

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

CAMPUS TIMÓTEO

Documentação de desenvolvimento de projeto
de software

Pedilândia

Alunos: André Marcelino e Leonam T. Vasconcelos

Engenharia de computação - 2019.2

Timóteo - MG

22 de Agosto de 2019

Sumário

1	Glossário	3
2	Materiais de referência	3
3	Descrição do minimundo do projeto	5
4	Requisitos do sistema	6
4.1	Requisitos funcionais	6
4.1.1	Descrição dos atores	6
4.1.2	Lista de funcionalidades	7
4.1.3	Diagrama de casos de uso	10
4.2	Requisitos não funcionais	11
4.2.1	Usabilidade	11
4.2.2	Confiabilidade	11
4.2.3	Desempenho	12
4.2.4	Segurança	12
4.2.5	Padrões	12
4.2.6	Hardware e software	13
4.3	Diagrama de classes	13

1 Glossário

- **API** - Application Programming Interface (Interface de programação para aplicações);
- **CRUD** - Create Read Update and Delete (Cadastrar Pesquisar Atualizar e Excluir);

2 Materiais de referência

Tabela 1: Tabela de materiais de referência.

Tipo de material	Referência
Documentação	<i>TensorFlow</i> site: https://www.tensorflow.org/overview/ github: https://github.com/tensorflow/tensorflow
	<i>Angular</i> site: https://angular.io/docs/ github: https://github.com/angular/angular
	<i>Angular Material</i> site: https://material.angular.io/ github: https://github.com/angular/material2
	<i>GraphQL</i> site: https://graphql.org/ github: https://github.com/graphql
	<i>Flutter</i> site: https://flutter.dev/ github: https://github.com/flutter/flutter
	<i>Firebase</i> site: https://firebase.google.com/docs/ github: https://github.com/firebase/
	<i>Node.js</i> site: https://nodejs.org/en/docs/ github: https://github.com/nodejs/node
	<i>MongoDB</i> site: https://docs.mongodb.com/ github: https://github.com/mongodb/mongo
Continua na próxima página...	

Tabela 1 – continuação da página anterior

Tipo de material	Referência
	<i>Mongoose</i> site: https://mongoosejs.com/docs/api.html github: https://github.com/Automattic/mongoose
	<i>Typegoose</i> site: https://www.npmjs.com/package/typegoose github: https://github.com/szokodiakos/typegoose
	<i>NPM</i> gerenciador de pacotes site: https://www.npmjs.com/ github: https://github.com/npm/cli
	<i>Spring Framework</i> site: https://spring.io/ github: https://github.com/spring-projects/spring-framework
	Documentações das respectivas dependências dos softwares acima;
Livro	Node.js Design Patterns https://subscription.packtpub.com/book/web_development/9781785885587
	TensorFlow Machine Learning Cookbook https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781786462169
	Building Machine Learning Projects with TensorFlow https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781786466587
	Statistics for Machine Learning https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788295758
	Machine Learning Algorithms https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781785889622
Site	Flutter https://hackernoon.com/scalable-app-structure-in-flutter-dad61a4bc389
Continua na próxima página...	

Tabela 1 – continuação da página anterior

Tipo de material	Referência
Vídeo	Flutter https://www.youtube.com/watch?v=hRz9JSiZm0Y&list=PLlBnICoI-g-dMEDwxyb01I0DteEvouE9c

Fonte: Elaborada pelos autores.

3 Descrição do minimundo do projeto

No setor médico de pediatria existem diversas situações e inconvenientes que acontecem na prestação de serviços. Um dos problemas é que os pais não possuem conhecimento preciso a respeito do histórico médico de seus filhos, pois nem todos planos de saúde disponibilizam as informações de diagnósticos clínicos e resultados de exames. Além disso, podem acontecer outras situações como :

- O médico pode não estar bem informado à respeito da situação / histórico de seu paciente, o que cria empecilhos para uma consulta mais eficaz;
- Em caso de houver necessidade de mudança do plano de saúde, é provável que todo o histórico médico do pacientes seja perdido, pois os planos de saúde utilizam bases de dados próprias.

Nos anos iniciais da vida de uma criança, seu desenvolvimento é acompanhado por diversos indicadores, como: Peso, altura, desenvolvimento da fala, capacidade de manter-se em pé e andar, coordenação motora fina, média e grossa, controle urinário, capacidade de leitura e escrita, desenvolvimento de habilidades como tocar um instrumento musical e andar de bicicleta. Além disso, também é essencial o acompanhamento do histórico médico, que envolve acometimento por doenças, exames realizados, laudos e diagnósticos.

Um bom acompanhamento da vida médica de uma criança envolve considerar todo o contexto de sua vida desde o nascimento. Por exemplo: Ao saber que a criança já sofreu de alguma doença muito específica aos três anos de idade, é essencial, para o médico ter consciência do ocorrido, pois permite inferir que o paciente esteja sofrendo alguma possível consequência.

4 Requisitos do sistema

4.1 Requisitos funcionais

4.1.1 Descrição dos atores

Nº	Nome	Descrição	Frequência de uso	Proficiência em informática
1	Adm. de clínica	Ator que coordena as funcionalidades que requerem credenciais mais elevadas. Está no topo da hierarquia de permissões do sistema no contexto de uma clínica médica.	Diária	Alta
2	Secretário	Ator que realiza funções de agendamento e que opera o núcleo do sistema, para organizar horários dos pacientes	Diária	Média
3	Enfermeiro	Ator que tem acesso às informações sobre os pacientes e pode inserir informações sobre novas consultas realizadas	Diária	Baixa
4	Médico	Ator que tem acesso às informações dos pacientes e pode inserir informações sobre novas consultas realizadas.	Diária	Baixa

Nº	Nome	Descrição	Frequência de uso	Proficiência em informática
5	Pais de crianças	Ator que tem acesso às informações inseridas sobre as consultas realizadas em seu filho, além de ter uso de inserção de algumas informações úteis a respeito do desenvolvimento do mesmo	Variável	Baixa
6	Adm. do sistema	Ator que tem acesso total à todos os controles do sistema. Esse ator faz parte do núcleo de analista de sistemas e gerencial da empresa que fornece o Software como um Serviço	Alta	Alta

Tabela 3: Tabela de descrição dos atores. Fonte: autores.

4.1.2 Lista de funcionalidades

Tabela 4: Lista de funcionalidades do software.

Núm.	Nome	Descrição	Tipo	Atores
1	Cálculo de dosagens de remédio	Realizar o cálculo de dosagens de cada medicamento baseando-se em características físicas do paciente	1	3, 4
2	Login	Controle de acesso dos atores	3	1, 2, 3, 4, 5, 6
3	CRUD de médicos	Cadastro, pesquisa, atualização e deleção de médicos	1	1
Continua na próxima página				

Tabela 4 – continuação da página anterior

Núm.	Nome	Descrição	Tipo	Atores
4	CRUD de enfermeiros(as)	Cadastro, pesquisa, atualização e deleção de enfermeiros(as)	1	1
5	CRUD de clientes	Cadastro, pesquisa, atualização e deleção de clientes	1	1
6	CRUD de clínicas	Cadastro, pesquisa, atualização e deleção de clínicas	1	6
8	Cadastro	Cadastro realizado pelos usuários que são pais das crianças	1	5
10	CRUD de exames	Cadastro, pesquisa, atualização e deleção de exames	1	6
11	CRUD de tipos de consulta	Cadastro, pesquisa, atualização e deleção de tipos de consultas	1	6
12	CRUD de tipos de diagnósticos	Cadastro, pesquisa, atualização e deleção de tipos de diagnósticos	1	6
13	CRUD de doenças possíveis	Cadastro, pesquisa, atualização e deleção de doenças possíveis	1	6
13	CRUD de sintomas de doenças	Cadastro, pesquisa, atualização e deleção de sintomas de doenças	1	6
14	CRUD de medicamentos	Cadastro, pesquisa, atualização e deleção de medicamentos	1	6
15	Agendamento	Funcionalidade que permite gerenciar agendamentos de consultas	4	2, 4, 5
Continua na próxima página				

Tabela 4 – continuação da página anterior

Núm.	Nome	Descrição	Tipo	Atores
16	Reportagem de finalização de consulta	Nessa funcionalidade o usuário finalizará a consulta informando os dados colhidos pela mesma: Dados da consulta, possível doença que tenha sido diagnosticada, informações usadas para o histórico médico da criança	2, 3, 4	
17	Publicação de informações	<ul style="list-style-type: none"> – Semelhante ao post em redes sociais. – Publicação realizada pela mãe, médico e secretária. – Foto, ou vídeo. – Sentimentos (Celebrações, sentindo-se feliz). – Marcar outras pessoas. – Check-in. – Evento ocorrido: <ul style="list-style-type: none"> – Criança começou a falar. – Criança começou a andar. – Aprendeu a ler/escrever. – Entrou na escola. 	2	1, 2
20	Relatório de atendimentos realizados no mês por cada médico	Exibe quais atendimentos foram realizados no mês por cada médico	2	1, 2

Fonte: Elaborada pelos autores.

Legenda:

Tabela 5: Tipos de funcionalidades.

Tipo	Nome
1	Entrada de dados
2	Saída de dados
3	Controle de acesso
4	Controle
5	Rotina noturna

Fonte: Elaborada pelos autores.

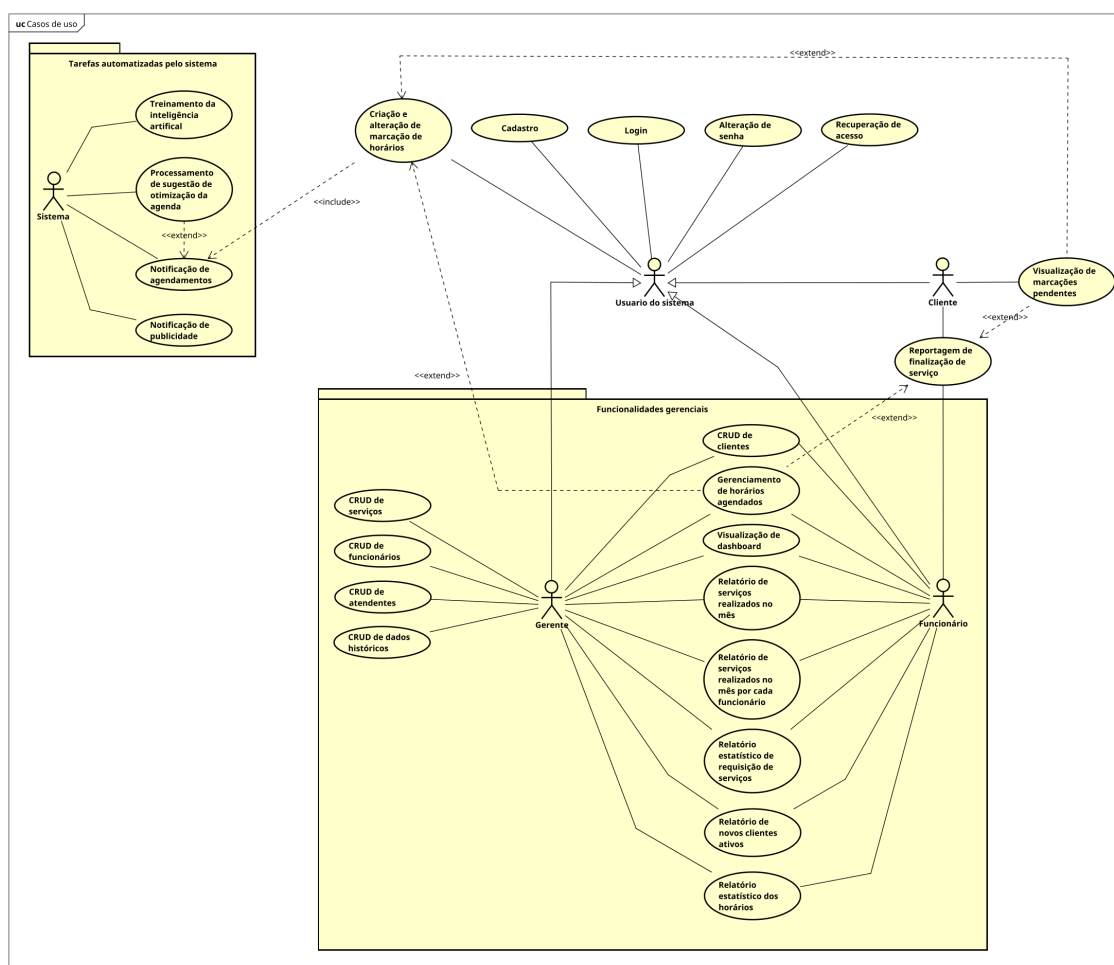
Tabela 6: Atores do sistema.

Número	Nome
1	Adm. de clínica
2	Secretário
3	Enfermeiro
4	Médico
5	Pais de crianças
6	Adm. do sistema

Fonte: Elaborada pelos autores.

4.1.3 Diagrama de casos de uso

Figura 1: Diagrama de casos de uso.



Fonte: Autores.

4.2 Requisitos não funcionais

4.2.1 Usabilidade

4.2.1.1 Interface com o Usuário

O sistema deve prover uma interface amigável e intuitiva, de forma que o tempo de aprendizado médio será conforme citado na tabela 7.

Tabela 7: Tabela de descrição dos atores.

Grupo	Tempo de aprendizagem
Funcionalidades gerenciais (CRUD's de serviços, funcionário, atendentes, entre outras semelhantes)	Uma hora
Funcionalidades cotidianas (marcação de horários, gerenciamento de horários agendados, entre outras semelhantes)	Cinco minutos

Fonte: Elaborada pelos autores.

Prioridade: **Importante.**

4.2.2 Confiabilidade

4.2.2.1 Tempo de Retorno após Falha

Após uma falha que indisponibilize temporariamente o acesso ao sistema, o deverá estar disponível em no máximo 10 minutos.

Prioridade: **Importante.**

4.2.2.2 Disponibilidade

O sistema deve estar disponível 24 horas por dia, sete dias por semana. Serão usadas técnicas de integração e desenvolvimento contínuo para que as manutenções não prejudiquem o correto funcionamento do sistema em produção.

Prioridade: **Essencial.**

4.2.3 Desempenho

4.2.3.1 Número de Acessos

O sistema deve ser capaz de atender em média 1.000.000 usuários por dia, com até 10.000 usuários simultâneos em picos de acesso.

Prioridade: **Essencial**.

4.2.3.2 Tempo de resposta para os Usuários

O tempo máximo de resposta do sistema a uma operação simples de um usuário deve ser de até 3 segundos.

Prioridade: **Desejável**.

4.2.4 Segurança

4.2.4.1 Privacidade de Acesso

A autenticação no sistema será feita por meio do recurso *Authentication* da plataforma *Firebase*, a qual fornece modernos e seguros recursos para autenticação e identificação dos usuários.

Prioridade: **Essencial**.

4.2.5 Padrões

4.2.5.1 Padrão de Codificação

A implementação do Sistema server-side (Node.js) será baseada no Padrão de API dinâmica do GraphQL.

Prioridade: **Desejável**.

4.2.6 Hardware e software

4.2.6.1 Linguagem de Implementação

O sistema deverá ser implementado em Dart e Javascript, utilizando os conceitos de programação funcional e orientada a serviços.

Prioridade: **Importante.**

4.2.6.2 Banco de Dados

O banco de dados utilizado será o MongoDB.

Prioridade: **Importante.**

4.2.6.3 Ferramentas

O sistema será desenvolvido com auxílio das seguintes ferramentas case: Astah Community 7.0 para análise e projeto em UML, Visual Studio Code para o desenvolvimento em Javascript e Dart, LaTeX e overleaf para a confecção da documentação.

Prioridade: **Importante.**

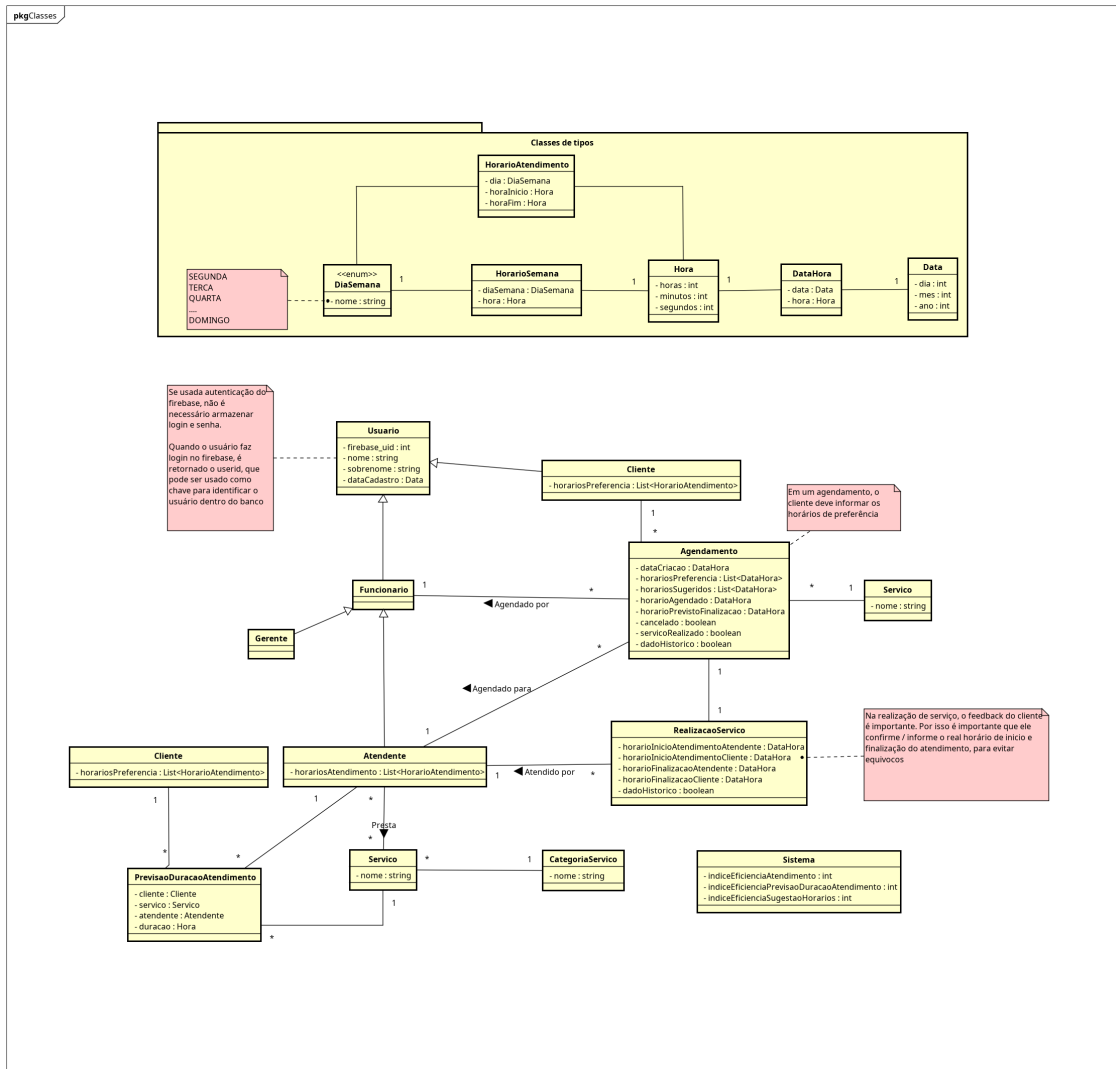
4.2.6.4 Plataforma de Funcionamento

O sistema será desenvolvido para as plataformas *Android*, *iOS* e *Web*.

Prioridade: **Essencial.**

4.3 Diagrama de classes

Figura 2: Diagrama de classes.



Fonte: Autores.