ambiente do Módulo 08. Para geração deste relatório, o usuário deverá selecionar o período (dia/mês/ano inicial e dia/mês/ano final). Caso o período corresponda a apenas uma semana, os dias serão representadas pelo eixo y e os setores da instituição pelo eixo x. Caso o período corresponda a um mês ou mais, os dias serão representadas pelo eixo y e os setores da instituição pelo eixo x no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: coluna (padrão), barra, linear e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #10	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre o número de produtos recebidos pela instituição a partir de doações, usando como base o registro de doações do Módulo 08. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, os tipos de produtos serão representados pelo eixo x e a quantidade pelo eixo y. Caso o período corresponda a dois meses ou mais as doações serão representadas pelo eixo x e a quantidade pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: linear (padrão), barra, coluna e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #11	Este sistema deverá permitir aos gestores a geração do relatório gráfico com legendas e estatístico de tabela sobre o valor recebido pela instituição por doações, usando como base o registro de incidentes e mortalidade do Módulo 08. Para geração deste relatório, o usuário deverá selecionar o período (mês/ano inicial e mês/ano final). Caso o período corresponda a apenas um mês, as semanas serão representadas pelo eixo x e o valor pelo eixo y. Caso o período corresponda a dois meses ou mais as semanas serão representadas pelo eixo x e o valor pelo eixo y no gráfico anual. O usuário pode selecionar o gráfico para ser de um dos seguintes tipos: linear (padrão), barra, coluna e pizza. É possível imprimir em PDF e enviar por e-mail o relatório em questão.
RF #12	O sistema deverá permitir que apenas gestores acessem os relatórios.
RN #13	Processo de Software: É necessário que o Processo de Software a ser utilizado seja o Redmine + Kanban + TortoiseSVN.
RN #14	Linguagem PHP: É necessário que se adote a linguagem PHP, pois é uma

	poderosa linguagem de programação.
RN #15	Design Pattern MVC: Será utilizado o design pattern MVC. Esse padrão de projeto será adotado pois facilita a reutilização do código e facilita também a forma com que a implementação será gerenciada, permitindo que cada camada possa ser construída de forma quase que completamente independente. Não será utilizada nenhuma ferramenta de geração de código.
RN #16	Requisitos de Sistema de Ajuda e de Documentação de Usuário On-line: É necessário disponibilizar, através da Web, a documentação do desenvolvimento do Protótipo do de Projeto do Módulo 09, para propiciar a troca de informações e experiências entre todos os grupos participantes da disciplina.
RN #17	Este sistema deverá ser capaz de propiciar uma interface de fácil entendimento ao usuário, com opções simples, e intuitivas.
RN #18	A ferramenta que será utilizada para a geração de gráficos será a GoogleAPI ou semelhante, pois é uma ferramenta simples e que possui grande quantidade de material para estudo na internet.

Fonte: elaboração própria

(Nota: RF = requisito funciona; RN = requisito não-funcional)

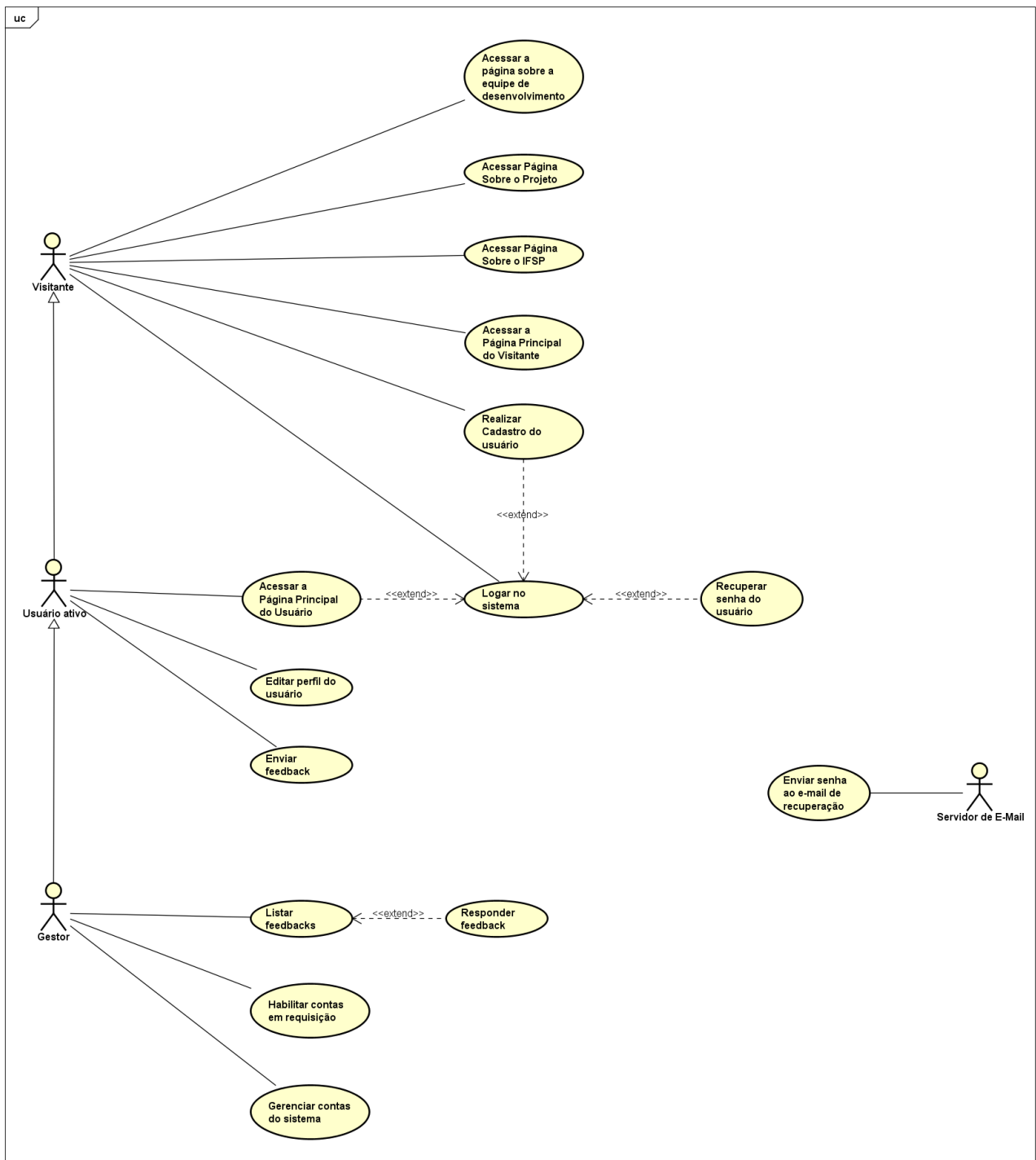
Apesar da grande quantidade de informações contidas nas tabelas acima, os analistas, desenvolvedores e outros membros, como os administradores de banco de dados, foram capazes de exercer suas respectivas funções com base nelas, dado que estavam em cerca de 60 pessoas no total, mas é importante mencionar, além disso, que os analistas foram os principais responsáveis nessa parte específica; isso se deve pelo motivo de eles terem tido de se preocupar com os clientes e fazer o estudo detalhado sobre os requisitos para que não houvesse ambiguidades (explicado em 2.1.2.2), ou seja, os possíveis erros e a falta de acurácia nas tabelas, que tiveram de ser corrigidos nessa parte para que pudesse então montar os diagramas de casos de uso que tendiam a não apresentar problemas.

### 3.2 UML e diagramas de caso de uso do projeto Gerações

De acordo com o UML, os diagramas de caso de uso são abrangidos pelos diagramas comportamentais e eles podem ser montados com base no levantamento de requisitos (explicado em 2.1.1 e 2.1.1.1), e isso significa que os requisitos do projeto Gerações presentes nas 9 tabelas do capítulo 3.1 tiveram **respectivamente** os seguintes diagramas de casos de uso explicados:



Figura 7: Diagramas de casos de uso do módulo 1 (referentes à tabela 5)

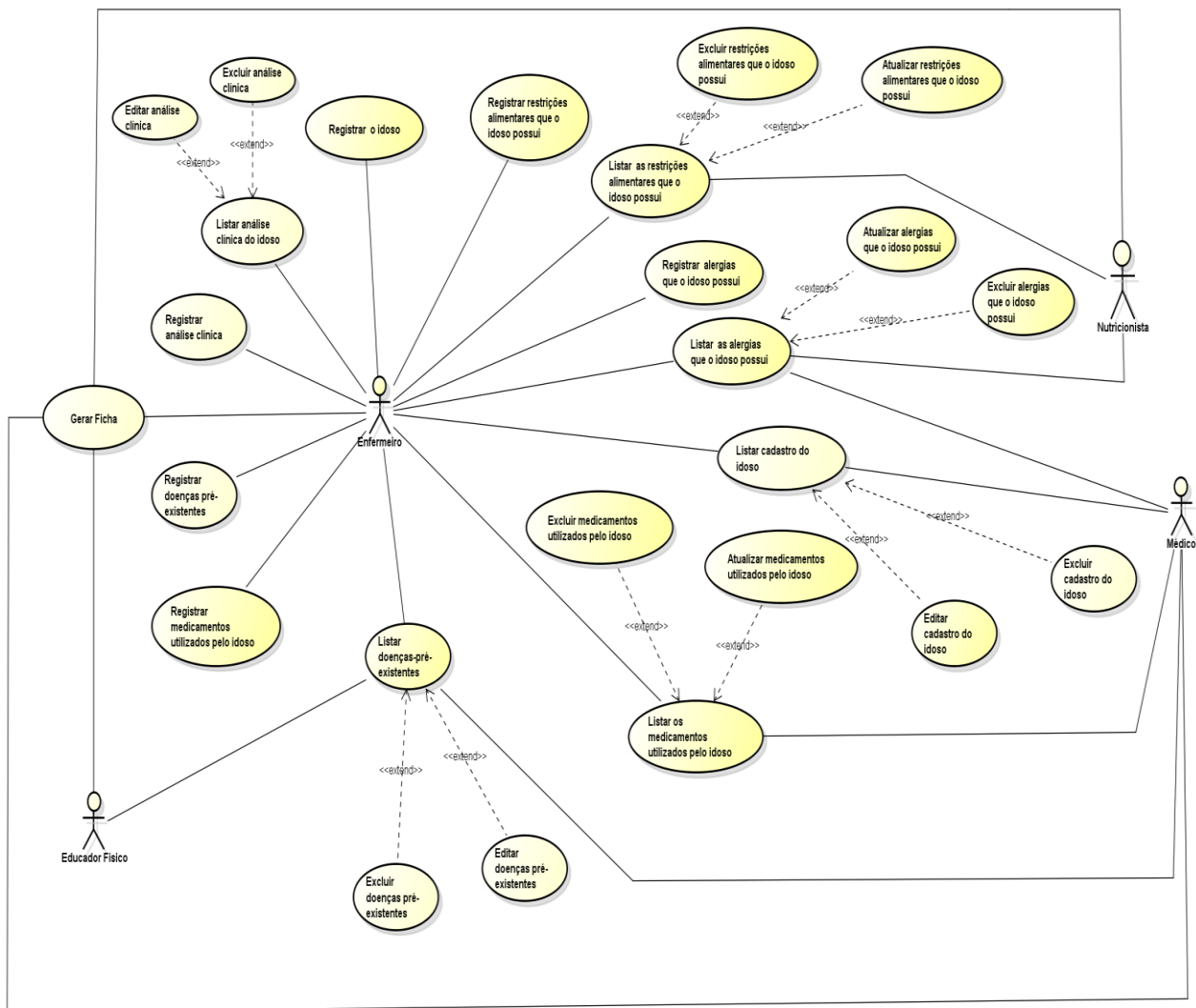


Fonte: elaboração própria

De acordo com a figura 7, referente à tabela 5, o visitante acessa a página sobre a equipe de desenvolvimento, acessa a página sobre o projeto, acessa a página sobre o IFSP e acessa a página principal do visitante; ademais, ele realiza o cadastro do usuário, podendo, após isso, opcionalmente logar no sistema e também recuperar a senha caso deseje. O usuário ativo faz tudo que o visitante

faz; porém, apesar disso, acessa a página principal do usuário podendo análoga e opcionalmente logar no sistema e recuperar a senha. O gestor faz tudo que o visitante e o usuário ativo fazem e ainda habilita contas em requisição, gerencia contas do sistema e, por fim, lista os *feedbacks*, podendo responder aos *feedbacks*. O servidor de *e-mail* envia a senha ao *e-mail* de recuperação.

Figura 8: Diagrama de casos de uso do módulo 2 (referentes à tabela 6)



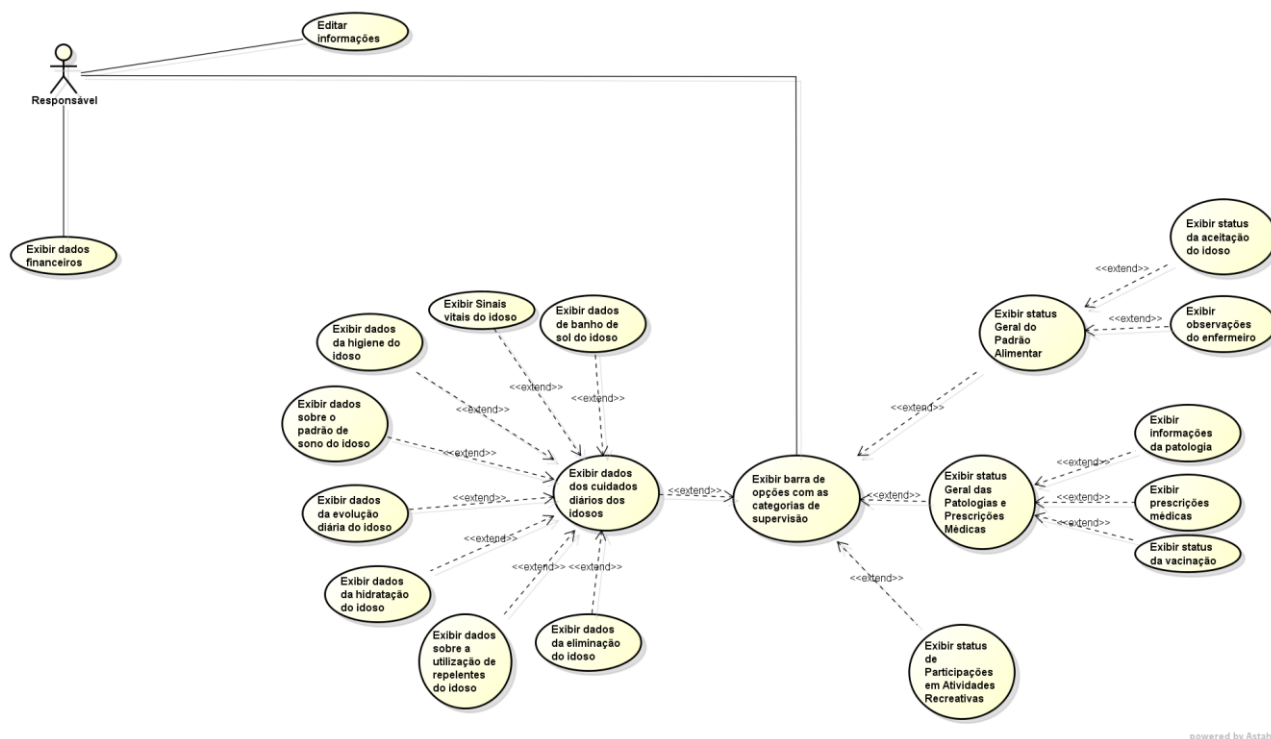
powered by Astah

Fonte: elaboração própria

Com base na figura 8, referente à tabela 6, o enfermeiro registra o idoso, registra restrições alimentares que o idoso possui, lista as restrições alimentas que o idoso possui (podendo, caso necessário, excluir as restrições alimentares dele ou atualizá-las), registra alergias que o idoso possui, lista as alergias que o idoso possui (podendo opcionalmente atualizar as alergias que ele possui tal como exclui-las), lista o cadastro do idoso (podendo opcionalmente editar ou excluir o cadastro do idoso), lista os medicamentos utilizados pelo idoso (podendo opcionalmente excluir ou

atualizar os medicamento utilizados pelo idoso), lista as doenças pré-existentes (podendo opcionalmente excluir ou deletar as doenças pré-existentes), registra medicamentos utilizados pelo idoso, registra doenças pré-existentes, gera ficha, registra análise clínica, lista a análise clínica do idoso (podendo, opcionalmente, então, editar ou excluí-la). O nutricionista lista as restrições alimentares que o idoso possui (podendo opcionalmente excluir ou atualizar as restrições alimentares que o idoso possui) e lista as alergias que o idoso possui (podendo opcionalmente atualizar ou excluir as alergias que o idoso possui). O médico lista os medicamentos utilizados pelo idoso (podendo opcionalmente excluir ou atualizar os medicamentos utilizados pelo idoso), lista as doenças pré-existentes (podendo opcionalmente excluir ou editar as doenças pré-existentes) e gera ficha. O educado físico também lista as doenças pré-existentes (podendo opcionalmente excluir ou editar as doenças pré-existentes) e gera ficha.

Figura 9: Diagrama de casos de uso do módulo 3 (referentes à tabela 7)



Fonte: elaboração própria

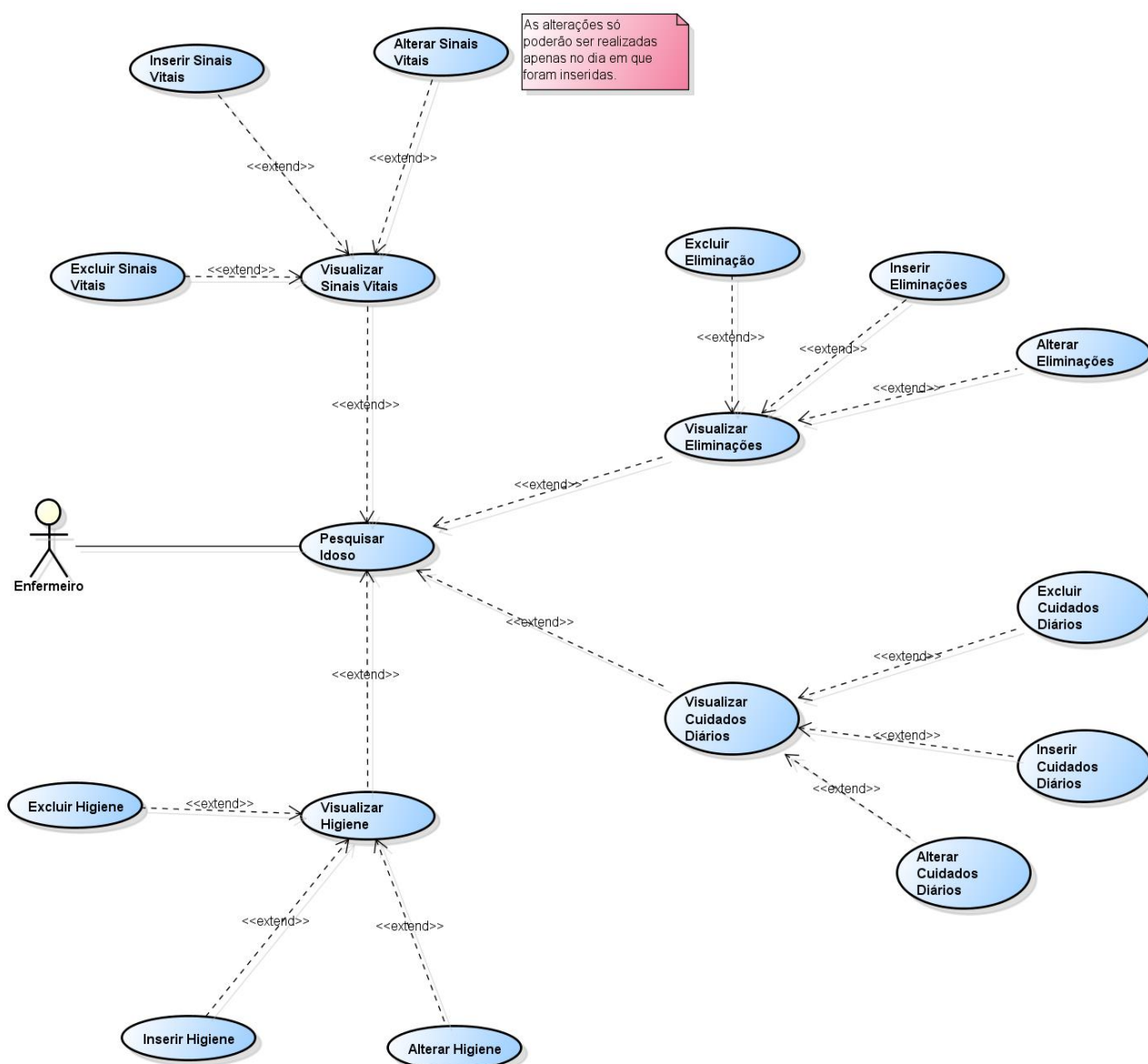
Em relação à figura 9, referente à tabela 7, o responsável edita informações, exibi dados financeiros e, por fim, exibi barra de opções com as categorias de supervisão, que vai-lhe disponibilizar, caso queira, as seguintes ações para serem aplicadas:

- Exibir dados dos cuidados diários dos idosos. Caso os exiba, poderá, se ele quiser, exibir dados de banho de sol do idoso, exibir sinais vitais do idoso, exibir dados de higiene do idoso, exibir dados sobre o padrão de sono do idoso, exibir dados de evolução diária do idoso,

exibir dados da hidratação do idoso, exibir dados sobre a utilização de repelentes do idoso ou exibir dados de eliminação do idoso);

- Exibir *status* geral do padrão alimentar. E caso o exiba, poderá, se ele quiser, exibir o *status* de aceitação do idoso ou exibir as observações do enfermeiro;
- Exibir *status* geral das patologias e prescrições médicas. E caso o exiba, poderá, se ele quiser, exibir informações da patologia, exibir prescrições médicas ou exibir status das vacinações.
- Exibir *status* de participações em atividades recreativas.

Figura 10: Diagrama de casos de uso do módulo 4 (referentes à tabela 8)



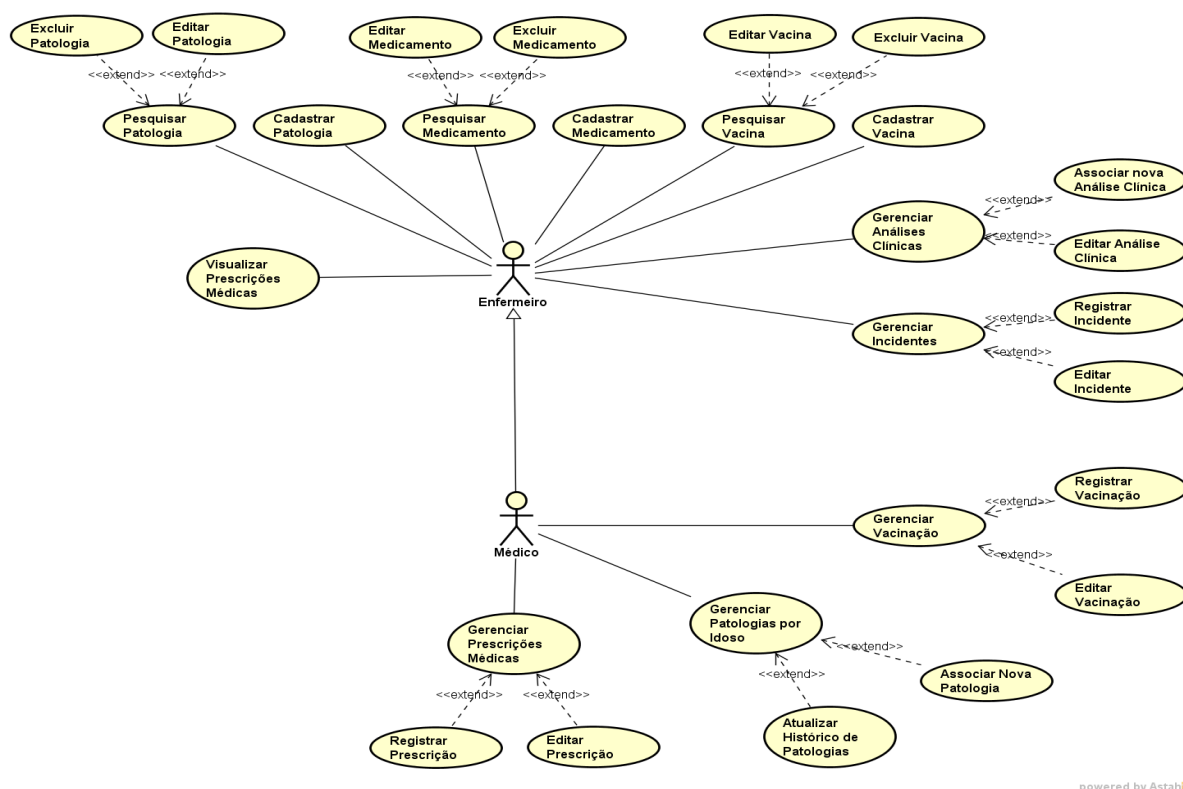
powered by Astah

Fonte: elaboração própria

De acordo com a figura 10, referente à tabela 8, o enfermeiro pesquisa o idoso, podendo opcionalmente:

- Visualizar sinais vitais. E caso os visualize, poderá, caso queira, inserir sinais vitais, alterá-los ou excluí-los;
- Visualizar a higiene. E caso a visualize, poderá, caso queira, inserir higiene, alterá-la ou excluí-la;
- Visualizar os cuidados diários. E caso os visualize, poderá, caso queira, inserir cuidados diários, alterá-los ou excluí-los;
- Visualizar as eliminações. E caso as visualize, poderá, caso queira, inserir eliminações, alterá-las ou excluí-las.

Figura 11: Diagrama de casos de uso do módulo 5 (referentes à tabela 9)



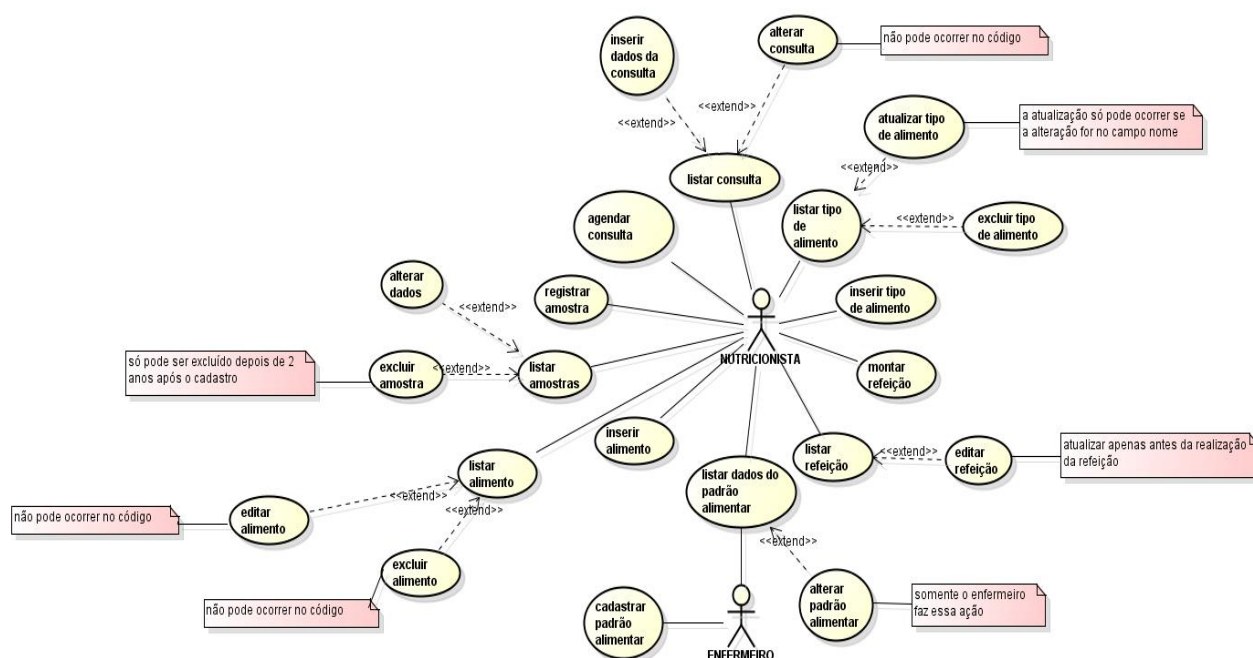
powered by Astah

Fonte: elaboração própria

Em relação à figura 11, referente à tabela 9, o enfermeiro visualiza prescrições médicas, pesquisa patologia (podendo opcionalmente excluí-la ou editá-la), cadastra patologia, pesquisar medicamento (podendo opcionalmente excluí-lo ou editá-lo), cadastra medicamento, pesquisa vacina (podendo opcionalmente excluí-la ou editá-la) e cadastra vacina, gerencia análises clínicas (podendo opcionalmente editá-la ou associar uma nova análise clínica), gerencia incidentes

(podendo registrar um incidente ou editá-lo), já o médico, além fazer tudo que o enfermeiro faz, gerencia prescrições médicas (podendo opcionalmente registrá-las ou editá-las), gerencia patologias por idoso (podendo opcionalmente atualizar o histórico de patologias ou associar uma nova patologia) e gerenciar vacinação (podendo opcionalmente editá-la ou registrar uma vacinação).

Figura 12: Diagrama de casos de uso do módulo 6 (referentes à tabela 10)

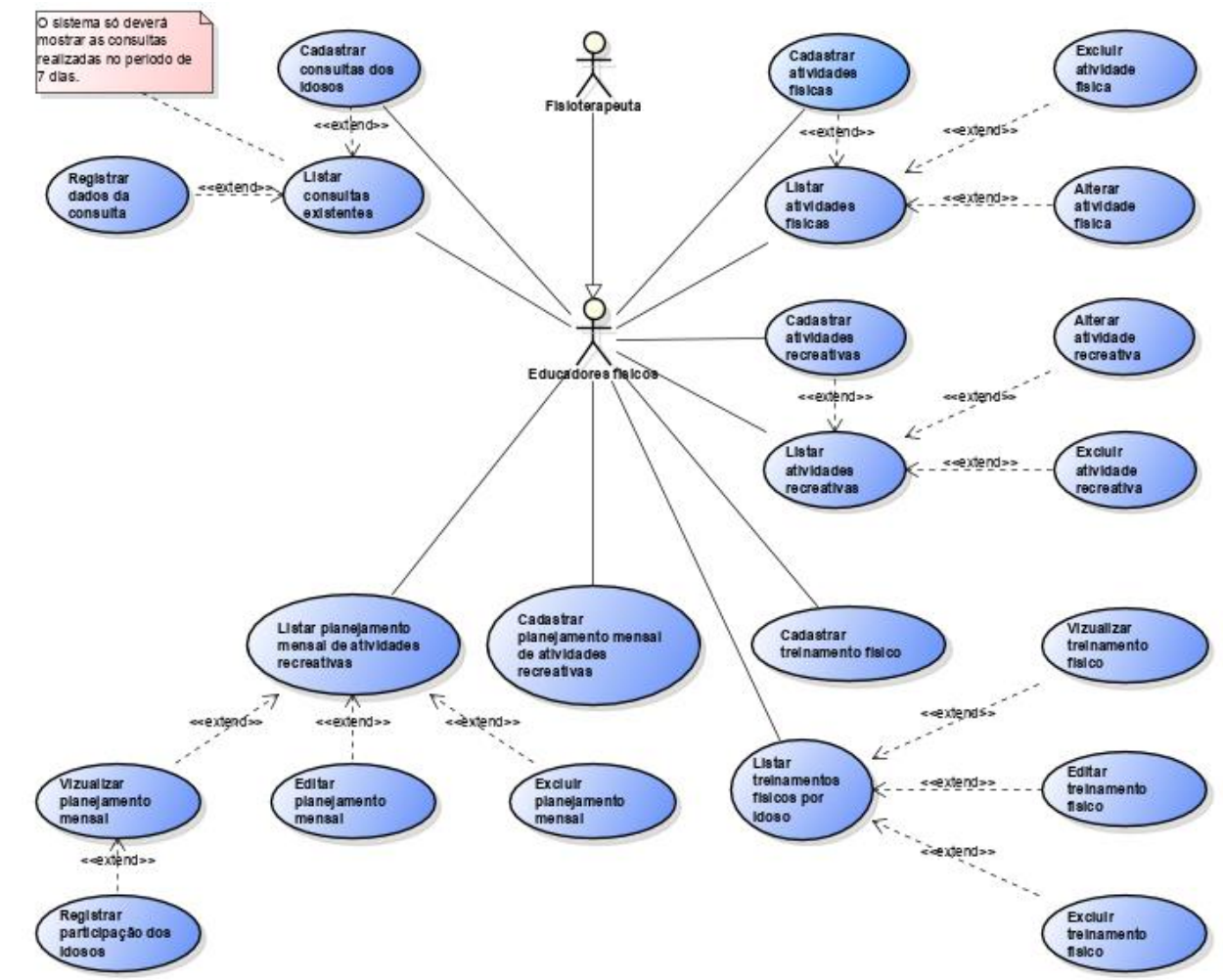


Fonte: elaboração própria

Com base na figura 12, referente à tabela 10, o nutricionista lista a refeição (podendo opcionalmente editá-la somente antes da realização da refeição), monta a refeição, insere o tipo de alimento, lista o tipo de alimento (podendo opcionalmente atualizar o seu tipo que só pode ser efetuado se a alteração for no campo nome ou excluir o seu tipo), lista a consulta (podendo opcionalmente inserir os dados da consulta ou alterar a consulta que não deve ser alterada no código), agenda consulta, registra amostra, lista amostras (podendo opcionalmente alterar seus dados ou, depois de dois anos após o cadastro, excluir a amostra), lista o alimento (podendo opcionalmente editar o alimento que não pode ser editado nem excluído no código), insere o alimento e lista os dados do padrão alimentar. O enfermeiro também lista esses dados do padrão alimentar; entretanto, como diferença, ele pode alterá-los; ele também cadastra o padrão alimentar.



Figura 13: Diagrama de casos de uso do módulo 7 (referentes à tabela 11)

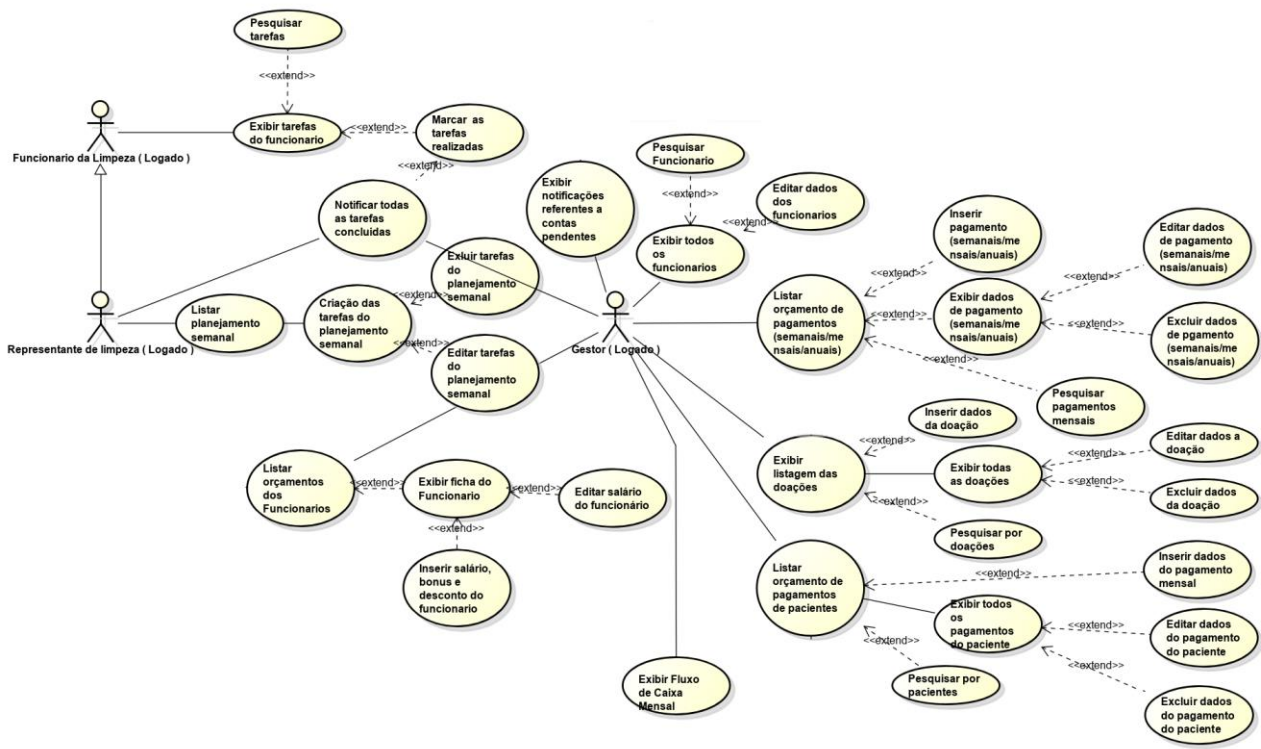


Fonte: elaboração própria

De acordo com figura 13, referente à tabela 11, os educadores físicos listam as consultas existentes, mas só serão mostradas aquelas que foram realizadas no período de 7 dias, e ainda poderão opcionalmente registrar dados da consulta ou cadastrar consultas dos idosos. Ademais, eles cadastram as atividades físicas; e, quando listam as atividades físicas, podem opcionalmente cadastrá-las, alterá-las ou excluí-las. Eles cadastram as atividades recreativas, e apesar disso, caso necessário, podem cadastrá-las quando as listam; e, também, quando as listam, podem opcionalmente alterá-las ou excluí-las. E eles, outrossim, cadastram o treinamento físico, cadastram o planejamento mensal de atividades recreativas, listam os treinamentos físicos por idoso (podendo opcionalmente visualizar o treinamento físico, editá-lo ou excluí-lo), listam o planejamento mensal de atividades recreativas podendo ter a escolha, caso queiram, de:

- Editar ou excluir o planejamento semanal;
- Visualizar o planejamento mensal. E caso o visualize, poderá, caso necessário, registrar as participações dos idosos.

Figura 14: Diagrama de casos de uso do módulo 8 (referentes à tabela 12)



Fonte: elaboração própria

De acordo com a figura 14, referente à tabela 12, o gestor, quando logado no sistema, tem a opção de exibir todos os funcionários e, caso necessário, poderá pesquisar o funcionário ou editar dados dos funcionários. E apesar disso, ele também lista o orçamento de pagamentos semanais, mensais e anuais podendo, então, opcionalmente:

- Inserir o pagamento semanal, mensal e anual;
- Exibir dados de pagamentos semanais, mensais e anuais. Caso exiba, poderá, caso necessário, editar dados de pagamentos semanais, mensais e anuais ou excluí-los;
- Pesquisar somente pagamento mensais.

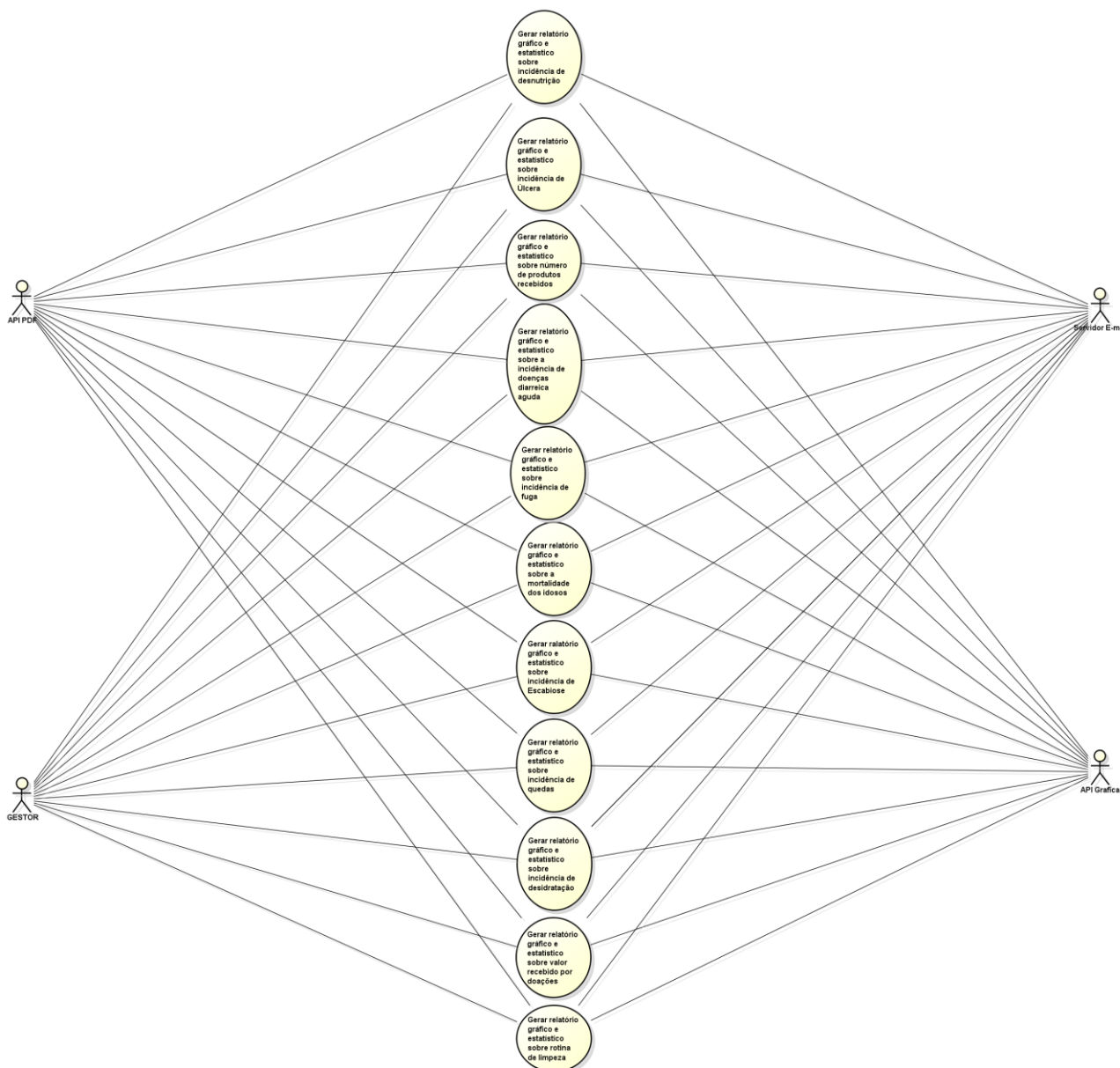
Ele também tem a opção de exibir a listagem das doações podendo opcionalmente inserir dados da doação, ou pesquisar doações ou exibir todas as doações (podendo opcionalmente editar ou excluir os dados a doação). Ademais, ele lista o orçamento de pagamentos de pacientes podendo opcionalmente inserir dados do pagamento mensal, pesquisar os pacientes. Ainda nessa etapa, ele tem a opção de exibir todos os pagamentos do paciente e, então, opcionalmente, editar ou excluir os dados do pagamento do paciente).

Ele exibe o fluxo de caixa mensal, edita tarefas do planejamento semanal, lista orçamento do funcionário (podendo opcionalmente inserir salário bônus e desconto do funcionário ou editar o salário do funcionário), notifica todas as tarefas concluídas, já o funcionário da limpeza, quando



logado no sistema, exibe as tarefas do funcionário podendo opcionalmente pesquisar as tarefas ou marcar as tarefas realizadas. E caso ele as marque, ele poderá, caso necessário, notificar todas as tarefas concluídas. O representante da limpeza (logado), que faz tudo que o funcionário da limpeza faz, tem analogamente a opção de notificar todas as tarefas prontas e ainda lista o planejamento semanal, cria as tarefas do planejamento semanal (podendo opcionalmente excluí-las ou editá-las).

Figura 15: Diagrama de casos de uso do módulo 9 (referentes à tabela 13)



Fonte: elaboração própria

Tanto o API PDF, quanto o gestor, API gráfica e o servidor *e-mail* geram relatório gráfico e estatístico sobre o/a:

- Incidência de desnutrição;

- Incidência de úlcera;
- Número de produtos recebidos;
- Incidência de doenças diarreicas agudas;
- Incidência de fuga;
- Incidência de escabiose;
- Incidência de quedas;
- Incidência de desidratação;
- Valor recebido por doações;
- Rotina de limpeza.

Além disso tudo, é necessário saber que todos os diagramas de casos de uso foram feitos com bastante atenção e que os analistas de todos os módulos fizeram um documento mais complexo, como o da tabela 2 (páginas 12 e 13) referentes aos requisitos levantados (e também e aos diagramas de casos de uso) para que pudessem encarregar os membros da equipe como os desenvolvedores a fazer a parte de programação do *software*. E na medida que tudo fosse sendo programado, os desenvolvedores comunicavam os analistas para que estes testassem o que havia sido feito em uma determinada data e anotassem as falhas e os sucessos como na tabela 4 e 5 pertencentes respectivamente às páginas 15 e 16; porém, nem tudo havia sido feito a tempo, dado que a parte de programação demanda de muito tempo. E além disso, o projeto e o seu preço calculado via aplicativo privado será ainda negociado com o cliente interessado em comprar o *software* do projeto Gerações sob a direção dos professores Breno Romano e Luiz Francisco da matéria de PDS do IFSP de São João da Boa Vista em 2019. Em poucas palavras, o analista é a pessoa que aplica o UML para que os desenvolvedores e outros membros possam usá-los como base em suas atividades com segurança livre ambiguidades.

## 4 Conclusão

Como especificado na contextualização (página 6), o *software* Gerações fora criado pelas cerca de 60 pessoas, em conjunto com os professores da matéria de PDS dos 4º anos vespertinos, Breno Romano e Luiz Francisco, para que São João da Boa Vista conseguisse manter um *status* como a cidade brasileira, contendo entre 50 mil e 100 mil habitantes, que melhor oferece qualidade de vida para idosos, dado que ele seria usado por uma instituição de longa permanência.

E em relação à criação desse *software*, ora também referido como projeto, é importante relatar, como objetivo geral, os quão significativos foram os Analistas de Sistemas para o auxílio e vantagem do desenvolvimento e funcionamento do *software* através da aplicação do UML e a etapa principal de processo de desenvolvimento de *software*.

Entretanto, nesse TFC, foi necessário explicar antes como um analista comum aplicava o UML e a etapa de processo de desenvolvimento de *software* em um projeto qualquer sem ser o Gerações. E, então, devido a isso, foi necessário explicar também o que era um analista de sistemas, informando que ele era um componente humano que exercia o seu papel na área de TI.

Após ter explanado o que era um analista, foi necessário elucidar o que era o UML, dizendo que era uma linguagem usada por componentes humanos que trabalham na área de TI.

Assim que foi revelado o que era UML, foi esclarecido quais eram as utilidades dos seus diagramas de casos de uso, expondo que eles, além de estarem abrangidos no diagrama comportamental, tinham a utilidade e função de expor a funcionalidade apresentada a um sistema que, por fim, seria planejado.

Tendo finalizado essa parte, foi-se necessário, outrossim, exemplificar como se aplicava o processo de desenvolvimento de *software* (que abrangia o processo de levantamento de requisito, estudo detalhado e testes de *software*); esse processo se tratava de os analistas e os membros de um projeto qualquer terem de conversar com os clientes para levantar requisitos bem como fazer o estudo detalhado para saber se havia ambiguidades nos requisitos para que o desenvolvedor, então, pudesse programar com base nos requisitos corrigidos durante esse processo. E assim que o desenvolvedor acabava de programar tal etapa, o analista aplicava o teste de *software* sobre os códigos dele a fim de encontrar *bugs* ou problemas de programação.

Assim que tudo isso foi feito, isso foi analogamente aplicado no projeto Gerações conforme o capítulo 3, ou seja, tiveram-se, em poucas palavras, de levantar-lhe requisitos resultando em 9 tabelas bem como montar 9 diagramas de caso de uso. Não foi necessário explicar mais a fundo, dado que o capítulo 2 era um ótimo exemplar.

Tendo feito isso, é possível concluir que o objetivo do trabalho, de fato, foi concluído dado que nele foi explicado que é o analista de sistema quem aplicou UML e o processo de desenvolvimento de software no projeto Gerações.

Contudo, houve pontos negativos pelo motivo de o projeto Gerações ainda não ter sido desenvolvido completamente nem testado, e isso afetou parte do processo de desenvolvimento de *software* como dito no final do subcapítulo 3.2, deixando ainda inviável para venda, que ainda estaria para ser revista. A parte boa era que foi possível realizar todos os diagramas de caso de uso propostos pelo UML.

## 5 Referências Bibliográficas

- [1] G1 São Carlos e Araraquara. **Pesquisa aponta São João da Boa Vista como melhor cidade para idosos**, 2017. 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: 06 de set. de 2019
- [2] Guia da Carreira. **O que faz um analista de sistemas?**, 2018. Disponível em: <<https://www.guiadacarreira.com.br/carreira/o-que-faz-um-analista-de-sistemas-2/>>. Acesso em: 24 de set. de 2019
- [3] BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**: 3. ed. Rio de Janeiro: Editora Elsevier, 2014
- [4] CARODOSO, C. **UML na prática: do problema ao sistema**. Rio de Janeiro Editora Ciência Moderna Ltda., 2003
- [5] Anhanguera Educacional Faculdade Anhanguera de Valinhos Ciência da Computação. **Engenharia de Software e Análise de Sistemas Atividades Práticas Supervisionadas**, 2014. Disponível em: <<https://www.passeidireto.com/arquivo/6199872/atps-engenharia-de-software>>. Acesso em 16 de out. de 2019.
- [6] IBM Knowledge Center. **Relacionamentos de Associação**, 2013. Disponível em: <[https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH\\_9.5.0/com.ibm.xtools.modeler.doc/topics/casn.html](https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/casn.html)>. Acesso em 18 de out. de 2019.
- [7] IBM Knowledge Center. **Relacionamentos de Generalização**, 2013. Disponível em: <[https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH\\_9.5.0/com.ibm.xtools.modeler.doc/topics/cgeneral.html](https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cgeneral.html)>. Acesso em 18 de out. de 2019.
- [8] IBM Knowledge Center. **Relacionamentos de Inclusão**, 2013. Disponível em: <[https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH\\_9.5.0/com.ibm.xtools.modeler.doc/topics/cinclusion.html](https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cinclusion.html)>. Acesso em 18 de out. de 2019.

br/SS5JSH\_9.5.0/com.ibm.xtools.modeler.doc/topics/cinclud.html>. Acesso em 18 de out. de 2019.

[9] IBM Knowledge Center. **Relacionamentos de Extensão**, 2013. Disponível em: <[https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH\\_9.5.0/com.ibm.xtools.modeler.doc/topics/cextend.html](https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cextend.html)>. Acesso em 18 de out. de 2019.

[10] VIEGAS, Júlio. **Teste de Software: Introdução, conceitos básicos e tipos de testes**, 2016. Disponível em: <<https://blog.onedaytesting.com.br/teste-de-software/>>. Acesso em 26 de out. de 2019.