



FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

LEONARDO MARTINS DE OLIVEIRA

NATÁLIA AKINA UESUGI

RAMON LACAVA GUTIERREZ GONÇALES

NFC HEALTH SYSTEM

Americana, SP

2018



FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

LEONARDO MARTINS DE OLIVEIRA

NATÁLIA AKINA UESUGI

RAMON LACAVA GUTIERREZ GONÇALES

NFC HEALTH SYSTEM

Projeto de Conclusão de Curso apresentado como requisito parcial para obtenção do grau Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade de Tecnologia de Americana.

Orientador: Prof. Dr. Kleber de Oliveira Andrade

Área de concentração: Engenharia de Software

Americana, SP

2018

RESUMO

Este trabalho se trata de uma plataforma de saúde digital unificada, que permite melhor gestão de informações de saúde e processos inteligentes. A plataforma visa auxiliar o dia a dia das instituições de saúde e pacientes, buscando diminuir a quantidade de erros na área médica através de uma coleção consistente de dados do paciente, possibilitando que o sistema atue em qualquer instituição de saúde, e que tenha uma melhor eficiência e eficácia em atendimentos, sejam estes comuns ou de urgência e emergência. O trabalho foi realizado em colaboração com a universidade de Durban, na África do Sul. A metodologia empregada durante o decorrer do trabalho foi o SCRUM, que visa a transparência, dinamicidade e agrega valor ao produto final. Foram desenvolvidos dois aplicativos para dispositivos móveis e um para computador, que realizam gestão de exames, diagnósticos, dados de saúde, medicamentos, instituições, médicos(as), dentre outros. Diversos requisitos foram coletados de forma dinâmica com as equipes sul africanas para possibilitar a integração dos sistemas tanto no Brasil quanto na África do Sul. Todo o desenvolvimento do sistema se voltou para agregar valor aos processos e interfaces de usuário (se focando em facilidade de uso e experiência de usuário). Os resultados foram dois aplicativos publicados na Google Play e um sistema computadorizado, sendo que os três estão em fase de testes na África do Sul. Conclui-se que o sistema poderá auxiliar muito no ambiente da saúde, facilitando a gestão, fornecendo processos inteligentes e uma maior agilidade no atendimento, permitindo com que pacientes possuam acesso a seus dados de saúde, e com que funcionários de saúde possuam uma maior facilidade e uma maior quantidade de dados relevantes para análise durante os atendimentos.

Palavras Chave: Saúde; Sistema; Internacionalização; Prontuário Médico; NFC; Arduino; Android; Java

ABSTRACT

This work is about a unified digital healthcare platform, that allows better management of health information and intelligent process. The platform aims help the day by day of the health institutions and patients, seeking to reduce the quantity of errors in the medical area through a consistent collection of patient data, allowing the system to work in any health institution, and that it has a better efficiency and effectiveness in attendance, be these usual or urgency or emergency. The work was made with collaboration with the Durban University, at South Africa. The methodology used during the work was SCRUM, that aims transparency, dynamism and add value to the final product. Were developed two applications for mobile dispositivo and one for computer, that perform management of exams, diagnostics, health data, medicines, institutions, medics, among others. Several requirements were collected dynamically with the South African teams, to be able to integrate the systems in Brazil and South Africa. All the system development has turned to add value to process and user interfaces (focusing on ease of use and user experience). The result were two applications published on Google Play and a computer system, being that the three are undergoing testing in South Africa. It is concluded that the system can help very much in the health ambient, facilitating the management, giving intelligent processes and a greater agility in attendance, allowing the patients have access to their health data, and the health workers have a greater facility and a greater amount of relevant data for analysis during care.

Keywords: Health; Internationalization; Medical Records; NFC; Arduino; Android; Java

AGRADECIMENTOS

Nossos agradecimentos a todos que contribuíram, direta ou indiretamente, para a realização deste trabalho, em especial:

Aos professores (em especial a nosso orientador Kleber) e coordenadores do Curso de Análise e Desenvolvimento de Sistemas, pela dedicação e competência na ajuda, além de transferir os conhecimentos adquiridos por toda a vida a nós. Também a instituição Centro Paula Souza que nos deu a oportunidade de ter uma formação em tecnologia e nos avaliando através desse trabalho.

Aos nossos familiares, por sempre nos apoiar e incentivar na elaboração do trabalho.

Aos nossos amigos, que nos ajudaram e aguentaram todos os dias.

Por fim, gostaríamos de agradecer a Deus, que durante toda a nossa vida tem nos ajudado e acompanhado nos momentos difíceis.

À memória de Deleusa Francisco de Oliveira, que tudo que fiz, me tornei e conquistei foi graças ao esforço e investimentos que mesmo sem poder ela realizou em mim. Que se não fosse por ela, não seria o que sou hoje. Esse trabalho, é uma das pequenas coisas, que mostra o meu retorno e gratidão a ela. Irei sempre fazer o máximo possível para orgulha-la.

“Para onde agora? Aonde a imaginação nos levar”
(Stan Lee)

SUMÁRIO

1	INTRODUÇÃO.....	17
2	PROJETO DO NFC HEALTH SYSTEM	24
2.1	Levantamento de Requisitos	24
2.1.1	Requisitos Funcionais.....	25
2.1.2	Requisitos Não Funcionais	26
2.2	Recursos e Ferramentas	27
3	MODELAGEM	32
3.1	Casos De Uso.....	32
3.2	Documentação dos Casos de Uso	34
3.2.1	Documentação dos Casos de Uso	38
3.3	Diagrama de Classe	61
3.3.1	<i>API</i> do Sistema de Saúde.....	61
3.3.1.1	<i>API</i> de Usuários	62
3.3.1.2	<i>API</i> de Pacientes	64
3.3.1.2	<i>API</i> de Médico(a).....	67
3.3.1.3	<i>API</i> de Enfermeiro(a).....	69
3.3.1.4	<i>API</i> de Instituição de Saúde	70
3.3.1.5	<i>API</i> de Trilha de Auditoria.....	72
3.3.1.6	<i>API</i> de Erro.....	74
3.3.1.6	Miscelânea	75

3.3.2 Sistema de Saúde (Versão Desktop).....	76
3.3.2.1 Funcionalidade de <i>Login</i>	76
3.3.2.2 Funcionalidade de Navegação	78
3.3.2.2 Funcionalidade de Gerenciamento de instituições de saúde	79
3.3.2.3 Funcionalidade de Selecionar instituição de saúde.....	80
3.3.2.4 Funcionalidade de Trilha de Auditoria	80
3.3.2.4 Funcionalidade de <i>Log</i> de Erros.....	81
3.3.2.5 Funcionalidade de Alterar Perfil	82
3.3.2.6 Funcionalidade de Notificação	83
3.3.2.7 Funcionalidade de Gerenciar usuários	83
3.3.2.8 Funcionalidade de Gerenciar pacientes	85
3.3.2.9 Funcionalidade de Gestão do paciente	88
3.3.3 Sistema de Enfermagem (Aplicativo).....	91
3.3.3.1 Funcionalidade de <i>Login</i> e Navegação	92
3.3.3.2 Funcionalidade de seleção de instituição de saúde	94
3.3.3.3 Funcionalidade de tela inicial de listagem de novos pacientes	94
3.3.3.4 Funcionalidade de tela inicial de listagem de atendimentos em processo	95
3.3.4 Sistema do Paciente (Aplicativo)	96
3.3.4.1 Funcionalidade de inicialização do sistema.....	98
3.3.4.2 Funcionalidade de cadastro de paciente	99
3.3.4.3 Funcionalidade de tela inicial	100

3.3.4.4 Funcionalidade de listagem de instituições de saúde próximas	100
3.3.4.5 Funcionalidade de listagem de médicos(as)	101
3.3.4.6 Funcionalidade de gerenciamento de medicamentos	102
3.3.4.7 Funcionalidade de sincronização	103
3.3.4.8 Funcionalidade de gestão de exames	104
3.4 Diagrama de Sequência	105
3.5 Banco de Dados	105
3.5.1 Diagrama de Entidade e Relacionamento (DER)	105
3.5.2 Dicionário de Dados	108
3.5.3 Circuito do sistema arduino	117
4 DESENVOLVIMENTO	121
4.1 Etapas de Desenvolvimento	122
4.1.1 Entrega 1	122
4.1.2 Entrega 2	124
4.1.3 Entrega 3	126
4.1.4 Entrega 4	127
4.1.5 Entrega 5	129
4.1.6 Entrega 6	131
4.1.7 Entrega 7	132
4.1.8 Entrega 8	134
4.2 Interfaces de Usuário.....	135

4.2.1 Sistema <i>desktop</i>	136
4.2.2 Aplicativo do paciente	174
4.2.3 Aplicativo do(a) Enfermeira.....	180
5 CONSIDERAÇÕES FINAIS.....	184
REFERÊNCIAS.....	186
APENDICE	189
APENDICE A – Equipes da universidade de Durban	190
APENDICE B – Dados levantados sobre algumas questões realizadas com a universidade de Durban	191
APENDICE C – Aplicativos na Google Play	195
APENDICE D – Reuniões	196

LISTA DE FIGURAS

Figura 1 - CloudAMQ e RabbitMQ	31
Figura 2 - Hierarquia de usuários	33
Figura 3 - Diagrama de caso de uso do login do usuário	34
Figura 4 - Diagrama de caso de uso do administrador do sistema	35
Figura 5 - Diagrama de caso de uso do administrador da instituição de saúde	36
Figura 6 - Diagrama de caso de uso enfermeira/paramédico.....	36
Figura 7 - Diagrama do caso de uso do(a) médico(a)	37
Figura 8 - Diagrama de caso de uso paciente.....	38
Figura 9 - Diagrama de classe da API de Usuários.....	62
Figura 10 - Trecho de código demonstrando a conexão com o banco de dados	64
Figura 11 - Diagrama de classe da API do paciente	65
Figura 12 - Diagrama de classe da API de médico (a).....	67
Figura 13 - Diagrama de Classe da API do(a) Enfermeiro(a).....	69
Figura 14 - Diagrama de Classe da API de instituição de saúde	71
Figura 15 - Diagrama de Classe da API de trilha de auditoria	73
Figura 16 - Diagrama de Classe da API de Erro	74
Figura 17 - Diagrama de classe da funcionalidade de Login.....	77
Figura 18 - Diagrama de classe da navegação do sistema.....	78
Figura 19 - Diagrama de classe do gerenciamento de instituições de saúde	79
Figura 20 - Diagrama de classe da seleção de instituição de saúde.....	80
Figura 21 - Diagrama de classe da trilha de auditoria	81
Figura 22 - Diagrama de classe de log de erros.....	81
Figura 23 - Diagrama de classe de alterar perfil.....	82

Figura 24 - Diagrama de classe de notificação	83
Figura 25 - Diagrama de classe de gerenciar usuários	84
Figura 26 – Diagrama de classe de adicionar usuário na instituição de saúde	85
Figura 27 - Diagrama de classe de gerenciamento de pacientes.....	86
Figura 28 - Diagrama de classe da janela de autenticação.....	87
Figura 29 - Diagrama de classe de adicionar novo paciente.....	87
Figura 30 - Diagrama de classe de gerenciamento das doenças do paciente	88
Figura 31 - Diagrama de classe de gestão de diagnósticos do paciente	89
Figura 32 - Diagrama de classe de gerenciamento de exames	90
Figura 33 - Diagrama de classe de gerenciamento de medicamentos.....	90
Figura 34 - Diagrama de classe de gerenciar equipamentos de acesso	91
Figura 35 - Diagrama de classe de Login e Navegação.....	93
Figura 36 - Diagrama de classe de seleção de instituição de saúde.....	94
Figura 37 - Diagrama de classe de listagem de novos pacientes na fila	95
Figura 38 - Diagrama de classe de gerenciar lista de atendimentos em progresso ..	96
Figura 39 - Diagrama de classe da funcionalidade de inicialização	98
Figura 40 - Diagrama de classe de cadastro de paciente	99
Figura 41 - Diagrama de classe da tela inicial do sistema	100
Figura 42 - Diagrama de classe de listagem de instituições de saúde próximas	101
Figura 43 - Diagrama de classe de listagem de médicos(as).....	102
Figura 44 - Diagrama de classe de gerenciamento de medicamentos.....	103
Figura 45 - Diagrama de classe de sincronização.....	104
Figura 46 - Diagrama de classe de exames	104
Figura 47 - Diagrama de sequência da leitura do equipamento <i>NFC</i>	106
Figura 48 - DER do banco de dados armazenado na Azure	107

Figura 49 - DER do banco de dados local no dispositivo móvel.....	108
Figura 50 - Circuito.....	118
Figura 51 - Definição da interface de comunicação	119
Figura 52 - <i>Setup</i>	120
Figura 53 - <i>Loop</i>	120
Figura 54 – Gráfico de <i>Burndown</i> da entrega 1.....	124
Figura 55 – Gráfico de <i>Burndown</i> da entrega 2.....	125
Figura 56 – Gráfico de <i>Burndown</i> da entrega 3.....	127
Figura 57 – Gráfico de <i>Burndown</i> da entrega 4.....	128
Figura 58 – Gráfico de <i>Burndown</i> da entrega 5.....	130
Figura 59 – Gráfico de <i>Burndown</i> da entrega 6.....	131
Figura 60 – Gráfico de <i>Burndown</i> da entrega 7.....	133
Figura 61 – Gráfico de <i>Burndown</i> da entrega 8.....	135
Figura 62 - <i>Login</i> do sistema <i>desktop</i>	136
Figura 63 - Tela inicial do <i>desktop</i> do usuário administrador do sistema	137
Figura 64 – Menu opções do usuário administrador do sistema	137
Figura 65 - Tela de instituições médicas	138
Figura 66 – Janela adicionar instituição de saúde.....	138
Figura 67 – Mensagem de confirmação de adição de instituição.....	139
Figura 68 – Janela editar instituição de saúde	139
Figura 69 – Mensagem de confirmação de edição da instituição.....	140
Figura 70 - Mensagem de exclusão	140
Figura 71 – Mensagem de confirmação de exclusão de instituição	140
Figura 72 – Tela gerenciar usuários da instituição de saúde	141
Figura 73 – Janela adicionar usuário administrador do sistema	141

Figura 74 – Janela adicionar usuário da instituição.....	142
Figura 75 – Mensagem de confirmação de adição de usuário	142
Figura 76 – Janela de envio de convite para usuário já existente	142
Figura 77 – Mensagem de convite enviado com sucesso.....	143
Figura 78 – Janela visualizar usuário	143
Figura 79 – mensagem de exclusão de usuário.....	144
Figura 80 – Mensagem de usuário removido da instituição com sucesso	144
Figura 81 -Tela de log de erros	144
Figura 82 – Tela de log de atividades	145
Figura 83 – Janela de atualização de dados da conta do usuário administrador do sistema.....	146
Figura 84 – Mensagem de usuário administrador do sistema alterado com sucesso	146
Figura 85 - Tela de notificações usuário administrador do sistema.....	147
Figura 86 - <i>Login</i> do sistema <i>desktop</i>	147
Figura 87 – Tela de seleção de instituição	148
Figura 88 - Tela inicial das instituições de saúde	149
Figura 89 - Menu opções do usuário administrador da instituição do sistema	149
Figura 90 – Tela selecionar nova instituição de saúde.....	150
Figura 91 - Tela de gerenciamento de usuário da instituição de saúde	150
Figura 92 – Janela adicionar usuário administrador do sistema para instituição	151
Figura 93 – Janela adicionar usuário da instituição.....	151
Figura 94 – Mensagem de confirmação de adição de usuário.....	152
Figura 95 – Janela para envio de convite para usuário já existente.....	152
Figura 96 – Mensagem de convite enviado com sucesso.....	152

Figura 97 – Janela visualizar usuário	153
Figura 98 – Janela de exclusão de usuário	153
Figura 99 – Mensagem usuário removido da instituição com sucesso	153
Figura 100 – Janela de atualização de dados do usuário administrador da instituição de saúde	154
Figura 101 – Mensagem de usuário administrador do sistema alterado com sucesso	154
Figura 102 - Tela de notificações usuário administrador do sistema.....	155
Figura 103 - <i>Login</i> do sistema <i>desktop</i>	155
Figura 104 – Tela de seleção de instituição	156
Figura 105 - Tela inicial do usuário médico	156
Figura 106 - Tela de notificações do usuário médico	157
Figura 107 - Menu opções do usuário médico	157
Figura 108 - Tela de seleção de instituição de saúde do usuário médico	158
Figura 109 - Tela de atendimentos do médico	158
Figura 110 - Janela de solicitação de código de acesso	159
Figura 111 - Tela com as informações do paciente.....	160
Figura 112 - Tela de doenças.....	161
Figura 113 - Janela de adição de doenças	161
Figura 114 - Mensagem de doença adicionada com sucesso	162
Figura 115 - Tela de edição de doença	162
Figura 116 - Mensagem de atualização de doença realizada com sucesso	162
Figura 117 - Janela de exclusão de doença.....	163
Figura 118 - Mensagem de doença excluída com sucesso	163
Figura 119 - Tela de diagnósticos	163

Figura 120 - Janela de visualização de diagnóstico	164
Figura 121 - Janela de informações do médico.....	164
Figura 122 - Janela da instituição de saúde.....	165
Figura 123 - Janela de procedimento de enfermagem.....	165
Figura 124 - Janela de adição de diagnóstico.....	166
Figura 125 - Mensagem de diagnóstico adicionado com sucesso	166
Figura 126 - Tela de exames.....	167
Figura 127 - Janela de visualização de exames.....	167
Figura 128 - Janela do médico que realizou o exame	168
Figura 129 - Janela da instituição de saúde	168
Figura 130 - Janela adicionar exames	169
Figura 131 - Mensagem de exame adicionado com sucesso.....	169
Figura 132 - Tela de medicamentos	170
Figura 133 - Tela de equipamentos de acesso	170
Figura 134 - Janela de inserção de equipamento	171
Figura 135 - Janela de solicitação do equipamento de acesso <i>NFC</i>	171
Figura 136 - Janela de edição de equipamento	172
Figura 137 - Janela de confirmação de exclusão de equipamento de acesso	172
Figura 138 - Janela de adição de paciente	173
Figura 139 - Janela de atualização de usuário médico	173
Figura 140 - Tela de autenticação.....	174
Figura 141 - Tela inicial do sistema.....	175
Figura 142 - Menu	175
Figura 143 - Atualização dos dados do paciente.....	176
Figura 144 - Tela de instituições	177

Figura 145 - Tela de médicos(as).....	178
Figura 146 - Tela de exames.....	178
Figura 147 - Medicamentos.....	179
Figura 148 - Tela de carregamento e autenticação	180
Figura 149 - Tela de seleção de instituições de saúde	181
Figura 150 - Estrutura principal do aplicativo do(a) enfermeiro(a).....	181
Figura 151 - Adição de diagnóstico	182
Figura 152 - Paciente na tela inicial	182
Figura 153 - Lista de atendimentos em progresso	183
Figura 154 - Explicação sobre o código de identificação sul africano	192

LISTA DE TABELAS

Tabela 1 - Tabela de sistemas de saúde relacionados	21
Tabela 2 – Comparativo de funcionalidades entre os sistemas apresentados e o NFC Health System.....	22
Tabela 3 - Requisitos funcionais do projeto.	25
Tabela 4 – Requisitos não funcionais do projeto.....	26
Tabela 5 – Caso de uso “Efetuar login”.....	39
Tabela 6 - Caso de uso “Gerenciar instituições de saúde”.....	39
Tabela 7 - Caso de uso “Gerenciar usuários”.....	41
Tabela 8 - Caso de uso “Gerenciar log de erros”	43
Tabela 9 – Caso de uso “Gerenciar trilha de auditoria”.....	43
Tabela 10 - Caso de uso " Gerenciar perfil"	44
Tabela 11 - Caso de uso "Selecionar instituição de saúde"	45
Tabela 12 - Caso de uso "Gerenciar notificações"	45
Tabela 13 - Caso de uso " Visualizar atendimentos".....	46
Tabela 14 - Caso de uso " Gerenciar paciente"	46
Tabela 15 - Caso de uso " Gerenciar dados de saúde"	48
Tabela 16 - Caso de uso "Gerenciar doenças"	49
Tabela 17 - Caso de uso " Gerenciar diagnósticos"	50
Tabela 18 - Caso de uso "Gerenciar exames"	51
Tabela 19 - Caso de uso " Visualizar medicamentos".....	52
Tabela 20 - Caso de uso "Solicitar procedimentos"	53
Tabela 21 - Caso de uso " Gerenciar procedimentos"	54
Tabela 22 - Caso de uso "Visualizar histórico médico"	55

Tabela 23 - Caso de uso " Visualizar instituições de saúde"	55
Tabela 24 - Caso de uso " Visualizar médicos"	56
Tabela 25 - Caso de uso " Gerenciar medicamentos".....	57
Tabela 26 - Caso de uso " Manter exames"	58
Tabela 27 - Caso de uso " Gerenciar perfil"	59
Tabela 28 - Caso de uso " Realizar cadastro".....	60
Tabela 29 - Caso de uso "Sincronizar dados"	60
Tabela 30 - Especificação dos pacotes utilizados	61
Tabela 31 - Recursos da <i>API</i> de usuário.....	63
Tabela 32 - Formato do retorno do JSON com base nas classes de retorno.....	64
Tabela 33 - Recursos da <i>API</i> de paciente	65
Tabela 34 - Recursos disponibilizados pela <i>API</i> de médico(a).....	68
Tabela 35 - Recursos disponibilizados pela <i>API</i> de enfermeiro(a)	70
Tabela 36 - Recursos disponibilizados pela <i>API</i> de instituição de saúde	71
Tabela 37 - Recursos disponibilizados pela <i>API</i> de trilha de auditoria	73
Tabela 38 - Recursos disponibilizados pela <i>API</i> de erro	75
Tabela 39 - Especificação dos pacotes utilizados	76
Tabela 40 - Tabela de permissões	78
Tabela 41 - Especificação dos pacotes utilizados	91
Tabela 42 - Especificação dos pacotes utilizados	96
Tabela 43 - Dicionário de dados da entidade AUDIT_TRAIL	109
Tabela 44 - Dicionário de dados da entidade DIAGNOSIS	109
Tabela 45 - Dicionário de dados da entidade DIAGNOSIS_PROCEDURE	110
Tabela 46 - Dicionário de dados da entidade DISEASE	110
Tabela 47 - Dicionário de dados da entidade ERROR_LOG	110

Tabela 48 - Dicionário de dados da entidade EXAM.....	111
Tabela 49 - Dicionário de dados da entidade EXAM_ATTACHMENT	111
Tabela 50 - Dicionário de dados da entidade HEALTH_INSTITUTION	111
Tabela 51 - Dicionário de dados da entidade MEDICINE	112
Tabela 52 - Dicionário de dados da entidade NURSE	112
Tabela 53 – Dicionário de dados da entidade NURSE_HAS_SPECIALIZATION ..	112
Tabela 54 - Dicionário de dados da entidade NURSE_SPECIALIZATION	113
Tabela 55 - Dicionário de dados da entidade PATIENT	113
Tabela 56 - Dicionário de dados da entidade PATIENT_HAS_DISEASE	113
Tabela 57 - Dicionário de dados da entidade PATIENT_HAS_TAG	113
Tabela 58 - Dicionário de dados da entidade PATIENT_USE_MEDICINE	114
Tabela 59 - Dicionário de dados da entidade PHYSICIAN.....	114
Tabela 60 - Dicionário de dados da entidade PHYSICIAN_ATTENDANCE	114
Tabela 61 - Dicionário de dados da entidade PHYSICIAN_HAS_SPECIALIZATION	115
Tabela 62 - Dicionário de dados da entidade PHYSICIAN_ SPECIALIZATION.....	115
Tabela 63 - Dicionário de dados da entidade USER_AND_HEALTH_INSTITUTION	115
Tabela 64 - Dicionário de dados da entidade USER_HEALTH.....	116
Tabela 65 - Dicionário de dados da entidade PATIENTUSES_MEDICINE do SQLite	116
Tabela 66 - Dicionário de dados da entidade PATIENTUSES_MEDICINE_IN_HOUR do SQLite	117
Tabela 67 - Componentes utilizados	117
Tabela 68 - Ligações entre o Arduino e o módulo PN532.....	118

Tabela 69 -Planejamento realizado da primeira entrega	123
Tabela 70 -Planejamento realizado da segunda entrega	124
Tabela 71 -Planejamento realizado da terceira entrega.....	126
Tabela 72 - Planejamento realizado da quarta entrega.....	128
Tabela 73 -Planejamento realizado da quinta entrega	129
Tabela 74 -Planejamento realizado da sexta entrega	131
Tabela 75 -Planejamento realizado da sétima entrega	132
Tabela 76 -Planejamento realizado da oitava entrega	134
Tabela 77 - Equipes	190

1 INTRODUÇÃO

A tecnologia vem estando cada vez mais presente na realidade da população, e está se tornando algo cada vez mais natural, virando uma extensão da realidade (TIEGHI, 2014). Muitas pessoas já utilizam recursos tecnológicos para melhorar a vida de muitos portadores de diversos tipos de problemas (TIEGHI, 2014). Como por exemplo, com o advento de novas tecnologias que auxiliam na área médica, uma grande quantidade de dados sobre o câncer tem sido coletada e está disponível para comunidades de pesquisa médica (KOUROU, 2014).

Foi realizada uma pesquisa pelo GE Healthcare Worldwide¹ em 2014 com cerca de dez mil pessoas distribuídas em dez países, tais como Estados Unidos, Brasil, China, Índia e Reino Unido. Dentre algumas tecnologias que a população brasileira acredita que será importante no futuro em relação a saúde, estão os seguintes itens:

- Tecnologias de monitoramento de saúde fora do hospital (95%).
- Equipamentos portáteis (94%).
- Sistemas para diminuir o número de cancelamentos e atrasos nos atendimentos (94%).
- Gravação e armazenamento digital dos dados dos pacientes (93%).

Também é possível perceber que grande parte dos entrevistados acreditam que a tecnologia irá possibilitar ainda mais benefícios no atendimento ao paciente, tais como:

- Criação de equipamentos de diagnósticos mais rápidos (83%).
- Tecnologias para reduzir o tempo de espera em atendimentos (81%).
- Acesso instantâneo dos médicos aos dados de saúde do paciente (74%).
- Acesso por parte do paciente aos seus próprios dados médicos (67%).

A saúde vem enfrentando diversos desafios, como o aumento de doenças crônicas, orçamentos reduzidos, envelhecimento da população, dentre diversos outros. Os resultados da pesquisa analisada anteriormente comprovam que as

¹ GE Healthcare Worldwide é uma empresa que fabrica e distribui agentes de diagnóstico por imagem - <http://www3.gehealthcare.com.br/>

condições políticas, sociais e econômicas do país interferem grandemente na infraestrutura médica fornecida por cada local (Portal Hospitais Brasil, 2015).

Com o crescente aumento de atendimentos de emergência e urgência no Brasil, muitas vezes causados principalmente por violência, acidentes de trânsito e doenças cardiovasculares, surge uma grande necessidade de um atendimento específico visando a realização dos primeiros socorros da forma mais rápida e eficiente possível (ROCHA, 2012). De acordo com a Organização Pan-Americana de Saúde², os serviços de saúde de emergência e urgência visam prolongar a vida da vítima ou realizar a prevenção de consequências críticas, necessitando ser proporcionados imediatamente.

O atendimento do paciente de urgência ou emergência é dividido em três passos realizados de modo sucessivo, sendo eles o atendimento na cena do acidente, durante o transporte e no centro hospitalar (GUIDO, 1995). O grande problema é que muitas vezes falta informações do paciente que acaba de sofrer um acidente, e em diversas vezes o mesmo muitas vezes chega ao centro hospitalar sem nenhum acompanhante.

Seria extremamente útil uma ferramenta de fácil acesso de informações durante uma emergência, muitas vezes o paciente chega ao hospital sem acompanhante, e não temos nenhuma informação, e embora nós oferecemos todo tipo de cuidado, podemos cometer algum erro, como por exemplo, o paciente é diabético e ele é medicado com soro glicosado. Seria muito bom não correr este risco. (SCHIAVINATO, 2018).

Embora os serviços de saúde trabalhem duro para fornecer um serviço de alta qualidade e segurança, as vezes as pessoas acabam sendo muito prejudicadas. A Organização Mundial da Saúde consta que um dos desafios globais na área médica é o cuidado da saúde de modo inseguro. Todos os dias milhões de pessoas utilizam serviços de saúde, portanto é extremamente necessário a redução de danos causados por atendimentos de forma incorreta, que podem causar hospitalizações desnecessárias e em alguns casos, desde a incapacidade e até a morte.

Embora não haja um consenso sobre a definição de erros de medicação, o Conselho Nacional de Coordenação dos Estados Unidos para Erros de Medicação e Prevenção define um erro de medicação do seguinte modo:

² Organização Pan Americana de saúde é uma organização internacional especializada em saúde - <https://www.paho.org/bra/>

Any preventable event that may cause or lead to inappropriate medication use or patient harm while the medication is in the control of the health care professional, patient, or consumer. Such events may be related to professional practice, health care products, procedures, and systems, including prescribing, order communication, product labelling, packaging, and nomenclature, compounding, dispensing, distribution, administration, education, monitoring, and use (World Health Organization, 2016).³

Estimar as taxas de prevalência de erros de medicação é uma tarefa difícil mediante a grande variedade de definições e sistemas de classificação utilizados. Um estudo realizado no México observou que 58% das prescrições continham erros de dosagem para a maioria dos casos (27,6%). No Reino Unido, um estudo observou que 12% dos pacientes podem ser afetados por prescrição ou monitoramento de forma errada ao longo de um ano. Estes dados apresentados pela Organização Mundial da Saúde mostram como os erros de medicamentos são um problema global.

Devido aos problemas apresentados e a grande necessidade de proporcionar um atendimento eficiente, rápido e seguro ao paciente, juntamente a grande necessidade de compartilhar dados essenciais dos pacientes entre hospitais para o devido atendimento, o presente projeto propõe o desenvolvimento de um sistema que auxilie toda a área hospitalar a obter informações do paciente através de uma ficha médica armazenada de forma digital, auxiliando assim todo o corpo médico no processo de tomada de decisão, evitando possíveis erros médicos devido à falta de informação. Além disso, o paciente terá acesso a seu prontuário médico sempre quando for necessário.

O sistema também fornecerá mecanismos para que o paciente possa realizar o monitoramento constante de sua situação de saúde, assim como visualizar sua ficha médica. De acordo com o código de ética médica, é vedado ao médico negar o acesso ao prontuário médico, ficha clínica ou similar, salvo quando ocasionar riscos ao paciente ou a terceiros. Apesar do termo “prontuário médico”, o documento é de propriedade do paciente.

³ Qualquer evento evitável pelo uso de medicação inapropriada que possa causar ou levar o paciente a sofrer danos, enquanto a medicação está no controle do profissional de saúde, paciente ou consumidor. Tais eventos podem estar relacionados a prática profissional, produtos de saúde, procedimentos e sistemas, incluindo prescrição, comunicação de pedidos, rotulagem de produto, embalagem, nomenclatura, composição, distribuição, administração, educação, monitoramento e uso (World Health Organization, 2016. Tradução nossa).

Todas as informações relacionadas a saúde do paciente serão guardadas de forma sigilosa, sendo que apenas o paciente e a área médica terão acesso às informações, assim como consta na Constituição Federal e no Código Penal, que garantem a privacidade do indivíduo (CONSELHO FEDERAL DE MEDICINA, 1999).

O preenchimento do prontuário médico é obrigação e responsabilidade intransferível do médico, com exceção aos hospitais de ensino, onde os alunos de medicina realizam o preenchimento sob supervisão e responsabilidade de médicos (professores de medicina ou hospital de ensino) (CONSELHO REGIONAL DE MEDICINA DO ESTADO DE SANTA CATARINA, 2000). Sendo assim, apenas o médico poderá preencher e alterar as informações presentes no prontuário. A edição do documento por qualquer outra pessoa é uma prática antiética e ilegal, sendo condenável o preenchimento por alguém que não esteja habilitado perante o Conselho de Medicina.

O sistema irá realizar tanto a leitura quanto gravação do prontuário médico utilizando a tecnologia *NFC*. O *NFC* é, de modo geral um conjunto de tecnologias sem fio de curto alcance (normalmente de cerca de dez centímetros) que consiste em três elementos básicos: um leitor, uma antena e uma etiqueta para o armazenamento de dados. Dentre as principais aplicações que utilizam a tecnologia *NFC* atualmente, é possível citar o comércio eletrônico, identificação pessoal e aceleramento de conexões.

A fim de se obter uma melhor visão das necessidades do sistema, do ambiente de atuação e de se realizar um levantamento de informações, foram analisados alguns sistemas relacionados a área da saúde que possuem uma forma de atuação semelhante ao sistema do presente documento. A Tabela 1 apresenta uma relação dos sistemas analisados.

Tabela 1 - Tabela de sistemas de saúde relacionados

Sistema	Funcionalidades e características do sistema	Preço	Localização, Atendimento e serviços
Medicina Direta	<ul style="list-style-type: none"> -Não existe carência ou período mínimo de contrato. -Tecnologia em Nuvem. -Aplicativo para IPHONE e IPAD, e Android (não nativo). -5 Gigabytes/s de armazenamento de dados. -100 SMS por usuários por mês. -E-mails ilimitados. -Ficha do paciente completa. -Diagnósticos. -Upload de imagens e exames. -Atestados. -Receitas. -Cadastro de medicamentos. -Gerenciamento financeiro. -Sistema de agendamento. 	<p>Para a inclusão do serviço é necessária uma taxa de adesão no primeiro mês de cerca de R\$ 599,00. Também é necessário pagar o valor unitário de cada usuário do sistema. Conforme mais usuários adicionados, mais desconto é adicionado no valor do prontuário médico, como por exemplo, 1 usuário possui um custo de R\$ 139,00, o que totaliza R\$ 139,00, enquanto 20 usuários possuem um custo unitário de R\$ 75,00 que totalizam R\$ 1500,00. Descontos são aplicativos caso seja feita a aquisição de outras soluções da empresa.</p> <p>Pagamento por cartão de crédito, sem a necessidade do aluguel da máquina de cartão. Conforme é feito a compra de mais serviços do cliente, mais é aplicado um desconto encima do valor total. Sempre é cobrado uma taxa de adesão.</p>	Localização Rua Américo Brasiliense, 1923 CJ 1602 Chácara Santo Antônio - São Paulo - SP CEP: 04715-005 Atendimento O suporte é total via telefone, TeamViewer e E-mail. Serviços A empresa fornece o serviço de Prontuário médico eletrônico, desenvolvimento de websites para médicos, consultórios e clínicas, juntamente ao serviço de divulgação utilizando Google Adwords.
MedicalBox	<ul style="list-style-type: none"> -Agenda. -Pacientes. -Prontuário médico. -Relatório. -Bulário. -Atendimentos. 	<p>É fornecido um mês de período gratuito. Após o término do período de teste, é feito a cobrança de R\$ 98,00 mensais, sendo possível cancelar a qualquer momento.</p> <p>Para a utilização da ferramenta é necessário informar o cartão de crédito.</p>	Localização Av. Dr. Francisco Pereira Lopes, 1701 Lojas 17 e 18 Parque Santa Monica, São Carlos/SP CEP 13564-002 Atendimento Suporte é realizado via chat rapidamente dentro do sistema, sendo possível tirar qualquer dúvida a qualquer hora. Serviços A MedicalBox fornece serviços de agendamentos inteligentes, com possibilidade de anotações, automação de tarefas (enviando SMSs para os pacientes e e-mail), backups diários das informações, prontuário médico eletrônico e gestão de pacientes. Eles também fornecem um serviço de bulário e o CID-10, que é sobre o código internacional de doenças.
IClinic	<ul style="list-style-type: none"> -Agenda. -Cadastro de pacientes. -Prontuário eletrônico. -Prescrição eletrônica. -Faturamento TISS. -Envio de lembretes. -Controle financeiro. -Relatório financeiro. -Envio de SMS. 	<p>A empresa possui três planos, sendo eles o <i>starter</i> (R\$ 79,00 mensal), o <i>plus</i> (R\$ 99,00 mensal) e o <i>pro</i> (R\$ 119,00 mensal). Esse valor é o custo do uso do sistema por cada profissional de saúde, por exemplo, se 10 profissionais de saúde utilizarem o plano <i>starter</i>, o sistema custaria em torno de R\$ 799,00 mensal.</p> <p>O pagamento deve ser feito por cartão e o valor do plano é multiplicado pela quantidade de pacientes. O plano assinado pode ser mensal ou anual. No plano anual ocorre um desconto.</p>	Localização Ribeirão Preto/SP e São Paulo/SP Atendimento O atendimento é via chat, e-mail ou telefone. O atendimento é feito de maneira rápida. Eles também possuem uma central de suporte. Serviço Dentre os serviços fornecidos estão a agência, pacientes, finanças, relatórios, envio de SMS, CID 10 e bulário.
Shosp	<ul style="list-style-type: none"> -Agendamento Online. -Prontuário Eletrônico. -Faturamento TISS. -Controle financeiro. -SMS e E-mail para notificar paciente. -Múltiplas clínicas. -Emissão de Nota fiscal de serviço. -Controle de estoque. -Fila de espera 	<p>A empresa possui três planos, sendo que um deles é o experimental, sendo gratuito que não possui suporte a múltiplas clínicas.</p> <p>O outro é o <i>fellowship</i>, que custa R\$ 69,94 por prestador (profissional de saúde cadastrado), enquanto o último plano é o <i>Excelence</i>, que custa em torno de R\$ 99,00 por prestador. O custo do treinamento é de R\$ 1500,00.</p> <p>É possível pagar com cartão de crédito ou boleto bancário. O usuário recebe e-mails para lembrá-lo das faturas.</p> <p>Todo o suporte é ilimitado, sem custo adicional, via chat, e-mail, helpdesk e telefone. Além disso, há atualizações periódicas do software e atualizações de hardware para melhor performance, backups automáticos e armazenamento ilimitado de dados.</p>	Localização Rio de Janeiro - RJ: Av. Churchill, 94, Centro Atendimento O atendimento é por telefone, e-mail, chat e helpdesk. Serviço Dentre os serviços estão agenda médica, agendamento online de pacientes, prontuário eletrônico, faturamento TISS, controle financeiro e envio de SMS e e-mail.
IMedicina	<ul style="list-style-type: none"> -Prontuário Eletrônico. -Finanças e Estatísticas. -Agenda e Agendamento Online. -Relacionamento com Paciente. -Automação de E-mail e SMS. -Implementação acompanhada. -Conteúdo relativo a marketing e divulgação. 	<p>Existem três planos:</p> <ul style="list-style-type: none"> -Plano <i>start</i>: R\$ 397 mensal. -Plano <i>full</i>: R\$ 497 mensal. -Plano <i>advanced</i>: Customizado. <p>Os planos são mensais. A cada plano o usuário ganha mais recursos. O plano <i>start</i> possui recursos essenciais ao atendimento. Já os outros planos envolvem recursos relacionados a sites, gestão de anúncios, produção de artigos, pautas estratégicas para conquistar mais clientes, dentre outros recursos.</p>	Localização Feito em Belo Horizonte – MG Atendimento Atendimento por e-mail, telefone e chat. Serviço O imedicina fornece serviços como indicadores relevantes (atendimentos realizados, número de novos pacientes), finanças simples, gestão de pacientes, criação de site, juntamente a serviços de divulgação de conteúdo e marketing.

Fonte: Elaborado pelo autor

Levando estes aspectos em consideração, foi elaborada a Tabela 2 mostrando as principais diferenças entre os sistemas analisados e o NFC Health System. Note que, a sigla S1 destacada representa o sistema desenvolvido neste trabalho, enquanto S2, S3, S4, S5 e S6, representam, respectivamente, Medicina Direta, MedicalBox, IClinic, Shosp e IMedicina.

Tabela 2 – Comparativo de funcionalidades entre os sistemas apresentados e o NFC Health System.

Funcionalidade	S1	S2	S3	S4	S5	S6
Prontuário eletrônico	X	X	X	X	X	X
Gestão de pacientes	X	X	X	X	X	X
Gestão de diagnósticos	X	X	X	X	X	X
Gestão de exames	X					
Gestão de medicamentos	X	X				
Busca por médicos próximos	X					
Agendamento de atendimento		X			X	X
Busca por instituições de saúde próximas	X					
Funcionalidades de enfermagem	X					
Sistema de envio de e-mails e SMS		X		X	X	
Segurança baseada em equipamentos de acesso e/ou códigos de acesso	X					
Bulário			X	X		
Histórico resumido de cada paciente	X					
Lista de unificada de todos os procedimentos um paciente passou, independente da instituição de saúde	X					
Lista de doenças	X			X	X	

Fonte: Elaborado pelo autor

Este trabalho tem como objetivo criar um *software* que auxilie funcionários de saúde durante os atendimentos, provendo as informações disponíveis necessárias (que são filtradas para garantir o sigilo) em relação a saúde do paciente, aumentando assim a eficiência do atendimento e fornecendo uma menor possibilidade de erros. O software também visa aumentar a comunicação entre instituições de saúde no que diz respeito as fichas médicas dos pacientes, assim aumentando também a velocidade de atendimento. Por fim, o sistema visa fornecer ao paciente meios pelo qual ele possa

gerenciar seus documentos de saúde de modo totalmente digital, tendo acesso a uma série de recursos adicionais. Quanto aos objetivos específicos são os seguintes:

- Utilizar a metodologia SCRUM e expor os procedimentos realizados;
- Fornecer informações de saúde do paciente em momento de atendimento.
- Aumentar os cuidados de emergência que podem ser realizados.
- Criar uma ficha médica *online* globalizada.
- Apresentar as conclusões finais e exibir as possibilidades para trabalhos futuros.

O restante do trabalho está organizado em três capítulos conforme descrição a seguir: Capítulo 2 apresenta o projeto do sistema desenvolvido, o Capítulo 3 descreve o desenvolvimento do projeto, e por fim, as considerações finais juntamente com as diversas possibilidades de trabalhos futuros são apresentadas no Capítulo 4.

2 PROJETO DO NFC HEALTH SYSTEM

O software conta com quatro módulos, um sistema *desktop* e três aplicativos Android. A versão *desktop* é acessada principalmente por médicos(as), administradores do sistema e da instituição de saúde. O sistema *desktop* atualmente se encontra desenvolvido.

O paciente terá um aplicativo para Android, onde terá acesso a todas suas informações, tais como ficha médica, exames, histórico médico, médicos(as) que possuem acesso a sua ficha, medicamentos em uso, instituições de saúde e médicos (as) mais próximos(as). O aplicativo do paciente atualmente se encontra desenvolvido.

O segundo aplicativo para Android é voltado para enfermeiros(as). Através dele poderão realizar a gestão dos procedimentos nos pacientes que se encontram na fila de atendimento da instituição de saúde. Este aplicativo também já se encontra desenvolvido.

O terceiro aplicativo é utilizado por paramédicos (as). No aplicativo, eles apenas podem visualizar a ficha de emergência do paciente que conterá informações como nome, endereço, RG, CPF, tipo sanguíneo, doenças, alergias e medicamentos em uso. Este aplicativo atualmente se encontra em fase de planejamento.

Para agilizar o acesso às informações, o sistema conta com leitores e *tags NFC*, que, assim como já dito, é uma tecnologia que faz transmissão de dados de curto alcance sem a utilização de fios.

2.1 Levantamento de Requisitos

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 3 apresenta os requisitos funcionais deste projeto.

Tabela 3 - Requisitos funcionais do projeto.

Identificação	Requisito Funcional	Prioridade
RF001	Um(a) administrador(a) deve ser capaz de gerir instituições de saúde	Essencial
RF002	Um(a) administrador(a) deve ser capaz de gerir usuários de cada instituição.	Importante
RF003	Um(a) administrador(a) da instituição de saúde deve ser capaz de gerir os usuários da respectiva instituição de saúde.	Essencial
RF004	Um(a) médico(a) deve ser capaz de gerir seus atendimentos.	Essencial
RF005	Um(a) médico(a) deve ter acesso aos dados de saúde do paciente, tais como dados gerais, doenças, exames, diagnósticos e medicamentos utilizados pelo paciente.	Essencial
RF006	O sistema deve gerar uma lista dos atendimentos realizados em um dia específico.	Importante
RF007	Cada usuário paciente que utiliza o sistema deve possuir um equipamento de acesso ao qual deverá possuir um código de vinte e um dígitos.	Essencial
RF008	Um(a) médico(a) deverá poder enviar um paciente para a fila de espera de procedimentos da instituição de saúde	Essencial
RF009	Um(a) enfermeiro(a) deve ter acesso a fila de procedimentos da instituição de saúde.	Essencial
RF010	Um(a) enfermeiro(a) deve poder realizar procedimentos no paciente	Essencial
RF011	Um(a) administrador(a) do sistema deverá poder acessar a trilha de auditoria do sistema.	Essencial
RF012	Um(a) administrador(a) deve ser capaz de gerar uma lista de erros que ocorreram.	Desejável
RF013	Um paciente deverá poder ver seu histórico médico resumido	Essencial
RF014	Um paciente deverá ter acesso a seus exames	Essencial
RF015	Um paciente deverá ter acesso a seus exames	Essencial

RF016	Um paciente deverá poder acessar as instituições de saúde cadastradas no sistema mais próximas	Essencial
RF017	Um paciente deverá poder acessar médicos(as) mais próximos cadastrados no sistema.	Essencial
RF018	Todos os perfis de usuário deverão poder editar suas informações pessoais.	Importante
RF019	Um paciente deverá poder pesquisar por médicos baseado em especializações.	Desejável
RF020	Um(a) paramédico(a) deverá ter acesso a uma ficha resumida de saúde do paciente para o atendimento.	Importante
RF021	Os sistemas devem suportar internacionalização para o Brasil e África do Sul.	Essencial
RF022	Um(a) paramédico(a) deverá poder adicionar procedimentos realizados no paciente.	Importante
RF023	Um paciente deve poder gerenciar seus medicamentos e os horários que deseja utiliza-los.	Essencial
RF024	Um paciente deve ter acesso a um código secreto de quatro dígitos caso não possua um equipamento de acesso para a realização de sua consulta.	Essencial
RF025	Um paciente deve ser capaz de sincronizar os medicamentos armazenados localmente com a nuvem.	Desejável

Fonte: Elaborado pelo autor

2.1.2 Requisitos Não Funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 4 apresenta os requisitos não funcionais deste projeto.

Tabela 4 – Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Design atrativo e de alta responsividade nas interfaces de usuário	Usabilidade	Essencial
RNF002	O sistema deve ser de facilmente utilizado de modo que os erros dos usuários sejam minimizados.	Usabilidade	Importante
RNF003	O sistema deverá retornar respostas aos usuários em no máximo 10 segundos.	Desempenho	Desejável
RNF004	O sistema de webservices deverá ter um baixo consumo de memória para bom funcionamento na Microsoft Azure.	Desempenho	Importante
RNF005	Deve ser utilizado Microsoft Azure como plataforma.	Ambiental	Essencial

RNF006	Deve ser utilizado o Sistema gerenciador de banco de dados SQL Server.	Ambiental	Essencial
RNF007	O sistema deve ser desenvolvido na linguagem Java (tanto o aplicativo, quanto os <i>webservices</i>).	Desenvolvimento	Essencial
RNF008	Deve ser utilizado o Google Cloud Platform para se utilizar de recursos de tradução, geolocalização e mapas	Ambiental	Essencial
RNF009	Os sistemas devem atender aos códigos de ética da saúde definidos pelo Brasil e África do Sul.	Ético	Essencial
RNF010	Apenas o paciente e usuários habilitados devem poder acessar os dados de saúde do paciente.	Segurança	Essencial

Fonte: Elaborado pelo autor

2.2 Recursos e Ferramentas

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do NFC Health System.

- **Android Studio:** ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android, baseado no IntelliJ IDEA⁴, que oferece um ambiente unificado para o desenvolvimento de aplicativos. No qual é possível desenvolver, fazer debugs, testes e interfaces para smartphones e tablets Android e dispositivos Android Wear, Android TV e Android Auto (ANDROID, 2017).
- **Eclipse:** É uma plataforma aberta que facilita o processo de desenvolvimento, fornecendo ferramentas para codificação, construção, execução e *debug* de aplicações (UMBC, 2008). Esta ferramenta suporta o desenvolvimento em diversas linguagens e foi utilizada para o desenvolvimento dos *webservices* do sistema.
- **SQL Server Management Studio:** O SSMS é um ambiente integrado que é utilizado para gerenciamento da infraestrutura de SQL, atuando do SQL Server para o Banco de dados SQL do Azure. Fornece ferramentas para configurar, monitorar e gerenciar as instâncias do SQL, e para implantar, monitorar e atualizar componentes da camada de dados (Microsoft, 2018).

⁴ IntelliJ IDEA é um JAVA IDE da empresa JetBrains. Disponível em: <<https://www.jetbrains.com/idea/>>.

- **Netbeans:** Permite que de forma rápida e fácil seja desenvolvido aplicações em Java *deskop, mobile, web*, assim como aplicações HTML5, JavaScript e CSS. A IDE também fornece boas ferramentas para PHP, C e C++. É uma plataforma de código aberto com uma vasta comunidade de usuários e desenvolvedores (Netbeans, 2018). É utilizado para o desenvolvimento do sistema *desktop*.
- **NFC (*Near Field Communication*):** Tecnologia de curto alcance, distância de cerca de 4 a 20 centímetros (TORRES, 2008), alta frequência e baixa largura de banda, permitindo a comunicação sem fio através de dois dispositivos habilitados com NFC (COSKUN, 2011).
- **Adesivos NFC:** Equipamentos de acesso utilizados, comumente conhecidos como *INLAY*, possuem apenas *DIE* (pastilha de silício onde se encontra o circuito do *NFC*). Ao adicionar um encapsulamento plástico ao *INLAY*, o resultado é uma *tag* com maior durabilidade e resistência, porém esta possui um custo maior do que as que não estão encapsuladas (CUNHA, 2016).
- **Módulo leitor *RFID NFC PN532*:** O PN532 é um módulo de alta integração para comunicação sem contato que atua na frequência de 13.56 MHz. O módulo de transmissão utiliza um conceito de modulação e demodulação completamente integrados para diferentes tipos de comunicação. O módulo suporta tanto leitura quanto escrita e funciona em uma distância de até 7 centímetros. É utilizado para a leitura dos equipamentos de acesso dos pacientes.
- **Arduino:** O Arduino é um placa eletrônica de código aberto criado pelo time Massimo Banzi, David Cuartilles, Tom Igoe, Gianluca Martino, e David Mellis com o objetivo de desenvolver hardwares micro controladores de fácil uso que estaria disponível a todos (BARRETT, 2013).
- **REST:** *REST (Representational State Transfer)* define um conjunto de princípios arquiteturais os quais podem ser utilizados para projetar *web services* focados em recursos do sistema, incluindo como os estados dos recursos são endereçados e transferidos através do protocolo HTTP⁵ por uma ampla variedade de clientes escritos em diferentes linguagens (RODRIGUEZ,

⁵ O HTTP (*Hypertext Transfer Protocol*) é um protocolo no nível de aplicação para informações distribuídas, colaborativas e de hipermídia

2008). Também pode ser definido como um estilo híbrido derivado de vários estilos de arquitetura baseados em rede e combinado com restrições adicionais que definem uma interface de conexão uniforme (FIELDING, 2000). Este conjunto de princípios arquiteturais foi escolhido devido à grande popularidade que vem ganhando em meio aos desenvolvedores, juntamente a diversas implementações nas mais variadas linguagens de programação e bibliotecas (RODRIGUEZ, 2008).

- **Android:** É um Sistema Operacional para dispositivos móveis que foi criado com base em Linux de código aberto (ANDROID DEVELOPERS, 2018). Atualmente, o Android pode ser encontrado em diversos dispositivos, como smartphones, relógios, TVs e até mesmo carros. De acordo com a pesquisa realizada pelo jornal EL PÁIS em Abril de 2017, o Android vem cada vez mais crescendo seus números de usuários, fazendo com que seja ultrapassado o número de usuários de Windows (MENDIOLA, 2017).
- **Java:** É uma tecnologia utilizada para desenvolvimento de aplicações, que podem ser *Web*, *Desktop* ou até mesmo *mobile*, sendo orientada a objetos, compilada e interpretada. Atualmente o Java é uma das tecnologias mais utilizadas do mundo, que de acordo com a Oracle, cerca de 97% dos *Desktops* executam Java e 89% dos *desktops* nos Estados Unidos também o executam. Esses números se dão ao fato de que essa tecnologia foi projetada para permitir o desenvolvimento de aplicativos portáteis de alto desempenho e para que se possa abranger todas as plataformas possíveis (ORACLE, 2018).
- **C:** É uma linguagem de médio nível (permitindo manipulação de bits, bytes e endereços), de grande portabilidade , estruturada e compilada (ou seja, o código em C é traduzido em um *código-objeto* que pode ser entendido pelo computador) e muito utilizado na programação de sistemas operacionais, compiladores, gerenciadores de banco de dados, dentre diversas outras aplicações (SCHILD, 1997). A linguagem foi utilizada para o desenvolvimento do sistema em Arduino.
- **Armazenamento em nuvem:** Armazenamento em nuvem é a disponibilização de capacidade de armazenamento em hardwares remotos (data-centers⁶), fazendo assim que não exista mais a necessidade do gerenciamento de

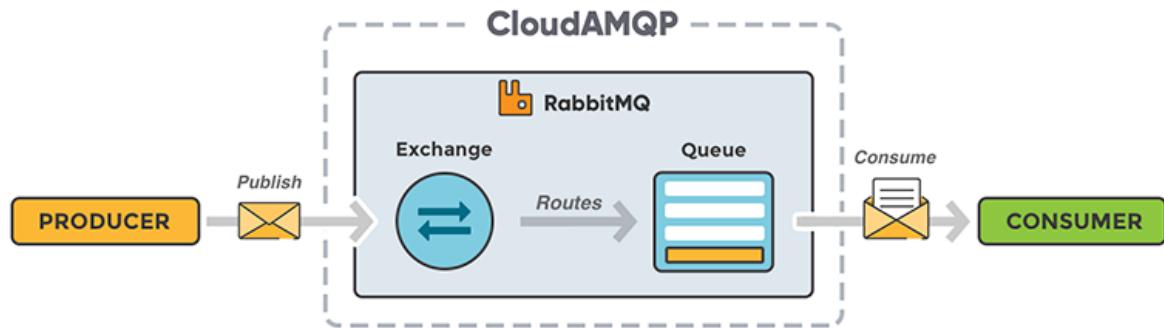
⁶ Centrais de armazenamento de dados

hardware para armazenamento de dados(DRAGO et al., 2012).Existem muitas empresas que provêm armazenamento em nuvem como: Microsoft, Google, Amazon, dentre outros (DRAGO et al., 2012). Para este trabalho de graduação estaremos utilizando a Microsoft Azure para a utilização do serviço de armazenamento em nuvem.

- **Azure:** É uma plataforma em na nuvem para hospedar aplicativos e simplificar o processo de desenvolvimento. A Microsoft Azure integra diversos serviços de nuvem muito uteis para desenvolver, testar, implantar e gerenciar aplicativos, tudo aproveitando a computação em nuvem (Microsoft, 2017).
- **SQL Server:** É uma das partes centrais da plataforma de dados da Microsoft, sendo líder em sistemas de gerenciamento de banco de dados operacionais. É um sistema de gerenciamento de banco de dados relacional altamente escalável (Quackit, 2018).
- **Apache Tomcat:** É um sistema de código aberto que implementa diversas tecnologias, tais como Java Servlet, Java Server Pages, dentre outras (Apache Tomcat, 2018). Ela é utilizada para o desenvolvimento dos *webservices*.
- **RabbitMQ:** É um intermediário de mensagens construído em Erlang, que implementa o protocolo AMQP, responsável por aceitar e encaminhar mensagens. Basicamente, um produtor envia uma mensagem, que através de uma área de troca (*Exchange*) é encaminhada a uma fila, onde esta mensagem deverá ser consumida por um consumidor (RabbitMQ, 2018). É utilizado para gerenciamento das filas de atendimento dos(as) enfermeiros(as).
- **CloudAMQP:** É um serviço que gerencia os servidores que executam RabbitMQ na nuvem (CloudAMQP, 2018).
- **Maven:** É uma ferramenta de gestão de projetos e se baseia no conceito de projeto de objeto de modelo (*POM*). O maven pode gerenciar a criação, relatório e documentação de um projeto a partir de um centro de informações (Apache Maven, 2018). É a ferramenta utilizada para gestão dos projetos desenvolvidos em Java.
- **Pusher Channels:** Fornece comunicação em tempo real entre servidor, aplicativo e dispositivos, sendo usado para notificações, chats, jogos, *web*, internet das coisas, e outros sistemas que utilizam comunicação em tempo real (Pusher, 2018).

- **Astah UML:** É uma ferramenta que suporta os requerimentos da UML 2.x para construção de diagramas de classe, caso de uso, sequencia, máquina de estado, atividade, componente, dentre outros (ASTAH, 2018). É a ferramenta utilizada para a construção dos diagramas de classe e caso de uso.
- **Fritzing:** É um sistema para automação de design eletrônico, com o objetivo de fornecer ferramentas fáceis para documentar e compartilhar projetos de computação física (FRITZING, 2018). É a ferramenta utilizada para construção do circuito do sistema Arduino.
- **Arduino IDE:** Esta ferramenta de código aberto faz com que seja fácil desenvolver o código e realizar o *upload* para a placa utilizada (Arduino, 2018). É a ferramenta utilizada para o desenvolvimento do sistema do Arduino.

Figura 1 - CloudAMQP e RabbitMQ



Fonte: Cloud AMQP

3 MODELAGEM

Na fase da modelagem é feita a documentação do aplicativo, se tratam de diagramas que facilitam na compreensão do projeto de forma padronizada.

A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language*⁷ (UML) para modelar os casos de uso e o diagrama de classe.

3.1 Casos De Uso

Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

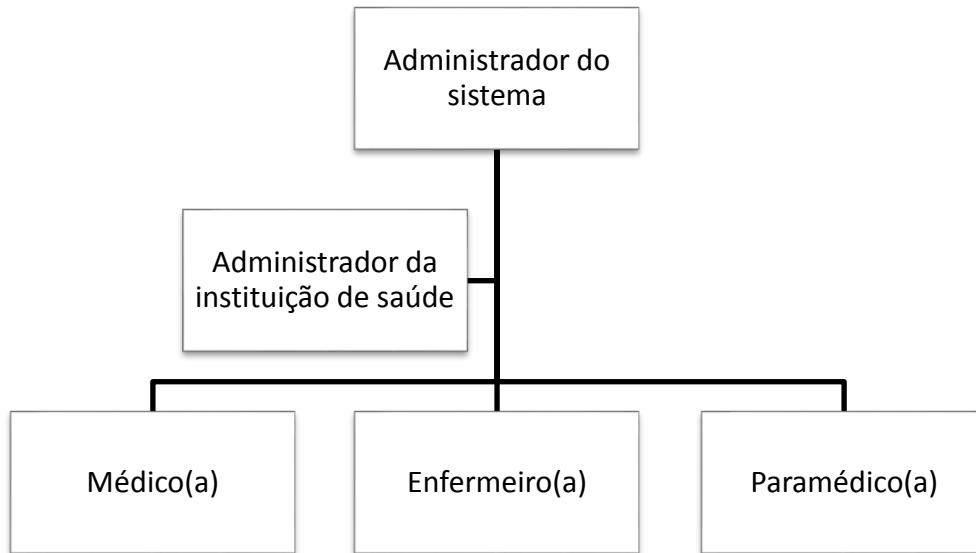
Os atores que interagem com o sistema são: o Usuário (Paciente, Médico, Enfermeira, Paramédico, Administrador da Instituição de Saúde e Administrador do sistema), API de Tradução, API de Geolocalização, API do Pusher, API do RabbitMQ Sistema de Saúde DB Lite, Sistema de Saúde DB, e a API do Sistema de saúde. A API do sistema de saúde se trata do sistema para o acesso ao banco de dados, sendo um *webservice REST*⁸.

- **Usuário** é o ator que representa os utilizadores do sistema. Um usuário pode acessar diferentes tipos de funcionalidade baseado em seu perfil. A Figura 2 apresenta um diagrama hierárquico dos usuários do sistema baseado em níveis hierárquicos.

⁷ *Unified Modeling Language* ou Linguagem Unificada de Modelagem (UML) é uma linguagem padrão para modelagem e documentar os sistemas orientados a objetos.

⁸ REST (*Representational State Transfer*) define um conjunto de princípios arquiteturais os quais podem ser utilizados para projetar *web services* focados em recursos do sistema

Figura 2 - Hierarquia de usuários



Fonte: Elaborado pelo autor

Vale ressaltar que todos os tipos de usuários, são ao mesmo tempo pacientes, como por exemplo, um administrador do sistema é ao mesmo tempo um paciente, assim como um médico é um paciente.

- **API de Tradução** representa o ator da *API* que permite a interação entre o sistema e o Google Cloud Platform⁹, permitindo o acesso a tradução dos nomes de medicamentos.
- **API de Geolocalização** representa a *API* que permite acessar serviços para obtenção de latitude e longitude das instituições de saúde, assim como acessar recursos de mapas, localizações, etc.
- **API do Pusher** representa a *API* conectada ao serviço da empresa Pusher¹⁰, que permite utilizar recursos de notificação em tempo real. Este serviço permite realizar notificações do sistema, e auxiliar o RabbitMQ¹¹ na notificação de eventos de atendimento.

⁹ Google Cloud Platform fornece acesso a diversos recursos utilizando computação em nuvem através da infraestrutura fornecida pela Google.

¹⁰ Pusher é uma empresa líder em tecnologias de tempo real.

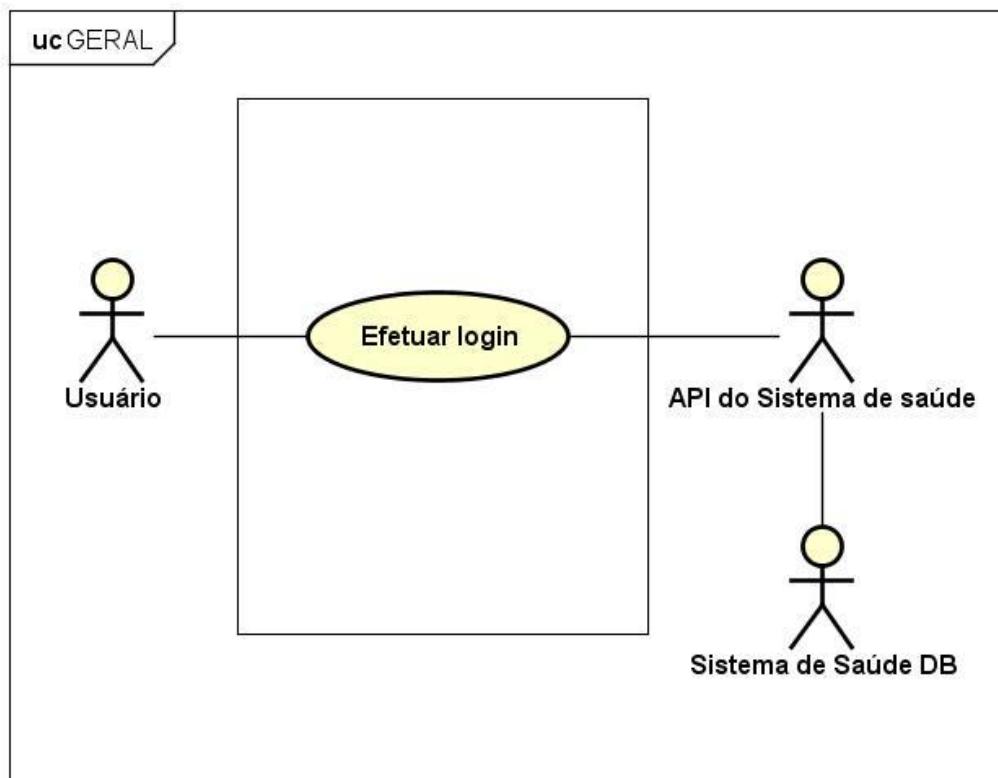
¹¹ RabbitMQ é um sistema de código livre que implementou o protocolo AMQP grandemente utilizado para gerenciamento de filas.

- **API do RabbitMQ** representa a API que permite acessar a lista de pacientes em estado de espera em uma dada instituição de saúde para serem atendidos pelos(as) enfermeiros(as).
- **Sistema de saúde DB** representa o banco de dados da Azure¹², onde o sistema armazena todas as informações necessárias.
- **Sistema de Saúde DB Lite** representa o banco de dados local do celular, onde o sistema armazena informações relativas a medicamentos.
- **API do sistema de saúde** representa o webservice REST que permite o acesso ao banco de dados na Azure.

3.2 Documentação dos Casos de Uso

A Figura 3 apresenta o caso de uso para a entrada do usuário no sistema.

Figura 3 - Diagrama de caso de uso do login do usuário

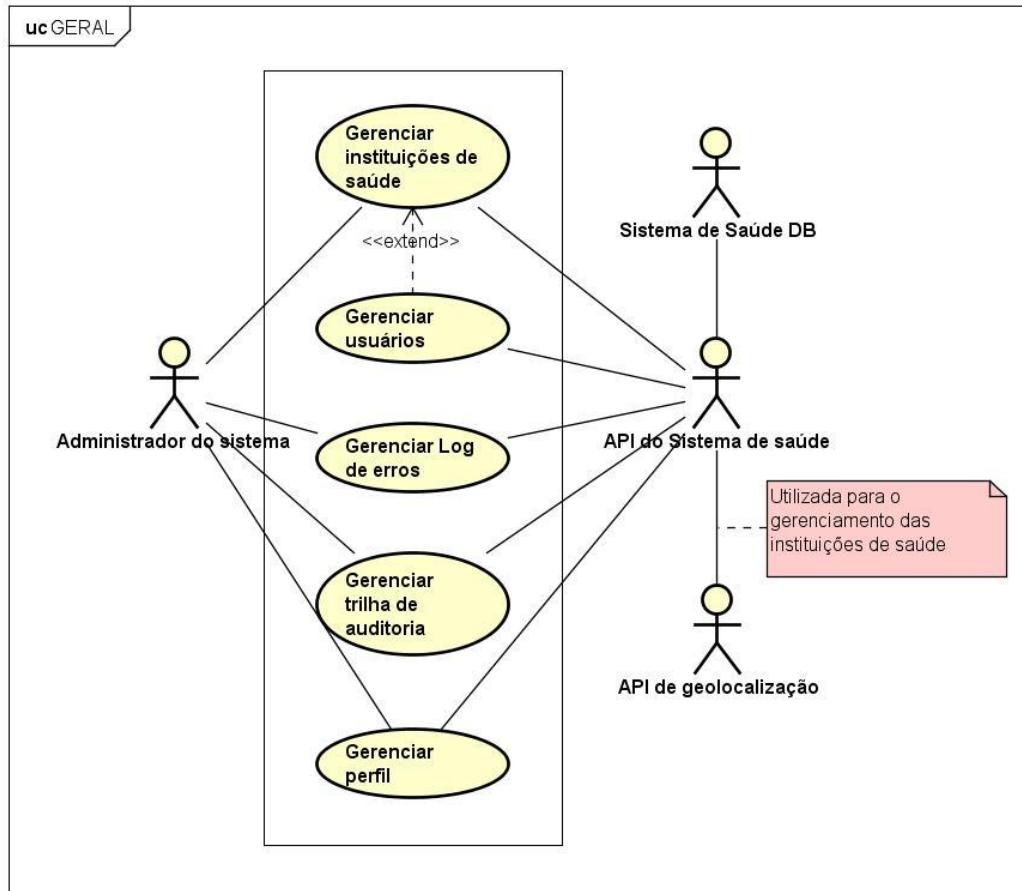


Fonte: Elaborado pelo autor

¹² A Microsoft Azure é uma plataforma baseada em computação em nuvem que permite gerenciar aplicativos, serviços, bancos de dados, etc.

A Figura 4 apresenta o caso de uso do administrador do sistema.

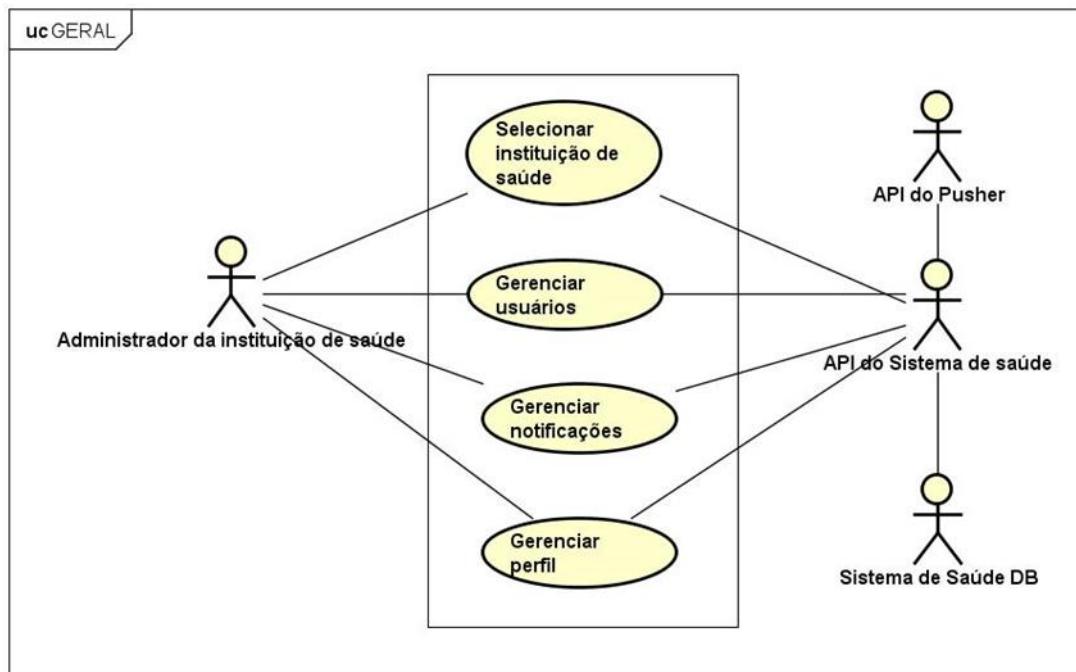
Figura 4 - Diagrama de caso de uso do administrador do sistema



Fonte: Elaborado pelo autor

A Figura 5 apresenta o caso de uso do administrador da instituição

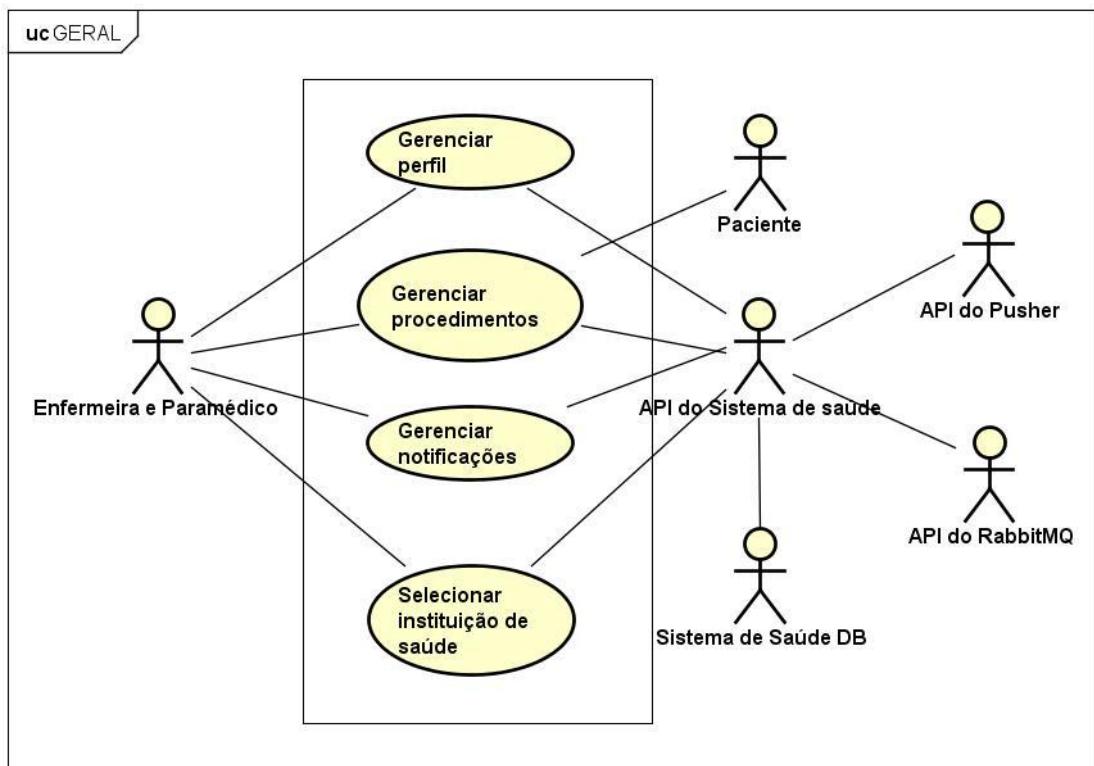
Figura 5 - Diagrama de caso de uso do administrador da instituição de saúde



Fonte: Elaborado pelo autor

A Figura 6 apresenta o caso de uso do usuário enfermeira/paramédico.

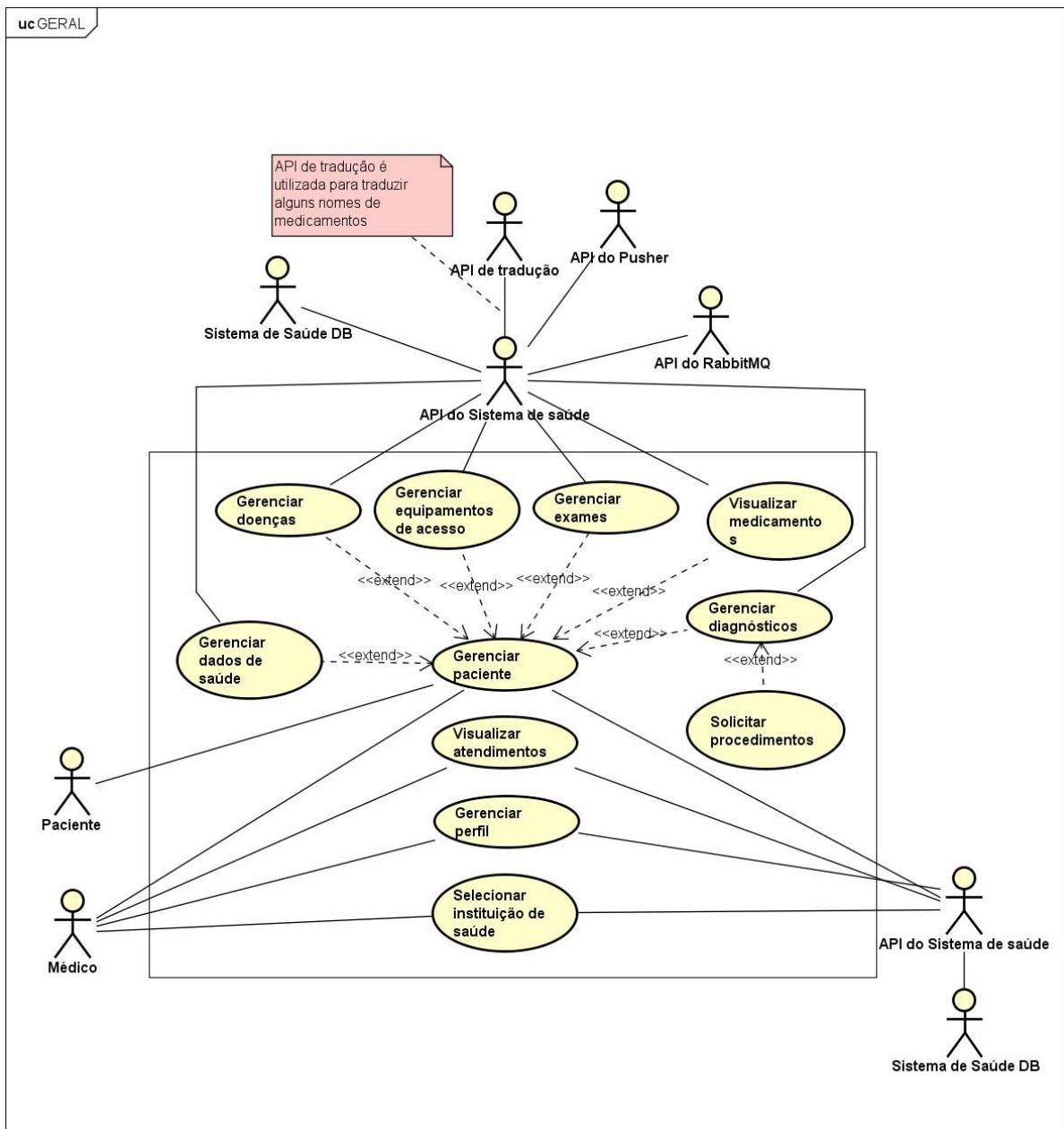
Figura 6 - Diagrama de caso de uso enfermeira/paramédico



Fonte:
Elaborado pelo autor

A Figura 7 apresenta o caso de uso do usuário médico.

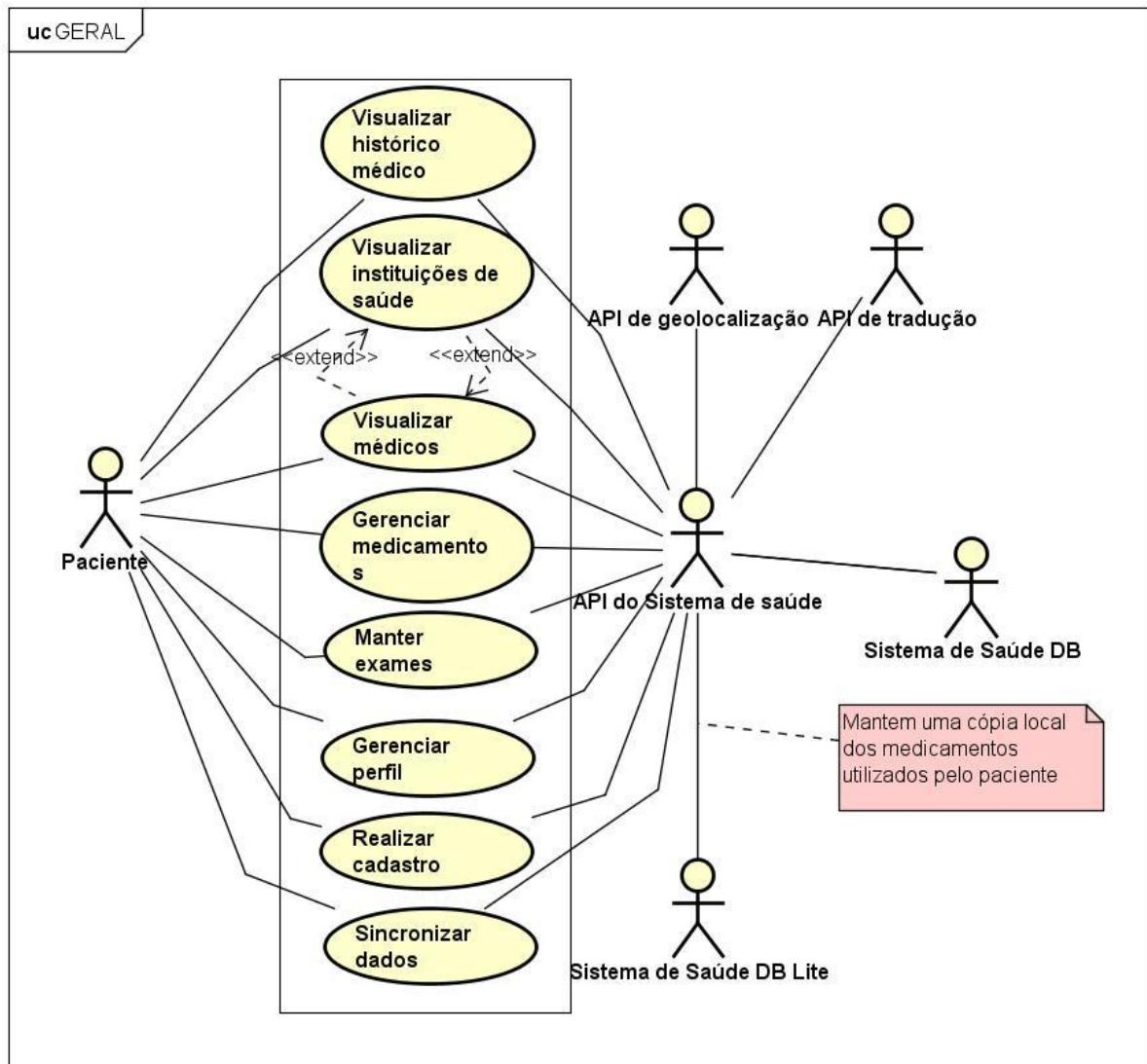
Figura 7 - Diagrama do caso de uso do(a) médico(a)



Fonte: Elaborado pelo autor

A Figura 8 apresenta o caso de uso do usuário paciente.

Figura 8 - Diagrama de caso de uso paciente



Fonte: Elaborado pelo autor

No subcapítulo 3.2.1 será apresentado à documentação dos casos de uso do projeto deste trabalho.

3.2.1 Documentação dos Casos de Uso

Cada funcionalidade dos diagramas de casos de uso será descrita na Tabela 5 à Tabela 29.

Tabela 5 – Caso de uso “Efetuar login”.

Nome do caso de uso	Efetuar login
Atores envolvidos	Usuário, API do sistema de saúde, Sistema de saúde DB.
Objetivo	Este caso de uso descreve os passos do login para os usuários administrador do sistema, administrador da instituição, médico, enfermeira e paramédico.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O usuário clica em login.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações de login e senha no Sistema de saúde DB e as autentica.
	3. Após a autenticação, o sistema redireciona para a página inicial do aplicativo.
Validações	Para o login seja efetuado, o usuário deve entrar com seu usuário e senha.

Fonte: Elaborado pelo autor

Tabela 6 - Caso de uso “Gerenciar instituições de saúde”

Nome do caso de uso	Gerenciar instituições de saúde
Atores envolvidos	Administrador do sistema/Administrador do sistema de saúde, API do sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos de gerenciamento das instituições de saúde pelo administrador do sistema.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Administrador do sistema/Administrador do sistema de saúde clica em instituições de saúde.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações das instituições cadastradas no Sistema de Saúde DB e as retorna na tela em forma de lista.
3. O Administrador do sistema/Administrador do sistema de saúde faz uma busca por nome/código de instituição.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca o nome/código da instituição de saúde no Sistema de Saúde DB e retorna na tela as

	instituições com nomes/códigos parecidos na tela em forma de lista.
5.O Administrador do sistema/Administrador do sistema de saúde clica no botão adicionar.	
7. O Administrador do sistema/Administrador do sistema de saúde preenche as informações e clica no botão confirmar.	6. O sistema abre uma janela com as informações necessárias para que seja adicionado uma nova instituição ao sistema.
9. O Administrador do sistema/Administrador do sistema de saúde seleciona uma instituição e depois clica no botão editar.	8. A API do sistema de saúde insere as informações no Sistema de Saúde DB, caso o processo tenha sido realizado com sucesso o sistema exibe uma mensagem de confirmação na tela, caso tenha ocorrido algum erro uma mensagem de erro é exibida.
11. O Administrador do sistema/Administrador do sistema de saúde edita as informações que deseja e clica no botão confirmar.	10. A API do sistema busca as informações dessa instituição no Sistema de Saúde DB, depois as retorna e uma nova janela com campos editáveis.
13.O Administrador do sistema/Administrador do sistema de saúde seleciona uma instituição e clica em apagar.	12. A API do sistema grava as informações alteradas no Sistema de Saúde DB, depois de salvas as informações o sistema uma mensagem de confirmação na tela, caso ocorra algum erro uma mensagem de erro é exibida.
15.O Administrador do sistema/Administrador do sistema de saúde clica em “sim” ou “não”.	14. Uma mensagem de confirmação será exibida.
17. O Administrador do sistema/Administrador do sistema de saúde seleciona uma instituição e clica no botão de gerenciar usuários.	16. Caso o usuário tenha clicado em “sim” a API do sistema de saúde irá apagar as informações desta instituição de saúde do Sistema de Saúde DB, exibindo uma mensagem de confirmação de exclusão após essa instituição ter sido apagada. Caso clique em “não” a mensagem de confirmação será fechada.
Validações	18. A API do sistema e saúde irá buscar a lista de usuários cadastrados nessa instituição no Sistema de Saúde DB e as retornará na tela em formato de lista.
	Para cadastro e alteração, é verificado se todos os dados da instituição foram preenchidos corretamente, respeitando as máscaras e formatos pré-definidos.

	Para exclusão, é checado se algum item da lista realmente está selecionado e também se aquele determinado item é o qual o usuário deseja excluir.
--	---

Fonte: Elaborado pelo autor

Tabela 7 - Caso de uso “Gerenciar usuários”

Nome do caso de uso	Gerenciar usuários
Atores envolvidos	Administrador do sistema/Administrador da instituição de saúde, API do sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos de gerenciamento de usuários pelo administrador do sistema e administrado das instituições de saúde
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Administrador do sistema/Administrador da instituição de saúde seleciona uma instituição e clica em gerenciar usuários.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações dos usuários cadastrados nessa instituição no Sistema de Saúde DB e as retorna em formato de lista na tela.
3. O Administrador do sistema/Administrador da instituição de saúde faz uma busca por nome/documento do funcionário.	
	4. O sistema realiza uma requisição á API do sistema de saúde que busca o nome/documento do funcionário no Sistema de Saúde DB e retorna na tela os funcionários com nomes/documento semelhantes em formato de lista.
5.O Administrador do sistema/Administrador da instituição de saúde clica no botão adicionar.	
	6. O sistema abre uma janela com as informações necessárias para que seja adicionado um novo funcionário a instituição de saúde.
7. O Administrador do sistema/Administrador da instituição de saúde preenche as informações e clica no botão confirmar.	
	8. A API do sistema de saúde insere as informações no Sistema de Saúde DB, caso o processo tenha sido realizado com sucesso o sistema exibe uma mensagem de confirmação na tela, caso tenha ocorrido algum erro uma mensagem de erro é exibida.
9. O Administrador do sistema/Administrador da instituição de saúde clica no botão enviar convite.	
	10. O sistema abre uma nova janela com o campo de e-mail para ser preenchido.

11. O Administrador do sistema/Administrador da instituição de saúde preenche o e-mail e clica em botão enviar.	
	12. A API do sistema de saúde irá enviar um convite a este funcionário já cadastrado via e-mail para participar desta instituição.
13.O Administrador do sistema/Administrador da instituição de saúde seleciona um funcionário e clica em apagar.	
	14. Uma mensagem de confirmação será exibida.
15.O Administrador do sistema/Administrador da instituição de saúde clica em "sim" ou "não".	
	15. Caso o usuário tenha clicado em "sim" a API do sistema de saúde irá apagar as informações deste funcionário do Sistema de Saúde DB, exibindo uma mensagem de confirmação de exclusão após essa instituição ter sido apagada. Caso clique em "não" a mensagem de confirmação será fechada.
16. O Administrador do sistema/Administrador da instituição de saúde seleciona um funcionário e clica no botão de visualização.	
	17. A API do sistema de saúde busca as informações desse funcionário no Sistema de Saúde DB e as retorna em uma nova janela sem a possibilidade de edição.
18. O Administrador do sistema/Administrador da instituição de saúde clica no botão adicionar usuário administrador da instituição parra essa instituição.	
	18. O sistema abre uma nova janela com as informações de e-mail e senha a serem preenchidas.
19. O Administrador do sistema/Administrador do sistema de saúde preenche o e-mail e senha e clica no botão confirmar.	
	10. A API do sistema de saúde grava as informações no Sistema de Saúde DB, após o registro das informações é exibida na tela uma mensagem de confirmação é exibida na tela. Caso ocorra algum erro uma mensagem de erro é exibida.
Validações	Para adicionar um usuário, é verificado se os dados foram preenchidos e estão no formato correto pré-definido. Também é checado se o mesmo já existe através do e-mail, pais e documento de identidade. Caso já exista e não faça parte desta instituição, um convite ao mesmo pode ser enviado. Outra checagem é a verificação de se um usuário do tipo paciente está sendo registrado como um funcionário de saúde, pois, neste caso, será atualizado seu tipo de usuário. Em todos os casos os e-mails e senhas são rigidamente validados, juntamente aos dados de cada

	usuário específico, como dados do(a) médico(a) (CRM/Número de prática), e do(a) enfermeiro(a) (COREN/SANC). Para exclusão de um usuário (desvincular o mesmo da instituição) é apresentado uma mensagem de confirmação.
--	---

Fonte: Elaborado pelo autor

Tabela 8 - Caso de uso “Gerenciar log de erros”

Nome do caso de uso	Gerenciar log de erros
Atores envolvidos	Administrador do sistema, API do sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos de gerenciamento do log de erros
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Administrador do sistema clica em log de erros.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações de erros no Sistema de Saúde DB e as retorna na tela em formato de lista.
3. O Administrador do sistema faz uma busca por data.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca os erros que ocorreram naquele dia no Sistema de Saúde DB e as retorna na tela em formato de lista.
Validações	É realizado o tratamento da data e hora no sistema, para que a mesma seja apresentada no formato da zona de tempo sul americana.

Fonte: Elaborado pelo autor

Tabela 9 – Caso de uso “Gerenciar trilha de auditoria”

Nome do caso de uso	Gerenciar trilha de auditoria
Atores envolvidos	Administrador do sistema, API do sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos de gerenciamento da trilha de auditoria
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema

1. O Administrador do sistema clica em log de atividades.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações das atividades realizadas no sistema no Sistema de Saúde DB e as retorna na tela em formato de lista.
3. O Administrador do sistema faz uma busca por data.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações das atividades que ocorreram naquele dia no Sistema de Saúde DB e as retorna na tela em formato de lista.
Validações	É realizado o tratamento da data e hora no sistema, para que a mesma seja apresentada no formato da zona de tempo sul americana.

Fonte: Elaborado pelo autor

Tabela 10 - Caso de uso " Gerenciar perfil"

Nome do caso de uso	Gerenciar perfil
Atores envolvidos	Usuário, API do sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos de gerenciamento de perfil do usuário
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O usuário clica em minha conta.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações do usuário no Sistema de Saúde DB e as retorna na tela uma nova janela para edição.
3. O usuário realiza as alterações necessárias e clica no botão confirmar.	
	4. O sistema realiza uma requisição à API do sistema de saúde que grava as informações alteradas no Sistema de Saúde DB e retorna uma mensagem de confirmação, caso ocorra algum erro uma mensagem de erro é exibida na tela.
Validações	É verificado se o usuário já existe com base no e-mail, documento de identidade e país do mesmo. Também é verificado se todos os dados preenchidos pelo usuário condizem com os padrões pré-definidos.

Fonte: Elaborado pelo autor

Tabela 11 - Caso de uso "Selecionar instituição de saúde"

Nome do caso de uso	Selecionar instituição de saúde
Atores envolvidos	Usuário, API do Pusher, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para selecionar uma instituição de saúde.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O usuário entra no sistema.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as instituições nas quais este usuário está registrado no Sistema de Saúde DB e as retorna na tela em formato de lista.
3. O usuário seleciona a instituição na qual deseja utilizar.	
	4. O sistema realiza uma requisição à API do sistema de saúde que registra a informação da instituição no Sistema de Saúde DB e retorna na página Home do sistema.
Validações	É verificado se o usuário possui alguma instituição de saúde vinculado ao mesmo. Caso tenha apenas uma instituição, o processo de escolha é automático.

Fonte: Elaborado pelo autor

Tabela 12 - Caso de uso "Gerenciar notificações"

Nome do caso de uso	Gerenciar notificações
Atores envolvidos	Usuário, API do Pusher, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para o gerenciamento das notificações.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O usuário clica no botão de notificação.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as notificações do usuário no Sistema de Saúde DB e as retorna na tela em formato de lista, a API do Pusher permite que notificações sejam alertadas em tempo real.

3. O usuário clica em aceitar convite da instituição de saúde	
	4. A API do sistema de saúde vincula o usuário a respectiva instituição de saúde, enviando os dados de vínculo ao Sistema de Saúde DB.
5. O usuário clica em negar convite da instituição de saúde.	
Validações	6. A API do sistema de saúde desvincula o usuário a respectiva instituição de saúde, enviando os dados ao Sistema de Saúde DB. É realizado uma verificação em tempo real que através de um canal alerta de novas mensagens estão sendo encaminhadas a este usuário.

Fonte: Elaborado pelo autor

Tabela 13 - Caso de uso " Visualizar atendimentos"

Nome do caso de uso	Visualizar atendimentos
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para visualizar os atendimentos.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico em sua respectiva instituição de saúde no Sistema de Saúde DB e as retorna na tela em formato de lista dos pacientes atendidos em um determinado dia.
3. O Médico clica em alterar data de atendimento.	
	4. O sistema exibe os atendimentos em formato de lista na data informada.
Validações	É verificado o médico e a instituição de saúde em que os atendimentos ocorreram, juntamente a data do atendimento.

Fonte: Elaborado pelo autor

Tabela 14 - Caso de uso " Gerenciar paciente"

Nome do caso de uso	Gerenciar paciente
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB, Paciente

Objetivo	Este caso de uso descreve os passos para o gerenciamento de paciente pelo médico.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e as retorna na tela.
3. O Médico clica no botão de autenticar paciente por código especial.	
	4. O sistema abre uma nova janela com o campo de código especial para ser preenchido.
5. O Médico preenche o código e clica em autenticar.	
	6. O sistema realiza uma requisição à API do sistema de saúde que busca as informações do paciente pelo código especial no Sistema de Saúde DB e as retorna na tela.
7. O Médico clica em pacientes.	
	8. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e os retorna na tela.
9. O Médico clica em criar paciente.	
	10. O sistema abre uma nova janela com os campos de criação de paciente para serem preenchidos.
11. O Médico preenche os campos e clica no botão confirmar.	
	12. O sistema solicitará ao usuário que aproxime o equipamento de acesso para a leitura de NFC.
13. O Médico aproxima o equipamento do leitor.	
	14. O sistema realiza uma requisição à API do sistema de saúde que irá vincular o equipamento ao usuário criado e registrará no Sistema de Saúde DB as informações do paciente, caso a operação seja realizada com sucesso será enviado um e-mail ao paciente com sua senha temporária e código secreto de acesso e será exibida uma mensagem de confirmação na tela, caso o contrário será exibida uma mensagem de erro.
15. O Médico clica no ícone do leitor NFC e o Paciente aproxima o equipamento de acesso ao leitor NFC.	

	16. O sistema faz a leitura do equipamento de acesso e faz uma requisição à API do sistema de saúde que busca as informações do paciente pelo equipamento de acesso no Sistema de Saúde DB e as retorna na tela.
Validações	<p>É validado se o paciente já foi registrado a partir do e-mail, pais e documento de identidade. Também é verificado se o equipamento de acesso que está sendo registrado já foi cadastrado.</p> <p>Também é realizado uma checagem em relação a todos os dados do usuário, verificando se os mesmos estão corretos e se são válidos (e-mail, identificação do paciente, dentre outros).</p> <p>Também se verifica se o equipamento de acesso foi ou não encontrado na base de dados.</p>

Fonte: Elaborado pelo autor

Tabela 15 - Caso de uso " Gerenciar dados de saúde"

Nome do caso de uso	Gerenciar dados de saúde
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para a edição dos dados da ficha básica do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e as retorna na tela em formato de lista.
3. O Médico clica sobre o paciente que deseja alterar.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e os retorna na tela com os campos bloqueados.
5. O Médico clica no ícone de cadeado para desbloquear os campos e os preenche, depois clica em confirmar.	
	6. O sistema realiza uma requisição à API do sistema de saúde registra as informações alteradas no Sistema de Saúde DB e retorna uma mensagem de confirmação, caso ocorra algum erro uma mensagem de erro será exibida.
Validações	É verificado se as informações condizem com as especificações de cada campo do banco de dados. Se verifica pesos mínimos e máximos, alturas mínimas e máximas, os tipos de cores possíveis a um ser

	humano dado a escala de Von Luschan, tipos sanguíneos e outros dados de saúde.
--	--

Fonte: Elaborado pelo autor

Tabela 16 - Caso de uso "Gerenciar doenças"

Nome do caso de uso	Gerenciar doenças
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para o gerenciamento de doenças de um paciente
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e os retorna na tela em formato de lista.
3. O Médico clica em cima do paciente que deseja gerenciar doenças.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e as retorna na tela.
5. O Médico clica em doenças.	
	6. O sistema exibe as doenças cadastradas em formato de lista.
7. O Médico clica no ícone de adição de doenças.	
	8. O sistema abre uma nova janela com os campos para a especificação da doença e anotações.
9. O Médico preenche as informações e clica em confirmar.	
	10. O sistema realiza uma requisição à API do sistema de saúde que registra as informações da nova doença do paciente no Sistema de Saúde DB e retorna uma mensagem de confirmação.
11. O Médico seleciona uma doença já cadastrada e clica em editar.	
	12. O sistema abre uma nova janela com as anotações desta doença disponíveis para edição.

13. O Médico faz as alterações necessárias e clica em confirmar.	
	14. O sistema realiza uma requisição à API do sistema de saúde que registra as informações alteradas no Sistema de Saúde DB e retorna uma mensagem de confirmação.
15. O Médico seleciona uma doença e clica no ícone de exclusão.	
	16. O sistema abre uma nova janela com uma mensagem de confirmação com as opções sim e não.
17. O Médico clica em "sim" ou "não"	
	18. Caso o Médico tenha clicado em sim o sistema realiza uma requisição à API do sistema de saúde que deleta a doença do Sistema de saúde DB e exibe uma mensagem de confirmação, caso contrário a janela é fechada.
Validações	É verificado se as doenças devem ser apresentadas em português ou inglês dado a linguagem do sistema operacional. Ao alterar ou excluir a doença, é verificado se algum item da tabela de doenças está selecionado.

Fonte: Elaborado pelo autor

Tabela 17 - Caso de uso " Gerenciar diagnósticos "

Nome do caso de uso	Gerenciar diagnósticos
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para gerenciar os diagnósticos.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e os retorna na tela em formato de lista.
3. O Médico clica sobre o paciente que deseja gerenciar o diagnóstico.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e as retorna na tela.

5. O Médico clica em diagnósticos.	
	6. O sistema exibe os diagnósticos em formato de lista.
7. O Médico clica no diagnóstico que deseja visualizar.	
	8. O sistema realiza uma requisição à API do sistema de saúde que busca as informações do diagnóstico selecionado no Sistema de Saúde DB e as exibe na tela: diagnóstico, médico o qual realizou, instituição e os procedimentos de enfermagem realizados.
9. O Médico clica em adicionar diagnóstico.	
	10. O sistema abre uma nova janela com os campos diagnóstico, assinatura e o botão procedimento.
11. O Médico preenche os dados necessários e clica em confirmar.	
	12. O sistema realiza uma requisição à API do sistema de saúde que registra as informações do diagnóstico no Sistema de Saúde DB e as exibe na listagem de diagnósticos.
Validações	<p>É checado se o diagnóstico possui uma assinatura, juntamente se o campo de diagnóstico foi devidamente preenchido.</p> <p>Também é checado se determinado exame possui procedimentos de enfermagem.</p> <p>Ao adicionar um diagnóstico, é possível que o médico solicite que procedimentos de enfermagem sejam realizados.</p>

Fonte: Elaborado pelo autor

Tabela 18 - Caso de uso "Gerenciar exames"

Nome do caso de uso	Gerenciar exames
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para gerenciar exames.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	

	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e os retorna na tela em formato de lista.
3. O Médico clica em cima do paciente que deseja gerenciar o diagnóstico.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e as retorna na tela.
5. O Médico clica em exames.	
	6. O sistema exibe os exames em formato de lista.
7. O Médico clica no exame que deseja visualizar.	
	8. O sistema realiza uma requisição à API do sistema de saúde que busca as informações do exame selecionado no Sistema de Saúde DB e as exibe na tela: descrição do exame, anexos, a assinatura do médico, médico e a instituição de saúde que a realizou.
9. O Médico clica em adicionar exame.	
	10. O sistema abre uma nova janela com os campos descrição, anexo e assinatura para serem preenchidos.
11. O Médico preenche os dados necessários e clica em confirmar.	
	12. O sistema realiza uma requisição à API do sistema de saúde que que registra as informações do exame no Sistema de saúde DB e as exibe na listagem de exames.
Validações	É checado o diagnóstico, assinatura e a quantidade de anexos, esta ultima a qual possui um limite de 5 anexos, sendo que cada um pode possuir um tamanho máximo de 20MB.

Fonte: Elaborado pelo autor

Tabela 19 - Caso de uso " Visualizar medicamentos"

Nome do caso de uso	Visualizar medicamentos
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para visualizar os medicamentos do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema

1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e os retorna na tela em formato de lista.
3. O Médico clica sobre o paciente que deseja visualizar os medicamentos.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e as retorna na tela.
5. O Médico clica em medicamentos.	
	6. O sistema exibe os medicamentos do paciente na tela em formato de lista.
Validações	É verificado a linguagem em que os medicamentos deverão ser apresentados, seguindo o padrão do sistema operacional.

Fonte: Elaborado pelo autor

Tabela 20 - Caso de uso "Solicitar procedimentos"

Nome do caso de uso	Solicitar procedimentos
Atores envolvidos	Médico, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para se solicitar um procedimento.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Médico clica em pacientes.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes deste médico no Sistema de Saúde DB e os retorna na tela em formato de lista.
3. O Médico clica em cima do paciente que deseja adicionar um procedimento.	
	4. O sistema realiza uma requisição a API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e as retorna na tela.
5. O Médico clica diagnósticos e em adicionar diagnóstico.	

	6. O sistema abre uma nova janela com os campos diagnóstico, assinatura e o botão procedimento.
7. O Médico preenche as informações de diagnóstico, assinatura e clica no botão de adicionar procedimentos.	
	8. O sistema realiza uma requisição à API do sistema de saúde que registra as informações do diagnóstico e do procedimento no Sistema de saúde DB e exibe em seguida a listagem de diagnósticos.
Validações	É verificado se o diagnóstico e assinatura são válidos.

Fonte: Elaborado pelo autor

Tabela 21 - Caso de uso " Gerenciar procedimentos"

Nome do caso de uso	Gerenciar procedimentos
Atores envolvidos	Enfermeira e Paramédico, API do Sistema de saúde, Sistema de Saúde DB, API do Pusher, API do RabbitMQ
Objetivo	Este caso de uso descreve os passos para o gerenciamento de procedimentos.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. A Enfermeira e/ou Paramédico entra no aplicativo.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca os pacientes aguardando na fila para serem realizados os procedimentos.
3. A Enfermeira e/ou Paramédico clica em um paciente específico para o atender.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as informações desse paciente no Sistema de Saúde DB e as retorna na tela.
5. A Enfermeira e/ou Paramédico clica em terminar procedimento.	
	6. O sistema realiza uma requisição à API do sistema de saúde que grava as informações do procedimento no Sistema de Saúde DB e em seguida a fila de atendimento é atualizada em tempo real pela API do RabbitMQ e do Pusher.
Validações	É verificado se os campos de procedimento estão válidos de acordo com o que foi pré-definido no banco de dados. Também são salvas as alterações no procedimento caso a enfermeira e/ou paramédico necessite sair da tela de procedimentos.

Fonte: Elaborado pelo autor

Tabela 22 - Caso de uso "Visualizar histórico médico"

Nome do caso de uso	Visualizar histórico médico
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para a visualização do histórico médico do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente realiza o login no sistema	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações do histórico médico do paciente no Sistema de Saúde DB e as exibe na tela em formato de linha do tempo.
Validações	São verificados os diagnósticos e exames realizados nos pacientes em determinadas instituições por médicos específicos.

Fonte: Elaborado pelo autor

Tabela 23 - Caso de uso " Visualizar instituições de saúde"

Nome do caso de uso	Visualizar instituições de saúde
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos para a visualização das instituições de saúde mais próximas do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente clica em instituições.	
	2. O sistema realiza uma requisição à API do sistema de saúde que juntamente com a API de geolocalização buscam no Sistema de Saúde DB as instituições mais próximas da localização do paciente e as retornam na tela em formato de lista.
3. O Paciente clica em ver médicos de uma instituição.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca os médicos cadastrados nessa instituição no Sistema de Saúde DB e os retorna na tela.

5. O Paciente clica no ícone de mapa.	
	6. O sistema realiza uma requisição à API do sistema de saúde que juntamente com a API de geolocalização buscam no Sistema de Saúde DB as instituições mais próximas do paciente e as exibem no mapa.
Validações	<p>É verificado se o GPS do dispositivo móvel do usuário está ativado, juntamente a disponibilidade de latitude e longitude.</p> <p>Também é realizado uma validação das instituições de saúde mais próximas com base em latitude e longitude das instituições e do paciente, se realizando cálculos através da formula de Haversine, o que permite o cálculo da distância em quilômetros entre o paciente e as respectivas instituições de saúde mais próximas.</p>

Fonte: Elaborado pelo autor

Tabela 24 - Caso de uso " Visualizar médicos"

Nome do caso de uso	Visualizar médicos
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB, API de geolocalização
Objetivo	Este caso de uso descreve os passos para a visualização dos médicos mais próximos do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente clica em médicos.	
	2. O sistema realiza uma requisição à API do sistema de saúde que juntamente com a API de geolocalização buscam no Sistema de Saúde DB os médicos mais próximos do paciente e as retornam na tela em formato de lista.
3. O Paciente clica em ver instituições.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca as instituições nas quais esse médico está cadastrado e as retorna na tela.
5. O Paciente faz uma busca por filtro de especializações.	
	6. O sistema realiza uma requisição à API do sistema de saúde que busca no Sistema de Saúde DB os médicos que correspondem ao filtro que o paciente solicitou e retorna na tela as informações em formato de lista.

7. O Paciente clica em médicos conhecidos.	
	8. O sistema realiza uma requisição à API do sistema de saúde que busca no Sistema de Saúde DB os médicos conhecidos do paciente e os retorna na tela em formato de lista.
Validações	<p>É verificado se o GPS do dispositivo móvel do usuário está ativado, juntamente a disponibilidade de latitude e longitude.</p> <p>Também é realizado uma validação das instituições de saúde mais próximas com base em latitude e longitude das instituições e do paciente, se realizando cálculos através da fórmula de Haversine, o que permite o cálculo da distância em quilômetros entre o paciente e as respectivas instituições de saúde mais próximas. A partir destas instituições, se obtém a lista de médicos.</p>

Fonte: Elaborado pelo autor

Tabela 25 - Caso de uso " Gerenciar medicamentos"

Nome do caso de uso	Gerenciar medicamentos
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB, Sistema de Saúde DB Lite
Objetivo	Este caso de uso descreve os passos para o gerenciamento de medicamentos do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente clica em medicamentos.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca no Sistema de Saúde DB Lite os medicamentos e os retorna na tela em formato de lista.
3. O Paciente clica adicionar medicamento.	
	4. O sistema redireciona o paciente para a tela de adição de medicamentos, nela é necessário preencher o nome, frequência, horário e notificação.
5. O Paciente preenche todos os campos e adiciona o medicamento.	
	6. O sistema realiza uma requisição à API do sistema de saúde e registra as informações no Sistema de Saúde DB Lite e no Sistema de Saúde DB e retorna todos os medicamentos na tela em formato de lista.
7. O Paciente clica em editar medicamento.	

	8. O sistema redireciona o paciente para a tela de edição de medicamentos, sendo nela é necessário preencher o nome, frequência, horário e notificação.
9. O Paciente altera as informações do medicamento e confirma.	
	10. O sistema realiza uma requisição à API do sistema de saúde e registra as informações alteradas no Sistema de Saúde DB Lite e no Sistema de Saúde DB e retorna todos os medicamentos na tela em formato de lista.
11. O Paciente clica em excluir.	
	12. O sistema exibe uma confirmação para excluir, apresentando as opções “sim” ou “não”.
13. O Paciente clica em “sim” ou “não”	
	14. Caso o Paciente clique em “sim” o sistema realiza uma requisição à API do sistema de saúde e apaga as informações do medicamento do Sistema de Saúde DB Lite e do Sistema de Saúde DB, caso o contrário retorna para a lista de medicamentos.
Validações	<p>É verificado se o medicamento selecionado pelo paciente realmente existe no banco de dados. Caso não exista, um novo medicamento é adicionado e enviado para validação na base de dados.</p> <p>Também é realizado uma verificação na frequência e horários que o dispositivo móvel deve alarmar o uso do medicamento.</p> <p>Ao desligar e ligar o dispositivo, também é realizado o processo de adicionar novamente todos os alarmes que foram definidos, já que no Android ao desligar um aparelho todos os alarmes registrados através da classe “AlarmManager” são perdidos ao desligar ou reiniciar o dispositivo.</p>

Fonte: Elaborado pelo autor

Tabela 26 - Caso de uso " Manter exames"

Nome do caso de uso	Manter exames
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para a visualização de exames.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente clica em exames.	

	2. O sistema realiza uma requisição à API do sistema de saúde que busca os exames no Sistema de Saúde DB e os exibe em formato de lista.
3. O Paciente clica em ver anexos de um exame.	
	4. O sistema realiza uma requisição à API do sistema de saúde que busca os anexos desse exame no Sistema de Saúde DB e os retorna na tela.
5. O Paciente clica em <i>download</i> de anexo.	
	6. O sistema realiza uma requisição à API do sistema de saúde que busca o anexo no Sistema de Saúde DB e realiza o <i>download</i> do arquivo.
Validações	É verificado se um exame possui ou não anexos, juntamente ao tipo do anexo que se está tentando realizar o <i>download</i> .

Fonte: Elaborado pelo autor

Tabela 27 - Caso de uso " Gerenciar perfil"

Nome do caso de uso	Gerenciar perfil
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para o gerenciamento do perfil do paciente.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente clica em editar perfil.	
	2. O sistema realiza uma requisição à API do sistema de saúde que busca as informações do paciente no Sistema de Saúde DB e as retorna na tela de modo editável.
3. O Paciente altera as informações necessárias e clica em confirmar.	
	4. O sistema realiza uma requisição à API do sistema de saúde que grava as alterações do perfil no Sistema de Saúde DB e retorna para a tela inicial do sistema.
Validações	É verificado se todos os dados preenchidos estão em acordo com as definições da base de dados. Também se verifica se o documento de identidade em um determinado país já está cadastrado, e se o e-mail já está em uso.

Fonte: Elaborado pelo autor

Tabela 28 - Caso de uso " Realizar cadastro"

Nome do caso de uso	Realizar cadastro
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para o cadastro do paciente no sistema pelo aplicativo.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente abre o aplicativo e clica em criar conta.	
	2. O sistema redireciona o paciente para a tela de cadastro com os campos que necessitam ser preenchidos.
3. O Paciente preenche as informações necessárias e clica em confirmar.	
	4. O sistema realiza uma requisição à API do sistema de saúde que grava as informações do usuário no Sistema de Saúde DB e cria o usuário Paciente.
Validações	É validado se todos os dados do paciente estão dentro dos padrões definidos no banco de dados, se verificando a consistência de e-mail, documento de identidade, senhas, se o usuário já existe, dentre outros.

Fonte: Elaborado pelo autor

Tabela 29 - Caso de uso "Sincronizar dados"

Nome do caso de uso	Sincronizar dados
Atores envolvidos	Paciente, API do Sistema de saúde, Sistema de Saúde DB
Objetivo	Este caso de uso descreve os passos para o cadastro do paciente no sistema pelo aplicativo.
Prioridade de desenvolvimento	Essencial
Ações do ator	Ações do Sistema
1. O Paciente clica em sincronizar dados.	
	2. O sistema realiza uma requisição à API do sistema de saúde que registra as alterações dos medicamentos no Sistema de Saúde DB com base no Sistema de Saúde DB Sqlite.

Validações	São verificados os dados do Sistema de Saúde DB Sqlite, e o Sistema de Saúde DB é atualizado através das requisições à API do Sistema de Saúde para se espelhar neste banco de dados local.
-------------------	---

3.3 Diagrama de Classe

O diagrama de classe é responsável por oferecer uma representação da estrutura e relações das classes como também as operações solicitadas pelos atores que servem de modelo para os objetos.

3.3.1 API do Sistema de Saúde

Para entender melhor a estrutura funcional do projeto, as próximas figuras apresentam as classes da API do Sistema de Saúde, sistema desenvolvido com o padrão arquitetural REST, sendo o *webservice* responsável pela conexão com o banco de dados para realizar as devidas operações. Na Tabela 30 é apresentado uma breve descrição dos pacotes utilizados.

Tabela 30 - Especificação dos pacotes utilizados

Nome do pacote	Descrição
com.healthsystem	Pacote base.
com.healthsystem.dao	Pacote onde ficam as classes de acesso ao banco de dados utilizando o padrão DAO ¹³ .
com.healthsystem.database	Classes utilitárias que criam uma conexão com o banco de dados da Azure utilizando o padrão Factory ¹⁴ .
com.healthsystem.filter	Classes utilizadas para realizar a lógica de autenticação das APIs criadas.
com.healthsystem.model	Classes de modelo que serão utilizadas para a lógica de negócio e troca de mensagens utilizando o formato JSON.
com.healthsystem.resource	Classes que definem o <i>webservice</i> e as operações e serviços que podem ser realizados através de requisições <i>HTTP</i> .
com.healthsystem.util	Classes utilitárias.

Fonte: Elaborado pelo autor

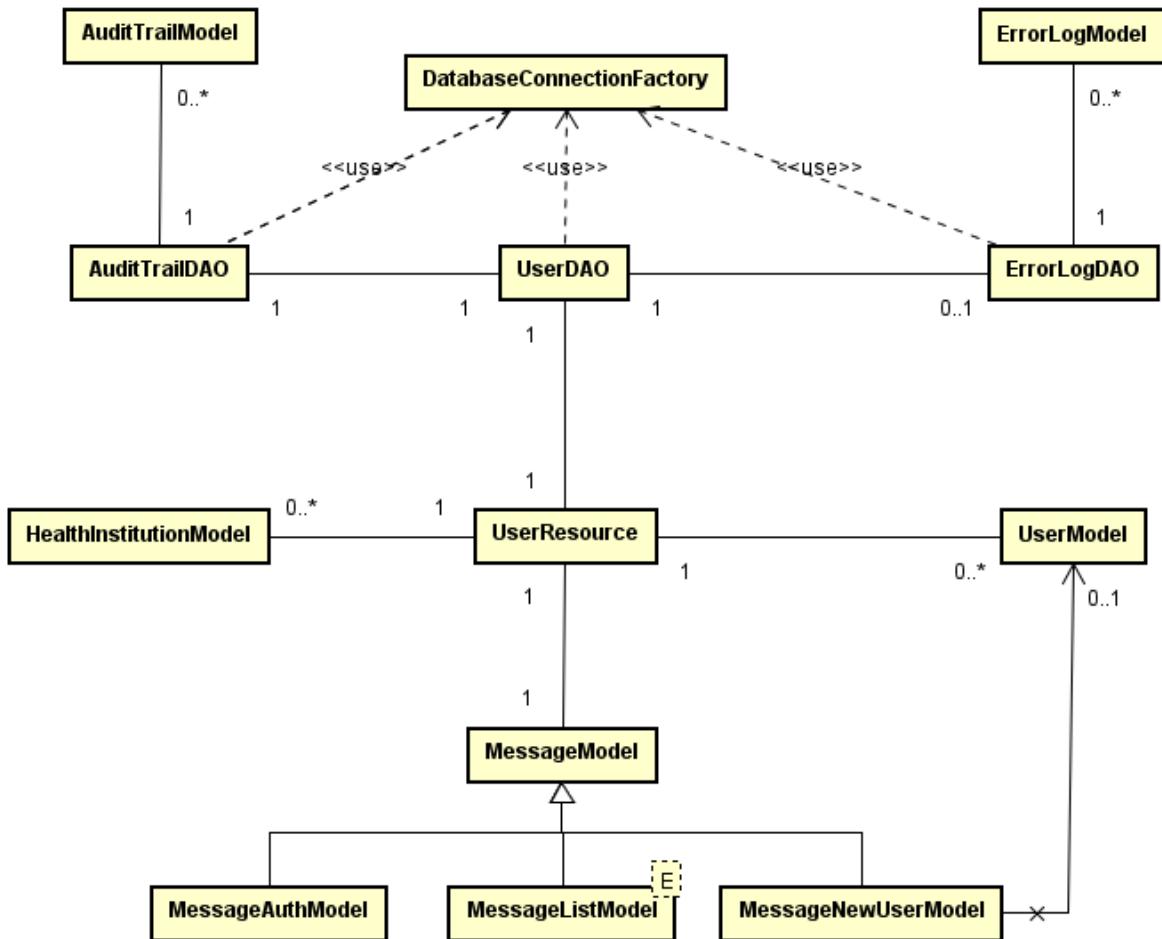
¹³ DAO (*Data Access Object*) é um padrão utilizado para persistência de dados utilizado para separar as regras de negócio das regras do banco de dados.

¹⁴ Factory é um padrão criacional que usa métodos para lidar com o problema de criar objetos sem precisar especificar uma classe do objeto que será criado.

3.3.1.1 API de Usuários

A API de usuários lida com operações que são comuns entre pacientes, enfermeiros(as), médicos (as), paramédicos (as). Na Figura 9 é apresentado o padrão estrutural das classes.

Figura 9 - Diagrama de classe da API de Usuários



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da API de usuário. A classe “UserDAO” permite que sejam gerenciados usuários, realizando operações como atualização e cadastro de dados comuns a todos usuários, checagens de usuários, dentre outras funções. A classe se comunica diretamente com o ator Sistema de Saúde DB. A classe “UserResource” especifica o endereço do webservice do usuário. A Tabela 31 especifica os recursos da API de usuário.

Tabela 31 - Recursos da API de usuário

Recursos disponibilizados pela API de usuários			
Endereço	Método	Autenticação	Descrição
/user	POST	Sim	Adiciona um usuário
/user/checkUserStatus	POST	Não	Checa se um usuário já está cadastrado
/user	PUT	Sim	Atualiza um usuário
/user/updateUserType	PUT	Sim	Atualiza o tipo de usuário (ex: paciente passa a ser médico).
/user	DELETE	Sim	Exclui um usuário
/user/{idUser}	GET	Sim	Obtém os dados do usuário
/user/secretCode	GET	Sim	Obtém o código secreto do usuário.
/user/search/{login}	GET	Sim	Obtem os dados do usuário a partir do login.
/user/auth	POST	Não	Autentica o usuário em relação as apis que possuem autenticação.
/user/healthinstitutionbind/{idUser}	GET	Sim	Obtém as instituições de saúde que o usuário está vinculado.
/user/sendPasswordMail	POST	Sim	Envia um e-mail com login e senha para o usuário recém-criado.
/user/uploadImage	PUT	Sim	Atualiza a imagem do usuário.
/user/image	GET	Não	Obtém a imagem do usuário.

Fonte: Elaborado pelo autor

A classe “DatabaseConnectionFactory” é responsável por criar uma conexão diretamente com o banco de dados hospedado na Microsoft Azure. Ela é utilizada por todas as classes que utilizam o padrão *DAO*, sendo uma importante dependência. A Figura 10 demonstra um trecho de código para conexão com o banco de dados.

Figura 10 - Trecho de código demonstrando a conexão com o banco de dados

```

1 package com.healthsystem.database;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class DatabaseConnectionFactory {
8     public static Connection getConnection() {
9         try {
10             Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
11
12             return DriverManager.getConnection("jdbc:sqlserver://healthsystem.database.windows.net:1433;" +
13                 + "database=HEALTHSYSTEM;user=user@healthsystem;" +
14                 + "password=password;encrypt=true;" +
15                 + "trustServerCertificate=false;" +
16                 + "hostNameInCertificate=*.database.windows.net;loginTimeout=30;");
17
18         } catch (SQLException | ClassNotFoundException e) {
19             throw new RuntimeException(e);
20         }
21     }
22 }

```

Fonte: Elaborado pelo autor

Os recursos retornam os dados através das requisições *HTTP* em formato JSON. O formato do retorno é definido pelas classes “MessageModel”, “MessageAuthModel”, “MessageListModel” e “MessageNewUserModel”, sendo que toda a aplicação retorna os dados com base nestas classes. Os respectivos atributos da classe representam os campos em JSON, assim como representado na Tabela 32.

Tabela 32 - Formato do retorno do JSON com base nas classes de retorno

Formato JSON retornado pelas classes			
MessageModel	MessageListModel	MessageNewUserModel	MessageAuthModel
{ “success”: true, “code”: 0, “description” : “Descrição” }	{ “success”: true, “code”: 0, “description” : “Descrição”, “list” : [] }	{ “success”: true, “code”: 0, “description” : “Descrição”, “id” : “”, “userModel” : {} }	{ “success”: true, “code”: 0, “description” : “”, “userId” : “”, “token” : “”, “userType” : “”, “username” : “”, “secretCode” : “” }

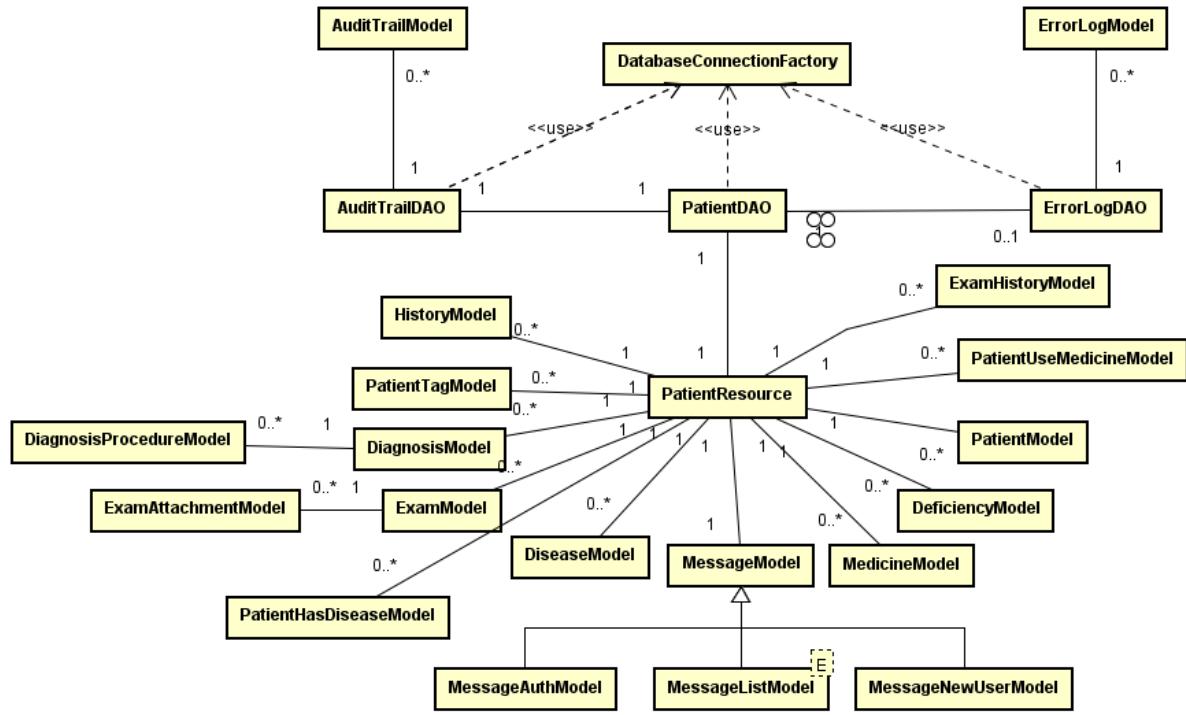
Fonte: Elaborado pelo autor

3.3.1.2 API de Pacientes

A *API* de pacientes realiza operações que possuem relação com pacientes, doenças, medicamentos, equipamentos de acesso, exames, diagnósticos, dentre

outros tópicos que podem ser relacionados ao paciente. Na Figura 11 é apresentado o padrão estrutural das classes.

Figura 11 - Diagrama de classe da API do paciente



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da *API* de paciente. A classe “PatientDAO” permite com que sejam gerenciados pacientes, medicamentos, doenças, equipamentos de acesso, dentre outras diversas funcionalidades. A classe de comunica diretamente com o ator Sistema de Saúde DB.

A classe “PatientResource” é de grande importância, já que ela especifica o endereço do webservice do paciente. A Tabela 33 apresenta uma lista dos recursos fornecidos por esta classe.

Tabela 33 - Recursos da API de paciente

Recursos disponibilizados pela API de paciente			
Endereço	Método	Autenticação	Descrição
/patient	POST	Sim	Adiciona um paciente
/patient	PUT	Sim	Altera um paciente
/patient/patientData	GET	Sim	Retorna os dados do paciente
/patient/tagRegister	POST	Sim	Adiciona um equipamento de acesso ao paciente

/patient/tagUpdate	PUT	Sim	Atualiza o equipamento de acesso
/patient/tagDelete	DELETE	Sim	Exclui o equipamento de acesso.
/patient/listPatientTag	GET	Sim	Lista os equipamentos de acesso do usuário
/patient/tagExist	GET	Sim	Verifica se o equipamento de acesso já está cadastrado.
/patient/tag	GET	Sim	Obtém o paciente a partir do equipamento de acesso.
/patient/diagnosis	POST	Sim	Adiciona um diagnóstico.
/patient/diagnosis	PUT	Sim	Altera um diagnóstico
/patient/diagnosis	DELETE	Sim	Exclui um diagnóstico
/patient/diagnosis	GET	Sim	Obtém uma lista de diagnósticos de um paciente.
/patient/diagnosis/procedures	GET	Sim	Obtém uma lista de procedimentos realizados em um diagnóstico.
/patient/disease	GET	Sim	Obtém uma lista de doenças cadastradas.
/patient/disease	POST	Sim	Adiciona uma doença ao paciente.
/patient/disease	PUT	Sim	Altera uma doença do paciente.
/patient/disease	DELETE	Sim	Exclui uma doença do paciente
/patient/patientDisease	GET	Sim	Obtém uma lista de doenças do paciente
/patient/exam	POST	Sim	Adiciona um exame
/patient/exam	PUT	Sim	Altera um exame
/patient/exam	DELETE	Sim	Exclui um exame
/patient/exam	GET	Sim	Obtém um exame
/patient/exam/attachment	POST	Sim	Adiciona um anexo ao exame
/patient/exam/attachment	DELETE	Sim	Exclui um anexo do exame
/patient/exam/attachment	GET	Sim	Obtém uma lista de anexos de um exame
/patient/listMedicines	GET	Sim	Lista os medicamentos cadastrados.
/patient/registerMedicine	POST	Sim	Adiciona um medicamento
/patient/bindMedicine	POST	Sim	Adiciona um medicamento ao paciente

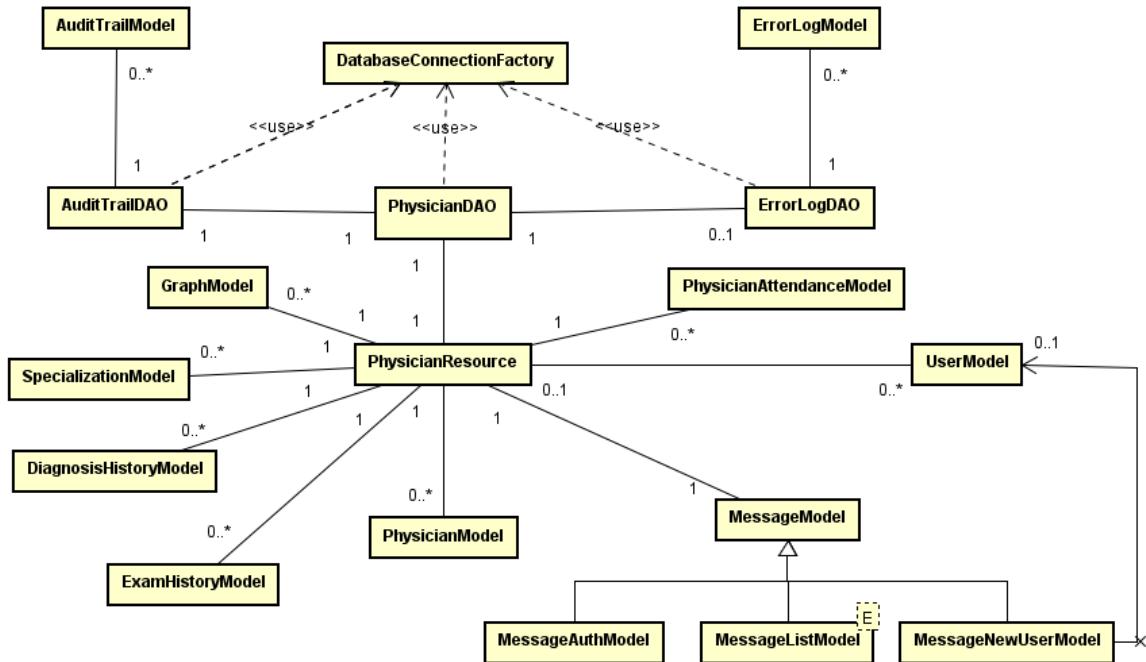
/patient/unbindMedicine/{idPatient}	DELETE	Sim	Desvincula um medicamento do paciente
/patient/listPatientMedicines	GET	Sim	Obtém uma lista de medicamentos do paciente.
/patient/diagnosis/image	GET	Sim	Obtém a assinatura do diagnóstico.
/patient/exam/attachment/document	GET	Sim	Obtém o anexo do exame.
/patient/listPatientHistory	GET	Sim	Obtém o histórico médico do paciente
/patient/listKnownPhysicians	GET	Sim	Lista os médicos conhecidos pelo paciente

Fonte: Elaborado pelo autor

3.3.1.2 API de Médico(a)

A API de médico(a) realiza operações que possuem relação com o(a) médico(a), tais como gerenciar médicos, listagem do histórico de atendimento realizado, listagem de médicos mais próximos da região, dentre outras operações. Na Figura 12 é apresentado o padrão estrutural das classes.

Figura 12 - Diagrama de classe da API de médico (a)



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da API de médico (a). A classe “PhysicianDAO” permite com que sejam gerenciados médicos(as), listado

especializações, histórico de atendimento, exames e diagnósticos realizados, dentre outras funcionalidades. A classe “PhysicianResource” especifica o endereço do webservice do(a) médico(a). A Tabela 34 apresenta uma lista dos recursos fornecidos por esta classe.

Tabela 34 - Recursos disponibilizados pela API de médico(a)

Recursos disponibilizados pela API de médico(a)			
Endereço	Método	Autenticação	Descrição
/physician	POST	Sim	Adiciona um médico(a)
/physician	PUT	Sim	Altera um médico(a)
/physician	GET	Sim	Retorna os dados do(a) médico(a)
/physician/userdata	GET	Sim	Retorna dados do usuário.
/physician/listSpecializations	GET	Sim	Lista as especializações do médico
/physician/specialization	GET	Sim	Lista todas as especializações de um determinado país
/physician/bindSpecialization	POST	Sim	Adiciona uma especialização ao médico
/physician/unbindSpecialization	DELETE	Sim	Remove uma especialização do médico.
/physician/listDiagnosisHistory	GET	Sim	Lista o histórico de diagnóstico do(a) médico(a) em uma instituição de saúde
/physician/listExamHistory	GET	Sim	Lista o histórico de diagnóstico do(a) médico(a) em uma instituição de saúde
/physician/listPhysicianInNearArea	GET	Sim	Lista os médicos próximos de uma

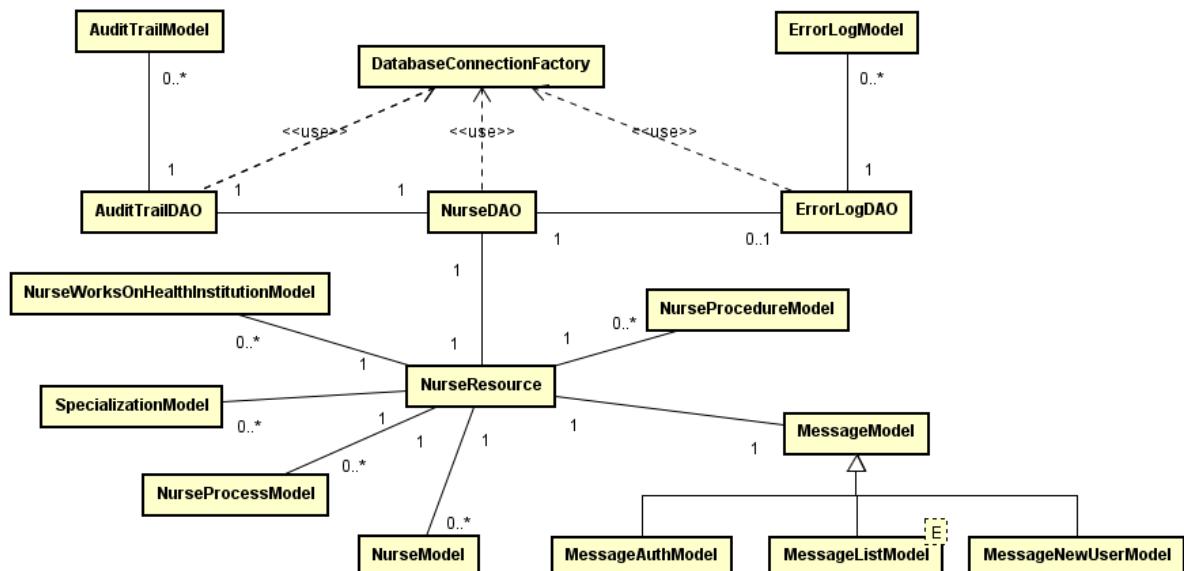
			determinada latitude e longitude
/physician/physicianAttendanceRegister	POST	Sim	Registra um atendimento médico em uma instituição de saúde.
/physician/listPhysicianAttendance	GET	Sim	Lista os atendimentos feitos pelo(a) médico(a) em uma instituição de saúde.
/physician/listAttendanceGraph	GET	Não	Lista os dados de atendimento do médico no mês corrente.

Fonte: Elaborado pelo autor

3.3.1.3 API de Enfermeiro(a)

A API de enfermeiro(a) realiza operações que possuem relação com o(a) enfermeiro(a), tais como gerenciar enfermeiros(as), adicionar procedimentos a um diagnóstico médico, adicionar pacientes a fila de atendimento, listar atendimentos realizados, dentre outras operações. Na Figura 13Figura 13 - Diagrama de Classe da API do(a) Enfermeiro(a) é apresentado o padrão estrutural das classes.

Figura 13 - Diagrama de Classe da API do(a) Enfermeiro(a)



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da *API* de enfermeiro (a). A classe “NurseDAO” permite fazer diversas operações, tais como adicionar, alterar, e obter dados de enfermeiros(as), gerenciar procedimentos e atendimentos, gerenciar especializações, e outras operações.

A classe “NurseResource” define todos os recursos e operações que podem ser realizados através do *webservice*. A Tabela 35 apresenta algumas operações que podem ser realizadas.

Tabela 35 - Recursos disponibilizados pela *API* de enfermeiro(a)

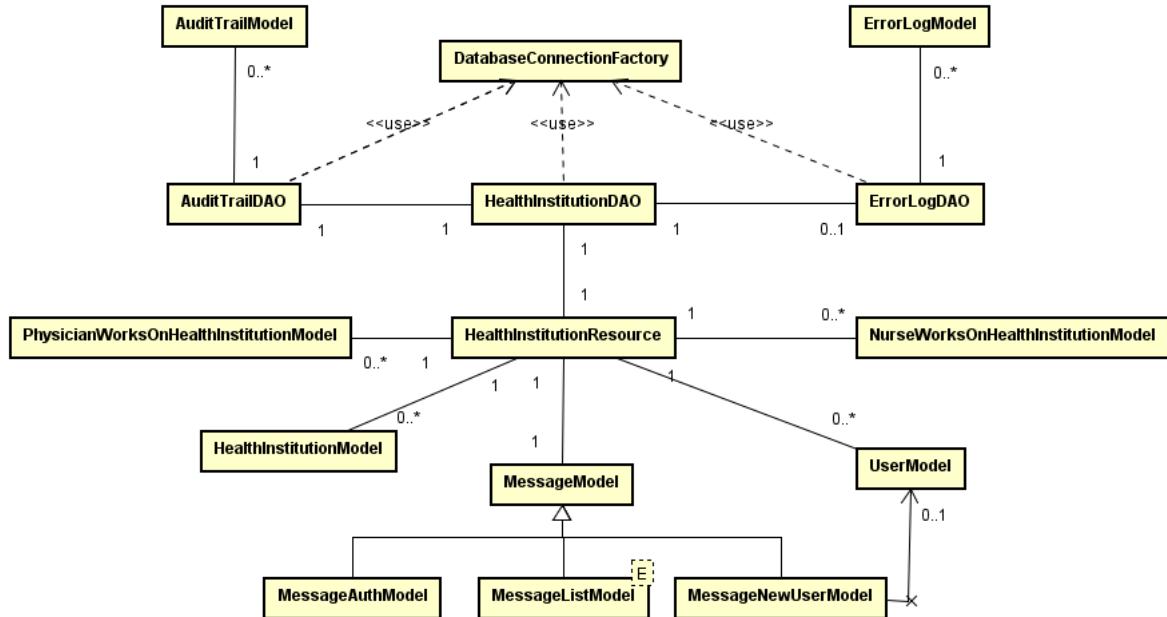
Recursos disponibilizados pela <i>API</i> de enfermeiro(a)			
Endereço	Método	Autenticação	Descrição
/nurse	POST	Sim	Adiciona um enfermeiro(a)
/nurse	PUT	Sim	Altera um enfermeiro(a)
/nurse	GET	Sim	Retorna os dados do(a) enfermeiro(a)
/nurse/listSpecializations	GET	Sim	Lista as especializações que o(a) enfermeiro(a) possui
/nurse/bindSpecialization	POST	Sim	Adiciona uma especialização a um(a) enfermeiro(a)
/nurse/unbindSpecialization	DELETE	Sim	Remove uma especialização do(a) enfermeiro(a)
/nurse/addAttendance	POST	Sim	Adiciona um atendimento
/nurse/updateAttendance	POST	Sim	Atualiza um atendimento
/nurse/listAttendance	GET	Sim	Lista os atendimentos pelo(a) enfermeiro(a) em uma instituição de saúde

Fonte: Elaborado pelo autor

3.3.1.4 *API* de Instituição de Saúde

A *API* de instituição de saúde gerencia instituições de saúde, lista instituições mais próximas dado uma posição geográfica, vincula um usuário a uma instituição de saúde, dentre outras operações. Na Figura 14 é apresentado o padrão estrutural das classes.

Figura 14 - Diagrama de Classe da API de instituição de saúde



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da *API* de instituição de saúde. A classe “*HealthInstitutionDAO*”, assim como as outras classes *DAO*, permite o acesso ao banco de dados. A classe permite gerenciar instituições, vincular usuários a instituições, adicionar pacientes a lista de espera dos(as) enfermeiros(as), listar a respectiva lista, dentre outras operações.

A classe “*HealthInstitutionResource*” define todos os recursos e operações que podem ser realizados através do *webservice*. A Tabela 36 apresenta algumas operações que podem ser realizadas.

Tabela 36 - Recursos disponibilizados pela API de instituição de saúde

Recursos disponibilizados pela API de instituição de saúde			
Endereço	Método	Autenticação	Descrição
/healthinstitution	POST	Sim	Adiciona uma instituição
/healthinstitution	PUT	Sim	Altera uma instituição
/healthinstitution	GET	Sim	Retorna uma lista de instituições de saúde dado uma posição geográfica.
/healthinstitution/list	GET	Não	Lista as instituições de saúde de modo paginado.
/healthinstitution	DELETE	Sim	Remove uma instituição

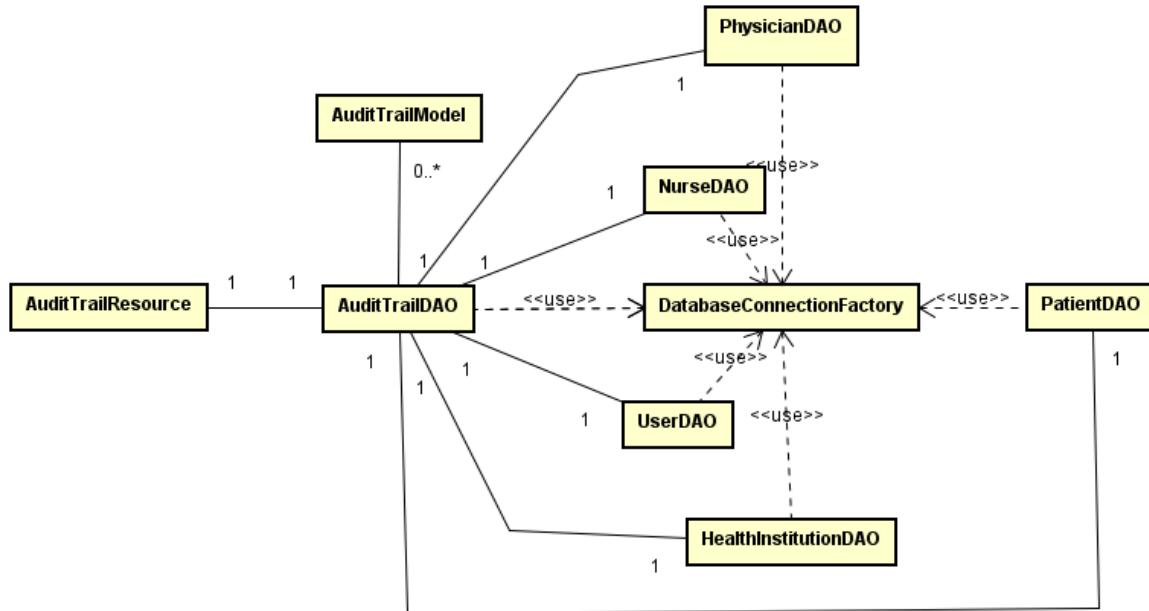
/healthinstitution/{idHealthInstitution}	GET	Sim	Obtém os dados de uma instituição de saúde
/healthinstitution/bind	POST	Sim	Vincula um usuário a uma instituição de saúde
/healthinstitution/updatebind	PUT	Sim	Atualiza o vínculo do usuário com a instituição
/healthinstitution/deletebind	DELETE	Sim	Remove o vínculo do usuário com a instituição
/healthinstitution/userlist/{idHealthInstitution}	GET	Sim	Retorna os usuários de uma dada instituição
/healthinstitution/image	GET	Sim	Retorna a foto da instituição de saúde
/healthinstitution/physician	GET	Sim	Retorna médicos(as) e suas respectivas especializações
/healthinstitution/nurse	GET	Sim	Retorna as enfermeiras de uma instituição de saúde
/healthinstitution/waitlist/{idHealthInstitution}	POST	Sim	Adiciona um diagnóstico a lista de espera do hospital
/healthinstitution/updatewaitlist/{idHealthInstitution}	POST	Sim	Atualiza a chegada de um paciente na lista do hospital

Fonte: Elaborado pelo autor

3.3.1.5 API de Trilha de Auditoria

A API de trilha de auditoria é responsável por manter um controle rígido a respeito das operações realizadas no sistema. Ela possui um registro do que, e quem realizou determinada operação, possibilitando assim grande rastreabilidade. Na Figura 15 é apresentado o padrão estrutural das classes.

Figura 15 - Diagrama de Classe da API de trilha de auditoria



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da *API* de trilha de auditoria. A classe “AuditTrailDAO” é responsável por adicionar eventos a trilha de auditoria, e listar os eventos que ocorreram dado uma determinada data.

A classe “AuditTrailResource” define o endereço do *webservice*. Os dados retornados por esta API são uma lista de eventos ocorridos no sistema, contendo informações sobre a data do evento, categoria, nome do evento, informações adicionais, id do usuário que realizou o evento, nome, login e foto do mesmo. Assim, é possível rastrear o usuário que realizou determinadas ações no sistema. Na Tabela 37 é apresentado a especificação do serviço do recurso.

Tabela 37 - Recursos disponibilizados pela API de trilha de auditoria

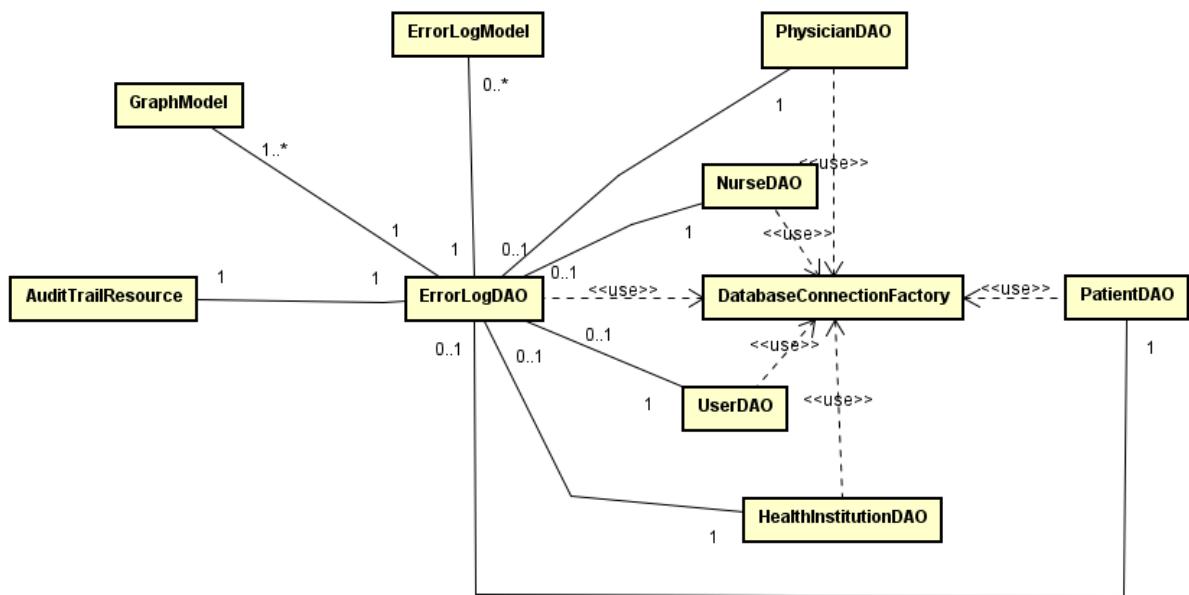
Recursos disponibilizados pela API de trilha de auditoria			
Endereço	Operação	Autenticação	Descrição
/auditTrail/listAuditTrail	GET	Sim	Lista a trilha de auditoria dada uma determinada data

Fonte: Elaborado pelo autor

3.3.1.6 API de Erro

A API de erro é responsável por catalogar todos os erros a nível de aplicação que ocorrem no sistema. Ela possui registros dos erros que ocorreram, facilitando o gerenciamento sistêmico. Na Figura 16 é apresentado o padrão estrutural das classes.

Figura 16 - Diagrama de Classe da API de Erro



Fonte: Elaborado pelo autor

Este diagrama apresenta a estrutura das classes da *API* de erro. Abaixo, será apresentado as características das classes mais importantes. A classe “ErrorLogDAO” é responsável por adicionar erros a nível de aplicação ao banco de dados, listar a quantidade de erros que ocorreram através de um gráfico, facilitando a visualização, e por fim, listar os erros em determinada data.

A classe “ErrorLogResource” irá definir as operações para a *API* de Erro. Ela possui dois serviços, sendo eles respectivamente, obter a quantidade de erros que ocorreram no mês atual para demonstrar estes dados em forma de gráfico, e, obter uma lista dos erros que ocorreram dado uma determinada data. A lista de erro contem a data do erro, uma mensagem, nome do arquivo, nome da classe, nome do método, número da linha, informações adicionais, identificação do usuário, nome, *login* e foto do mesmo. Na Tabela 38 é apresentado as operações da *API*.

Tabela 38 - Recursos disponibilizados pela API de erro

Recursos disponibilizados pela API de trilha de erro			
Endereço	Método	Autenticação	Descrição
/errorlog/list	GET	Não	Lista a quantidade de erros em forma de gráfico (considera o mês atual).
/errorlog/listErrors	GET	Sim	Lista os erros que ocorreram dado uma determinada data.

Fonte: Elaborado pelo autor

3.3.1.6 Miscelânea

Nesta categoria estão classes de grande utilidade, que são utilizadas como suporte para o desenvolvimento do sistema, tais como classes de criptografia, utilitários de imagem, gerenciadores de conexão, tradutores, dentre outros.

A classe “RabbitMQConnectionManager” é uma classe que permite gerenciar a conexão com o RabbitMQ. Possui métodos para se conectar a uma fila, realizar a publicação de um item, consumir as mensagens que são enviadas, e um método para encerrar a conexão.

A classe “Cryptography” é utilizada para gerar uma cadeia de caracteres codificada a partir de uma cadeia de caracteres comum, utilizando como função de hash criptográfico o SHA-256¹⁵.

A classe “AzureBlob” é responsável por enviar documentos digitais para o armazenamento em nuvem da Microsoft Azure. A classe possui desde métodos de criar um cliente para conexão com a Azure, até *upload*, *download* e remoção de documentos.

A classe “JWTTokenNeededFilter” realiza a autenticação das APIs que possuem este recurso. É verificado se o *token* de autenticação enviado realmente existe e condiz com um usuário ativo. Caso o contrário, será enviado uma resposta ao usuário informando a não autorização da requisição.

¹⁵ SHA-256 faz parte de uma família de funções de hash criptográfico, que são uma série de operações matemáticas que são realizadas nos dados.

3.3.2 Sistema de Saúde (Versão Desktop)

Para entender melhor a estrutura funcional do projeto do sistema de saúde na versão *desktop*, utilizada principalmente pelos(as) médicos(as), administradores e administradores da instituição de saúde, será apresentado a estrutura de classes deste sistema. A Tabela 39 apresenta uma breve descrição dos pacotes utilizados.

Tabela 39 - Especificação dos pacotes utilizados

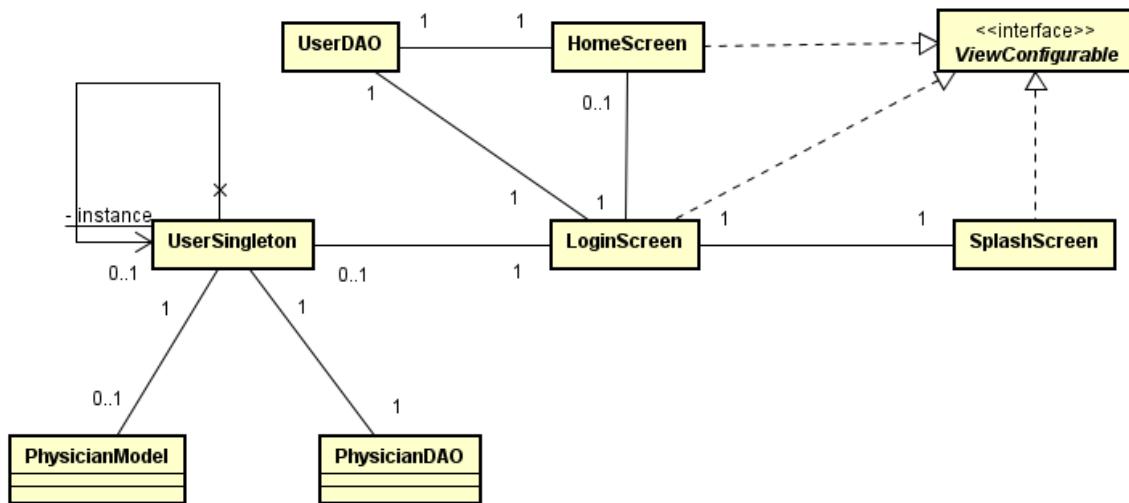
Nome do pacote	Descrição
com.healthsystem	Pacote base.
com.healthsystem.audittrail	Recursos e telas utilizados pela trilha de auditoria.
com.healthsystem.errorlog	Recursos e telas utilizados pelo log de erros.
com.healthsystem.healthinstitution	Recursos e telas utilizados pelo gerenciamento e seleção de instituições de saúde.
com.healthsystem.home	Recursos, telas e estrutura principal do sistema.
com.healthsystem.notification	Tela de notificação do sistema.
com.healthsystem.patient	Tela para gerenciamento do paciente, exames, diagnósticos, etc.
com.healthsystem.physician	Tela para gestão do(a) médico(a).
com.healthsystem.user	Tela para gestão de usuário
com.healthsystem.user.nurse	Recursos para gestão do(a) enfermeiro(a).
com.healthsystem.user.physician	Recursos para gestão do(a) médico(a).
com.healthsystem.user.specialization	Recursos para lidar com especializações.
com.healthsystem.util	Classes utilitárias.
com.healthsystem.util.azure	Classes utilitárias para utilizar a Microsoft Azure.
com.healthsystem.util.component	Classes de componentes gráficos desenvolvidos.
com.healthsystem.util.dataprovider	Classes fornecedoras de dados.

Fonte: Elaborado pelo autor

3.3.2.1 Funcionalidade de *Login*

A seguir, será apresentada a estrutura da função de *login* do sistema. A função de *login* é responsável por verificar se o usuário já está registrado no sistema, verificar o tipo de conta do mesmo (administrador, médico, e os outros perfis), e inicializar a tela principal do sistema.

Figura 17 - Diagrama de classe da funcionalidade de Login



Fonte: Elaborado pelo autor

Basicamente, uma tela inicial de carregamento é iniciada, e em seguida a tela de *login* é inicializada. A tela de *login* utiliza de classes como “UserDAO” e “UserSingleton” para, respectivamente, realizar a consulta do *login* e armazenar os dados em uma estrutura de classe *Singleton*¹⁶. Todas as classes demarcadas com a terminação “Screen”, “Panel” ou “Window” representam uma tela, painel ou janela do sistema.

A classe “UserSingleton” é responsável por armazenar os dados obtidos durante o *login* em uma estrutura que garante uma única instância da classe, e, por isso, foi utilizado o padrão de projeto *Singleton*.

A interface “ViewConfigurable” é responsável por estar lidando com os recursos gráficos do sistema em relação a decoração de janelas, centralização de telas, e outras características das janelas do sistema.

A classe “UserDAO” permite que sejam acessados todos os serviços disponibilizados pela API de Usuário, ou seja, serviços comuns a todos os tipos de perfis.

A classe “PhysicianDAO” permite que sejam acessados todos os serviços disponibilizados pela API de médico, tais como adicionar e atualizar médicos, listar especializações, e outros.

¹⁶ *Singleton* é um padrão de projeto, que garante que exista apenas uma instância de uma classe.

3.3.2.2 Funcionalidade de Navegação

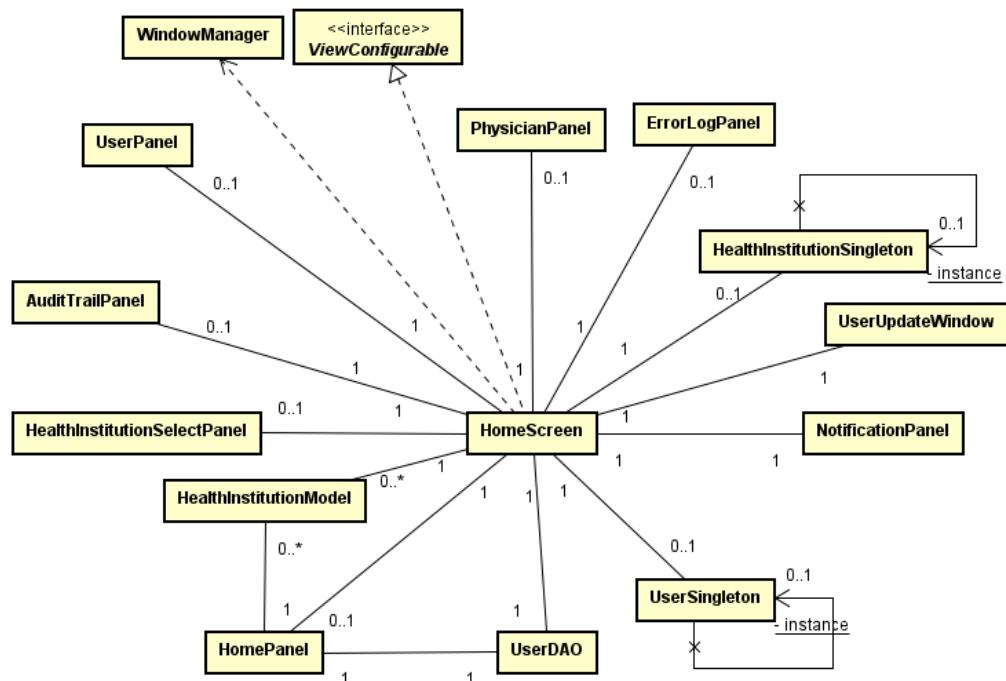
A função de navegação permite com que o usuário acesse as telas do sistema baseado em seu perfil. A Tabela 40 apresenta as telas que cada perfil de usuário possui permissão de acessar.

Tabela 40 - Tabela de permissões

Perfil	Telas
Administrador	Painel, notificação, log de erro, log de auditoria, gerenciamento de instituições de saúde e usuário e alteração de perfil.
Administrador da instituição de saúde	Painel, notificação, seleção da instituição de saúde, gerenciamento de usuário e alteração de perfil.
Médico	Painel, notificação, seleção da instituição de saúde, gerenciamento de pacientes, alteração de perfil.
Enfermeira	Painel, notificação, seleção da instituição de saúde, alteração de perfil.
Paramédico	Painel, notificação, seleção da instituição de saúde, alteração de perfil.
Paciente	Não possui acesso.

Fonte: Elaborado pelo autor

Figura 18 - Diagrama de classe da navegação do sistema



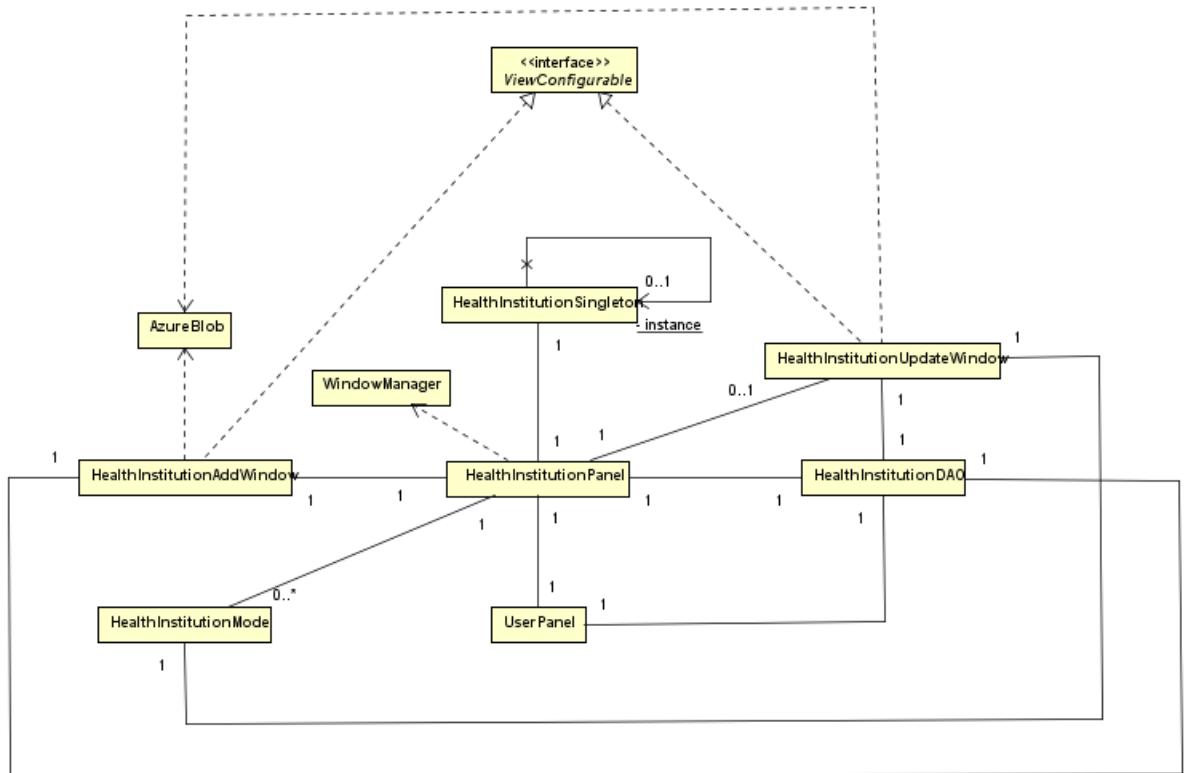
Fonte: Elaborado pelo autor

A estrutura de classes do sistema permite que a tela base do sistema, “HomeScreen”, identifique o tipo do perfil do usuário a partir da classe “UserSingleton”, e assim possa definir como será a naveabilidade do sistema. A classe utilizada para realizar a transição de telas é a “WindowManager”, sendo uma dependência necessária. Ao sistema iniciar, é carregado diretamente a classe “HomePanel” (caso o usuário faça parte de apenas uma instituição de saúde), ou a tela de seleção de instituições de saúde (caso faça parte de várias). Vale ressaltar que para o administrador do sistema, a classe carregada inicialmente é a “HomePanel”, já que o administrador do sistema não pode ser vinculado a nenhuma instituição de saúde.

3.3.2.2 Funcionalidade de Gerenciamento de instituições de saúde

A função de gestão de instituições de saúde pode ser apenas acessada pelo usuário administrador do sistema. Esta funcionalidade permite estar adicionando, alterando, excluindo, e listando instituições de saúde.

Figura 19 - Diagrama de classe do gerenciamento de instituições de saúde



Fonte: Elaborado pelo autor

A classe “HealthInstitutionPanel” possui uma listagem das instituições de saúde, possuindo realizar a adição, alteração e remoção de instituições. Também é possível, através desta tela, acessar os usuários de cada instituição de saúde. Uma classe utilizada como dependência é a “AzureBlob”, responsável por fazer o *upload* da foto da instituição de saúde.

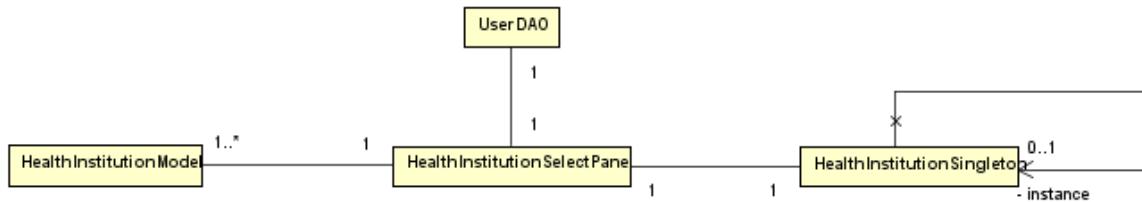
Vale ressaltar a classe “HealthInstitutionSingleton”, que funciona de maneira similar a “UserSingleton”, permitindo armazenar dados em uma estrutura que garanta uma instância única desta classe.

A classe “HealthInstitutionDAO” permite o acesso a todos os serviços definidos pela *API* de instituição de saúde, tais como adicionar, alterar, excluir, listar, dentre outras operações definidas.

3.3.2.3 Funcionalidade de Selecionar instituição de saúde

A função de selecionar instituição de saúde está presente a todos os tipos de usuários (com exceção do administrador do sistema e paciente), e permite que um usuário selecione a instituição de saúde que o mesmo faz parte, já que, por exemplo, um(a) médico(a) pode fazer parte de mais de uma instituição de saúde.

Figura 20 - Diagrama de classe da seleção de instituição de saúde



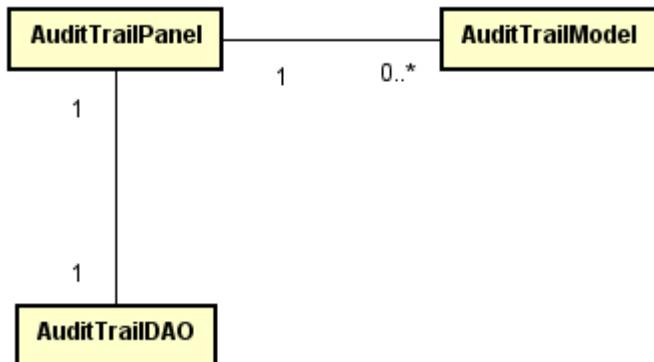
Fonte: Elaborado pelo autor

A classe “HealthInstitutionPanel” possui uma lista de instituições de saúde. Assim, a instituição que o usuário selecionar será armazenada na classe “HealthInstitutionSingleton”.

3.3.2.4 Funcionalidade de Trilha de Auditoria

A função de trilha de auditoria é apenas disponível ao administrador do sistema, e permite com que seja visualizado um log de todas as requisições realizadas na *API* do sistema, permitindo assim grande rastreabilidade das ações realizadas pelos usuários do sistema.

Figura 21 - Diagrama de classe da trilha de auditoria



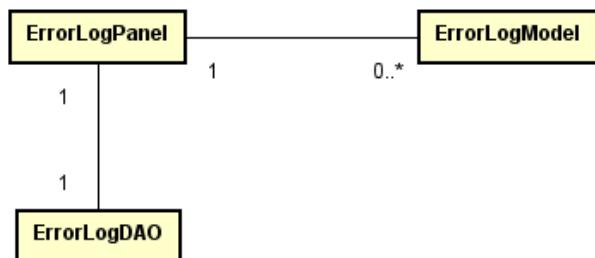
Fonte: Elaborado pelo autor

A classe “AuditTrailPanel” possui uma lista das ações realizadas no sistema. A classe “AuditTrailDAO” realiza a comunicação com todos os serviços definidos pela API de trilha de auditoria, possibilitando obter uma lista das ações realizadas dado uma determinada data.

3.3.2.4 Funcionalidade de *Log* de Erros

A função de *log* de erros está apenas disponível para o administrador do sistema, sendo possível através desta função, visualizar todos os erros a nível de aplicação que ocorreram no sistema em forma de tabela (onde os erros são apresentados com detalhes) ou gráfico (onde os erros são apresentados em um gráfico quantitativo, podendo ser visualizada na tela inicial do sistema).

Figura 22 - Diagrama de classe de log de erros



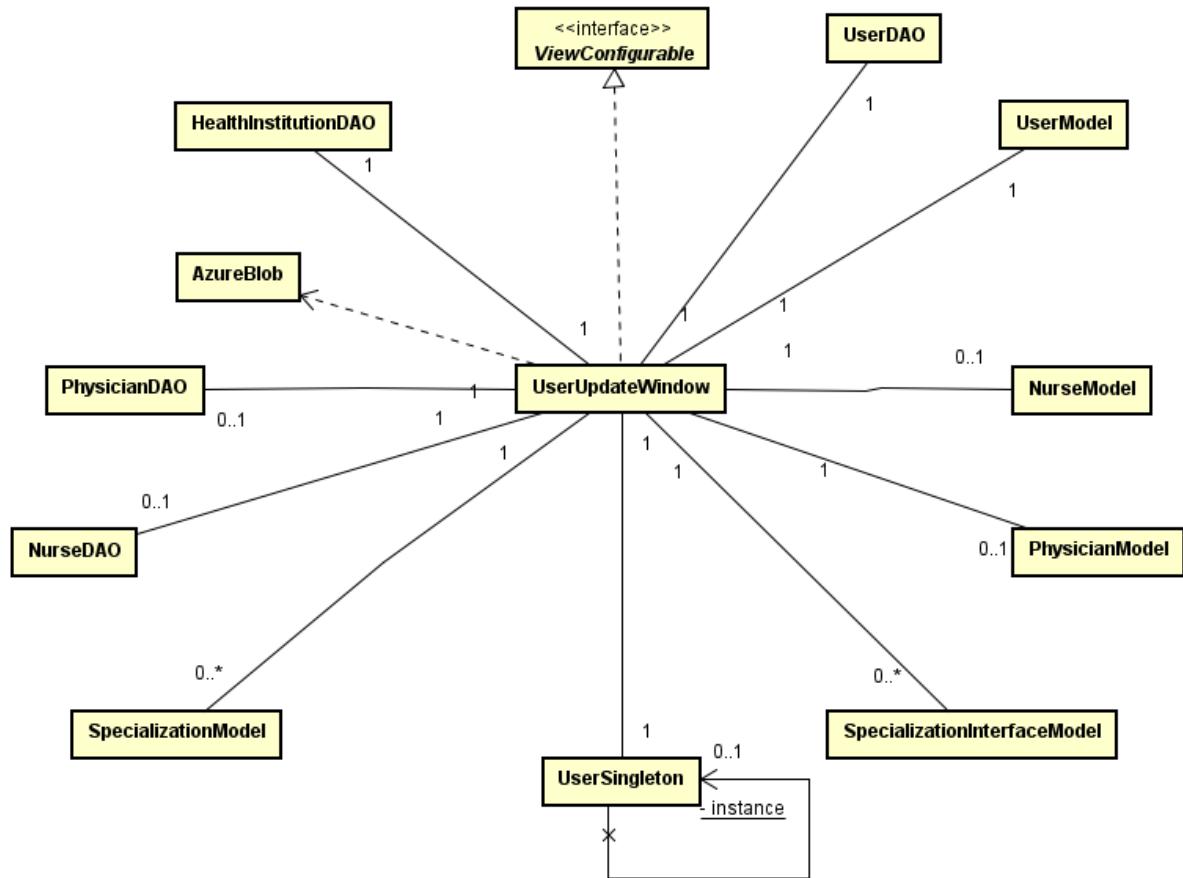
Fonte: Elaborado pelo autor

A classe “ErrorLogPanel” possui uma tabela detalhada de todos os erros que ocorreram no sistema, que são acessados através da classe “ErrorLogDAO”, que realiza a comunicação com a API de *log* de erros.

3.3.2.5 Funcionalidade de Alterar Perfil

A funcionalidade de alterar perfil está disponível a todos os usuários (com exceção do paciente), permitindo com que o usuário atualize seus dados.

Figura 23 - Diagrama de classe de alterar perfil



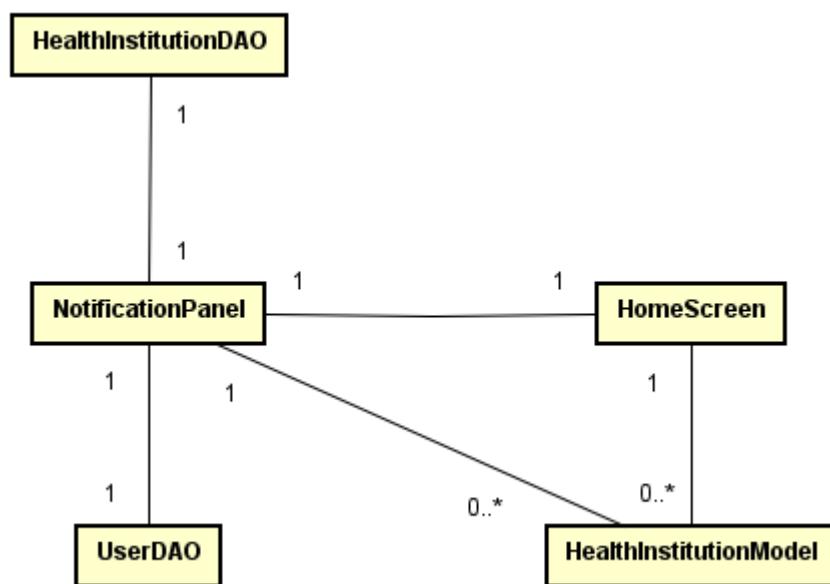
Fonte: Elaborado pelo autor

A janela principal (classe “UserUpdateWindow”) possui acesso a todas as classes *DAO* do sistema. A partir dos dados obtidos pela “UserSingleton”, se verifica o tipo de usuário, e assim, dado o tipo de usuário, se obtém a classe modelo do mesmo e se realiza os ajustes necessários na interface (por exemplo, para médicos(as), é necessário a edição do CRM, já para enfermeiros(as), é necessário o COREN). Na classe “NurseDAO”, por exemplo, existem métodos que tornam possível atualizar os dados do usuário, gerenciar especializações, entre outros métodos que se comunicam com a *API* de enfermeiro(a).

3.3.2.6 Funcionalidade de Notificação

A funcionalidade de notificação está presente em todos os perfis (exceto o do paciente) e, primariamente, é utilizado para usuários aceitarem convites realizados por instituições de saúde (como por exemplo, um médico faz parte de uma instituição de saúde A, porém, é contratado por uma instituição de saúde B, logo, é necessário que o mesmo aceite o convite feito pela instituição de saúde B, que estará em “Notificações”).

Figura 24 - Diagrama de classe de notificação



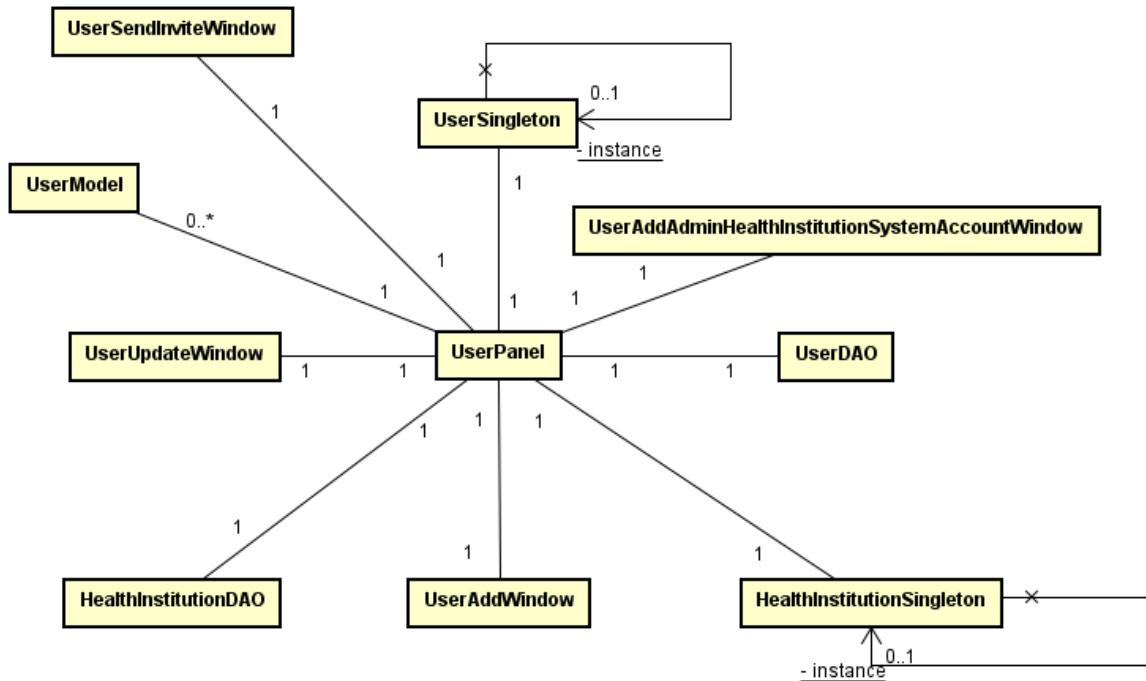
Fonte: Elaborado pelo autor

A classe “NotificationPanel” possui uma tabela de notificações que é acessada através das classes *DAO*. Vale ressaltar que as notificações funcionam em tempo real através da presença do serviço de tempo real Pusher. No Pusher, um usuário recebe eventos de notificação (no caso, *notification*), através de um canal de comunicação (a identificação do usuário), e assim é possível realizar a comunicação em tempo real.

3.3.2.7 Funcionalidade de Gerenciar usuários

A função de gerenciar usuários está disponível para administradores do sistema e da instituição de saúde. A função faz com que seja possível gerenciar usuários de uma determinada instituição de saúde, permitindo alterar, excluir, editar e visualizar dados dos usuários.

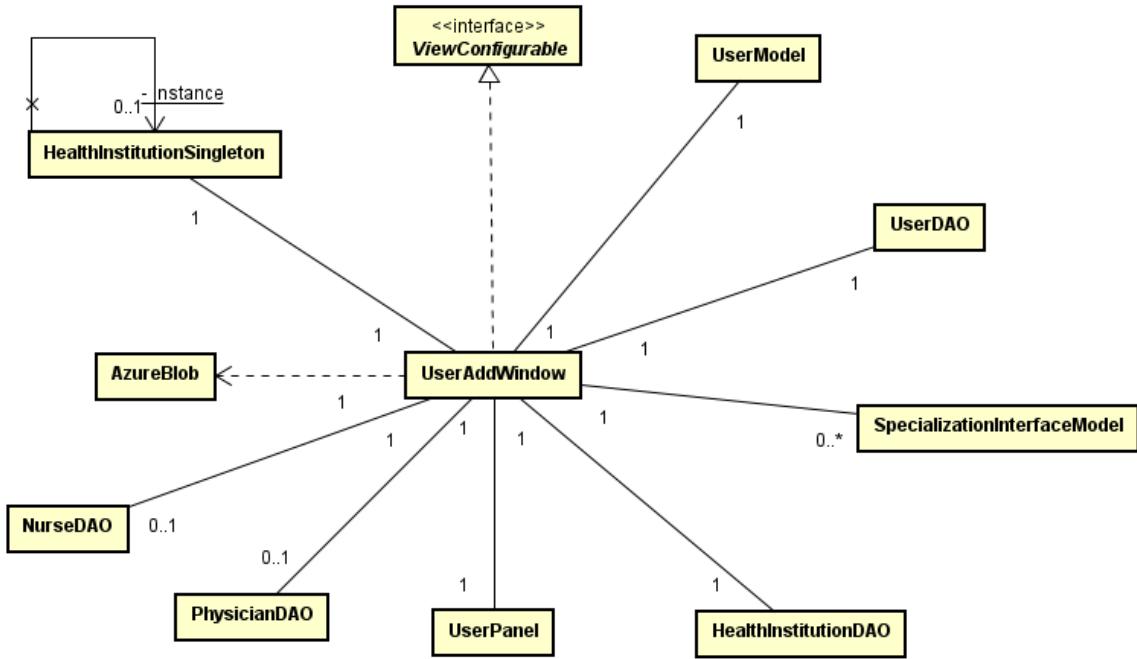
Figura 25 - Diagrama de classe de gerenciar usuários



Fonte: Elaborado pelo autor

A classe “UserPanel” apresenta uma tabela com uma lista de usuários vinculados a uma determinada instituição de saúde, acessada a partir da classe “HealthInstitutionSingleton”. Assim, a partir dessa tabela é possível adicionar (“UserAddWindow”), alterar (“UserUpdateWindow”), excluir, enviar um convite a um usuário ou adicionar um usuário sistemático administrador do hospital. Enviar um convite possui a seguinte funcionalidade: Dado um Médico que faz parte de uma instituição de saúde A, porem irá começar a trabalhar em uma instituição de saúde B, é dever do administrador da instituição de saúde B enviar um convite para que este médico faça parte da instituição desejada. Este processo vincula um usuário a uma instituição de saúde caso o usuário já exista, já que um usuário (médico, paramédico, enfermeira, administrador da instituição de saúde) pode pertencer a mais de uma instituição de saúde. Caso o usuário não exista, a conta deverá ser criada normalmente e o usuário já será automaticamente vinculado a instituição de saúde a qual o mesmo foi cadastrado.

Figura 26 – Diagrama de classe de adicionar usuário na instituição de saúde



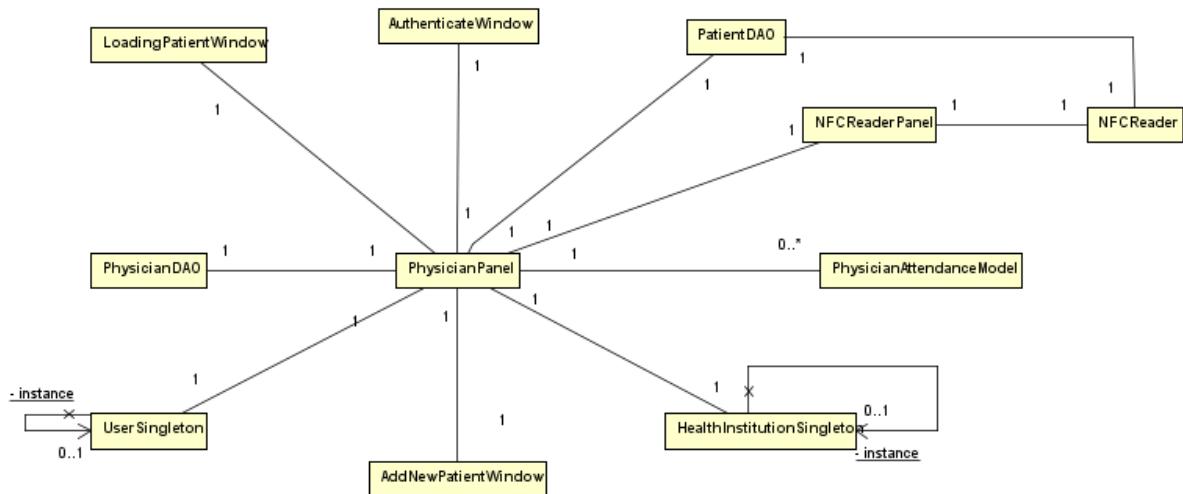
Fonte: Elaborado pelo autor

Para adicionar um novo usuário a uma instituição de saúde é utilizada a classe “UserAddWindow”. Adicionar o usuário por esta classe significa criar um usuário totalmente novo, e o vínculo com a instituição de saúde será realizado de modo automático. Esta classe possui acesso a “HealthInstitutionDAO” e a “UserDAO” para realização do vínculo e adição do usuário. A instituição de saúde a qual o usuário será adicionado é obtida a partir da classe “HealthInstitutionSingleton”. A edição do usuário é realizada na classe “UserUpdateWindow”, que é utilizada também para realizar a atualização do usuário. As outras operações de enviar convite e adicionar usuário administrador sistêmico possuem uma estrutura semelhante à estrutura apresentada. Um usuário sistêmico é um usuário administrador da instituição de saúde não humano, como por exemplo, uma conta *admin* do sistema operacional Linux.

3.3.2.8 Funcionalidade de Gerenciar pacientes

Esta funcionalidade apenas pode ser acessada pelo médico. Trata-se de uma tela que possui uma listagem dos pacientes atendidos, e funções para adicionar novos pacientes, e realizar a leitura dos equipamentos que utilizam *NFC* para assim estar realizando a consulta dos dados do paciente.

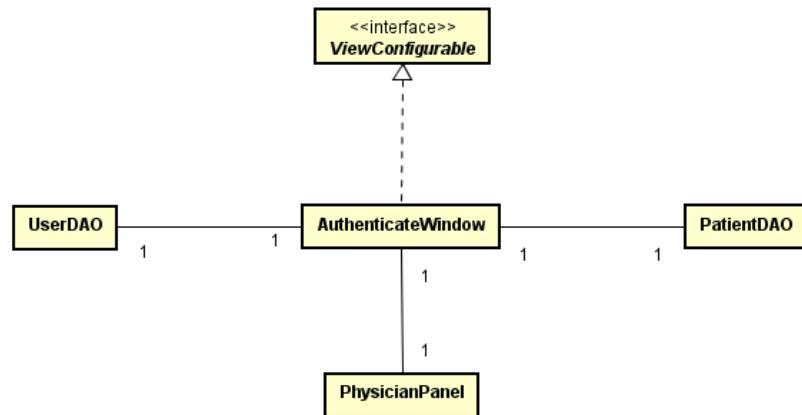
Figura 27 - Diagrama de classe de gerenciamento de pacientes



Fonte: Elaborado pelo autor

A classe “PhysicianPanel” possui uma lista dos pacientes que foram atendidos em uma determinada data, sendo possível realizar o filtro por data e nome do paciente. Por padrão, é carregado a data atual. Para adicionar um novo paciente, é utilizado a classe “AddNewPatientWindow”. Caso seja clicado sobre um dos pacientes da lista de atendimento, é possível acessar os dados médicos dos pacientes, e para isso, uma janela de carregamento é acionada utilizando a classe “LoadingPatientWindow”. Para obter o local do atendimento e quem o realizou são utilizadas as classes “UserSingleton” e “HealthInstitutionSingleton”. Caso o usuário não tenha presente o equipamento NFC, ele pode comunicar ao(a) médico(a) seu código secreto de acesso, um código de quatro caracteres, para que então, o(a) médico(a) possua acesso ao seu perfil.

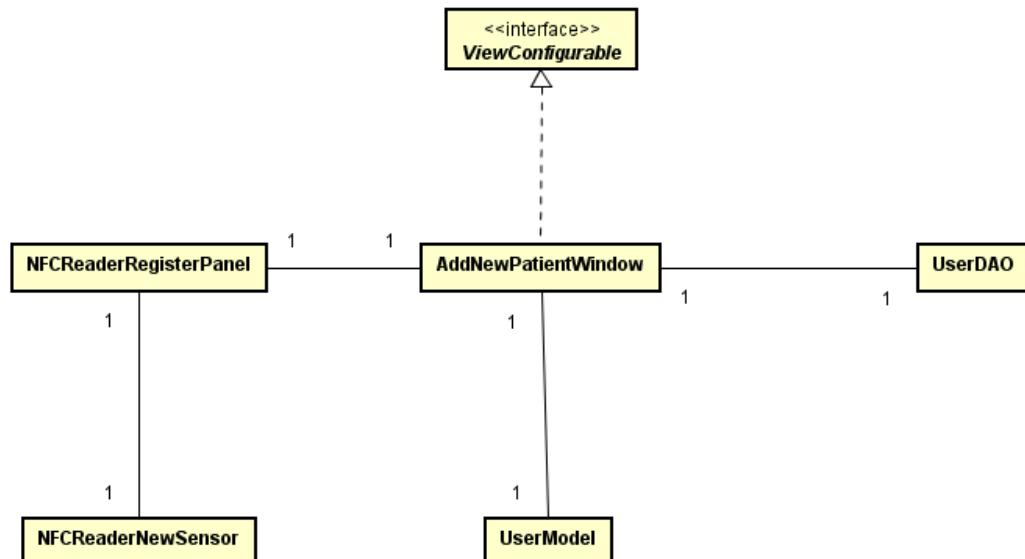
Figura 28 - Diagrama de classe da janela de autenticação



Fonte: Elaborado pelo autor

Para adicionar um novo paciente, basta informar os dados como *login*, nome, documento de identidade, pais, e, por fim, o equipamento de acesso que será utilizado.

Figura 29 - Diagrama de classe de adicionar novo paciente



Fonte: Elaborado pelo autor

Para realizar a leitura do equipamento *NFC*, é utilizado a classe "NFCReaderNewSensor", que através da biblioteca *JSSC*¹⁷ realiza a leitura do código armazenado no equipamento de acesso, e, em seguida, realiza a consulta do paciente

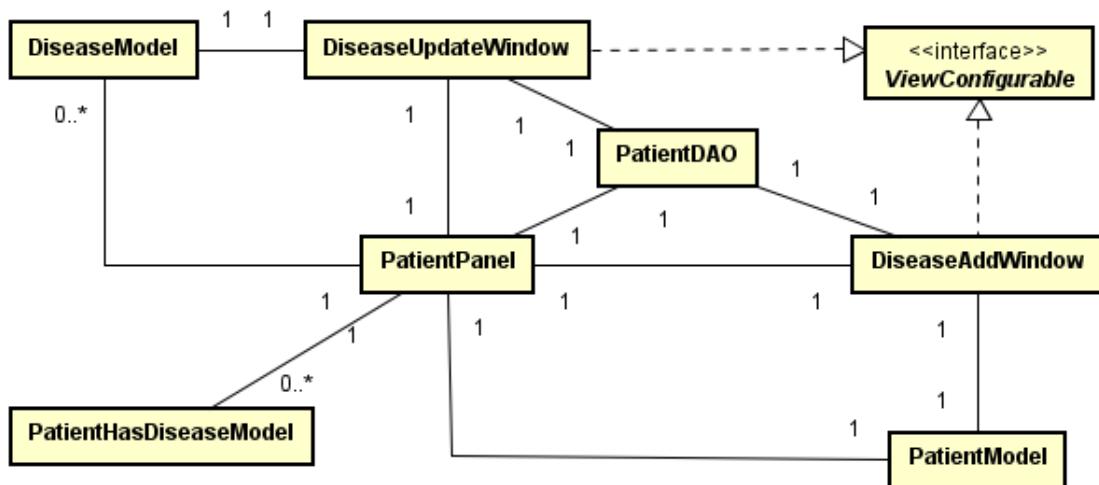
¹⁷ JSSC significa *Java Simple Serial Connector*, e é uma biblioteca muito utilizada para se comunicar com o Arduino utilizando Java.

no banco de dados. A tela de leitura do equipamento é representada pela classe “NFCReaderRegisterPanel”, responsável por alertar o andamento da ação. A classe “NFCReaderPanel” e “NFCReader” possuem a mesma estrutura que as respectivas classes “NFCReaderRegisterPanel” e “NFCReader”.

3.3.2.9 Funcionalidade de Gestão do paciente

A função de gestão de paciente só pode ser acessada por um(a) médico(a), e esta função apresenta ao(à) médico(a) uma série de dados sobre o paciente, tais como exames, diagnósticos, procedimentos, doenças, medicamentos e equipamentos de acesso. Dado a ampla quantidade de funções, os diagramas se apresentam em uma série para melhor explicação e compreensão da funcionalidade.

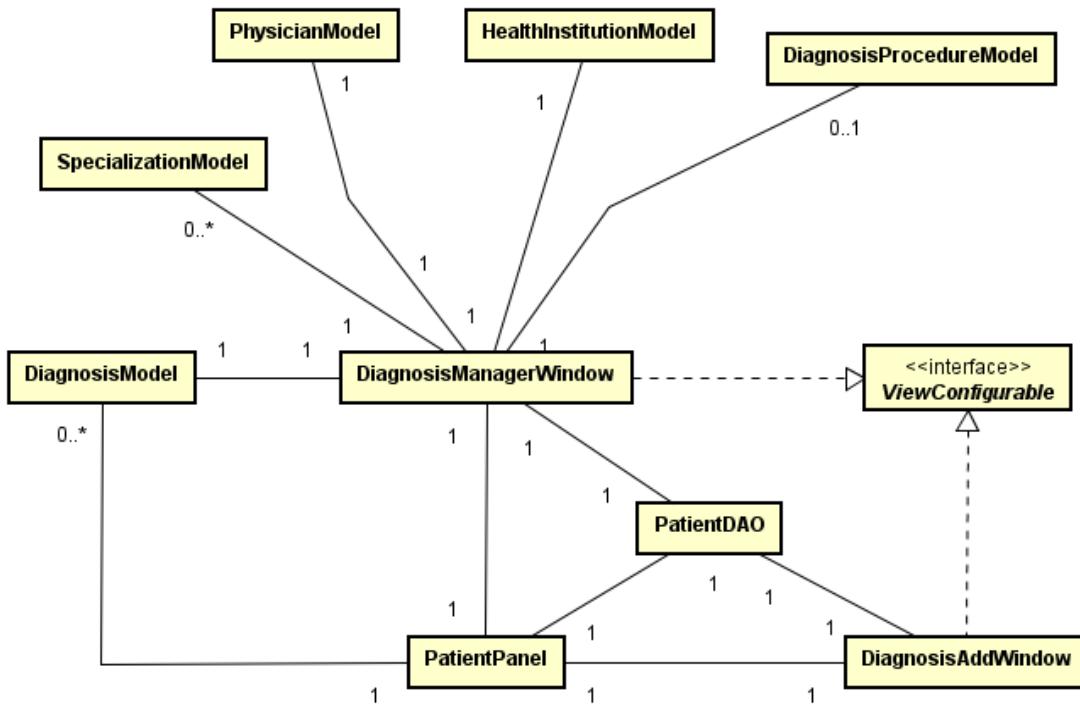
Figura 30 - Diagrama de classe de gerenciamento das doenças do paciente



Fonte: Elaborado pelo autor

A tela inicial carregada após a leitura do código do equipamento de acesso e autorização é a tela de paciente, representada pela classe “PatientPanel”. Esta é uma classe que representa listagem de todas as doenças, exames, diagnósticos, medicamentos e equipamentos de acesso. A partir desta tela, é possível adicionar, excluir e alterar uma doença. Para adicionar, é utilizado a classe “DiseaseAddWindow”, e para alterar, “DiseaseUpdateWindow”.

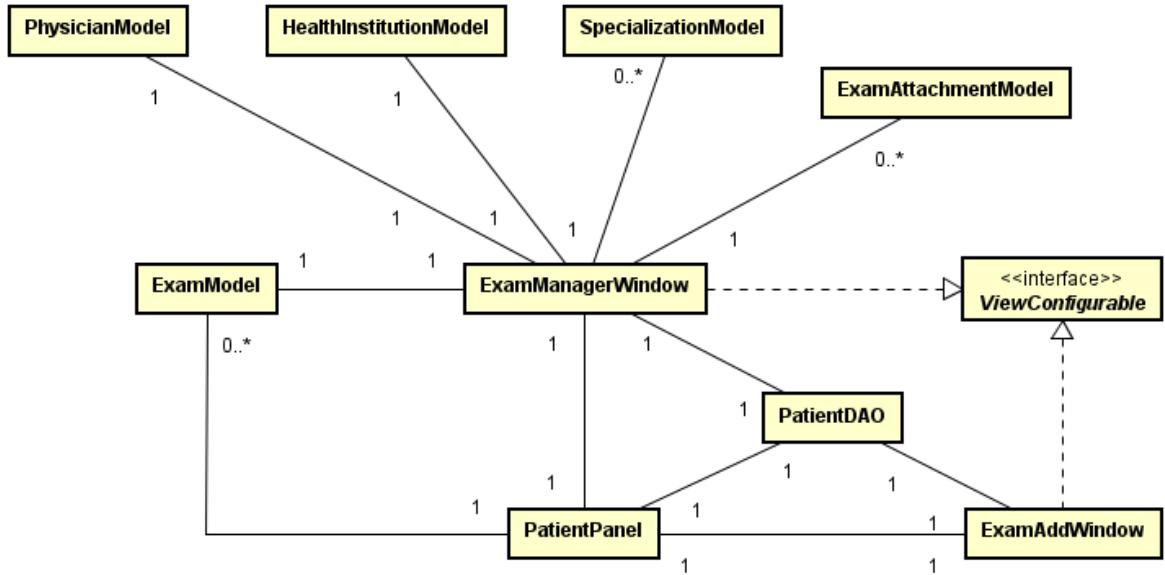
Figura 31 - Diagrama de classe de gestão de diagnósticos do paciente



Fonte: Elaborado pelo autor

A tela representada pela classe “PatientPanel”, assim como dito antes, possui uma lista de diagnósticos. A partir disto, é possível adicionar (“DiagnosisAddWindow”) e visualizar (“DiagnosisManagerWindow”) diagnósticos. Ao adicionar um diagnóstico em uma instituição de saúde, é possível solicitar que procedimentos de enfermagem sejam realizados. Este processo envia o paciente atual para uma fila de atendimento da instituição de saúde, no qual enfermeiros(as) possuem acesso, realizando todos os procedimentos necessários.

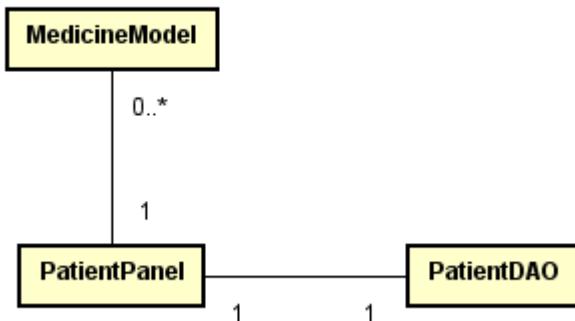
Figura 32 - Diagrama de classe de gerenciamento de exames



Fonte: Elaborado pelo autor

A classe “PatientPanel” possui uma lista com os exames realizados. Ao clicar em um exame para visualiza-lo, a classe “ExamManagerWindow” é invocada, e uma janela com todas as informações do exame, anexos, médico(a), especializações e instituição de saúde estará presente. Também é possível adicionar um exame através da classe “ExamAddWindow”.

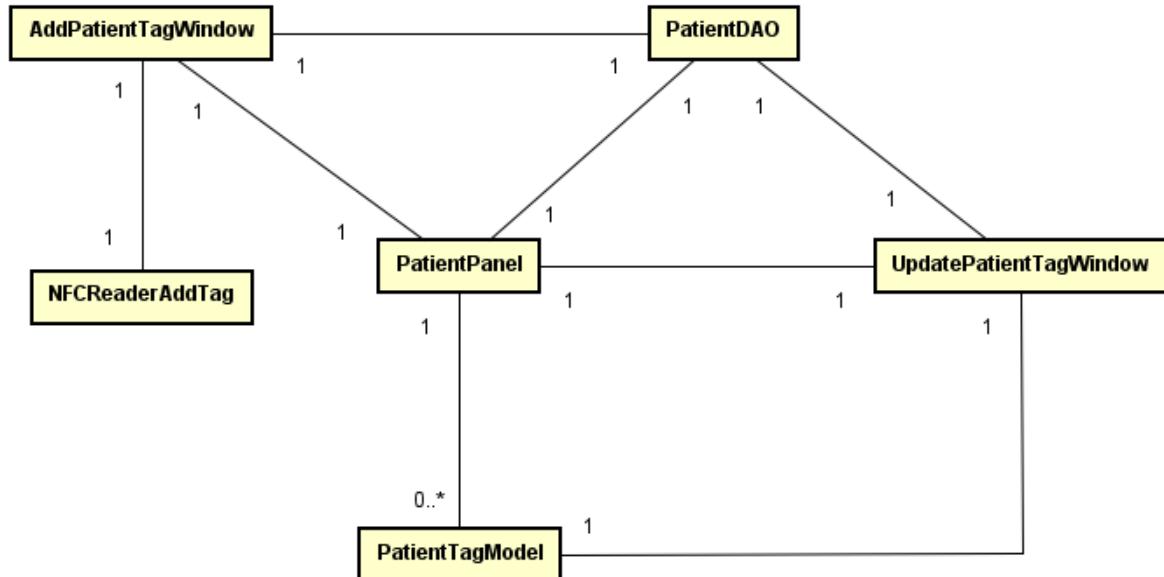
Figura 33 - Diagrama de classe de gerenciamento de medicamentos



Fonte: Elaborado pelo autor

A classe “PatientPanel” possui uma lista de medicamentos que o paciente utiliza, e que pode ser apenas visualizada, ou seja, o(a) médico(a) não pode adicionar nem alterar medicamentos.

Figura 34 - Diagrama de classe de gerenciar equipamentos de acesso



Fonte: Elaborado pelo autor

Os equipamentos de acesso podem ser cartões, pulseiras, colares e qualquer aparato que utilize da tecnologia *NFC*. A classe “PatientPanel” possui uma lista de equipamentos de acesso e a partir da mesma, é possível editar (“UpdatePatientTagWindow”) e excluir os equipamentos. A edição conta com adicionar nome e descrição específica para o equipamento de acesso. Também é possível realizar a adição utilizando a classe “AddPatientTagWindow”, que utiliza da classe “NFCReaderAddTag” para realizar o procedimento de leitura do equipamento, de forma idêntica ao que é demonstrada no diagrama de sequência de leitura do aparelho que possui *NFC*.

3.3.3 Sistema de Enfermagem (Aplicativo)

O sistema de enfermagem será utilizado por todos(as) enfermeiros(as) vinculados a uma instituição de saúde. Na Tabela 41 é apresentado a estrutura de classes do sistema.

Tabela 41 - Especificação dos pacotes utilizados

Nome do pacote	Descrição
com.tcc.nhsappmedic	Pacote base.
com.tcc.nhsappmedic.login	Possui a mecânica de login do sistema.

com.tcc.nhsappmedic.home	Possui uma classe, a controladora da tela inicial do aplicativo, que possui uma lista de pacientes que devem ser atendidos.
com.tcc.nhsappmedic.home.adapter	Possui uma classe adaptadora, responsável pela adaptação da interface gráfica, realizando o relacionamento entre os modelos e a classe “RecyclerView” ¹⁸ .
com.tcc.nhsappmedic.home.model	Classe de modelo que será utilizada pela controladora da tela inicial, a “HomeController”.
com.tcc.nhsappmedic.home.util	Classe de modelo que será responsável por armazenar um item da “RecyclerView”, ou seja, é basicamente um dos vários itens da lista.
com.tcc.nhsappmedic.institution	Controladora responsável por selecionar uma das várias instituições de saúde.
com.tcc.nhsappmedic.institution.adapter	Classe de adaptação da “RecyclerView” da controladora de instituições.
com.tcc.nhsappmedic.institution.model	Classe de modelo da instituição de saúde.
com.tcc.nhsappmedic.institution.util	Classe que armazena os itens da “RecyclerView” da instituição.
com.tcc.nhsappmedic.attendance	Controladora responsável por listar atendimentos em andamento.
com.tcc.nhsappmedic.attendance.adapter	Classe de adaptação da “RecyclerView” da controladora de atendimentos em andamento.
com.tcc.nhsappmedic.attendance.model	Classe de modelo dos atendimentos em andamento.
com.tcc.nhsappmedic.attendance.util	Classe que armazena os itens da “RecyclerView” dos atendimentos em andamento.
com.tcc.nhsappmedic.procedures	Classe controladora que permite a adição de procedimentos em um paciente.

Fonte: Elaborado pelo autor

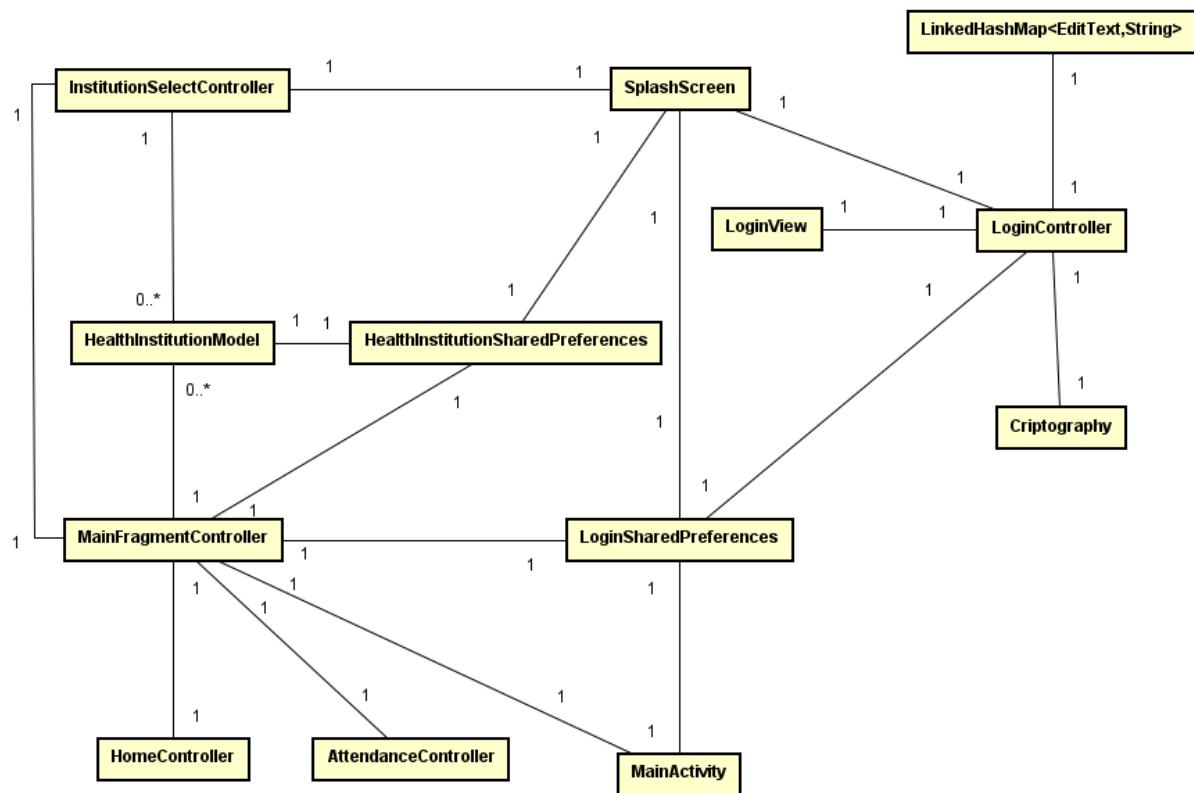
3.3.3.1 Funcionalidade de *Login* e Navegação

A estrutura de classes a seguir demonstra as funções de *login* juntamente a naveabilidade do sistema. De forma básica, o usuário realizará o *login*, e logo em seguida irá selecionar uma das várias instituições de saúde a qual o mesmo faz parte. Em seguida, após selecionar uma instituição, o usuário poderá visualizar os

¹⁸ “RecyclerView” basicamente é uma lista de elementos no Android que é utilizada para casos de grandes quantidades de dados ou grande frequência de mudança de dados. Veja mais em <https://developer.android.com/guide/topics/ui/layout/recyclerview>.

atendimentos que precisam ser realizados, a lista de atendimentos em processo (relacionado ao usuário do sistema, ou seja, um usuário apenas vê seus próprios atendimentos em progresso), e a opção de mudar de instituição de saúde. Ao clicar sobre um paciente (que estará na lista de atendimentos em andamento, ou na lista de novos atendimentos que precisam ser realizados) é possível adicionar procedimentos ao paciente. Os procedimentos estão diretamente ligados a um diagnóstico.

Figura 35 - Diagrama de classe de Login e Navegação



Fonte: Elaborado pelo autor

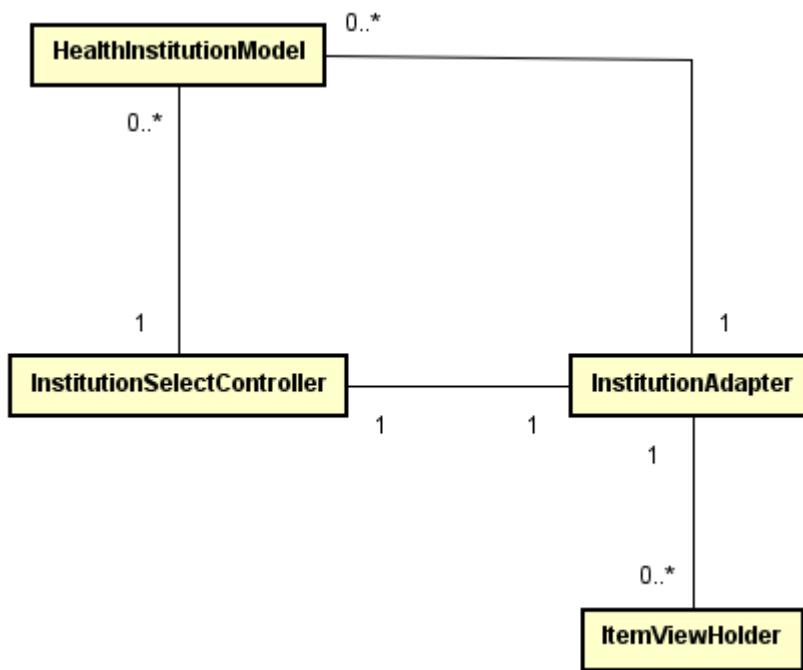
O aplicativo se inicia na tela de carregamento (“SplashScreen”), e já na mesma, é checado se o usuário já está autenticado no sistema. Caso esteja, o mesmo é enviado para a tela de seleção de instituições de saúde (“InstitutionSelectController”). Porém, caso não esteja, é enviado para a tela de Login (“LoginController”). Então, o usuário preencherá *login* e senha, e caso esteja cadastrado no sistema, será direcionado para a tela de seleção de instituições de saúde (“InstitutionSelectController”). Após selecionar a instituição, o usuário é encaminhado para a tela de listagem de novos pacientes que precisam ser atendidos (“HomeController”). Um(a) enfermeiro(a) pode iniciar um atendimento nesta tela, e caso precisar, poderá interromper o processo, podendo voltar a realizar o atendimento

novamente mais tarde na tela de atendimentos em progresso (“AttendanceController”).

3.3.3.2 Funcionalidade de seleção de instituição de saúde

A seleção da instituição de saúde é acessada tanto ao entrar no aplicativo, quanto no momento que o usuário está utilizando o sistema, podendo optar por trocar de instituição. É uma função que permite com que o usuário acesse as listas de pacientes de cada instituição.

Figura 36 - Diagrama de classe de seleção de instituição de saúde



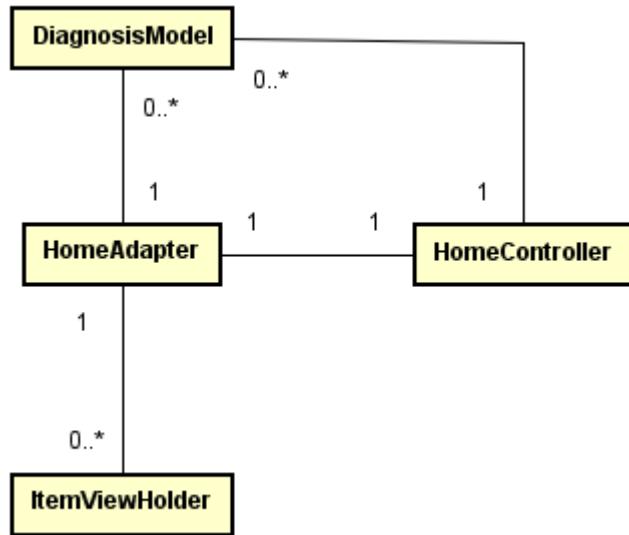
Fonte: Elaborado pelo autor

Basicamente, a tela representada pela classe “InstitutionSelectController” possui uma “RecyclerView”, ou seja, uma lista de itens (no caso, instituições de saúde) que podem ser utilizados pelo usuário. Cada um desses itens da lista representa um “ItemViewHolder”, que deve ser gerenciado por um adaptador, no caso, “InstitutionAdapter”. O adaptador irá criar itens conforme for necessário.

3.3.3.3 Funcionalidade de tela inicial de listagem de novos pacientes

A tela inicial é carregada após a seleção da instituição de saúde, e pode ser acessada a qualquer momento. Na tela inicial fica uma fila de pacientes que ainda não foram atendidos.

Figura 37 - Diagrama de classe de listagem de novos pacientes na fila



Fonte: Elaborado pelo autor

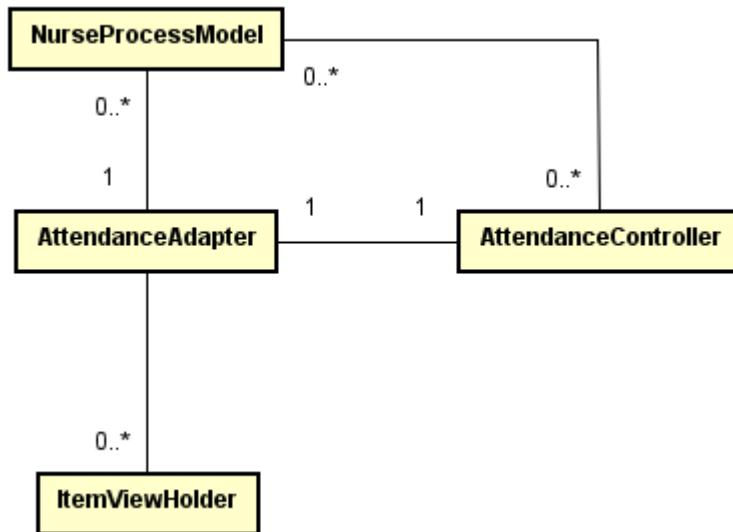
A classe controladora (“HomeController”) é responsável por obter uma lista de novos pacientes através do RabbitMQ. A controladora também possui uma “RecyclerView”, necessitando de um modelo, uma view de armazenamento de itens e um adaptador para o gerenciamento. Assim, é possível visualizar a fila em tempo real. No momento em que um(a) enfermeiro(a) clicar em um paciente, um evento de notificação é enviado para todas as filas, retirando o atual paciente da fila e o enviando para um *status* de atendimento em processo. Ao acessar o diagnóstico do paciente, será aberta uma tela para adicionar um procedimento. A classe “ProcedureController” representa esta tela.

3.3.3.4 Funcionalidade de tela inicial de listagem de atendimentos em processo

A tela de listagem de atendimentos em progresso é utilizada quando um atendimento a um novo paciente foi iniciado, porém, não pode ser continuado devido a alguma necessidade externa. Esses atendimentos ficam listados na tela de atendimentos em progresso.

A classe controladora (“AttendanceController”) obtém uma lista de pacientes com atendimento em progresso. A controladora também possui uma “RecyclerView”, que segue a mesma lógica utilizada pelo diagrama de listagem de novos pacientes.

Figura 38 - Diagrama de classe de gerenciar lista de atendimentos em progresso



Fonte: Elaborado pelo autor

3.3.4 Sistema do Paciente (Aplicativo)

O sistema do paciente será utilizado por todos os usuários pacientes. Na Tabela 42 é apresentado a estrutura de classes do sistema.

Tabela 42 - Especificação dos pacotes utilizados

Nome do pacote	Descrição
com.example.tcc.healthsystemapp	Pacote base.
com.example.tcc.healthsystemapp.login	Possui a mecânica de login do sistema.
com.example.tcc.healthsystemapp.exams	Possui todos os recursos para o gerenciamento de exames.
com.example.tcc.healthsystemapp.geoLocation	Utilitários de geolocalização.
com.example.tcc.healthsystemapp.home	Recursos para o funcionamento da tela principal.
com.example.tcc.healthsystemapp.home.adapter	Classe adaptadora da tela inicial do sistema.
com.example.tcc.healthsystemapp.home.util	Classes utilitárias para o funcionamento da tela inicial do sistema.
com.example.tcc.healthsystemapp.institutions	Recursos para o funcionamento da listagem das instituições de saúde mais próximas.
com.example.tcc.healthsystemapp.institutions.model	Classes de modelo para listagem das instituições.

com.example.tcc.healthsystemapp.institutions.adapter	Classe adaptadora das instituições de saúde.
com.example.tcc.healthsystemapp.institutions.info	Classe de informações sobre os(as) médicos(as) das respectivas instituições de saúde.
com.example.tcc.healthsystemapp.institutions.util	Classes utilitárias para o funcionamento das respectivas listagens.
com.example.tcc.healthsystemapp.medication	Recursos para o funcionamento da função de medicamentos.
com.example.tcc.healthsystemapp.medication.dao	Classe de acesso ao banco de dados de medicamentos do SQLite.
com.example.tcc.healthsystemapp.medication.model	Classe de modelo dos medicamentos.
com.example.tcc.healthsystemapp.medication.adapter	Classe adaptadora da listagem de medicamentos.
com.example.tcc.healthsystemapp.medication.db	Classes de definição do banco de dados local de medicamentos.
com.example.tcc.healthsystemapp.medication.info	Classe de informações de um respectivo medicamento.
com.example.tcc.healthsystemapp.medication.receiver	Utilitário para agendamento das notificações de medicamentos.
com.example.tcc.healthsystemapp.medication.register	Classe para registrar um novo medicamento.
com.example.tcc.healthsystemapp.medication.service	Classe utilitária para agendar novamente os horários dos medicamentos em caso de o celular reiniciar.
com.example.tcc.healthsystemapp.patient	Classes de base para funções do paciente.
com.example.tcc.healthsystemapp.patient.patientRegister	Classes para a mecânica de cadastrar um novo paciente.
com.example.tcc.healthsystemapp.patient.patientInfo	Classes para a obtenção das informações do paciente.
com.example.tcc.healthsystemapp.patient.country	Classes para obtenção do país do usuário.
com.example.tcc.healthsystemapp.physicians	Classes para listagem dos médicos mais próximos e médicos conhecidos.
com.example.tcc.healthsystemapp.physicians.model	Classes de modelo para a listagem dos médicos.
com.example.tcc.healthsystemapp.physicians.adapter	Classes adaptadoras para listagem de instituições de saúde e médicos(as).

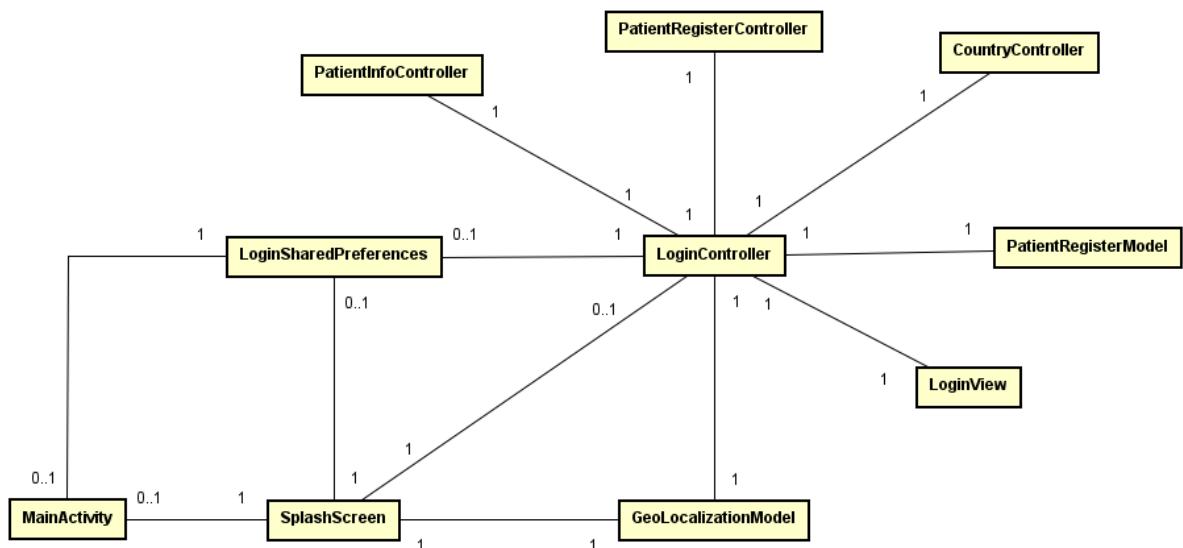
com.example.tcc.healthsystemapp.physicians.info	Classe de informações as instituições de saúde dos(as) respectivos(as) médicos(as).
com.example.tcc.healthsystemapp.physicians.util	Classes utilitárias para o funcionamento das respectivas listagens.
Util	Classes utilitárias.

Fonte: Elaborado pelo autor

3.3.4.1 Funcionalidade de inicialização do sistema

Ao aplicativo iniciar, a tela de carregamento “SplashScreen” é inicializada, sendo responsável por obter a localização do usuário, e carregar ou a tela de Login (“LoginController”), ou a tela principal (“MainActivity”) caso o usuário já esteja autenticado. Já na tela de login, o usuário pode optar por realizar o procedimento de login e entrar no aplicativo através da classe “LoginSharedPreferences”, que redireciona o usuário para a “MainActivity”. O paciente também possui a opção nesta tela de realizar um novo cadastro utilizando a classe “PatientRegisterController”. Na classe “LoginSharedPreferences” são salvos diversos tipos de dados, tais como identificadores de usuário, nomes, e informações importantes. A classe “MainActivity” representa a classe principal do sistema, invocada no momento em que o usuário se autentica com sucesso.

Figura 39 - Diagrama de classe da funcionalidade de inicialização

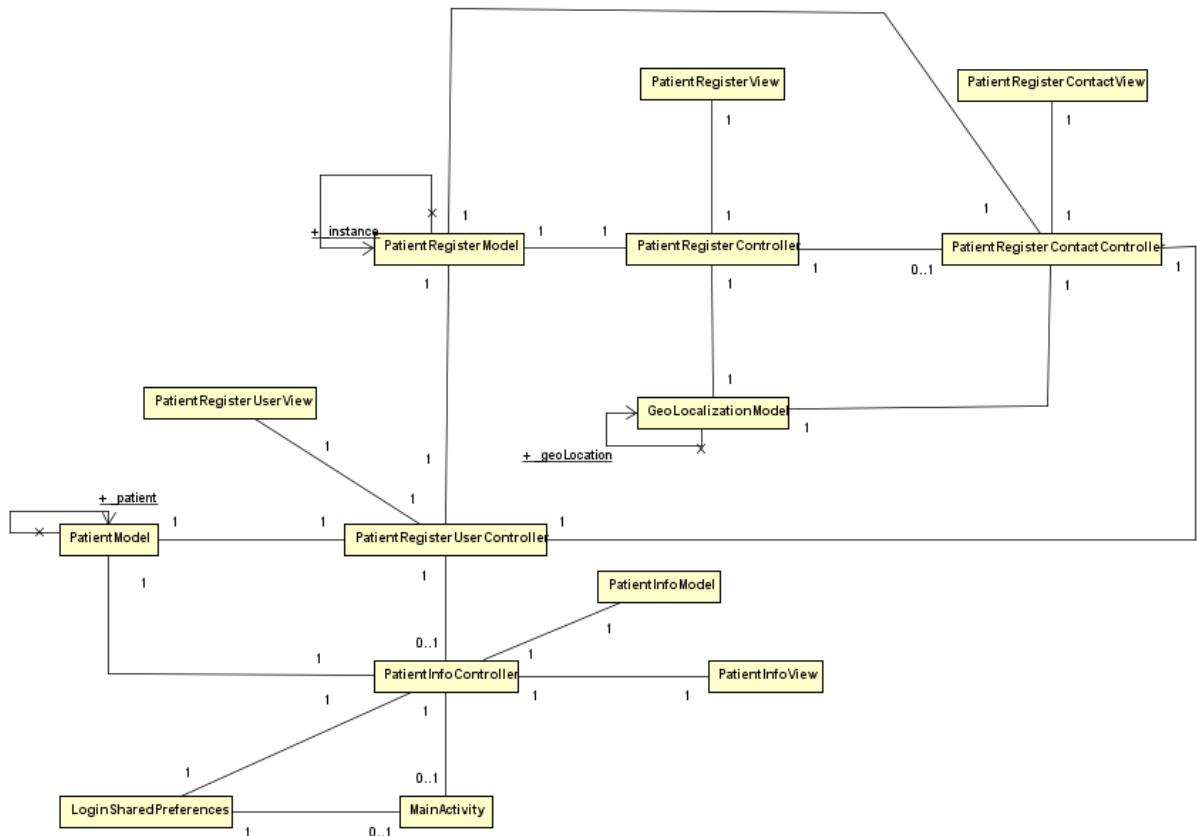


Fonte: Elaborado pelo autor

3.3.4.2 Funcionalidade de cadastro de paciente

Ao clicar em cadastrar um novo usuário, o paciente é direcionado para a tela representada pela classe “PatientRegisterController”, onde ele irá informar dados como nome, sexo, data de nascimento, e-mail e outros. Após confirmar os dados nesta tela, ele é direcionado para a tela a qual deverá preencher informações de contato (“PatientRegisterContactController”), como endereço, bairro, rua, cidade e outros campos. Caso todos os dados tenham sido preenchidos corretamente, o usuário é enviado para a tela de preenchimento de dados de usuário (“PatientRegisterUserController”), onde deverá preencher sua senha para ser utilizada no aplicativo. Por fim, ele é enviado a tela de preenchimento de informações de paciente (“PatientInfoController”), como nome dos pais, tipo sanguíneo, cor, peso e altura. Por fim, caso o cadastro tenha sido realizado com sucesso, o usuário é direcionado para a “MainActivity”.

Figura 40 - Diagrama de classe de cadastro de paciente

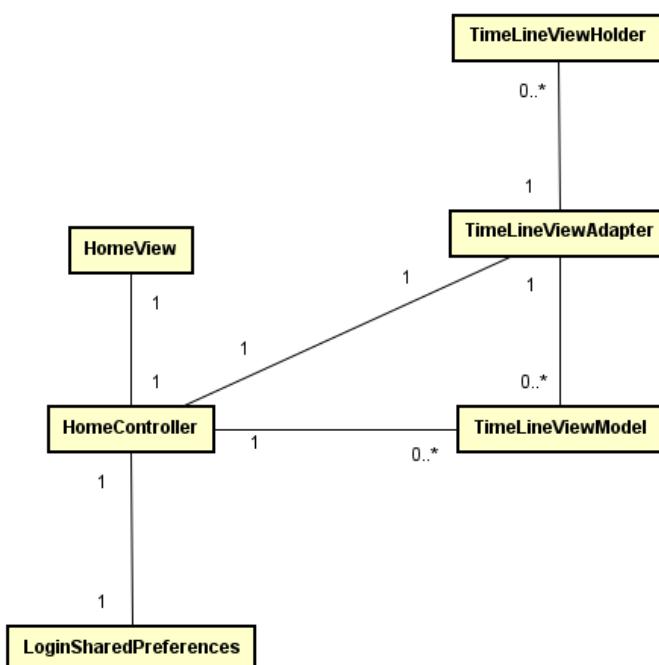


Fonte: Elaborado pelo autor

3.3.4.3 Funcionalidade de tela inicial

A tela inicial do sistema é representada pela classe “HomeController”, a qual possui uma “RecyclerView”, ou seja, uma lista de eventos de saúde que ocorreram com o paciente. Assim, para buscar estes respectivos eventos, foi utilizado como classe utilitária a “LoginSharedPreferences” para se obter o identificador do paciente. Para realizar o gerenciamento da “RecyclerView”, também se é utilizado uma classe adaptadora “TimeLineViewHolder”, e uma classe *view holder* “TimeLineViewHolder”.

Figura 41 - Diagrama de classe da tela inicial do sistema



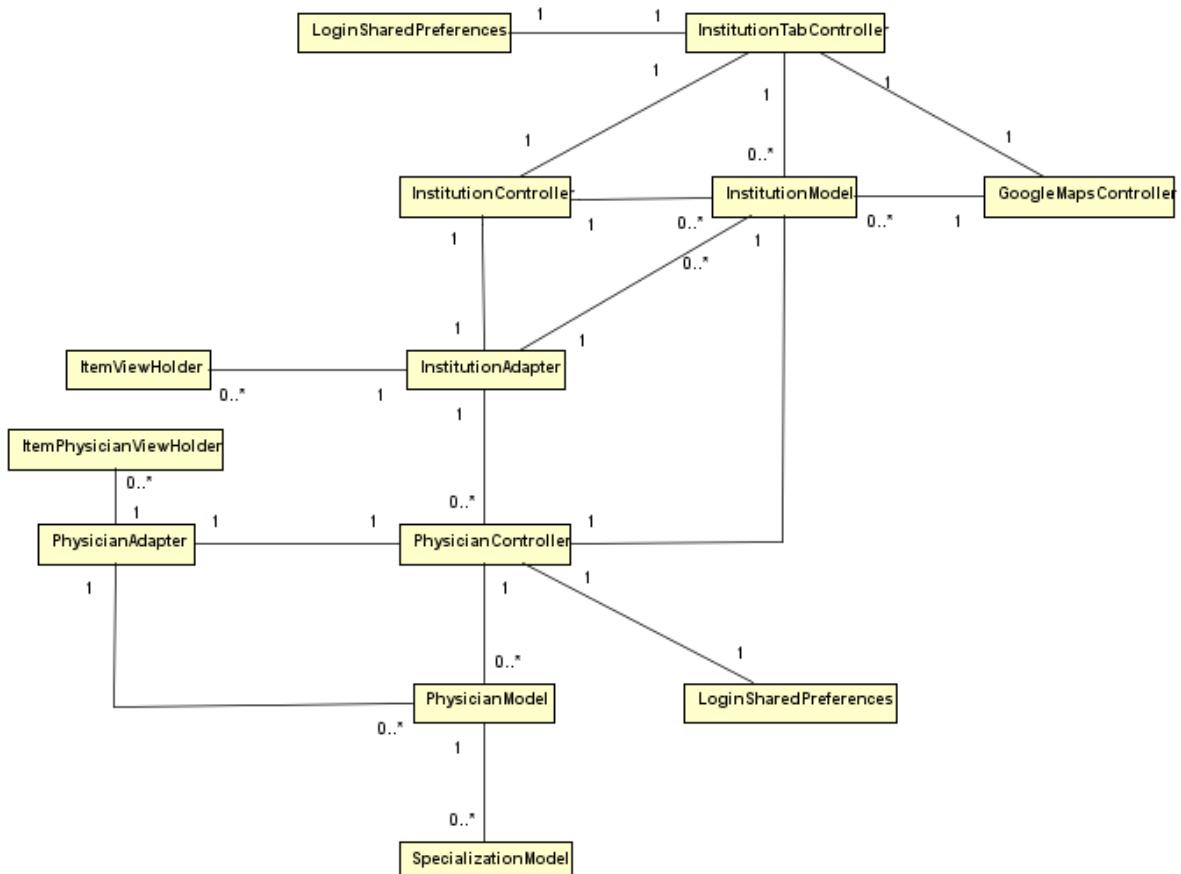
Fonte: Elaborado pelo autor

3.3.4.4 Funcionalidade de listagem de instituições de saúde próximas

A tela de listagem de instituições de saúde é representada pela classe “InstitutionTabController”, a qual é responsável por gerenciar as abas de listagem de instituições de saúde (“InstitutionController”), e do mapa com as instituições de saúde (“GoogleMapsController”). A classe “InstitutionController” possui uma “RecyclerView” para listagem das instituições, e possui uma classe de gerenciamento para a mesma, a “InstitutionAdapter”. Esta classe adaptadora possui diversos *view holders*, que possuem a interface propriamente dita. Cada *view holder* representa um item da lista. A listagem de médicos(as) das instituições de saúde parte da classe “InstitutionAdapter”. A classe que representa esta listagem é a classe

“PhysicianController”, que possui o mesmo funcionamento da classe “InstitutionController”, possuindo uma “RecyclerView”, classe adaptadora e *view holder*. A classe “PhysicianController” lista tanto os médicos quanto as especializações dos mesmos.

Figura 42 - Diagrama de classe de listagem de instituições de saúde próximas



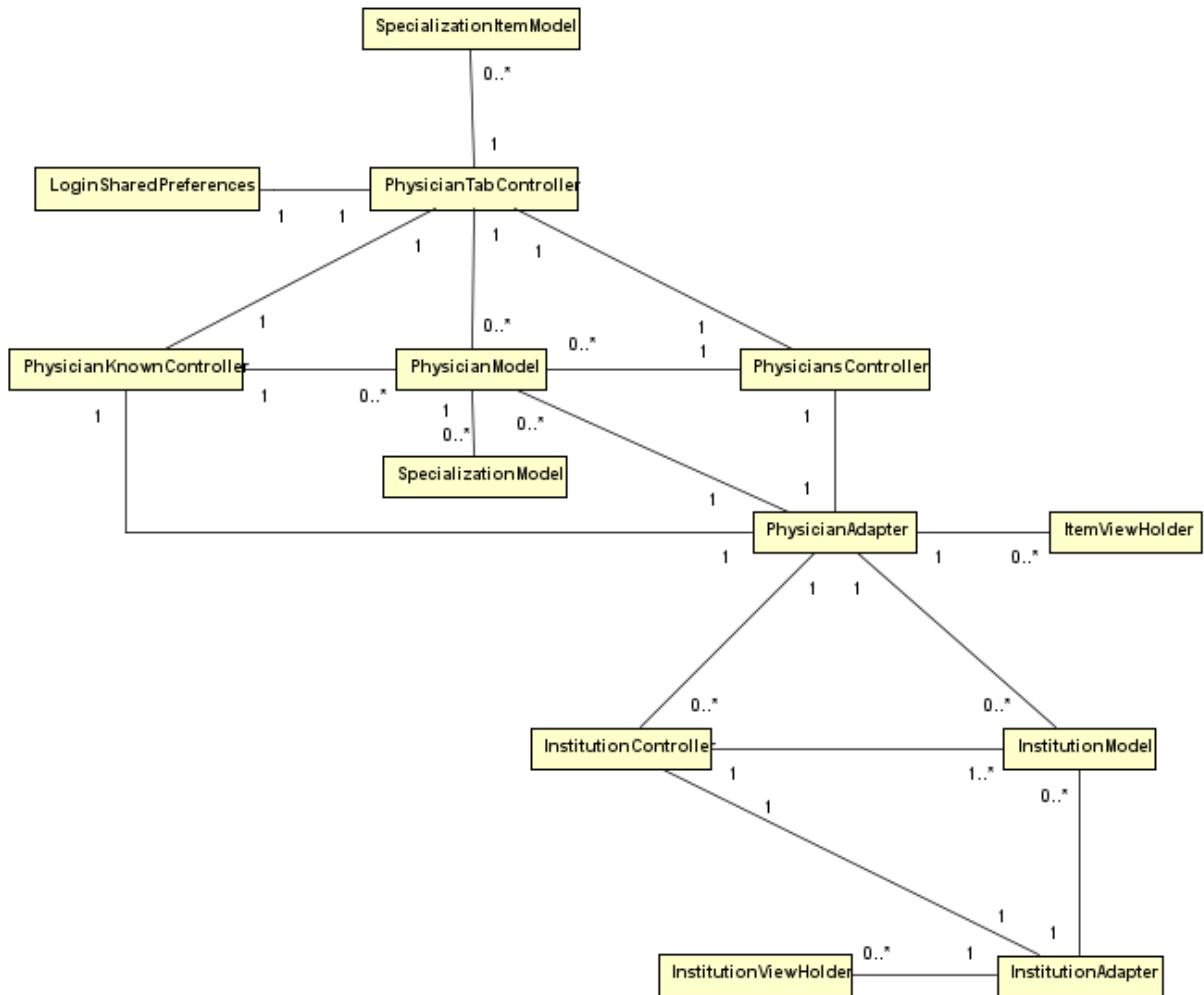
Fonte: Elaborado pelo autor

3.3.4.5 Funcionalidade de listagem de médicos(as)

A tela de listagem de médicos(as) é representado pela classe “PhysiciansTabController”, que possui duas abas, listagem de médicos(as) próximos(as), representado pela classe “PhysiciansController”, e listagem de médicos(as) conhecidos(as), representado pela classe “PhysicianKnownController”. Ambas estas classes possuem uma “RecyclerView”, e utilizam a mesma classe de gerenciamento, a “PhysicianAdapter”, a qual possui diversos itens da classe “ItemViewHolder”. Através desta classe adaptadora, é possível acessar todas as

instituições de saúde a qual um(a) médico(a) faz parte, tela a qual é representada pela classe `InstitutionController`, classe à qual possui uma classe adaptadora (“`InstitutionAdapter`”), a qual possui diversos *view holders*.

Figura 43 - Diagrama de classe de listagem de médicos(as)



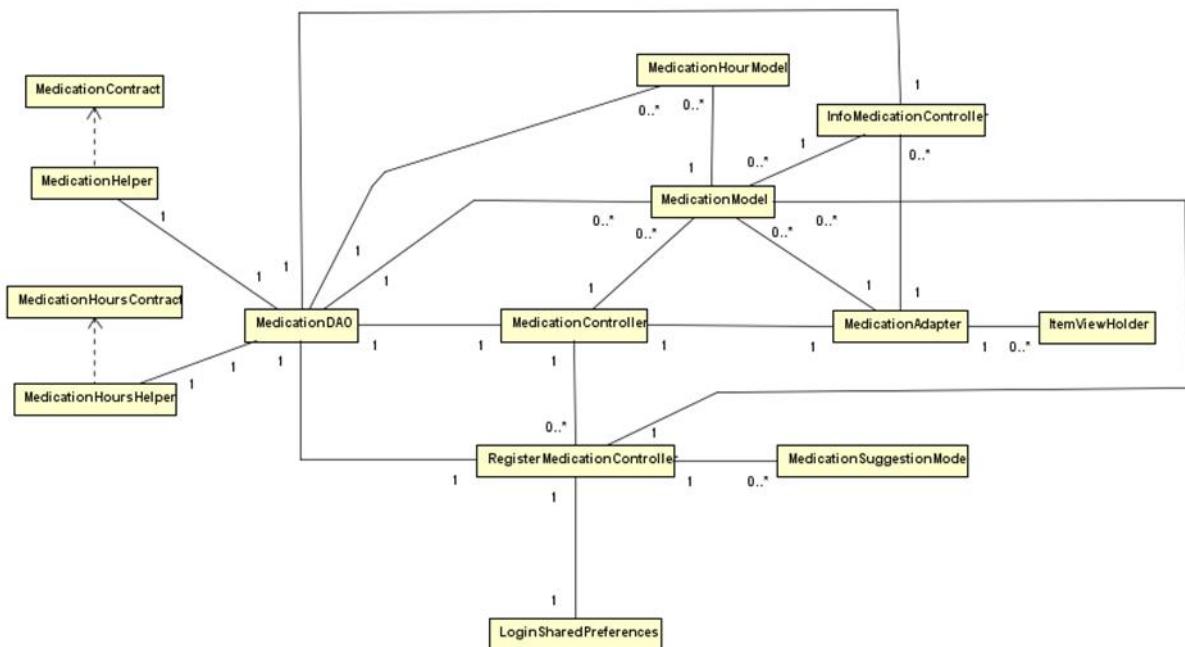
Fonte: Elaborado pelo autor

3.3.4.6 Funcionalidade de gerenciamento de medicamentos

A tela de listagem dos medicamentos que estão sendo gerenciados é representada pela classe “`MedicationController`”, que possui uma “`RecyclerView`”. Logo, ele possui uma classe adaptadora (“`MedicationAdapter`”), a qual possui diversas *view holders*. Através da classe adaptadora, é possível acessar as informações dos medicamentos que estão listados, e realizar a edição dos mesmos, através da classe “`InfoMedicationController`”. Também é possível adicionar um medicamento, através da classe “`RegisterMedicationController`”. As classes “`MedicationContract`” e

“MedicationHoursContract” definem a estrutura do banco de dados local de medicamentos. As classes “MedicationHelper” e “MedicationHoursHelper” são utilizadas para implementar a definição das classes de contrato no SQLite propriamente dito. A classe “MedicationDAO” é responsável por realizar operações no banco de dados.

Figura 44 - Diagrama de classe de gerenciamento de medicamentos



Fonte: Elaborado pelo autor

Existem também duas outras classes utilizadas que não estão listadas no diagrama. Uma delas, “AlarmNotificationReceiver”, é responsável pela realização das notificações no dispositivo móvel, alertando o paciente quando for necessário consumir determinado medicamento.

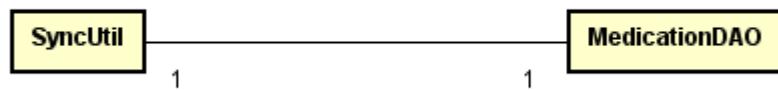
A outra classe é a “MedicationRebootService”. Quando um dispositivo Android é reiniciado, todos os alarmes agendados no mesmo dispositivo são perdidos. Esta classe é responsável por detectar a inicialização do dispositivo, e, logo em seguida, agendar todos os alarmes novamente.

3.3.4.7 Funcionalidade de sincronização

A funcionalidade de sincronização basicamente replica os dados do banco de dados local para a nuvem. Para isto, são utilizadas duas classes: “SyncUtil”,

responsável por enviar os medicamentos para serem sincronizados ao banco de dados, e “MedicationDAO”, para leitura dos medicamentos já cadastrados.

Figura 45 - Diagrama de classe de sincronização

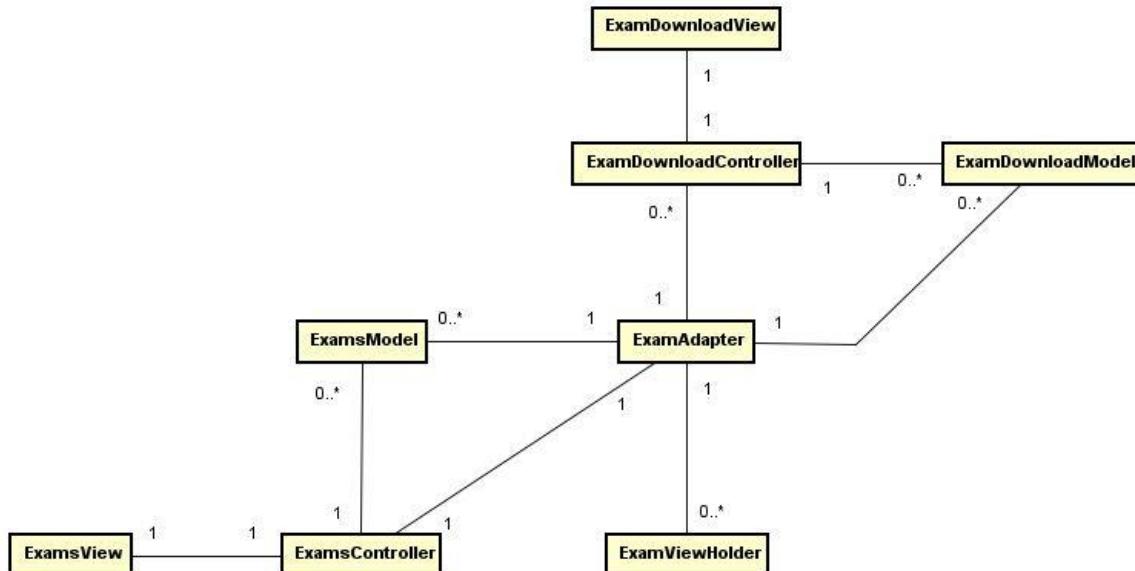


Fonte: Elaborado pelo autor

3.3.4.8 Funcionalidade de gestão de exames

A funcionalidade de gestão de exames permite com que o paciente possua acesso a sua respectiva lista de exames, permitindo acessar dados do local, quem realizou o exame, descrição e todos os anexos incluídos.

Figura 46 - Diagrama de classe de exames



Fonte: Elaborado pelo autor

A classe “ExamDownloadController” possui uma lista de exames, para realizar o gerenciamento da lista que utiliza de uma “RecyclerView”, é preciso de um adaptador (“ExamAdapter”), e através desse adaptador é possível acessar a lista de anexos.

3.4 Diagrama de Sequência

O diagrama de sequência é utilizado com o intuito de demonstrar a estrutura de um componente de software, sendo útil para exaltar a comunicação entre objetos durante a execução de uma tarefa, ornando-se visível a ordem de tempo no qual as mensagens são enviadas entre os objetos para a execução de uma determinada tarefa (PRESSMAN, 2011).

Com o intuito de demonstrar a comunicação com o sistema leitor de *NFC* desenvolvido utilizando Arduino, a biblioteca *JSSC* e Java, é demonstrado o passo a passo da obtenção dos dados do paciente a partir de um equipamento *NFC* através da Figura 47.

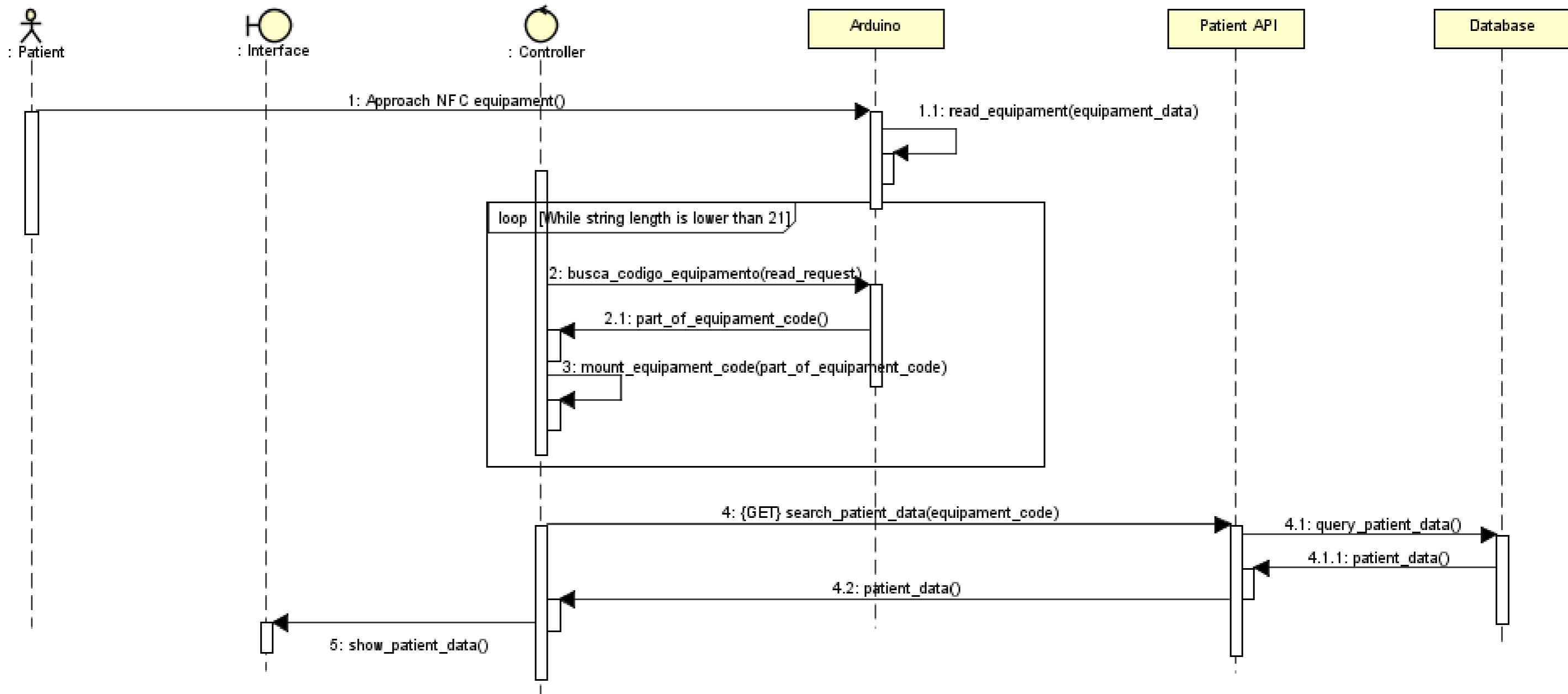
O ator (Paciente) aproxima o equipamento *NFC* no sistema Arduino. Logo em seguida, o sistema Arduino irá ler o equipamento utilizando o *NFC* como meio de comunicação. Após a leitura do código, o sistema *desktop* Java irá enviar consultas constantes ao Arduino através da biblioteca *JSSC*, para obter o código lido. Como o código não é retornado em sua forma completa, mas sim em partes sequenciais, o sistema *desktop* após realizar a leitura, precisa construir o código com base nas partes retornadas. Logo após a leitura terminar (quando a sequência do código atingir a quantidade de vinte e um caracteres), o sistema *desktop* envia uma requisição para a *API* de pacientes, enviando como parâmetro o código obtido através da leitura do equipamento. A *API* de pacientes, realiza a consulta no banco de dados e retorna os dados do paciente, que são retornados de volta para o sistema *desktop* e em seguida, enviados para a interface.

3.5 Banco de Dados

3.5.1 Diagrama de Entidade e Relacionamento (DER)

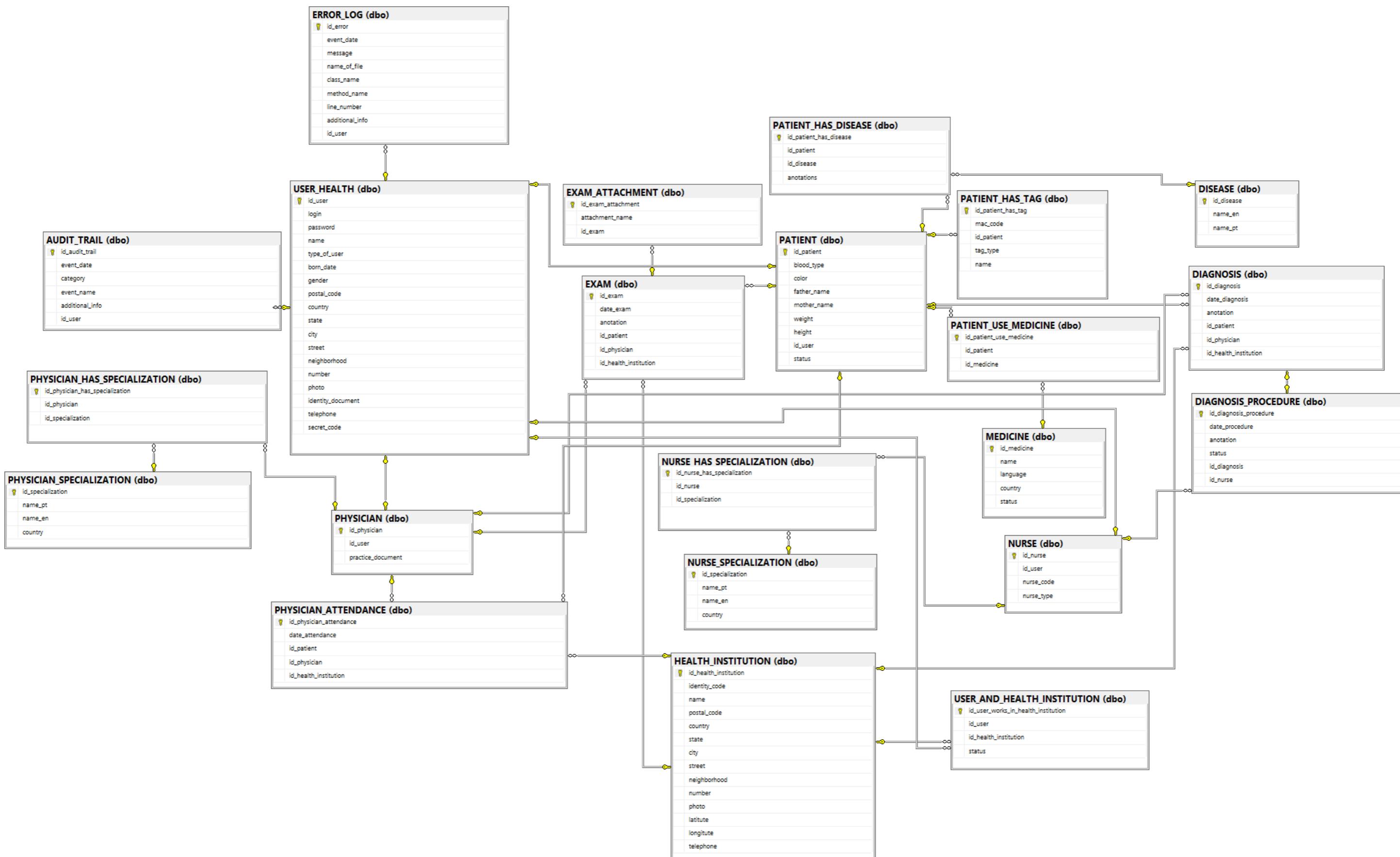
Diagrama Entidade Relacionamento (DER) é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação do Modelo de Entidades e Relacionamentos. Sua maior aplicação é para visualizar o relacionamento entre tabelas de um banco de dados, no qual as relações são construídas através da associação de um ou mais atributos destas tabelas. A Figura 48 apresenta o DER do sistema proposto.

Figura 47 - Diagrama de sequência da leitura do equipamento NFC



Fonte: Elaborado pelo autor

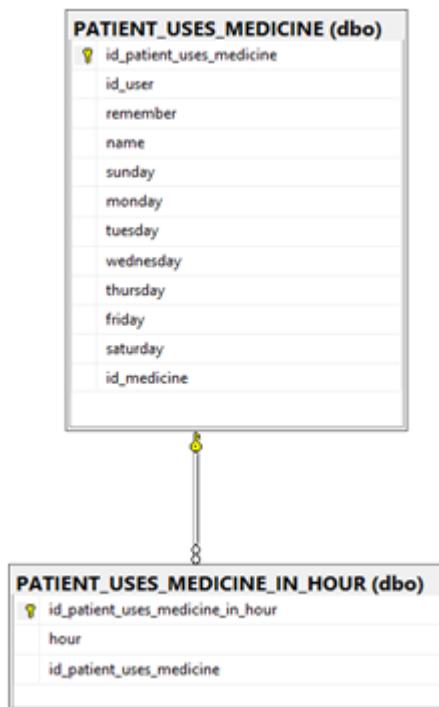
Figura 48 - DER do banco de dados armazenado na Azure



Fonte: Elaborado pelo autor

Já a Figura 49 apresenta o banco de dados local armazenado no SQLite.

Figura 49 - DER do banco de dados local no dispositivo móvel



Fonte: Elaborado pelo autor

3.5.2 Dicionário de Dados

O Dicionário de Dados (DD) consiste em uma lista organizada de todos os elementos de dados que são pertinentes para o sistema. As tabelas devem conter os seguintes campos:

Entidade: é o nome da entidade que foi definida no MER. A entidade é uma pessoa, objeto ou lugar que será considerada como objeto pelo qual temos interesse em guardar informações a seu respeito.

Atributo: Os atributos são as características da entidade cliente que desejamos guardar.

Classe: as classes podem ser: simples, composto, multivalorado e determinante. Simples indica um atributo normal. Composto indica que ele poderá ser dividido em outros atributos, como por exemplo, o endereço. Multivalorado é quando o valor do atributo poderá não ser único e determinante é um atributo que será usado como chave, como CPF, Código do cliente, etc.

Domínio: podem ser numéricos, texto, data e booleano. Podemos chamar também de tipo do valor que o atributo irá receber. A definição desses tipos deve seguir um processo lógico, exemplo: nome é texto, salário é numérico, data de nascimento é data e assim por diante.

Tamanho: define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo. Geralmente o tamanho é definido apenas para atributos de domínio texto.

Descrição: é opcional e pode ser usado para descrever o que é aquele atributo ou dar informações adicionais que possam ser usadas futuramente pelo analista ou programador do sistema.

As tabelas Tabela 43 até Tabela 66 apresentam o Dicionário de Dados de cada entidade.

Tabela 43 - Dicionário de dados da entidade AUDIT_TRAIL

Entidade: AUDIT_TRAIL				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_audit_trail	Determinante	Texto	36	Chave Primária da tabela de Auditoria.
Event_date	Simples	Data		Data em que ocorreu o evento.
Category	Simples	Texto	100	Categoria de pessoa que estava usando o sistema.
Event_name	Simples	Texto	200	Local onde ocorreu o evento.
Additional_info	Simples	Texto		Informações adicionais do evento.
Id_user	Simples	Texto	36	Código do usuário que acionou o evento.

Fonte: Elaborado pelo autor

Tabela 44 - Dicionário de dados da entidade DIAGNOSIS

Entidade: DIAGNOSIS				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_diagnosis	Determinante	Texto	36	Chave Primária da tabela de diagnósticos.
Date_diagnosis	Simples	Data		Data do diagnóstico.
Annotation	Simples	Texto		Anotações relativas ao diagnóstico.
Id_patient	Simples	Texto	36	Código do paciente a qual pertence o diagnóstico.
Id_physician	Simples	Texto	36	Código do médico que realizou o diagnóstico.
Id_health_institution	Simples	Texto	36	Código da instituição de saúde na qual o diagnóstico foi realizado.

Fonte: Elaborado pelo autor

Tabela 45 - Dicionário de dados da entidade DIAGNOSIS_PROCEDURE

Entidade: DIAGNOSIS_PROCEDURE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_diagnosis_procedure	Determinante	Texto	36	Chave Primária da tabela de procedimentos dos diagnósticos.
Date_procedure	Simples	Data		Data do procedimento.
Anotation	Simples	Texto		Anotações relativas ao procedimento.
Status	Simples	Texto	1	Status do diagnóstico.
Id_diagnosis	Simples	Texto	36	Código do diagnóstico médico.
Id_nurse	Simples	Texto	36	Código da enfermeira que realizou o procedimento.

Fonte: Elaborado pelo autor**Tabela 46 - Dicionário de dados da entidade DISEASE**

Entidade: DISEASE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_disease	Determinante	Texto	36	Chave Primária da tabela de doenças.
Name_en	Simples	Texto	100	Nome em inglês da doença.
Name_pt	Simples	Texto	100	Nome em português da doença.

Fonte: Elaborado pelo autor**Tabela 47 - Dicionário de dados da entidade ERROR_LOG**

Entidade: ERROR_LOG				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_error	Determinante	Texto	36	Chave Primária da tabela de logs de erros.
Event_date	Simples	Data		Data em que ocorreu o log.
Message	Simples	Texto		Mensagem do erro ocorrido.
Name_of_file	Simples	Texto	100	Nome do arquivo que gerou o erro.
Class_name	Simples	Texto	100	Nome da classe que gerou o erro.
Method_name	Simples	Texto	200	Método que gerou o erro.
Line_number	Simples	Numérico		Número da linha aonde ocorreu o erro.
additional	Simples	Texto		Informações adicionais sobre o erro.
Id_user	Simples	Texto	36	Código do usuário que gerou o erro.

Fonte: Elaborado pelo autor

Tabela 48 - Dicionário de dados da entidade EXAM

Entidade: EXAM				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_exam	Determinante	Texto	36	Chave Primária do exame.
Date_exam	Simples	Data		Data em que foi realizado o exame.
Anotation	Simples	Texto		Anotações do exame.
Id_patient	Simples	Texto	36	Código do paciente que realizou o exame.
Id_physician	Simples	Texto	36	Código do médico que solicitou o exame.
Id_health_institution	Simples	Texto	36	Código da instituição de saúde responsável pelo exame.

Tabela 49 - Dicionário de dados da entidade EXAM_ATTACHMENT

Entidade: EXAM_ATTACHMENT				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_exam_attachment	Determinante	Texto	36	Chave primária da tabela de anexos do exame.
Attachment_name	Simples	Texto	100	Nome do anexo.
Id_exam	Simples	Texto	36	Código do exame que possui o anexo.

Fonte: Elaborado pelo autor

Tabela 50 - Dicionário de dados da entidade HEALTH_INSTITUTION

Entidade: HEALTH_INSTITUTION				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_health_institution	Determinante	Texto	36	Chave Primária da tabela de Instituições de Saúde
Identity_code	Simples	Texto	20	Código de identificação da instituição.
Name	Simples	Texto	100	Nome da instituição.
Postal_code	Simples	Texto	8	Código postal da instituição.
Country	Simples	Texto	3	Pais da instituição.
State	Simples	Texto	100	Estado/Província da instituição.
City	Simples	Texto	100	Cidade da instituição.
Street	Simples	Texto	100	Rua em que fica localizada a instituição.
Neighborhood	Simples	Texto	50	Bairro.
Number	Simples	Texto	10	Número da instituição.
Photo	Simples	Texto	100	Foto da instituição.

Latitude	Simples	Numérico		Latitude da localização.
Longitude	Simples	Numérico		Longitude da localização.
Telephone	Simples	Texto	20	Telefone da instituição.

Fonte: Elaborado pelo autor

Tabela 51 - Dicionário de dados da entidade MEDICINE

Entidade: MEDICINE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_medicine	Determinante	Texto	36	Chave Primária do medicamento.
Name	Simples	Texto	100	Nome do medicamento.
Language	Simples	Texto	3	Linguagem que foi utilizada para inserir o medicamento (Inglês ou Português).
Country	Simples	Texto	3	Pais que foi inserido o medicamento.
Status	Simples	Texto	100	Status do medicamento (Ativo/Inativo).

Fonte: Elaborado pelo autor

Tabela 52 - Dicionário de dados da entidade NURSE

Entidade: NURSE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_nurse	Determinante	Texto	36	Chave Primária da tabela de enfermeira.
Id_user	Simples	Texto	36	Código do usuário que foi cadastrado como enfermeira.
Nurse_code	Simples	Texto	15	Código da enfermeira.
Nurse_type	Simples	Texto	2	Tipo de enfermeira.

Fonte: Elaborado pelo autor

Tabela 53 – Dicionário de dados da entidade NURSE_HAS_SPECIALIZATION

Entidade: NURSE_HAS_SPECIALIZATION				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_nurse_has_specialization	Determinante	Texto	36	Chave Primária da tabela de especialização da enfermeira.
Id_nurse	Simples	Texto	36	Código da enfermeira.
Id_specialization	Simples	Texto	36	Código da especialização atribuída a especialização da enfermeira.

Fonte: Elaborado pelo autor

Tabela 54 - Dicionário de dados da entidade NURSE_SPECIALIZATION

Entidade: NURSE_SPECIALIZATION				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_specialization	Determinante	Texto	36	Chave Primária da especialização.
Name_pt	Simples	Texto	100	Nome em português da especialização.
Name_en	Simples	Texto	100	Nome em inglês da especialização.
Country	Simples	Texto	3	País que possui a especialização em específico.

Fonte: Elaborado pelo autor**Tabela 55 - Dicionário de dados da entidade PATIENT**

Entidade: PATIENT				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_patient	Determinante	Texto	36	Chave Primária da tabela de paciente (código do paciente).
Blood_type	Simples	Texto	2	Tipo sanguíneo do usuário.
Color	Simples	Texto	2	Cor da pele do usuário.
Father_name	Simples	Texto	100	Nome do pai.
Mother_name	Simples	Texto	100	Nome da mãe.
Weight	Simples	Numérico		Peso do usuário.
Height	Simples	Numérico		Altura do usuário.
Id_user	Simples	Texto	36	Código do usuário.
Status	Simples	Texto	1	Status dizendo se o paciente está ativo ou inativo.

Fonte: Elaborado pelo autor**Tabela 56 - Dicionário de dados da entidade PATIENT_HAS_DISEASE**

Entidade: PATIENT_HAS_DISEASE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_patient_has_disease	Determinante	Texto	36	Chave Primária da tabela de doenças do paciente.
Id_patient	Simples	Texto	36	Código do paciente que possui a doença.
Id_disease	Simples	Texto	36	Código da doença possuída.
Anotações	Simples	Texto		Anotações relativa a doença do paciente.

Fonte: Elaborado pelo autor**Tabela 57 - Dicionário de dados da entidade PATIENT_HAS_TAG**

Entidade: PATIENT_HAS_TAG

Atributo	Classe	Domínio	Tamanho	Descrição
Id_patient_has_tag	Determinante	Texto	36	Chave Primária da tabela de <i>tag</i> do paciente.
Mac_code	Simples	Texto	36	Código MAC da <i>tag NFC</i> .
Id_patient	Simples	Texto	36	Código do paciente que possui o equipamento de acesso.
Tag_type	Simples	Texto	2	Tipo do equipamento de acesso.
Name	Simples	Texto	100	Nome do equipamento de acesso.

Fonte: Elaborado pelo autor

Tabela 58 - Dicionário de dados da entidade PATIENT_USE_MEDICINE

Entidade: PATIENT_USE_MEDICINE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_patient_use_medicine	Determinante	Texto	36	Chave Primária da tabela de medicamentos do paciente.
Id_patient	Simples	Texto	36	Código do paciente.
Id_medicine	Simples	Texto	36	Código do medicamento.

Fonte: Elaborado pelo autor

Tabela 59 - Dicionário de dados da entidade PHYSICIAN

Entidade: PHYSICIAN				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_physician	Determinante	Texto	36	Chave Primária da tabela de médicos.
Id_user	Simples	Texto	36	Código do usuário que foi cadastrado como médico.
Practice_document	Simples	Texto	20	Código de licença do médico.

Fonte: Elaborado pelo autor

Tabela 60 - Dicionário de dados da entidade PHYSICIAN_ATTENDANCE

Entidade: PHYSICIAN_ATTENDANCE				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_physician_attendance	Determinante	Texto	36	Chave Primária da tabela de atendimento médico.
Date_attendance	Simples	Data		Data em que ocorreu o atendimento.
Id_patient	Simples	Texto	36	Código do paciente que foi atendido.
Id_physician	Simples	Texto	36	Código do médico que realizou o atendimento.

Id_health_institution	Simples	Texto	36	Código da instituição de saúde onde foi realizado o atendimento.
-----------------------	---------	-------	----	--

Fonte: Elaborado pelo autor

Tabela 61 - Dicionário de dados da entidade PHYSICIAN_HAS_SPECIALIZATION

Entidade: PHYSICIAN_HAS_SPECIALIZATION				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_physician_has_specialization	Determinante	Texto	36	Chave Primária da especialização que o médico possui.
Id_physician	Simples	Texto	36	Código do médico que possui a especialização.
Id_specialization	Simples	Texto	36	Código da especialização possuída.

Fonte: Elaborado pelo autor

Tabela 62 - Dicionário de dados da entidade PHYSICIAN_SPECIALIZATION

Entidade: PHYSICIAN_SPECIALIZATION				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_specialization	Determinante	Texto	36	Chave Primária da especialização do médico.
Name_pt	Simples	Texto	100	Nome em português da especialização.
Name_en	Simples	Texto	100	Nome em inglês da especialização.
Country	Simples	Texto	3	Local que existe a especialização.

Fonte: Elaborado pelo autor

Tabela 63 - Dicionário de dados da entidade USER_AND_HEALTH_INSTITUTION

Entidade: USER_AND_HEALTH_INSTITUTION				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_user_works_in_health_institution	Determinante	Texto	36	Chave Primária do usuário da instituição.
Id_user	Simples	Texto	36	Código do usuário que pertence a instituição.
Id_health_institution	Simples	Texto	36	Código da instituição.
Status	Simples	Texto	1	Estado do usuário na instituição (ativo/inativo).

Fonte: Elaborado pelo autor

Tabela 64 - Dicionário de dados da entidade USER_HEALTH

Entidade: USER_HEALTH				
Atributo	Classe	Domínio	Tamanho	Descrição
Id_user	Determinante	Texto	36	Chave Primária da tabela de usuário.
Login	Simples	Texto	100	Login para acesso aos sistemas.
Password	Simples	Texto	64	Senha utilizada para acesso aos sistemas.
Name	Simples	Texto	100	Nome do usuário.
Type_of_user	Simples	Texto	2	Tipo de usuário (Admin, Médico, Paciente e outros).
Born_date	Simples	Data		Data de nascimento.
Gender	Simples	Texto	1	Gênero do usuário.
Postal_code	Simples	Texto	8	Código postal do usuário.
Country	Simples	Texto	3	País do usuário.
State	Simples	Texto	100	Estado do usuário.
City	Simples	Texto	100	Cidade do usuário.
Street	Simples	Texto	100	Rua do Usuário.
Neighborhood	Simples	Texto	50	Bairro.
Number	Simples	Texto	10	Número da residência do usuário.
Photo	Simples	Texto	10	Foto do usuário.
Identity_document	Simples	Texto	36	Documento de identidade do usuário.
Telephone	Simples	Texto	20	Telefone do usuário.
Secret_code	Simples	Texto	4	Código secreto do usuário.

Fonte: Elaborado pelo autor**Tabela 65 - Dicionário de dados da entidade PATIENT_USES_MEDICINE do SQLite**

Entidade: PATIENT_USES_MEDICINE				
Atributo	Classe	Domínio	Tamanho	Descrição
id_patient_uses_medicine	Determinante	Texto	36	Chave Primária da tabela.
id_user	Simples	Texto	36	Identificador do usuário.
remember	Simples	Numérico		Indica se o determinado medicamento deve ou não ser lembrado.
name	Simples	Texto		Nome do medicamento.
sunday	Simples	Texto		Verifica se é necessário alertar o usuário sobre o medicamento no Domingo.
monday	Simples	Numérico		Verifica se é necessário alertar o usuário sobre o medicamento na Segunda-feira.

tuesday	Simples	Numérico		Verifica se é necessário alertar o usuário sobre o medicamento na Terça-feira.
wednesday	Simples	Numérico		Verifica se é necessário alertar o usuário sobre o medicamento na Quarta-feira.
thursday	Simples	Numérico		Verifica se é necessário alertar o usuário sobre o medicamento na Quinta-feira.
friday	Simples	Numérico		Verifica se é necessário alertar o usuário sobre o medicamento na Sexta-feira.
saturday	Simples	Numérico		Verifica se é necessário alertar o usuário sobre o medicamento no Sábado.
id_medicine	Simples	Texto	36	Identificador do medicamento.

Fonte: Elaborado pelo autor

Tabela 66 - Dicionário de dados da entidade PATIENT_USES_MEDICINE_IN_HOUR do SQLite

Entidade: PATIENT_USES_MEDICINE_IN_HOUR				
Atributo	Classe	Domínio	Tamanho	Descrição
id_patient_uses_medicine_in_hour	Determinante	Texto	36	Chave Primária da tabela.
hour	Simples	Numérico		Indica o horário que o medicamento deve ser consumido.
id_patient_uses_medicine	Simples	Texto	36	Identificador da tabela PATIENT_USES_MEDICINE

Fonte: Elaborado pelo autor

3.5.3 Circuito do sistema arduino

O circuito do sistema Arduino representa uma importante função no sistema, realizando a função de identificação do paciente através de seu respectivo equipamento de acesso, permitindo uma maior velocidade na obtenção dos dados de saúde do paciente, e sendo um importante aspecto de segurança. A Tabela 67 apresenta os componentes utilizados, juntamente aos preços.

Tabela 67 - Componentes utilizados

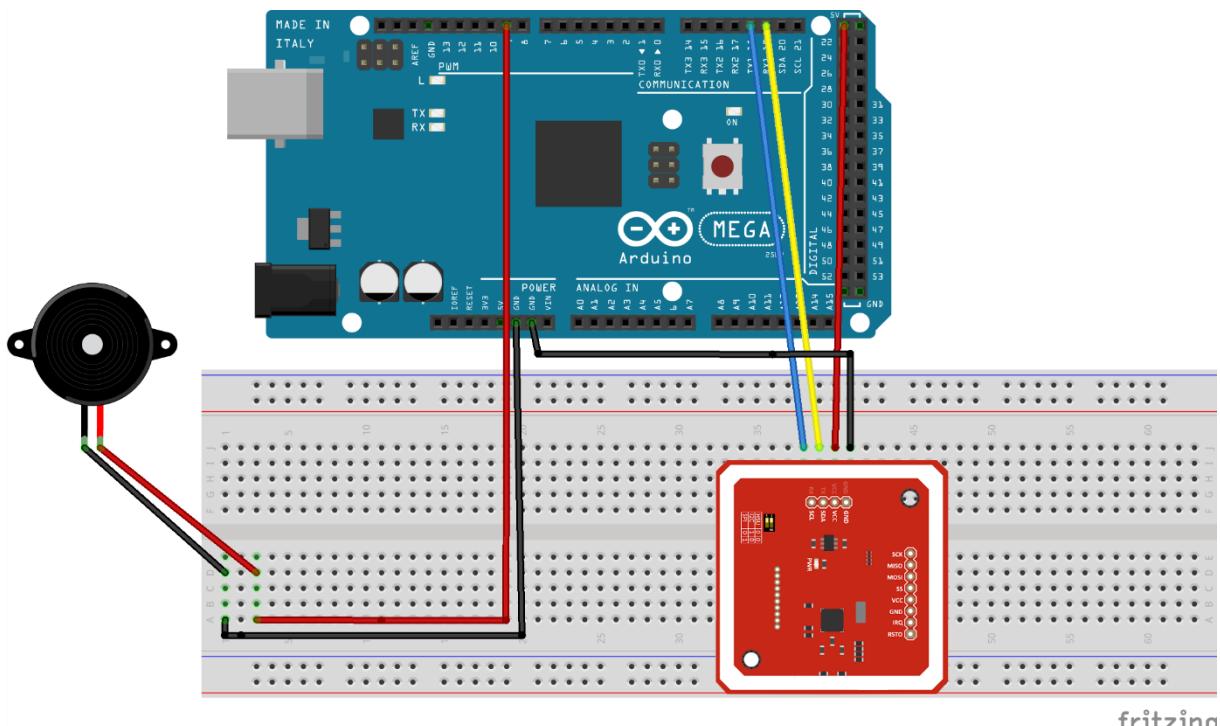
Nome do componente	Preço (R\$)
Arduino Mega	R\$ 94,90
Kit módulo leitor RFID NFC PN532	R\$ 117,90

Kit Jumpers Macho-Macho x65 unidades	R\$ 12,90
Protoboard 830 Pontos	R\$ 16,90
Buzzer ativo 5V	R\$ 3,40

Fonte: Elaborado pelo autor

A Figura 50 apresenta o circuito utilizado. Pode-se notar a presença do Arduino Mega, do módulo PN532 e de um Buzzer. O módulo PN532 é utilizado para obter o código de identificação do equipamento de acesso, e no momento que isto ocorrer, o *buzzer* irá alertar o usuário através do som.

Figura 50 - Circuito



Fonte: Elaborado pelo autor

A Tabela 68 demonstra as ligações entre o Arduino e o módulo PN532. Deve-se notar que dentre os módulos de comunicação disponíveis, estão HSU, I2C e SPI. Foi utilizado o modo de comunicação HSU na construção do sistema.

Tabela 68 - Ligações entre o Arduino e o módulo PN532

Arduino	PN532
GND	GND
5V	VCC
18 – TX1	SCL
19 – RX1	SDA

Fonte: Elaborado pelo autor

A Figura 51 demonstra o início do código utilizado no sistema. Primeiramente, é definido a interface de comunicação que será utilizada para realizar a leitura do equipamento. Por padrão, é utilizada a interface HSU.

Figura 51 - Definição da interface de comunicação

```

1  /* Definição da interface de comunicação*/
2  #if 0
3      #include <SPI.h>
4      #include <PN532_SPI.h>
5      #include "PN532.h"
6
7      PN532_SPI pn532spi(SPI, 10);
8      PN532 nfc(pn532spi);
9
10 #elif 1
11     #include <PN532_HSU.h>
12     #include <PN532.h>
13
14     PN532_HSU pn532hsu(Serial1);
15     PN532 nfc(pn532hsu);
16
17 #else
18     #include <Wire.h>
19     #include <PN532_I2C.h>
20     #include <PN532.h>
21     #include <NfcAdapter.h>
22
23     PN532_I2C pn532i2c(Wire);
24     PN532 nfc(pn532i2c);
25 #endif
26

```

Fonte: Elaborado pelo autor

A Figura 52 demonstra o código do *setup* inicial do sistema. Inicialmente, o *NFC* é inicializado, e tenta-se obter uma versão do *firmware* do mesmo a fim de verificar se o módulo PN532 foi encontrado. Em seguida, é configurado para que a placa tente um número máximo de tentativas até conseguir realizar a leitura do equipamento, prevenindo com que o sistema espere para sempre pelo equipamento de acesso (esse é o comportamento padrão do PN532). Por fim, se configura a placa para a realização de leitura dos equipamentos.

Figura 52 - Setup

```

27 void setup(void) {
28
29     Serial.begin(9600);
30     pinMode(9,OUTPUT);
31
32     nfc.begin();
33
34     uint32_t versiondata = nfc.getFirmwareVersion();
35     if (!versiondata) {
36         //PN532 não encontrado
37         while (1); // halt
38     }
39
40     //Setta um numero máximo de tentativas para a leitura do cartão
41     nfc.setPassiveActivationRetries(0xFF);
42
43     //configura a placa para leitura das tags
44     nfc.SAMConfig();
45
46 }
```

Fonte: Elaborado pelo autor

A Figura 53 realiza o *looping* de leitura dos equipamentos. Primeiro é declarado uma estrutura para armazenar o *UID* (identificador único) do equipamento, juntamente com o tamanho do mesmo. Em seguida, tenta-se realizar a leitura e impressão do código *UID*.

Figura 53 - Loop

```

48 void loop(void) {
49     boolean success;
50     uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 }; // Buffer para armazenar o UID
51     uint8_t uidLength; // Tamanho do UID (4 ou 7 bytes)
52
53     //Aguarda para a leitura. Quando a mesma for realizada, o UID será populado,
54     //e o uidLength vai indicar se o uid possui 4 ou 7 bytes.
55     success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength);
56
57     if (success) {
58         //Alerta o usuário
59         tone(9,1500, 500);
60
61         //Realiza a impressão do uid
62         for (uint8_t i=0; i < uidLength; i++)
63         {
64             Serial.print("0x");Serial.print(uid[i], HEX);
65             if(i != uidLength - 1){
66                 Serial.print("-");
67             }
68         }
69         Serial.println("");
70
71         delay(1000);
72     }
73     else
74     {
75         // Time out na leitura do cartão
76         noTone(9);
77     }
78 }
```

Fonte: Elaborado pelo autor

4 DESENVOLVIMENTO

Ao decorrer do projeto, será utilizado o SCRUM. SCRUM é um *framework* utilizado para o gerenciamento de produtos complexos, onde é possível empregar diversos processos e técnicas (SCRUM, 2013). No SCRUM, os times são compostos pelo *Product owner* (Universidades parceiras), time de desenvolvimento (Alunos) e o *Scrum master* (Orientador), sendo que os times são multifuncionais e auto organizáveis. O *Product owner* é responsável pelo produto em si, representando os interesses dos *stakeholders* para o time SCRUM (VMEDU INC, 2016). O time de desenvolvimento são profissionais que entregam uma versão utilizável que incrementam o produto ao fim de cada *sprint* (SCRUM, 2013). Já o *Scrum master* é quem garante que o SCRUM seja compreendido e aplicado (SCRUM, 2013), facilitando a interação do time, sendo um motivador e mentor do time (VMEDU INC, 2016).

O ciclo das *sprints* (tempo no qual uma versão incremental utilizável do produto será criada) devem durar um mês ou menos, e uma nova *sprint* será criada ao término da anterior (SCRUM, 2013). Todo trabalho que deve ser realizado deverá ser planejado na reunião de planejamento da *sprint*, e serão feitas reuniões diárias de 15 minutos para a sincronização das atividades e criação de planos para as próximas 24 horas (SCRUM, 2013).

Ao término de uma *sprint*, será realizada uma revisão da *sprint*, sendo esta uma reunião de 4 horas para um mês (VMEDU INC, 2016), variando a quantidade de horas dependendo do tamanho da *sprint* (SCRUM, 2013). Retrospectivas da *sprint* visando criar um plano de melhorias (SCRUM, 2013) para as próximas *sprints* também serão realizadas.

Durante o desenvolvimento do projeto, alguns artefatos que estarão presentes serão o *Backlog* do produto e da *sprint*. O *Backlog* do produto é uma lista ordenada de tudo o que deve ser feito no produto (SCRUM, 2013), sendo que este deve ser continuamente refinado pelo *Product owner* (VMEDU INC, 2016). Já o *Backlog* da *sprint* são os itens do *Backlog* de produto que foram selecionados para a *sprint*, visando entregar uma versão incremental do produto com estes itens para atingir o objetivo da *sprint* (SCRUM, 2016).

Assim, com a utilização deste *framework*, será possível realizar a tomada de decisão baseada na observação e em experimentos ao invés de um planejamento

inicial detalhado (VMEDU INC, 2016). Portanto, ideias como transparência, inspeção e auto-organização estarão presentes no projeto.

4.1 Etapas de Desenvolvimento

Para cada entregável foi convencionado realizar a divisão de tarefas de acordo com a similaridade entre elas. Foi decidido que cada entrega teria um prazo máximo de 15 dias. Resumidamente, as entregas ocorrem da seguinte forma.

- **Entrega 1:** Início do desenvolvimento (criação de telas e APIs);
- **Entrega 2:** Criação de perfis de usuários, notificação, internacionalização, testes e correções de bugs da primeira entrega.
- **Entrega 3:** Correção de bugs de perfil, criação de APIs (checar usuário), alterações nas funcionalidades já desenvolvidas e melhorias nas mesmas.
- **Entrega 4:** Correções, criação de listagens (exames e diagnósticos) e melhorias.
- **Entrega 5:** Tradução para o inglês de especializações médicas, finalização do desktop, criação de slides para o Canadá e melhorias voltadas para o aplicativo mobile.
- **Entrega 6:** Atualização do desktop para *Internal Frame*, criação de trilha de auditoria, correções e bugs e adição de funcionalidades no aplicativo do paciente.
- **Entrega 7:** Adição de funcionalidades para o aplicativo da enfermeira, inicio da melhoria do design, tradução de *strings*, padronização, melhorias e testes.
- **Entrega 8:** Entrega dos aplicativos e sistema desktop, criação dos manuais (Paciente e Enfermeira) e correções de bugs.

4.1.1 Entrega 1

No dia 1 de julho de 2018 o grupo se encontrou para realizar o planejamento da primeira entrega (15 dias de desenvolvimento – prazo 16 de julho de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a criação de telas iniciais do aplicativo e do sistema desktop, e criação de APIs (localização, autenticação e CRUD de instituição). A Tabela 69 apresenta

detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 69 -Planejamento realizado da primeira entrega

Atividade	Tempo (em dias)	Pontos
Criação de tela de Login app paciente	1	5
Criação de tela de <i>Splash</i> app paciente	1	5
Criação de tela de Home app paciente	1	5
Criação de tela de Prontuário app paciente	1	5
Criação de tela de Instituições app paciente	1	5
Criação de tela de Médicos app paciente	1	5
Criação de tela de Histórico médico app paciente	1	5
Criação de tela de Medicação app paciente	1	5
Criação de tela de Vacinação app paciente	1	5
<i>API</i> de localização	1	8
<i>API</i> de autenticação	1	8
<i>API</i> CRUD de instituição	2	9
Total	13	70

Fonte: Elaborado pelo autor

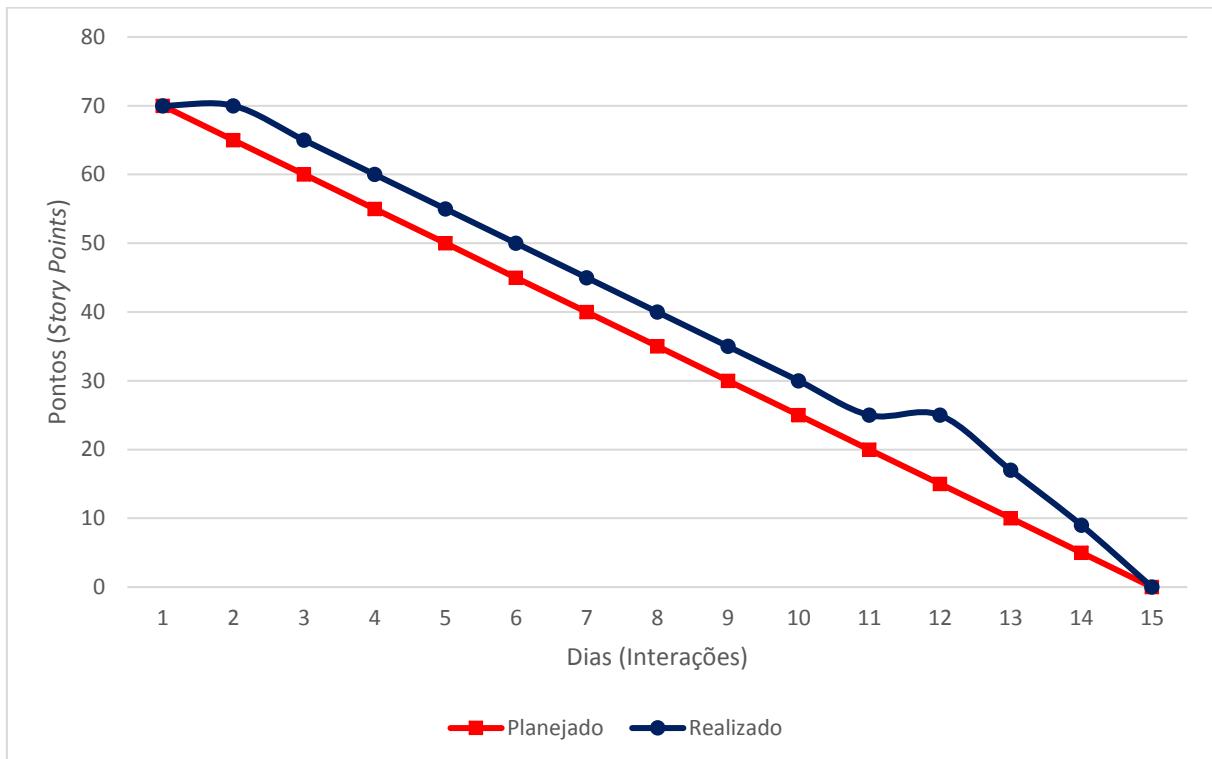
Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 54 apresenta o gráfico de *Burndown* da entrega 1, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que tudo ocorreu bem ao longo dos dias, não havendo nenhum problema.

No dia 16 de julho de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: 60% de telas a serem criadas foram bem-sucedidas, e *APIs* criadas com sucesso.
- O que deu errado: algumas telas não foram criadas devido à falta de requisitos que ainda não foram coletados (vacinação, histórico, etc) e também por dúvidas internas
- Ações de melhorias: criar uma forma de estabelecer mais contato para poder obter os requisitos e sanar dúvidas rapidamente, para que haja menos impacto no desenvolvimento do projeto.

- Observações: ainda não havia o contato com as equipes da África.

Figura 54 – Gráfico de Burndown da entrega 1.



Fonte: Elaborado pelo autor

4.1.2 Entrega 2

No dia 17 de julho de 2018 o grupo se encontrou para realizar o planejamento da segunda entrega (15 dias de desenvolvimento – prazo 01 de agosto de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a criação de perfis de usuário, criação de telas do app médico, e correções de bugs da primeira entrega. A Tabela 70 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 70 -Planejamento realizado da segunda entrega

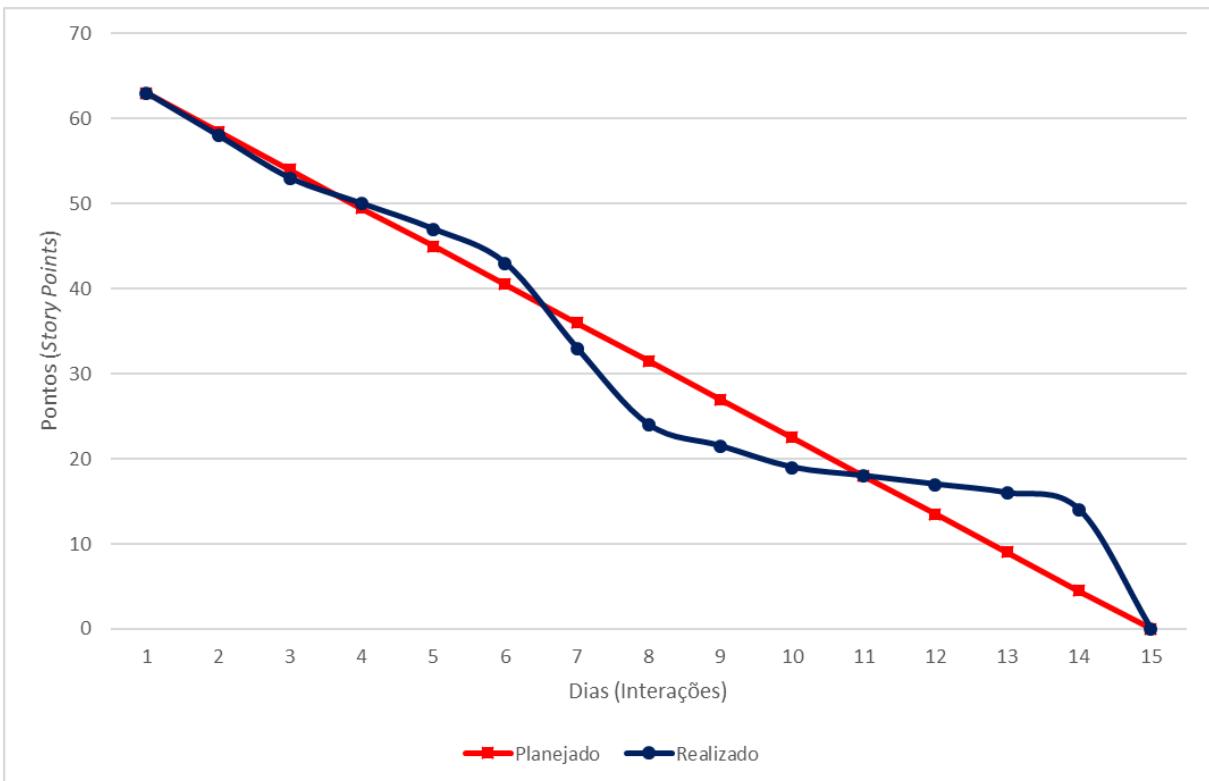
Atividade	Tempo (em dias)	Pontos
Criação do usuário médico	2	10
Criação do usuário administrador	3	10
Criação do usuário administrador da instituição médica	1	10
Notificação de atividade desktop	1	9
Criação de telas do app do médico	2	5
Internacionalização	4	5

Testes de <i>BUGS</i>	1	9
Criação de telas app paramédico	1	5
Total	15	63

Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 55 apresenta o gráfico de *Burndown* da entrega 2, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que tudo ocorreu bem ao longo dos dias, tendo uma queda brusca, porém o mesmo foi retomado concluindo assim a *sprint* com sucesso.

Figura 55 – Gráfico de *Burndown* da entrega 2.



Fonte: Elaborado pelo autor

No dia 01 de agosto de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: criação dos perfis de usuário, notificações, teste e internacionalização.
- O que deu errado: criação de telas do app do médico e do paramédico.
- Ações de melhorias: mais comunicação interna entre os integrantes do grupo para tomada de decisões mais rápidas e eficazes.

4.1.3 Entrega 3

No dia 1 de agosto de 2018 o grupo se encontrou para realizar o planejamento da terceira entrega (15 dias de desenvolvimento – prazo 15 de agosto de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a correção de *bugs*, API de cadastros, traduções e correções em geral no sistema. A Tabela 71 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 71 -Planejamento realizado da terceira entrega

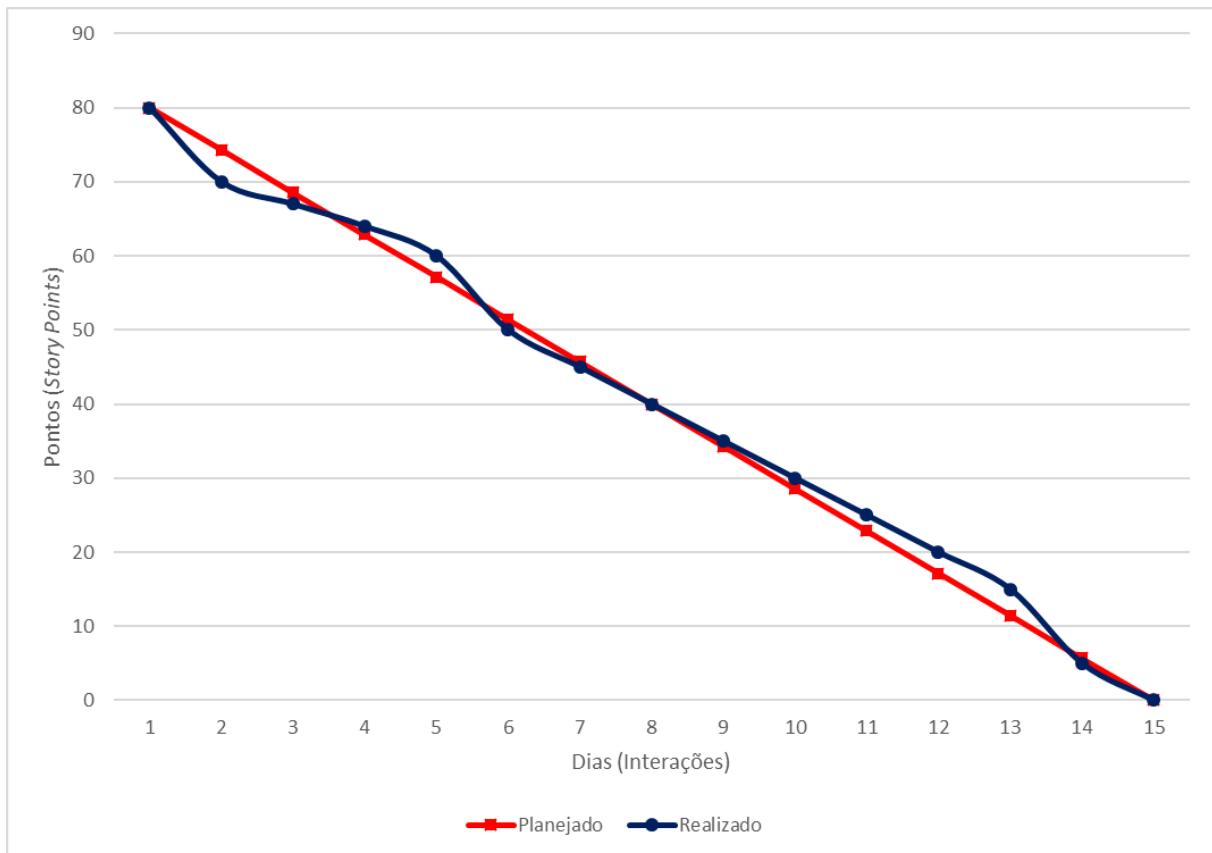
Atividade	Tempo (em dias)	Pontos
Correção de <i>BUGS</i> de perfil	0,5	5
API para cadastro Android (checagem de usuário)	3	10
Alterações em adicionar usuários	1	5
Correção do tamanho de anexos	0,5	5
Correção na adição de foto de assinatura digital	0,5	5
Correções de botões no <i>desktop</i>	0,5	5
Adição de pai e mãe biológicos	1	5
Visualização da especialização do médico em exames/diagnósticos	1	5
Tradução das especializações	1	5
Tela de carregamento	1	5
Ajustes de navegação do app	1	5
Verificação de CPF no cadastro de paciente do app	1	5
Adicionado <i>datepicker</i> no app	1	5
Melhoramento da navegação entre <i>fragments</i> e <i>activities</i> (<i>app</i>)	1	5
Puxar dados através do CEP (cadastro de paciente <i>app</i>)	1	5
Total	15	80

Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 56 apresenta o gráfico de

Burndown da entrega 3, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que durante o início da *sprint* houve um declínio, porém, o mesmo foi sanado e mantido durante o acontecimento da mesma.

Figura 56 – Gráfico de *Burndown* da entrega 3.



Fonte: Elaborado pelo autor

No dia 15 de agosto de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: todo o planejado da *sprint*.
- O que deu errado: nada.

4.1.4 Entrega 4

No dia 15 de agosto de 2018 o grupo se encontrou para realizar o planejamento da quarta entrega (15 dias de desenvolvimento – prazo 30 de agosto de 2018). Nesta

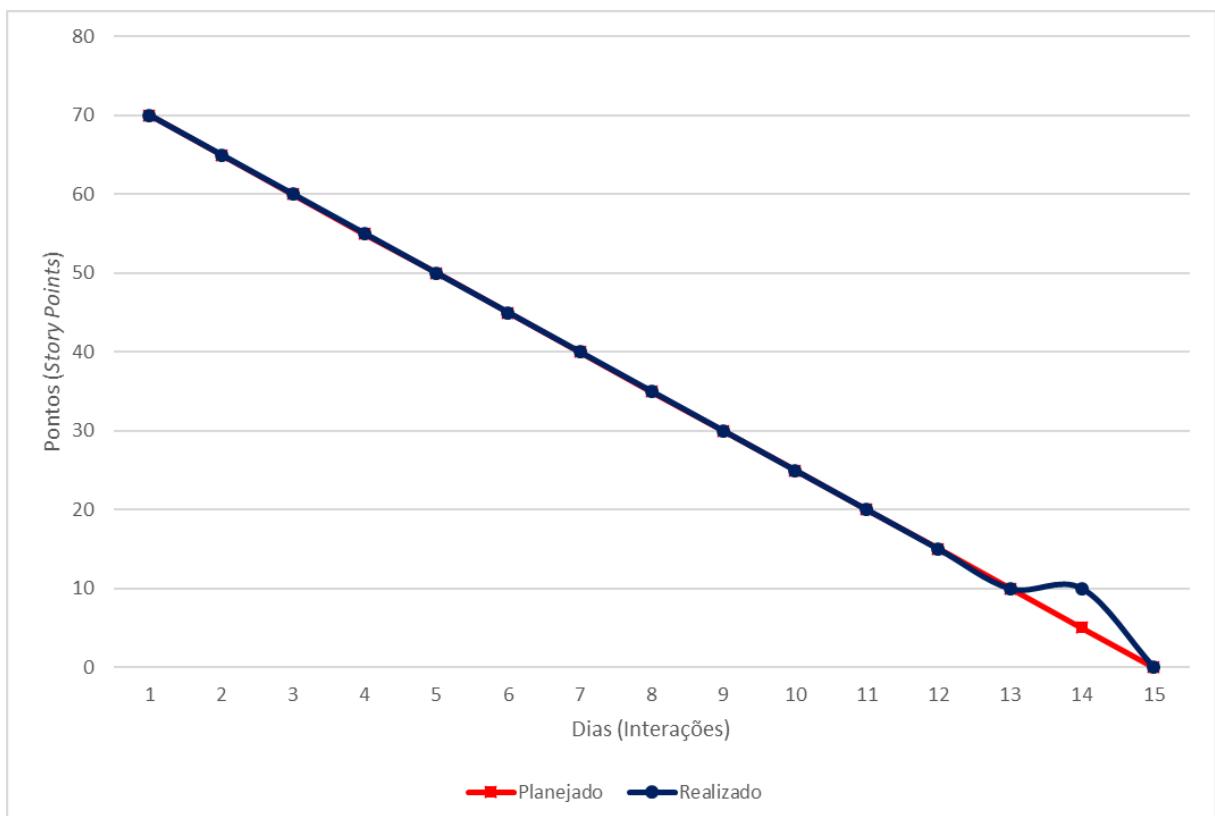
reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a correções, listagens e melhorias (atualizações). A Tabela 72 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 72 - Planejamento realizado da quarta entrega

Atividade	Tempo (em dias)	Pontos
Correções no salvamento de anexos de exames	1	5
Correções gerais do sistema e testes	1	5
Adicionar usuário paciente	2	10
Listagem de diagnósticos do médico no hospital	2	10
Listar exames realizados pelo médico no hospital	2	10
Correções de dados de saúde do médico	1	5
Correções no funcionamento do sensor	2	10
Atualização do ícone de espera do NFC para carregando	1	5
<i>Datapicker</i> inserido no app ao selecionar data de nascimento do usuário	1	5
Verificação de usuário já existente no app	1	5
Total	14	70

Fonte: Elaborado pelo autor

Figura 57 – Gráfico de Burndown da entrega 4.



Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 57 apresenta o gráfico de *Burndown* da entrega 4, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que durante a sprint o nível de planejado X realizado se manteve constante, tendo um leve aumento no planejado no fim da sprint.

No dia 30 de agosto de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: todos os itens planejados ocorreram sem problemas.
- O que deu errado: a verificação de usuário existente no aplicativo.
- Ações de melhorias: melhoria na comunicação interna e na modelagem das funcionalidades da aplicação.

4.1.5 Entrega 5

No dia 31 de agosto de 2018 o grupo se encontrou para realizar o planejamento da quinta entrega (15 dias de desenvolvimento – prazo 14 de setembro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a traduções de especializações médicas, finalização do desktop, criação de controle de medicamentos do aplicativo do paciente, correção de bugs do app. A Tabela 73 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

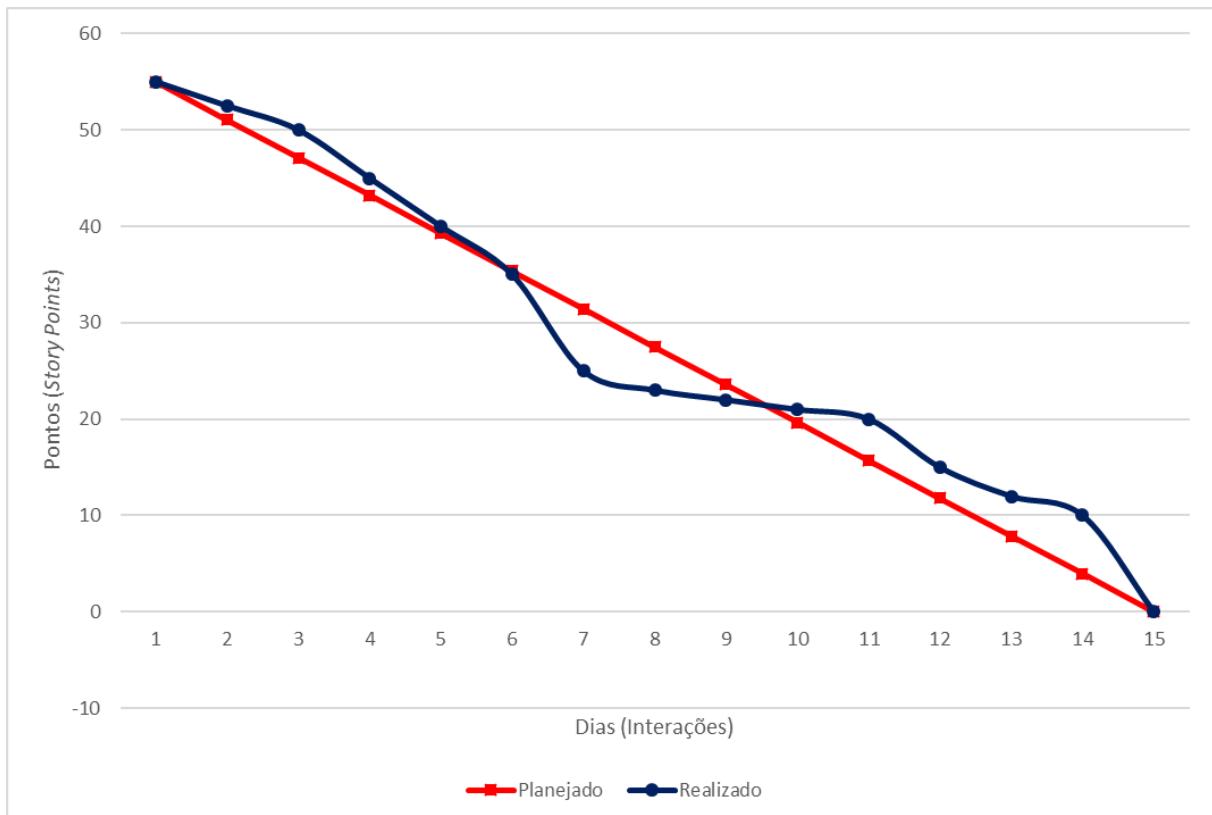
Tabela 73 -Planejamento realizado da quinta entrega

Atividade	Tempo (em dias)	Pontos
Tradução das especializações médicas	2	5
Finalização do <i>desktop</i>	3	15
Criação de controle de medicamentos do aplicativo paciente	5	15
Criação de <i>slide</i> para do Canadá	3	10
Correção de bugs do app	2	10
Total	15	55

Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 58 apresenta o gráfico de *Burndown* da entrega 5, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que durante a *sprint* houve uma queda bem grande no progresso do realizado, porém, após alguns dias houve o aumento do mesmo novamente.

Figura 58 – Gráfico de *Burndown* da entrega 5.



Fonte: Elaborado pelo autor

No dia 14 de setembro de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: todo o planejado da *sprint*.
- O que deu errado: falta de definição do que ser feito.
- Ações de melhorias: melhoria na maneira de como é definido as tarefas de cada membro do grupo.

4.1.6 Entrega 6

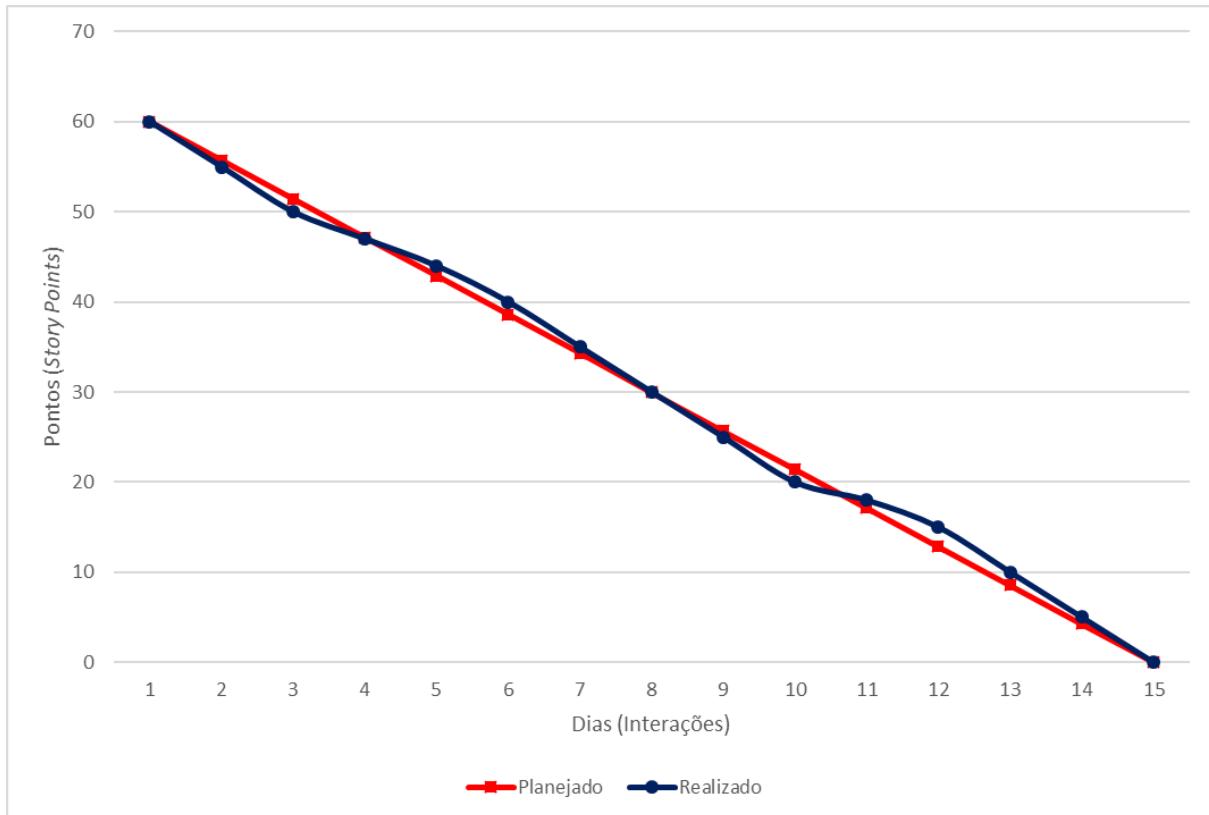
No dia 15 de setembro de 2018 o grupo se encontrou para realizar o planejamento do sexto entregável (15 dias de desenvolvimento – prazo 29 de setembro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a ajustes no desktop, auditoria e melhorias. A Tabela 74 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 74 -Planejamento realizado da sexta entrega

Atividade	Tempo (em dias)	Pontos
Modificação do desktop para <i>Internal Frame</i>	2	10
Criação da trilha de auditoria (normal e erros)	3	10
Correção das cores do aplicativo paciente	1	5
Adição da listagem de médicos no aplicativo paciente	3	15
Adição da visualização de exames no aplicativo paciente	3	10
Criação da Ata de reunião (07/06/2018)	1	5
Correção de vídeos para o <i>slide</i> do Canadá	2	5
Total	15	60

Fonte: Elaborado pelo autor

Figura 59 – Gráfico de *Burndown* da entrega 6.



Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 59 apresenta o gráfico de *Burndown* da entrega 6, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que durante a *sprint* o nível desejado de entrega foi bem estável, ocorrendo alguns picos, tanto para baixo quanto para cima.

No dia 29 de setembro de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: todos os itens planejados, exceto a correção de cores e criação da ata de reunião.
- O que deu errado: correção de cores e criação da ata de reunião.
- Ações de melhorias: anotações de tudo o que foi discutido em reuniões e centralizado em uma única pessoa a parte de design.

4.1.7 Entrega 7

No dia 30 de setembro de 2018 o grupo se encontrou para realizar o planejamento do sétimo entregável (15 dias de desenvolvimento – prazo 14 de outubro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a criação de funções para o aplicativo da enfermeira, internacionalização do app paciente, correção de bugs e testes. A Tabela 75 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 75 -Planejamento realizado da sétima entrega

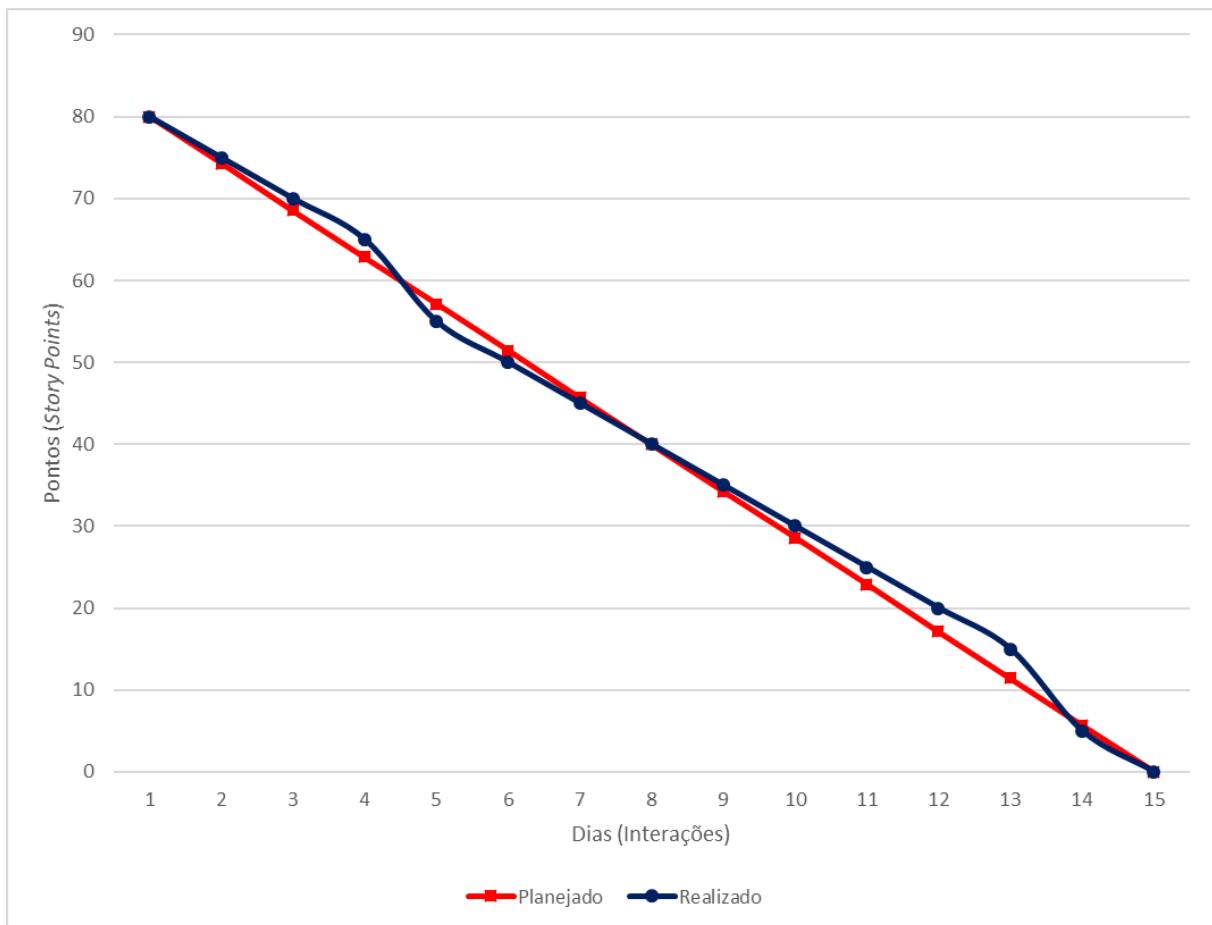
Atividade	Tempo (em dias)	Pontos
Criação da listagem de paciente do aplicativo do(a) enfermeiro(a)	3	15
Criação da listagem de hospitais vinculados do aplicativo do(a) enfermeiro(a)	2	15
Criação da adição de procedimentos do aplicativo do(a) enfermeiro(a)	3	15
Design	1	5
Tradução das <i>Strings</i> do aplicativo paciente	1	5

Correção da validação de conexão do aplicativo paciente	1	5
Alteração de foto de usuário do aplicativo paciente	1	5
Padronização do aplicativo paciente	1	5
Tratamento de erros do aplicativo paciente	1	5
Testes	1	5
Total	15	80

Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 60 apresenta o gráfico de *Burndown* da entrega 7, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que durante a *sprint* o nível desejado de entrega foi bem estável, não ocorrendo quedas bruscas.

Figura 60 – Gráfico de *Burndown* da entrega 7.



Fonte: Elaborado pelo autor

No dia 14 de outubro de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião,

a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: criação da aplicação da enfermeira, internacionalização, padronização do aplicativo do paciente.
- O que deu errado: alteração de foto do usuário no aplicativo do paciente, design.
- Ações de melhorias: melhoria na comunicação interna do grupo.

4.1.8 Entrega 8

No dia 15 de outubro de 2018 o grupo se encontrou para realizar o planejamento do sétimo entregável (15 dias de desenvolvimento – prazo 29 de outubro de 2018). Nesta reunião os membros definiram as atividades e seus níveis de dificuldades – representados por pontos. De modo geral, essas atividades estão relacionadas a correções, entrega dos aplicativos (paciente, enfermeira), entrega do sistema desktop, criação do manual e design. A Tabela 75 apresenta detalhadamente as atividades, seu tempo de realização em dias e sua respectiva pontuação.

Tabela 76 -Planejamento realizado da oitava entrega

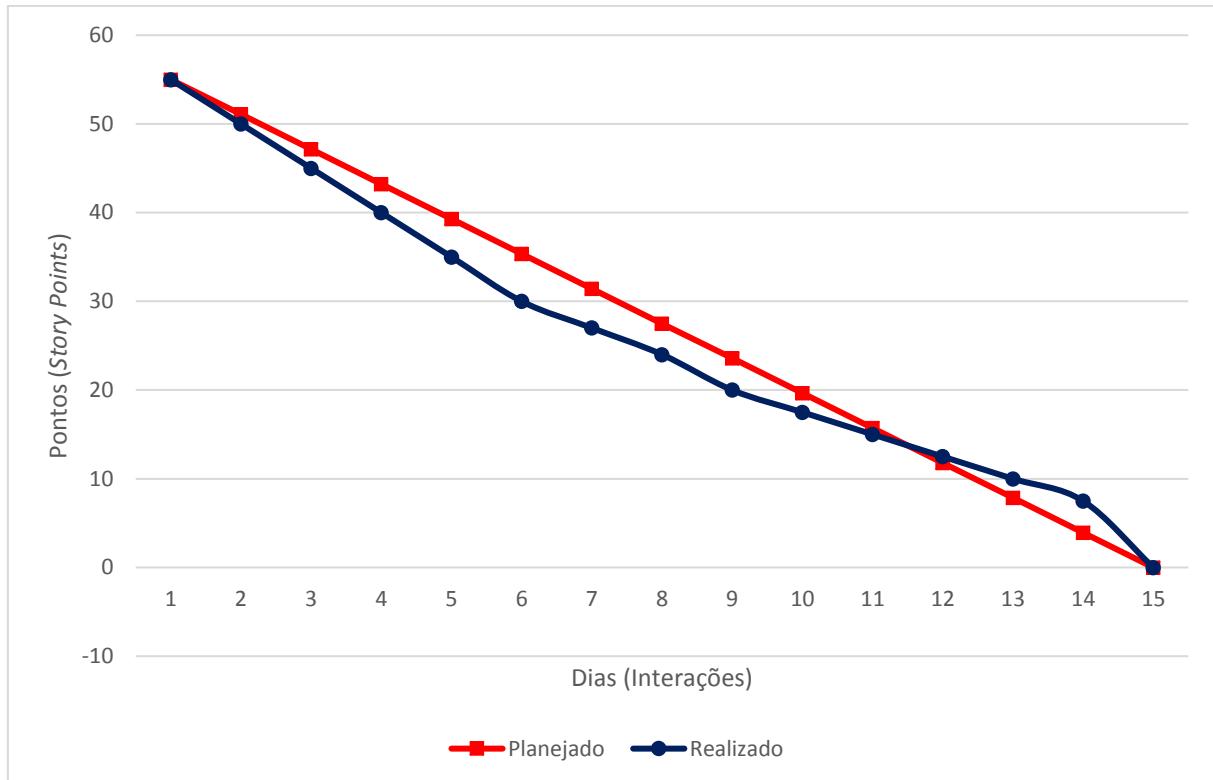
Atividade	Tempo (em dias)	Pontos
Correções nas notificações	2	10
Entrega do aplicativo da enfermeira	1	5
Entrega do aplicativo paciente	1	5
Entrega do <i>desktop</i>	1	5
Padronização de botões	1	5
Atualizações de design	3	10
Criação do manual Paciente	2	5
Criação do manual Enfermeira	2	5
Criação do manual desktop	2	5
Total	15	55

Fonte: Elaborado pelo autor

Durante os 15 dias de desenvolvimento, a equipe realizou baixas na pontuação conforme as atividades eram concluídas. A Figura 61 apresenta o gráfico de

Burndown da entrega 8, destacando o planejamento de baixas nos pontos e a baixas realizadas pela equipe. Percebe-se que durante a *sprint* foi realizado com sucesso o que foi planejado para a mesma.

Figura 61 – Gráfico de *Burndown* da entrega 8.



Fonte: Elaborado pelo autor

No dia 29 de outubro de 2018 a equipe se reuniu para realizar a revisão do entregável. Nesta reunião todos os membros estavam presentes. Durante a reunião, a equipe fez uma autoavaliação procurando detectar os principais pontos de acertos e erros, e juntos, propor ações de melhorias para o desempenho da equipe nos próximos entregáveis. Ao final, a equipe fez o seguinte resumo sobre algumas questões:

- O que deu certo: correções, entrega dos aplicativos e sistema desktop, manuais.
- O que deu errado: design.
- Ações de melhorias: definição de padrões para o layout.

4.2 Interfaces de Usuário

Foram desenvolvidas diversas interfaces de usuário. Os subcapítulos abaixo são apresentados em uma visão operacional de utilização do sistema.

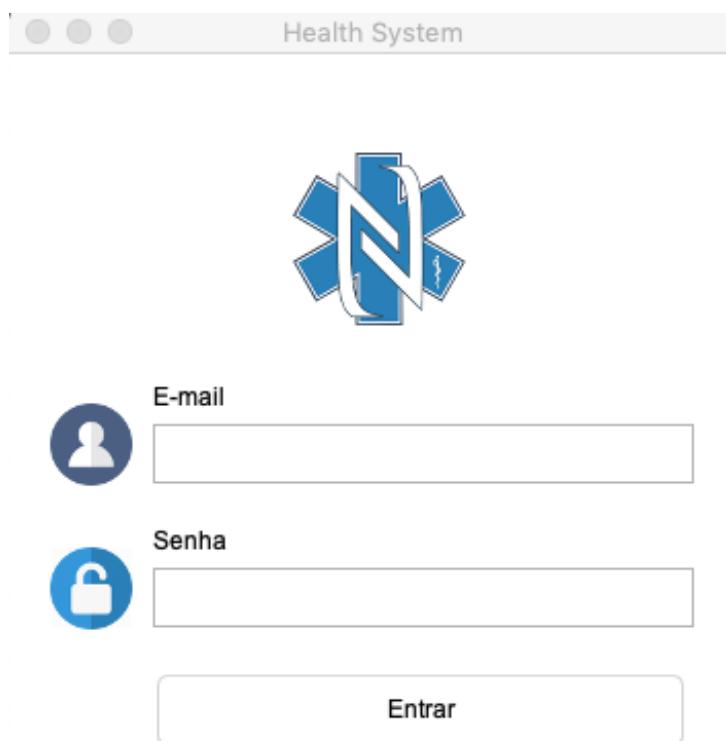
4.2.1 Sistema desktop

No sistema desktop existem três tipos de usuários, estes são: administrador do sistema, administrador da instituição e médico. Cada um destes usuários possui permissões e funções diferentes dentro do sistema.

4.2.1.1 Usuário administrador do sistema

Ao inicializar o sistema será exibida uma página de autenticação, nesta será solicitado o fornecimento do e-mail e senha.

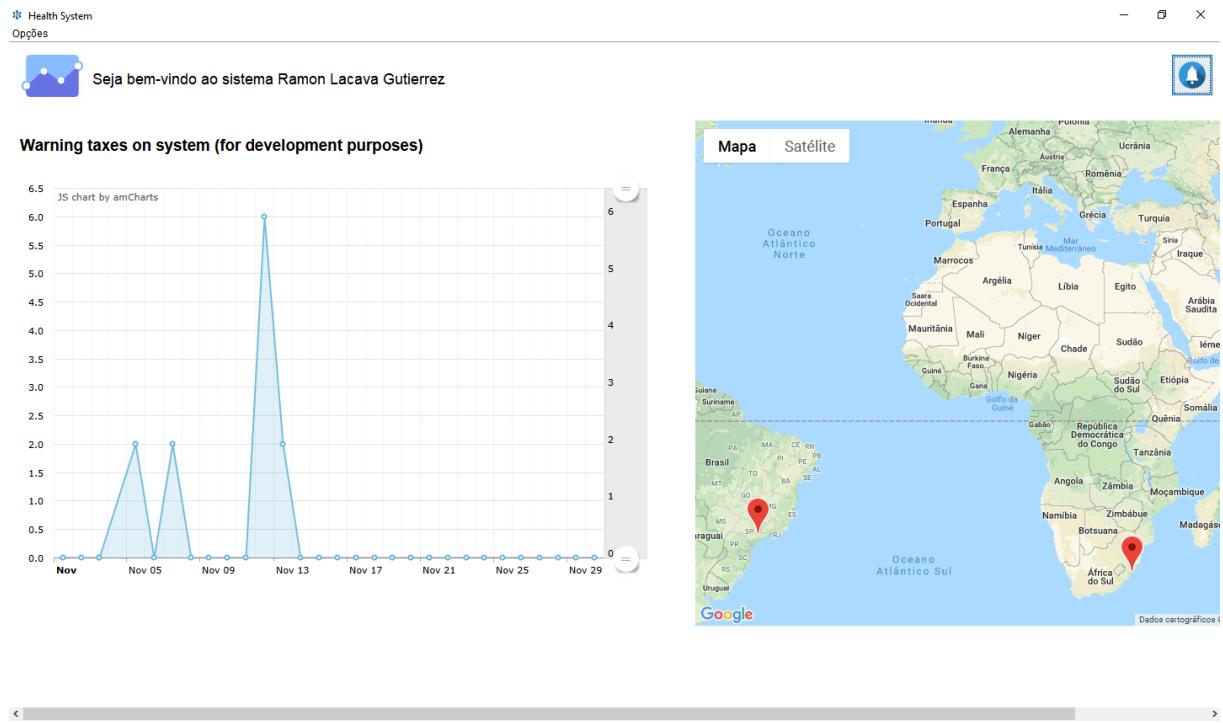
Figura 62 - Login do sistema desktop



Fonte: Elaborado pelo autor

Após a autenticação o usuário será direcionado para a tela inicial do sistema.

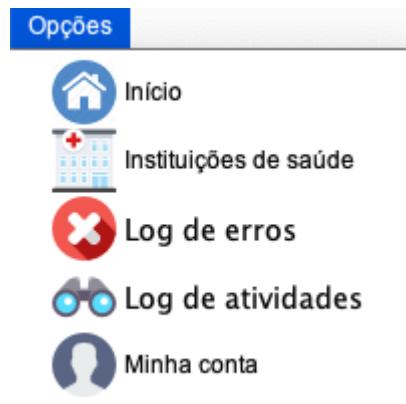
Figura 63 - Tela inicial do desktop do usuário administrador do sistema



Fonte: Elaborado pelo autor

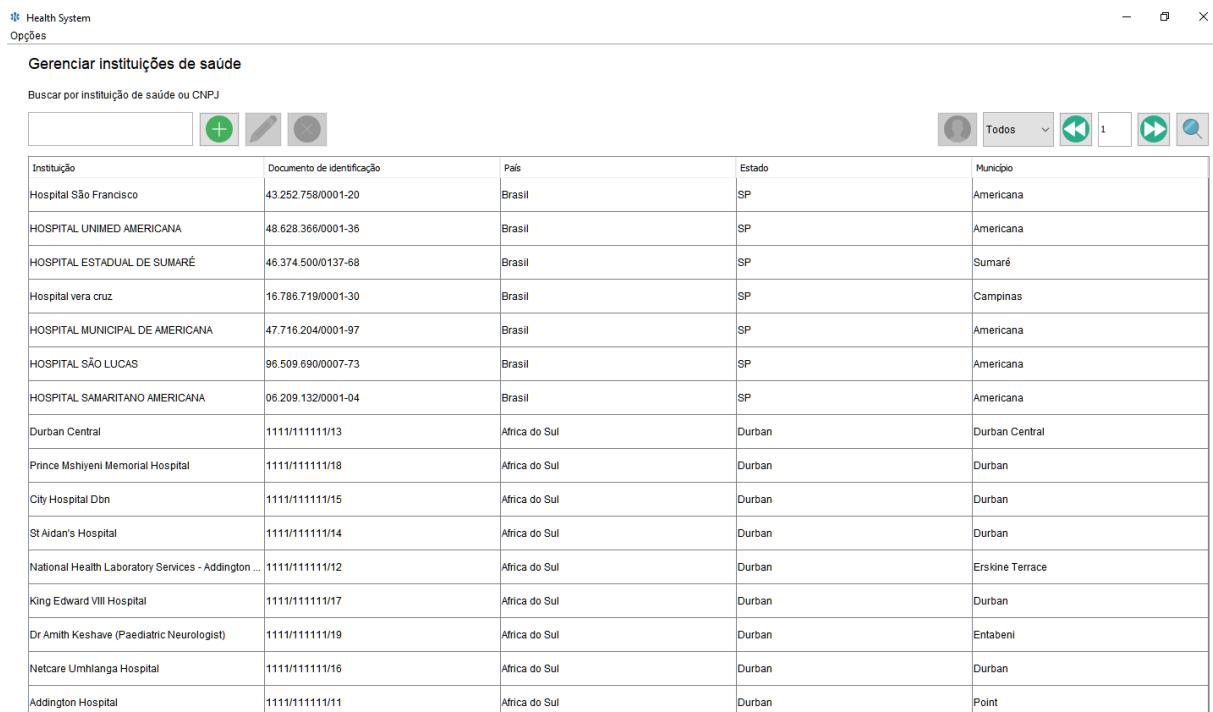
Nesta tela é possível ver um gráfico de erros e um mapa com as instituições cadastradas. Ao clicar no menu opções as opções de as seguintes alternativas apareceram: início, instituições de saúde, log de erros, log de atividades e minha conta.

Figura 64 – Menu opções do usuário administrador do sistema



Fonte: Elaborado pelo autor

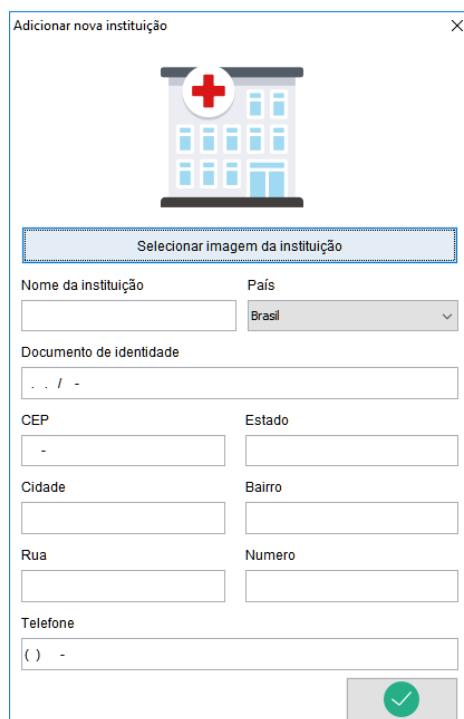
Ao escolher a opção instituições de saúde, uma tela com uma lista das instituições de saúde cadastradas será carregada.

Figura 65 - Tela de instituições médicas


Instituição	Documento de identificação	País	Estado	Município
Hospital São Francisco	43.252.758/0001-20	Brasil	SP	Americana
HOSPITAL UNIMED AMERICANA	48.628.366/0001-36	Brasil	SP	Americana
HOSPITAL ESTADUAL DE SUMARÉ	46.374.500/0137-68	Brasil	SP	Sumaré
Hospital vera cruz	16.786.719/0001-30	Brasil	SP	Campinas
HOSPITAL MUNICIPAL DE AMERICANA	47.716.204/0001-97	Brasil	SP	Americana
HOSPITAL SÃO LUCAS	96.509.690/0007-73	Brasil	SP	Americana
HOSPITAL SAMARITANO AMERICANA	06.209.132/0001-04	Brasil	SP	Americana
Durban Central	1111/111111/13	Africa do Sul	Durban	Durban Central
Prince Mshiyeni Memorial Hospital	1111/111111/18	Africa do Sul	Durban	Durban
City Hospital Dbn	1111/111111/15	Africa do Sul	Durban	Durban
St Aidan's Hospital	1111/111111/14	Africa do Sul	Durban	Durban
National Health Laboratory Services - Addington ...	1111/111111/12	Africa do Sul	Durban	Erskine Terrace
King Edward VIII Hospital	1111/111111/17	Africa do Sul	Durban	Durban
Dr Amith Keshave (Paediatric Neurologist)	1111/111111/19	Africa do Sul	Durban	Entabeni
Netcare Umhlanga Hospital	1111/111111/16	Africa do Sul	Durban	Durban
Addington Hospital	1111/111111/11	Africa do Sul	Durban	Point

Fonte: Elaborado pelo autor

Nesta página é possível realizar o gerenciamento das instituições médicas, podendo assim adicionar, editar, excluir e gerenciar os usuários das instituições. Clicando no botão adicionar uma nova janela irá se abrir.

Figura 66 – Janela adicionar instituição de saúde


Adicionar nova instituição

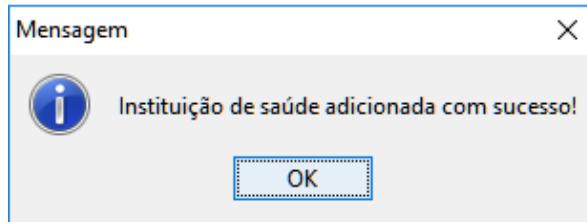
Selecionar imagem da instituição

Nome da instituição	País
<input type="text"/>	Brasil
Documento de identidade	<input type="text"/>
CEP	Estado
<input type="text"/>	<input type="text"/>
Cidade	Bairro
<input type="text"/>	<input type="text"/>
Rua	Numero
<input type="text"/>	<input type="text"/>
Telefone	<input type="text"/>
<input checked="" type="button"/>	

Fonte: Elaborado pelo autor

Nesta janela é possível adicionar uma imagem, um nome para a instituição, país, documento de identidade, CEP, estado, cidade, bairro, rua, número, telefone. Para adicionar uma nova instituição é preciso que seja preenchido todos os dados e que seja confirmada a ação, após estas ações a instituição será adicionada ao sistema.

Figura 67 – Mensagem de confirmação de adição de instituição



Fonte: Elaborado pelo autor

Ao selecionar uma instituição específica e clicar no botão editar uma nova janela com os dados da instituição irá ser aberta.

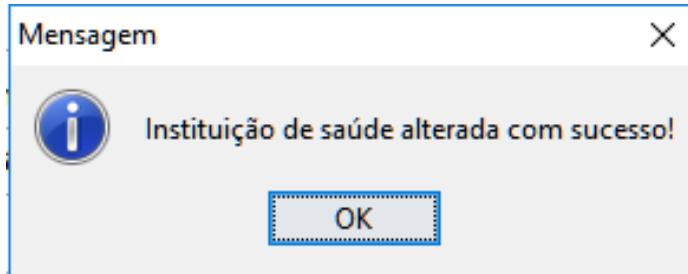
Figura 68 – Janela editar instituição de saúde

Nome da instituição		País
Hospital Unimed Campinas		Brasil
Documento de identidade		
46.124.624/0001-11		
CEP	Estado	
13035-420	SP	
Cidade	Bairro	
Campinas	Vila Industrial	
Rua	Número	
Rua São Carlos	369	
Telefone		
(19) 37739-310		

Fonte: Elaborado pelo autor

Após a modificação dos dados e a confirmação da ação uma mensagem irá aparecer e a instituição de saúde terá seus dados alterados no sistema.

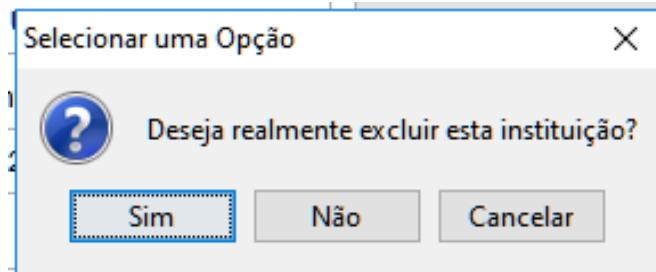
Figura 69 – Mensagem de confirmação de edição da instituição



Fonte: Elaborado pelo autor

Ao selecionar uma instituição e clicar no botão excluir uma nova janela irá se abrir, perguntando se o usuário realmente deseja realizar está ação.

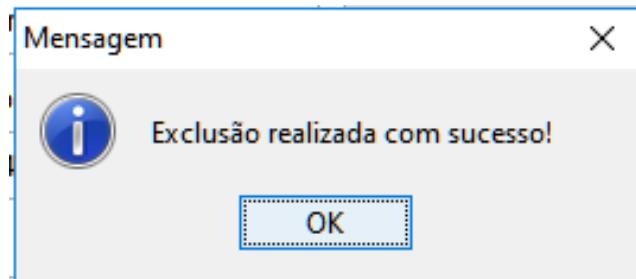
Figura 70 - Mensagem de exclusão



Fonte: Elaborado pelo autor

Caso o usuário clique em “sim”, uma mensagem de confirmação de exclusão irá se abrir e a instituição será apagada do sistema.

Figura 71 – Mensagem de confirmação de exclusão de instituição



Fonte: Elaborado pelo autor

Caso o contrário a janela apenas se fechará.

Ao selecionar uma instituição de saúde e clicar em gerenciar os usuários será carregada uma página com todos os usuários desta instituição.

Figura 72 – Tela gerenciar usuários da instituição de saúde

Foto	Nome	Documento de identidade	Tipo de usuário	Estado	Cidade
	adm@gmail.com	Não se aplica	1 - Administrador	SP	Americana
	unimed@gmail.com	Não se aplica	1 - Administrador	SP	Americana
	Salvador	050.876.348-77	2 - Médico	SP	Santa Bárbara D'Oeste
	Julia	546.291.080-09	4 - Enfermeira	SP	Santa Bárbara D'Oeste
	Drauzeo Varella	775.288.360-36	2 - Médico	SP	Santa Bárbara D'Oeste

Fonte: Elaborado pelo autor

Nesta tela é possível realizar a adição de um usuário administrador para a instituição, a adição de um usuário a instituição, enviar convite para participar da instituição, visualizar informação do usuário e exclusão de usuários.

Ao clicar em adicionar um usuário administrador para a instituição, uma nova janela se abrirá.

Figura 73 – Janela adicionar usuário administrador do sistema

Adicionar usuário administrador do sistema para esta instituição

E-mail

Senha

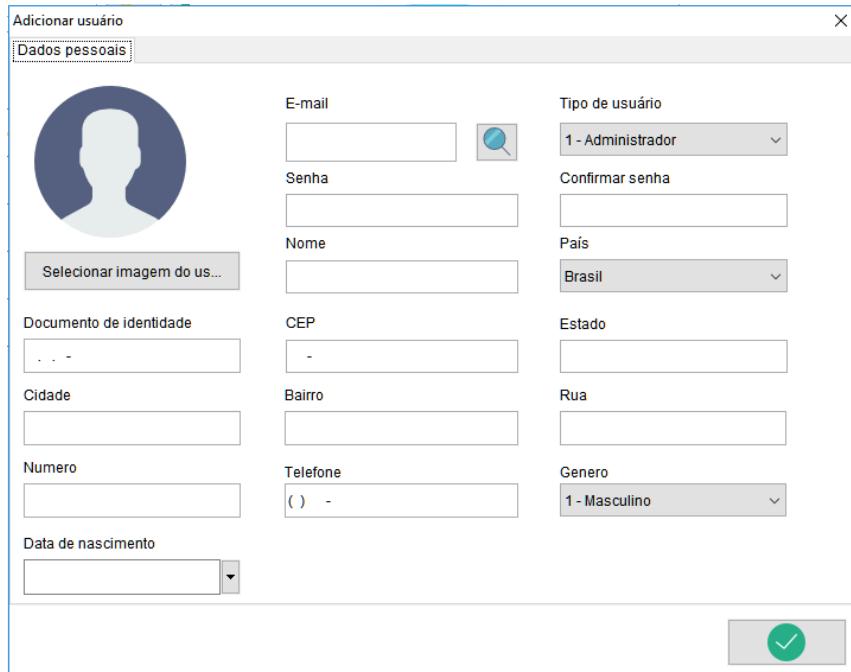
✓

Fonte: Elaborado pelo autor

Nesta janela é necessário o preenchimento do e-mail e senha para o usuário. Após o preenchimento dos dados e confirmar a ação uma mensagem de confirmação irá aparecer e o usuário aparecerá na lista de usuários da instituição de saúde.

Ao clicar em adicionar usuário uma nova janela se abrirá.

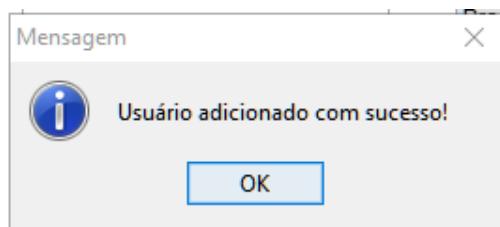
Figura 74 – Janela adicionar usuário da instituição



Fonte: Elaborado pelo autor

Nesta janela será pedido que o usuário preencha dados como: documento de identidade, nome, e-mail, senha, tipo de usuário, estado, país, bairro, rua, CEP, número, telefone, gênero e data de nascimento. Após o preenchimento dos campos e a confirmação das ações uma janela com a confirmação será exibida.

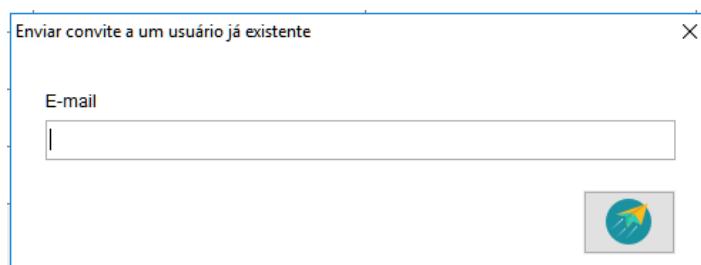
Figura 75 – Mensagem de confirmação de adição de usuário



Fonte: Elaborado pelo autor

Ao clicar em enviar convite para usuário já existente uma nova janela irá abrir.

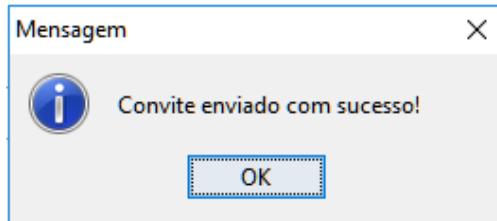
Figura 76 – Janela de envio de convite para usuário já existente



Fonte: Elaborado pelo autor

O sistema solicitará que o usuário insira o e-mail do usuário, assim que preenchido é necessário que clique no botão enviar para que um convite para este usuário seja enviado e uma mensagem de confirmação será exibida na tela.

Figura 77 – Mensagem de convite enviado com sucesso



Fonte: Elaborado pelo autor

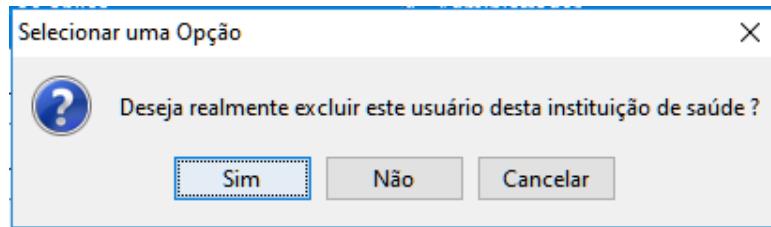
Ao selecionar um usuário e clicar em visualizar uma nova janela com as informações deste usuário se abrirá.

Figura 78 – Janela visualizar usuário

Fonte: Elaborado pelo autor

Ao selecionar um usuário e clicar em excluir uma nova janela irá se abrir com uma mensagem de confirmação com as opções sim e não.

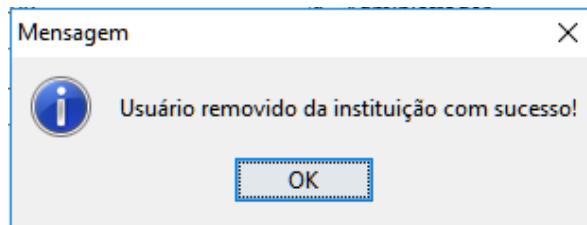
Figura 79 – mensagem de exclusão de usuário



Fonte: Elaborado pelo autor

Caso o usuário selecione a opção “sim” uma nova janela confirmando a ação será exibida e o usuário será apagado da instituição de saúde, caso o contrário a janela se fechará e o usuário será redirecionado para a lista de usuários cadastrados na instituição.

Figura 80 – Mensagem de usuário removido da instituição com sucesso



Fonte: Elaborado pelo autor

Voltando ao menu de opções, o usuário seleciona o *log* de erros. Uma nova tela será carregada com os *logs* de erros, sendo possível visualizar os erros ocorridos no sistema, os quais são filtrados por dia.

Figura 81 -Tela de log de erros

Foto	Nome	Data	Mensagem	Nome do arquivo	Nome da classe	Nome do método	Número da linha	Informações adicionais
		16/10/2018 22:13:28		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.ze2kas@gmail.com pass...
		16/10/2018 07:44:44		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.animesm213@gmail.com ...
		16/10/2018 20:10:27		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.ramonrunne@gmail.com ...
		16/10/2018 20:10:22		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.ramonrunne@gmail.com ...
		16/10/2018 23:22:38		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.ramonrunne@gmail.com ...
		16/10/2018 22:13:28		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.ze2kas@gmail.com pass...
		16/10/2018 07:44:44		SQLServerException.java	com.microsoft.sqlserver.jdbc...	makeFromDriverError	206	login.animesm213@gmail.com ...

Fonte: Elaborado pelo autor

Voltando ao menu de opções o usuário seleciona o *log* de atividades. Uma nova tela será carregada com todas as atividades realizadas no sistema, estas atividades de forma semelhante aos erros, também são filtradas por dia.

Figura 82 – Tela de log de atividades

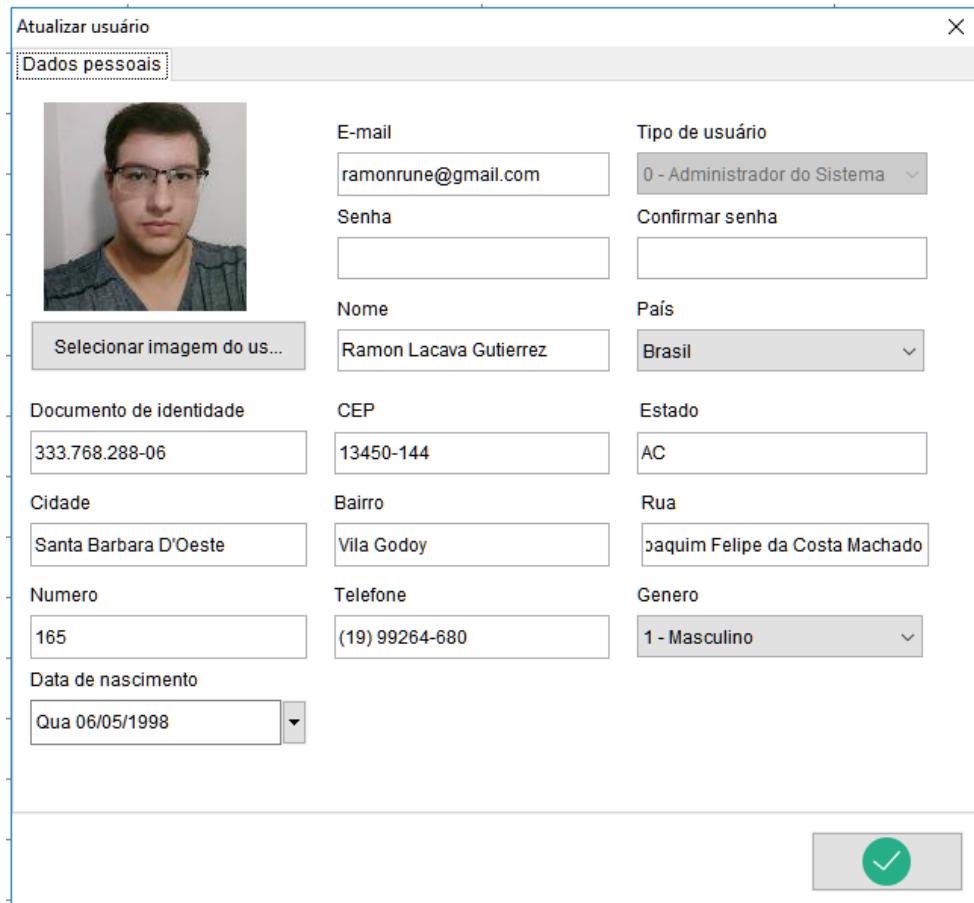
The screenshot shows a table titled 'Trilha de auditoria' (Audit Trail) with the following columns: Foto (Photo), Nome (Name), Data (Date), Categoria (Category), Evento (Event), and Informações adicionais (Additional Information). The data is as follows:

Foto	Nome	Data	Categoria	Evento	Informações adicionais
	Ramon Lacava Gutierrez	17/11/2018 10:43:12	HEALTH_INSTITUTION	DELETE_BIND_HEALTH_INSTITUTION	Id userab91e634-59dd-4625-bf0c-5802aef... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:36:47	PHYSICIAN	LIST_SPECIALIZATIONS	country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:36:46	PHYSICIAN	LIST_PHYSICIAN_SPECIALIZATIONS	idPhysician 2b55c8c6-8ac1-435b-9fa8-189df... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:36:43	PHYSICIAN	GET_PHYSICIAN	idUser caf069a3-1445-43e2-ah9e-47c69630... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:36:41	USER	GET_USER	userId caf069a3-1445-43e2-ab8e-47c69630... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:33:16	HEALTH_INSTITUTION	BIND_HEALTH_INSTITUTION	Id user982b19dd-4be7-a06e-f142f273... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:33:16	HEALTH_INSTITUTION	CONTAINS_BIND_HEALTH_INSTITUTION	Id user982b19dd-4be7-a06e-f142f273... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:33:11	USER	GET_USER_BY_LOGIN	login natalina@hotmail.com country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:54	HEALTH_INSTITUTION	LIST_BOUNDED_USERS_FROM_HEALTH_INSTITUTION	status 1 IdHealthInstitution 138a2cc4-0b02... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:49	HEALTH_INSTITUTION	CONTAINS_BIND_HEALTH_INSTITUTION	Id usere01b70d0-b6a5-4068-b1ce-82ffe58b... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:49	HEALTH_INSTITUTION	BIND_HEALTH_INSTITUTION	Id usere01b70d0-b6a5-4068-b1ce-82ffe58b... country BRA language PT
		17/11/2018 10:25:46	PATIENT	INSERT_PATIENT	PatientModel [idPatient=e6599cb4-4168-40c... country BRA language PT
		17/11/2018 10:25:45	PATIENT	CHECK_IF_USER_EXIST	idUser e01b70d0-b6a5-4068-b1ce-82ffe58b... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:41	USER	SEND_MAIL_PASSWORD	email 1234@hotmail.com secretCode #sb co... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:41	USER	INSERT_USER	UserModel [id=e01b70d0-b6a5-4068-b1ce-8... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:41	USER	CHECK_IF_USER_EXIST	login 1234@hotmail.com country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:25:41	USER	CHECK_IF_DOCUMENT_EXIST	document 52320324020 country BRA typeOf... country BRA language PT
	Ramon Lacava Gutierrez	17/11/2018 10:22:34	PHYSICIAN	LIST_SPECIALIZATIONS	country BRA language PT

Fonte: Elaborado pelo autor

Voltando ao menu de opções o usuário seleciona minha conta, uma nova janela irá se abrir com todas as informações do usuário prontas para a alteração.

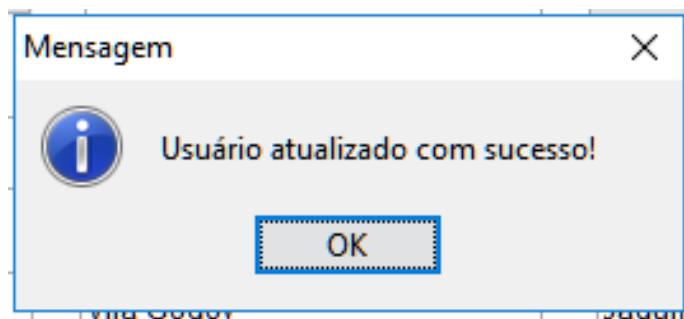
Figura 83 – Janela de atualização de dados da conta do usuário administrador do sistema



Fonte: Elaborado pelo autor

Após o usuário ter realizado as alterações e clicado no botão de confirmação uma nova janela com a confirmação da ação será exibida e os dados serão alterados no sistema.

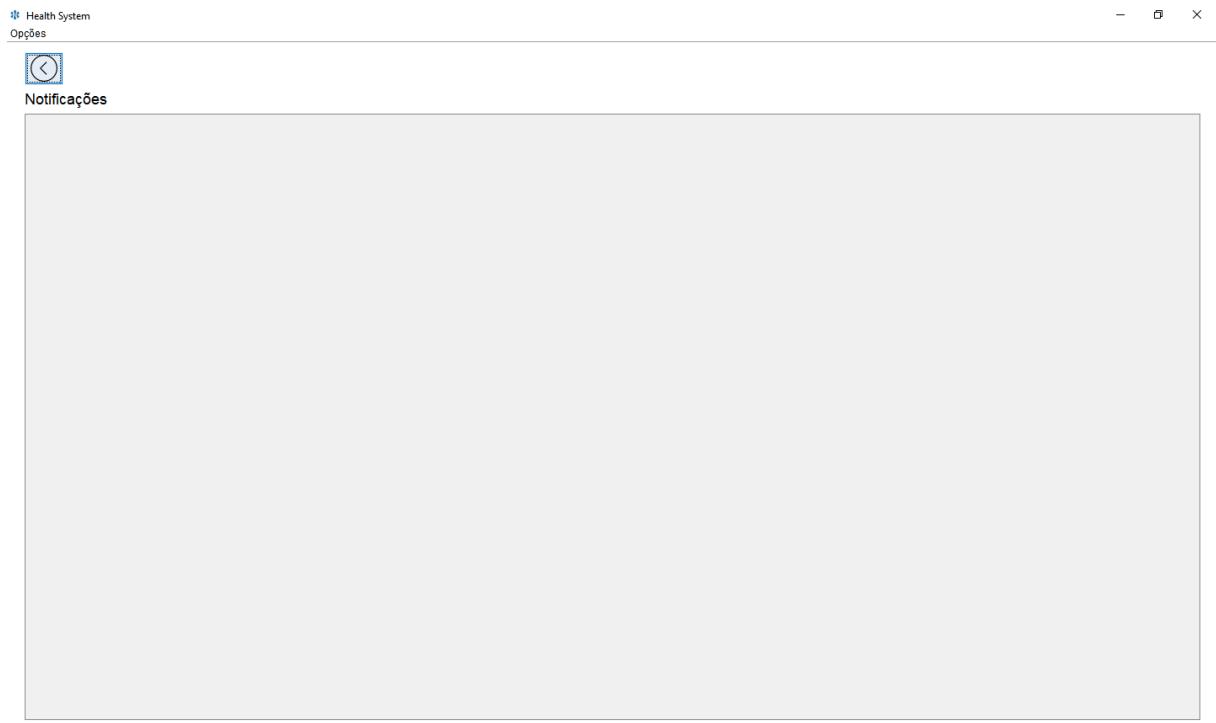
Figura 84 – Mensagem de usuário administrador do sistema alterado com sucesso



Fonte: Elaborado pelo autor

Na página inicial o usuário pode visualizar suas notificações clicando no ícone de notificações, que fará com que seja carregado uma tela com todas suas notificações.

Figura 85 - Tela de notificações usuário administrador do sistema

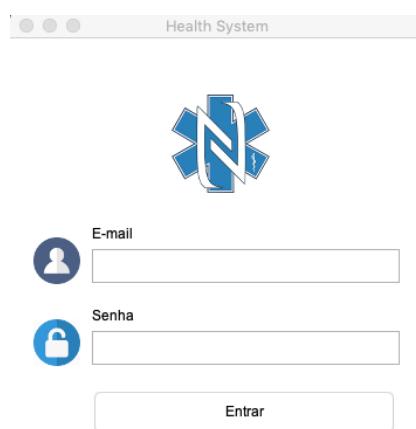


Fonte: Elaborado pelo autor

4.2.1.2 Usuário administrador da instituição de saúde

Ao inicializar o sistema serão solicitados e-mail e senha.

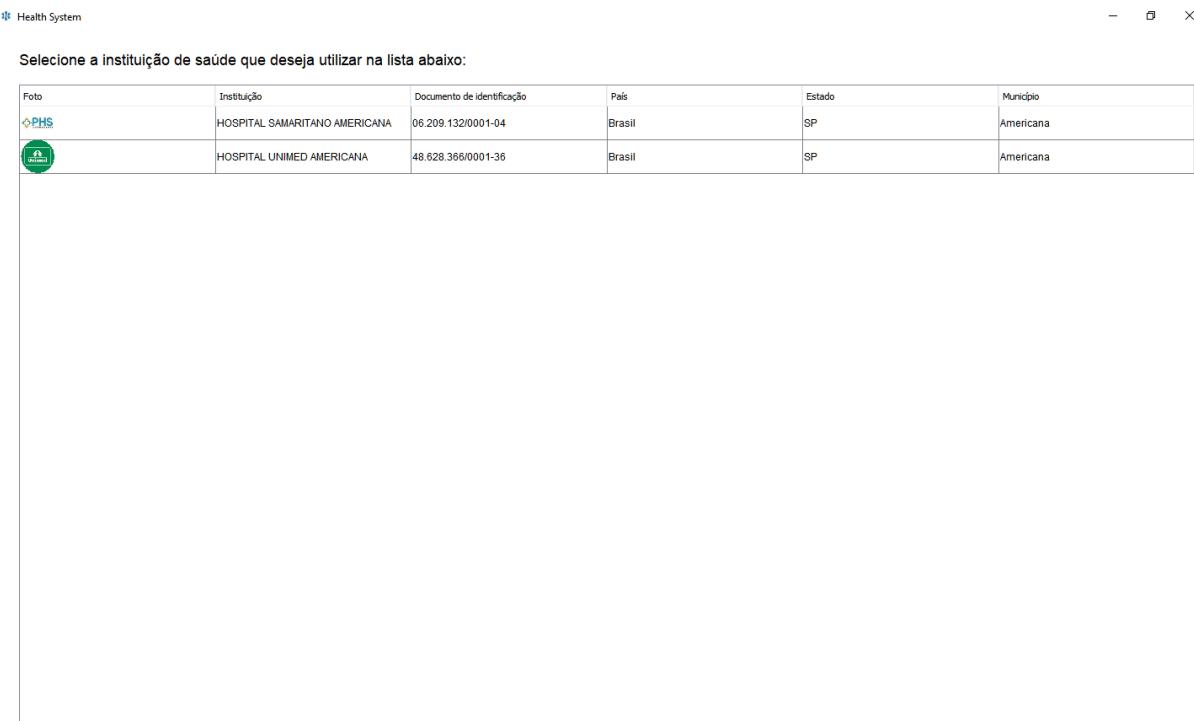
Figura 86 - Login do sistema desktop



Fonte: Elaborado pelo autor

Após a autenticação no sistema caso o usuário esteja cadastrado em mais de uma instituição a tela de seleção de instituição será exibida. Após selecionar a instituição, a página inicial do sistema será carregada.

Figura 87 – Tela de seleção de instituição



Fonte: Elaborado pelo autor

Caso o usuário esteja cadastrado em apenas uma instituição ele será direcionado para a tela inicial do sistema de forma direta.

Figura 88 - Tela inicial das instituições de saúde



Fonte: Elaborado pelo autor

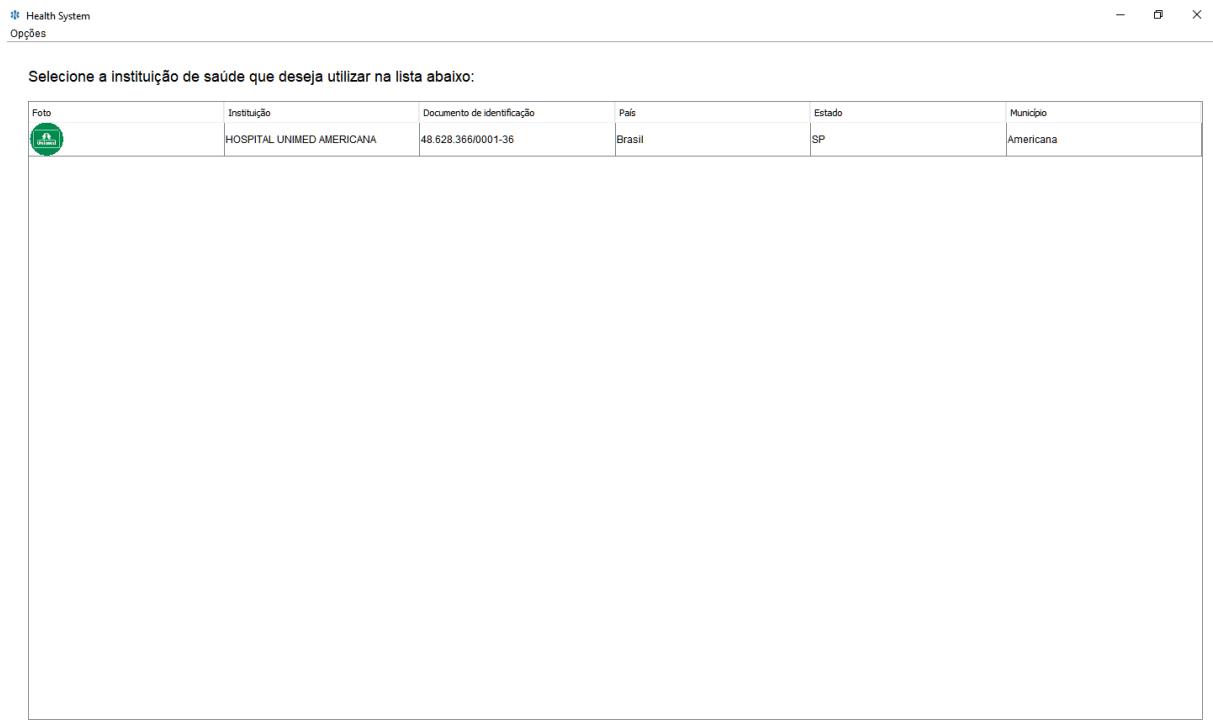
Ao clicar no menu de opções as seguintes opções aparecerão: início, instituições de saúde, usuários e minha conta.

Figura 89 - Menu opções do usuário administrador da instituição do sistema

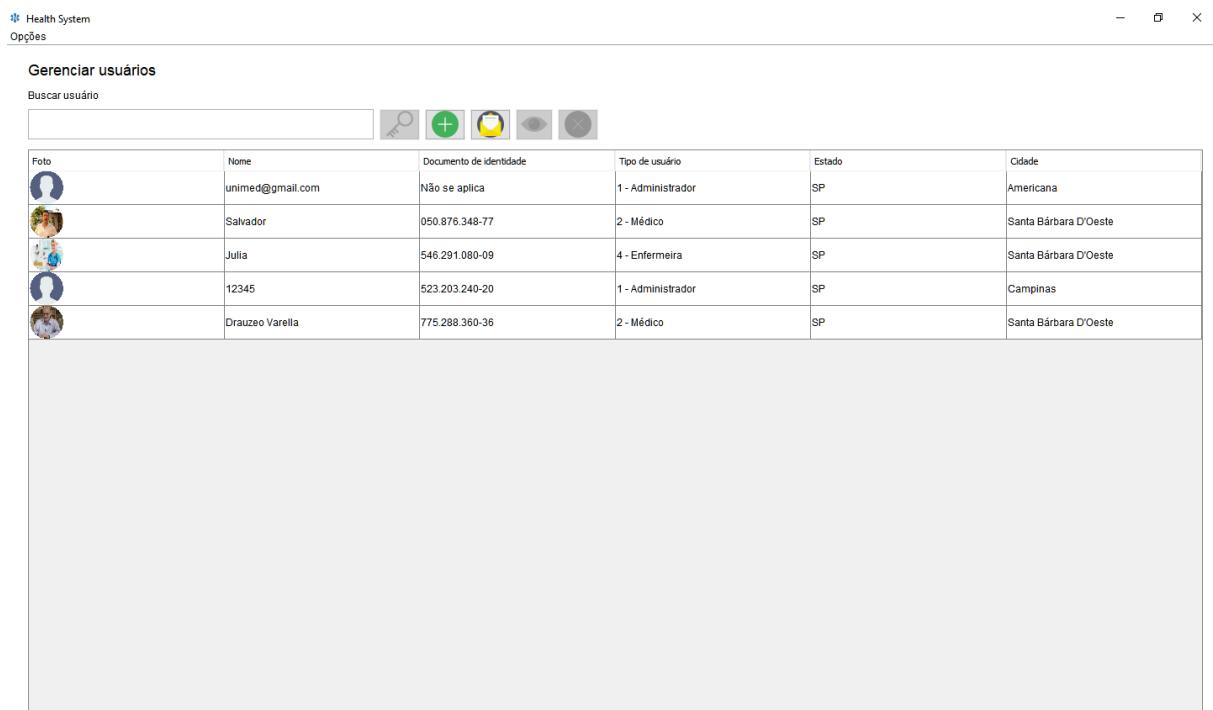


Fonte: Elaborado pelo autor

Ao clicar em instituições de saúde o usuário será direcionado novamente para a página de seleção de instituição.

Figura 90 – Tela selecionar nova instituição de saúde

Caso o usuário selecione a opção de usuário ele será redirecionado para a página de gerenciamento de usuários.

Figura 91 - Tela de gerenciamento de usuário da instituição de saúde

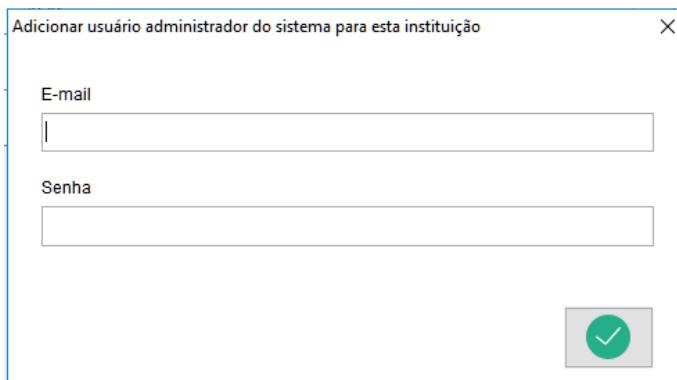
Fonte: Elaborado pelo autor

Neste tipo de usuário não é possível adicionar usuário administrador para a instituição, pois esta ação está disponível apenas para o usuário administrador do sistema.

Nesta tela é possível fazer a adição de um usuário administrador para a instituição, a adição de um usuário a instituição, enviar convite para participar da instituição, visualizar informação do usuário e exclusão de usuários.

Ao clicar em adicionar um usuário administrador para a instituição, uma nova janela se abrirá.

Figura 92 – Janela adicionar usuário administrador do sistema para instituição



Fonte: Elaborado pelo autor

Nesta janela é necessário o preenchimento do e-mail e senha para o usuário. Após o preenchimento dos dados e confirmar a ação, uma mensagem de confirmação irá aparecer e o usuário aparecerá na lista de usuários da instituição de saúde.

Ao clicar em adicionar usuário uma nova janela se abrirá.

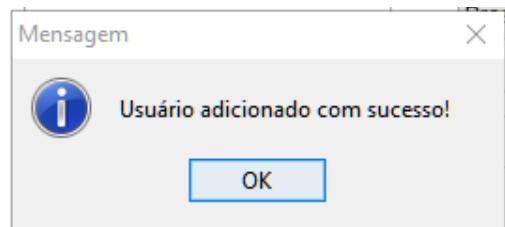
Figura 93 – Janela adicionar usuário da instituição

The screenshot shows a larger "Adicionar usuário" window. It includes a section for "Dados pessoais" with fields for "E-mail", "Senha", "Nome", "Documento de identidade", "Cidade", "Número", "Data de nascimento", "Tipo de usuário" (set to "1 - Administrador"), "Confirmar senha", "País" (set to "Brasil"), "Estado", "Bairro", "Telefone", "Rua", and "Genero" (set to "1 - Masculino"). There is also a placeholder for "Selecionar imagem do us...". A checkmark button is located at the bottom right.

Fonte: Elaborado pelo autor

Nesta janela será pedido que o usuário preencha dados como: documento de identidade, nome, e-mail, senha, tipo de usuário, estado, país, bairro, rua, CEP, número, telefone, gênero e data de nascimento. Após o preenchimento dos campos e a confirmação das ações uma janela com a confirmação será exibida.

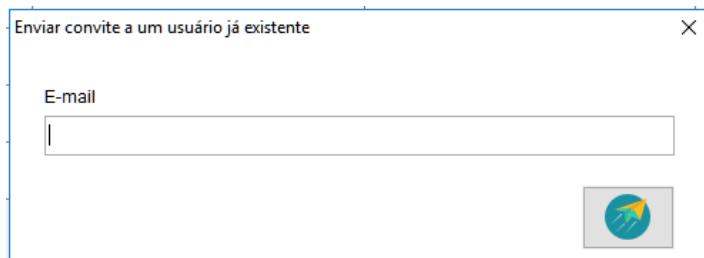
Figura 94 – Mensagem de confirmação de adição de usuário



Fonte: Elaborado pelo autor

Ao clicar em enviar convite para usuário já existente uma nova janela irá se abrir.

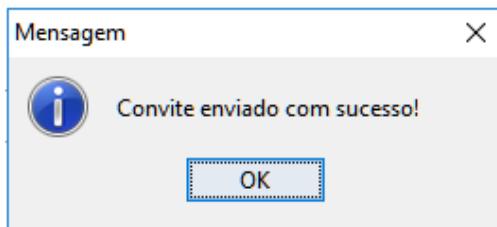
Figura 95 – Janela para envio de convite para usuário já existente



Fonte: Elaborado pelo autor

O sistema solicitará que o usuário insira um e-mail, e, assim que preenchido, é necessário que seja clicado no botão enviar, para que um convite para este usuário seja enviado e uma mensagem de confirmação seja exibida na tela.

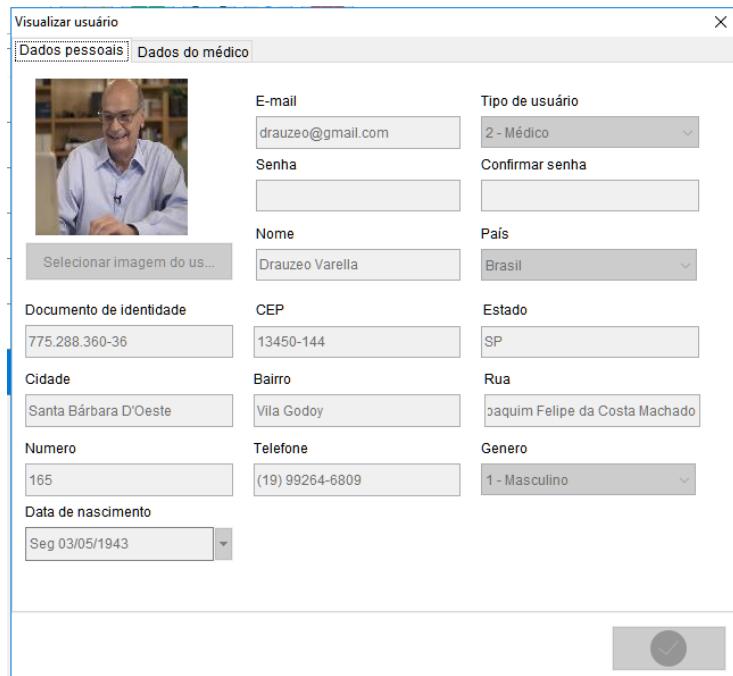
Figura 96 – Mensagem de convite enviado com sucesso



Fonte: Elaborado pelo autor

Ao selecionar um usuário e clicar em visualizar, uma nova janela com as informações deste usuário se abrirá.

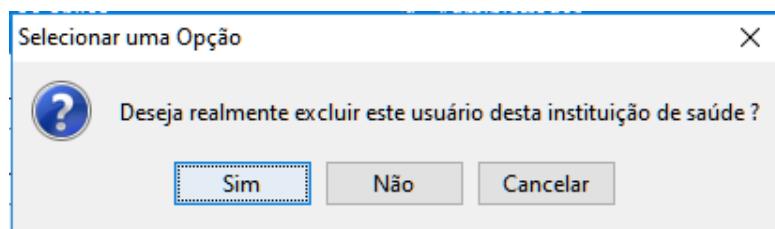
Figura 97 – Janela visualizar usuário



Fonte: Elaborado pelo autor

Ao selecionar um usuário e clicar em excluir, uma nova janela irá se abrir com uma mensagem de confirmação com as opções “sim” e “não”.

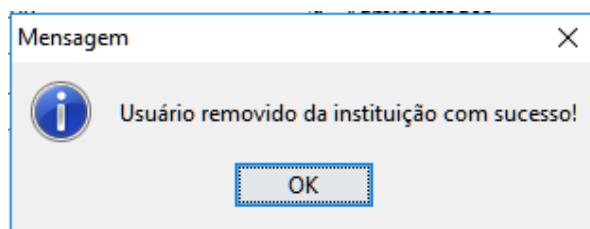
Figura 98 – Janela de exclusão de usuário



Fonte: Elaborado pelo autor

Caso o usuário selecione “sim”, uma nova janela confirmando a ação será exibida e o usuário será apagado da instituição de saúde, caso o contrário a janela se fechará e o usuário será redirecionado para a lista de usuários cadastrados na instituição.

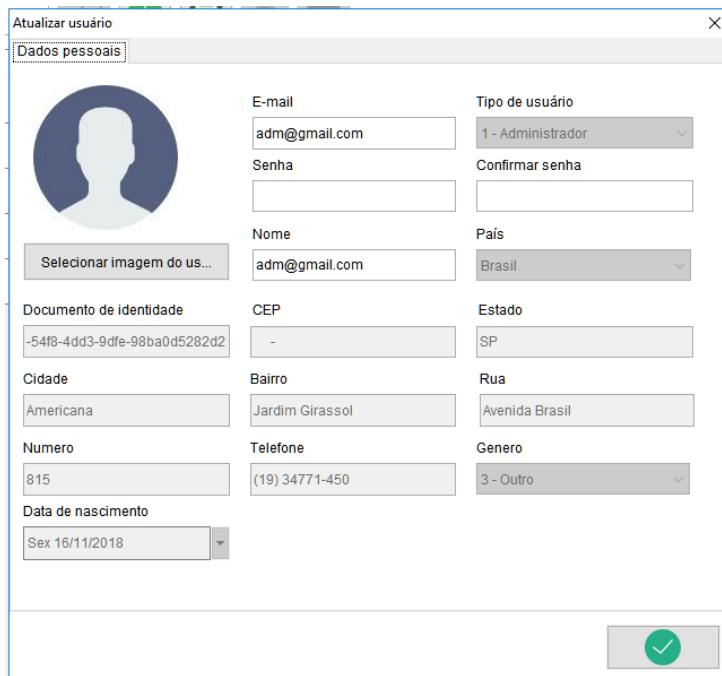
Figura 99 – Mensagem usuário removido da instituição com sucesso



Fonte: Elaborado pelo autor

No menu opções o usuário seleciona minha conta, uma nova janela com todas as informações do usuário é carregada.

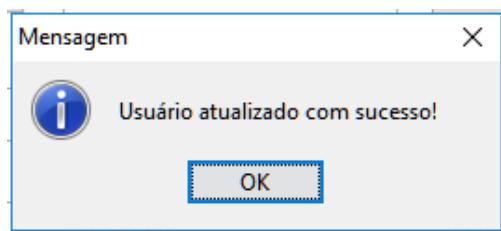
Figura 100 – Janela de atualização de dados do usuário administrador da instituição de saúde



Fonte: Elaborado pelo autor

Após feitas as alterações necessárias e a confirmação destas ações uma janela de confirmação é exibida e as informações do usuário são alteradas no sistema.

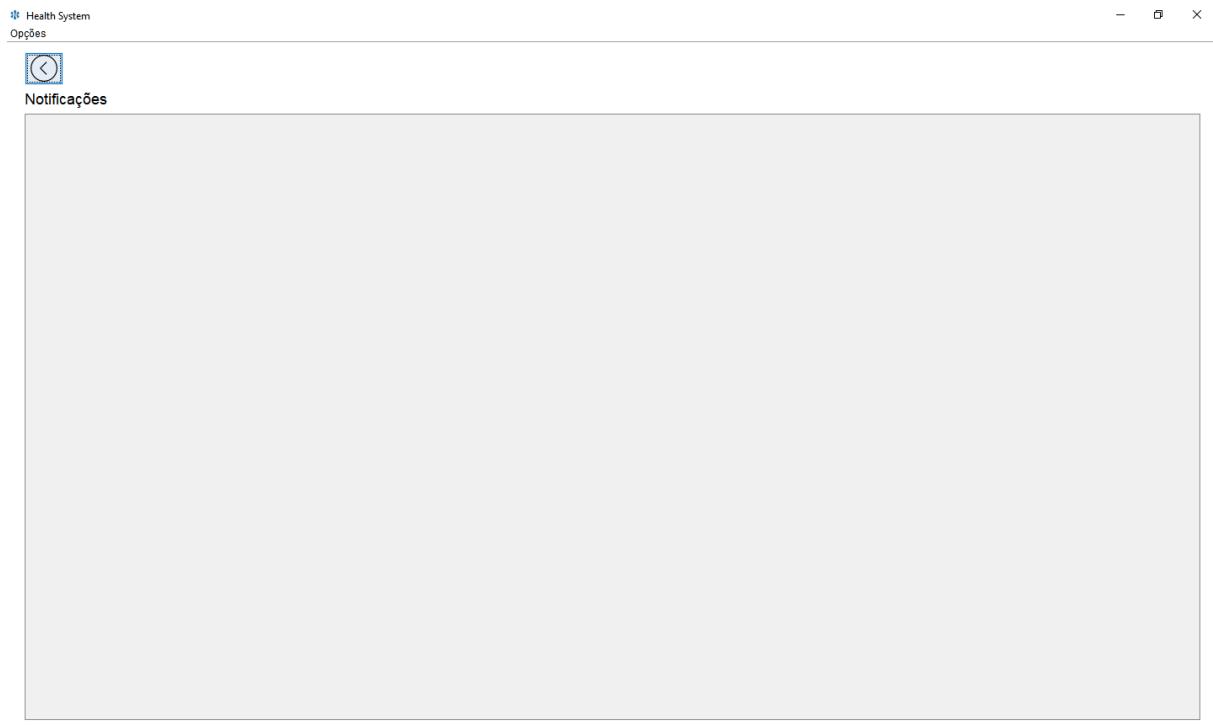
Figura 101 – Mensagem de usuário administrador do sistema alterado com sucesso



Fonte: Elaborado pelo autor

Na página inicial o usuário pode visualizar suas notificações clicando no ícone de notificações será carregado a página com todas suas notificações.

Figura 102 - Tela de notificações usuário administrador do sistema

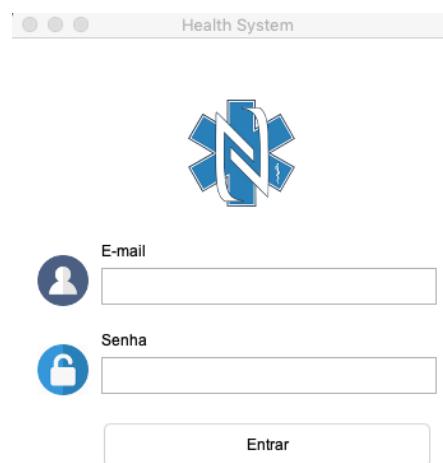


Fonte: Elaborado pelo autor

4.2.1.3 Usuário médico

Ao inicializar o sistema serão solicitados e-mail e senha.

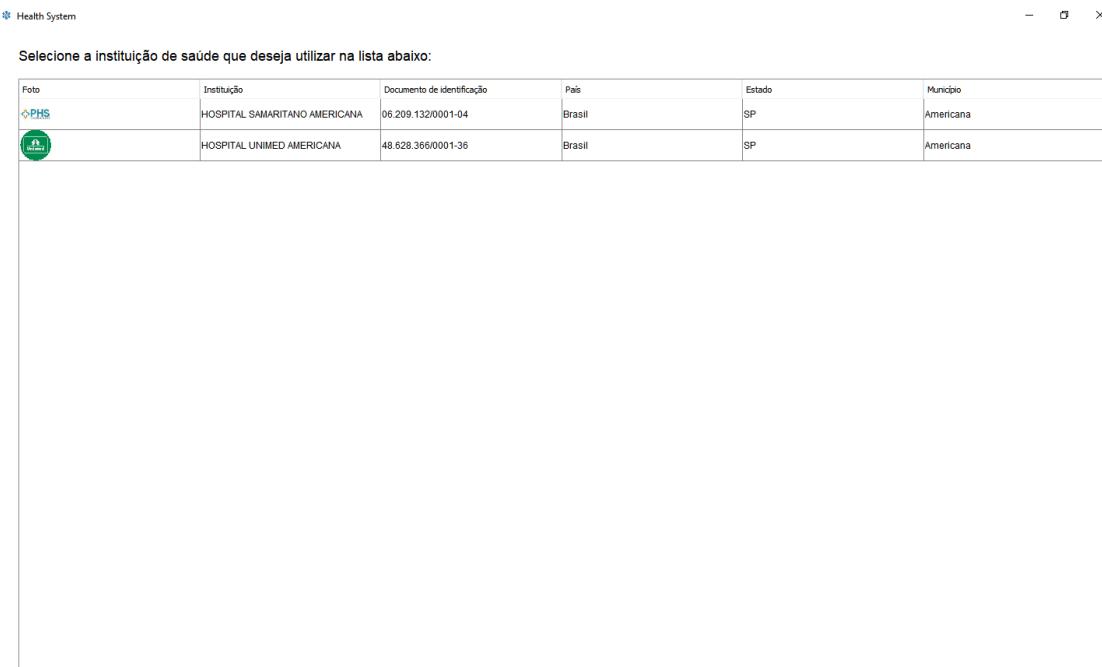
Figura 103 - Login do sistema desktop



Fonte: Elaborado pelo autor

Após a autenticação no sistema caso o usuário esteja cadastrado em mais de uma instituição a página de seleção de instituição será exibida, depois de selecionada a instituição a página inicial do sistema será carregada.

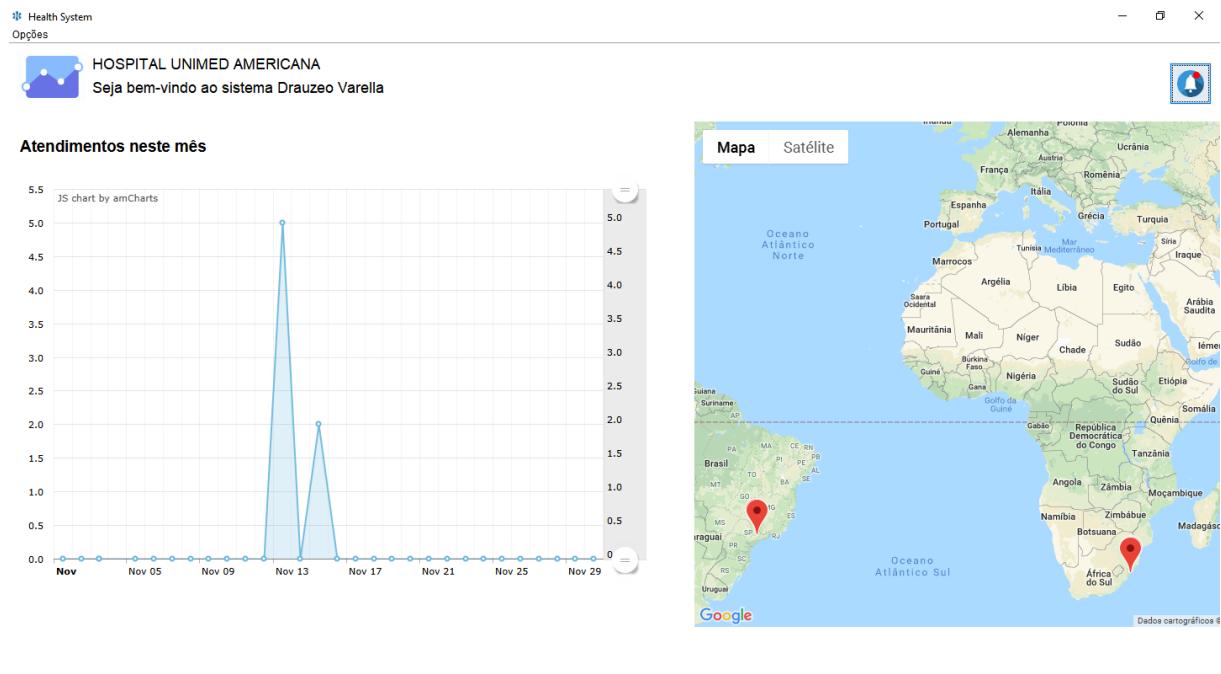
Figura 104 – Tela de seleção de instituição



Fonte: Elaborado pelo autor

Caso o usuário esteja cadastrado em apenas uma instituição ele será direcionado para a tela inicial do sistema de forma direta.

Figura 105 - Tela inicial do usuário médico

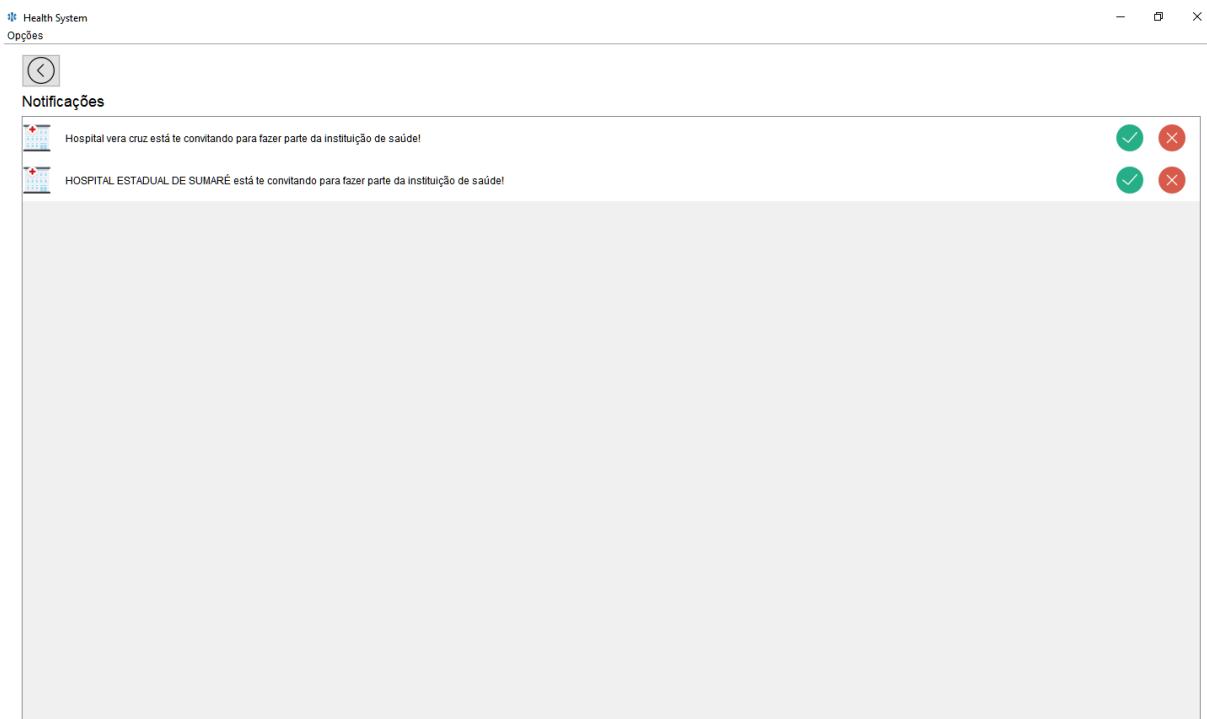


Fonte: Elaborado pelo autor

Nesta tela é possível analisar o número de atendimento por mês através do gráfico, visualizar as instituições na qual está cadastrado e ir para a páginas de notificações.

Clicando em notificações o usuário é redirecionado para a tela de notificações.

Figura 106 - Tela de notificações do usuário médico

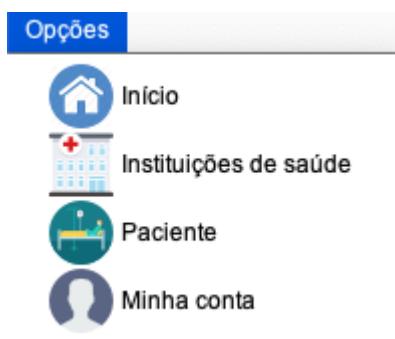


Fonte: Elaborado pelo autor

Nesta tela é possível ver uma lista das notificações do médico.

Ao selecionar o menu opções é possível visualizar as seguintes opções: início, instituições de saúde, paciente e minha conta.

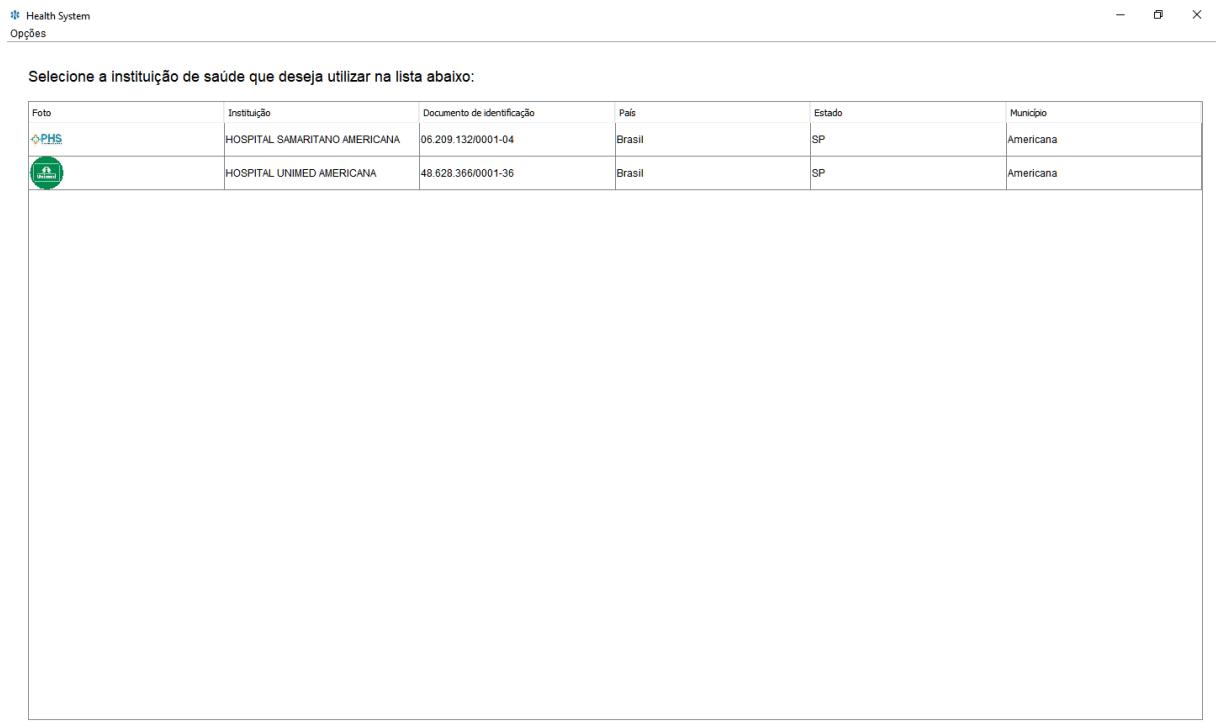
Figura 107 - Menu opções do usuário médico



Fonte: Elaborado pelo autor

Ao selecionar a opção instituição de saúde o usuário será direcionado novamente para a página de seleção de instituição de saúde.

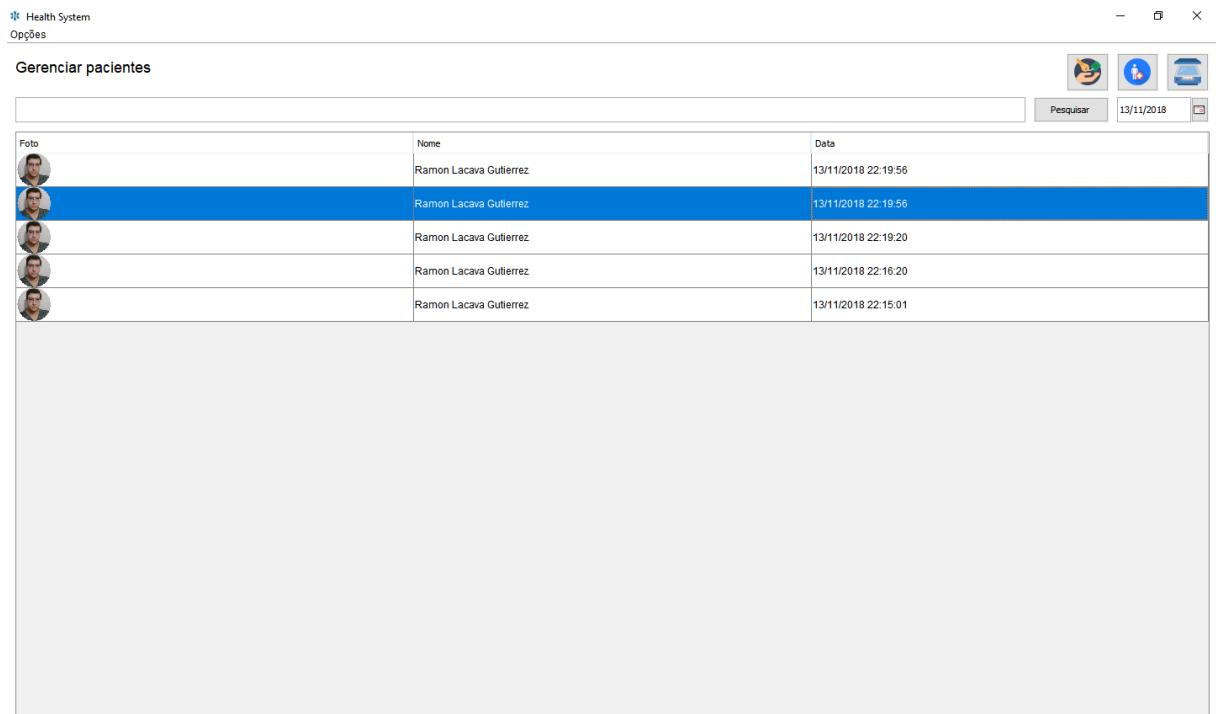
Figura 108 - Tela de seleção de instituição de saúde do usuário médico



Fonte: Elaborado pelo autor

Ao selecionar no menu de opções a opção paciente o sistema redirecionará o usuário para a tela de atendimentos do dia.

Figura 109 - Tela de atendimentos do médico



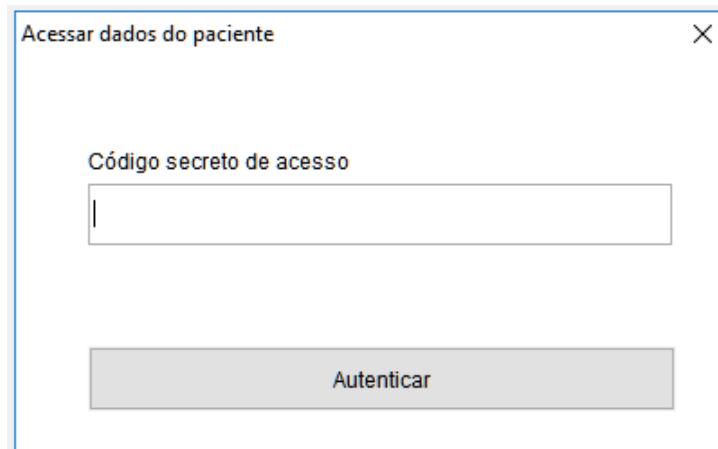
Fonte: Elaborado pelo autor

Nesta página é possível visualizar todos os atendimentos realizados no dia, buscar pacientes através do filtro, navegar entre os dias para visualizar os atendimentos dos dias anteriores, adicionar paciente, buscar paciente através do equipamento de acesso, buscar paciente através do código de acesso que fica disponível no aplicativo do paciente.

É possível realizar o acesso a ficha de um paciente através da leitura do equipamento de acesso NFC, clicando no paciente da lista ou acessando via código secreto de acesso.

Para realizar o acesso via código secreto é necessário clicar em buscar por código de acesso.

Figura 110 - Janela de solicitação de código de acesso



Fonte: Elaborado pelo autor

Nesta tela será solicitado o preenchimento do código de acesso do paciente, após a autenticação do código o sistema irá carregar as informações do paciente.

Figura 111 - Tela com as informações do paciente

The screenshot shows a web-based application for managing patient data. At the top, there's a header with the logo 'Health System' and a 'Opções' button. Below the header, a navigation bar includes tabs for 'Dados do paciente' (selected), 'Doenças', 'Diagnósticos', 'Exames', 'Medicamentos', and 'Equipamento de acesso'. The main content area displays a circular profile picture of a man with glasses. To the right of the profile are several input fields and dropdown menus:

- Nome do paciente:** Ramon Lacava Gutierrez
- Documento de identidade:** 333.768.288-06
- Idade:** 20
- Gênero:** 1 - Masculino
- País:** Brasil
- Tipo sanguíneo:** A+
- Peso (Kg):** 94.9
- Altura (Metros):** 1.7
- Cor:** A color picker shows a blue shade.
- Status:** Verificado
- Nome da mãe (Biológico):** Silvana Aparecida Lacava
- Nome do pai (Biológico):** Salvador Gutierrez

At the bottom right of the form are two buttons: a lock icon and a checkmark icon.

Fonte: Elaborado pelo autor

Nesta tela é possível fazer o gerenciamento das informações de saúde do paciente, desde o gerenciamento de diagnósticos, gerenciamento de doenças, gerenciamento de exames, visualização de medicamentos e o gerenciamento de equipamentos de acesso.

Ao clicar na aba doenças será carregada uma tela com uma lista de todas as doenças que o paciente possui.

Figura 112 - Tela de doenças| Doença | Anotação |
| --- | --- |
| Alergia | Espira quando tem muito pó |
| Alergia | a pó |
| Catapora | Na infância |

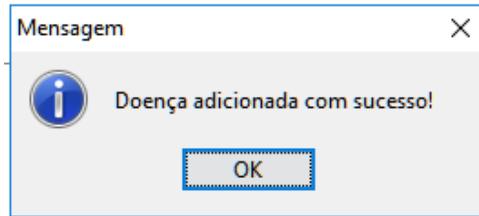
 The 'Alergia' row appears to be a duplicate entry."/>
Fonte: Elaborado pelo autor

Nesta tela é possível adicionar, editar e excluir doenças. Para adicionar uma doença é necessário que o usuário clique no botão adicionar, assim uma nova janela será aberta.

Figura 113 - Janela de adição de doenças
Fonte: Elaborado pelo autor

Nesta tela será necessário o preenchimento da doenças e caso necessário a descrição desta, após confirmada a ação será exibida uma mensagem de confirmação e a doença adicionada à lista.

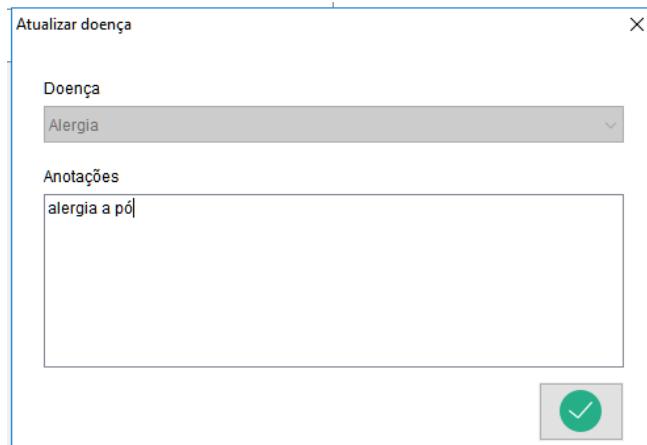
Figura 114 - Mensagem de doença adicionada com sucesso



Fonte: Elaborado pelo autor

Para fazer a edição de uma doença é necessário que o usuário selecione a doença na qual deseja fazer a edição e clique em editar, assim uma nova janela com os campos disponíveis para a edição se abrirá.

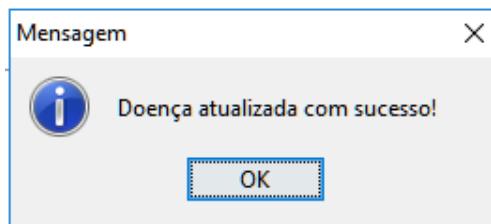
Figura 115 - Tela de edição de doença



Fonte: Elaborado pelo autor

Após a alteração dos dados uma janela de confirmação de sucesso será exibida.

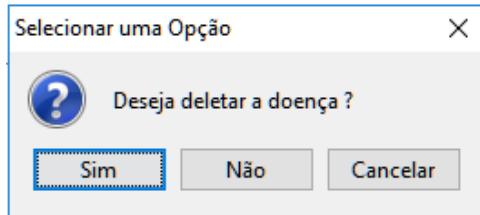
Figura 116 - Mensagem de atualização de doença realizada com sucesso



Fonte: Elaborado pelo autor

Para realizar a exclusão de uma doença é necessário que o usuário selecione a doença que deseja excluir e clique em excluir, assim uma janela de confirmação será exibida com as opções “sim” e “não”.

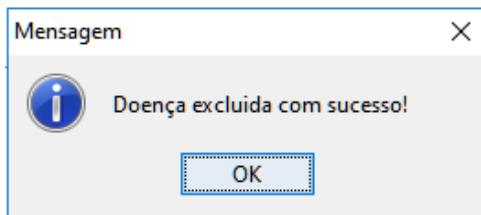
Figura 117 - Janela de exclusão de doença



Fonte: Elaborado pelo autor

Caso o usuário clique em “sim” uma nova janela irá se abrir confirmando a exclusão da doença da lista de doenças do paciente, caso o contrário a janela se fechará e o usuário será redirecionado para lista de doenças.

Figura 118 - Mensagem de doença excluída com sucesso



Fonte: Elaborado pelo autor

Ao selecionar a opção diagnóstico o usuário será direcionado para a tela de diagnósticos, nesta existe uma lista com todos os diagnósticos do paciente.

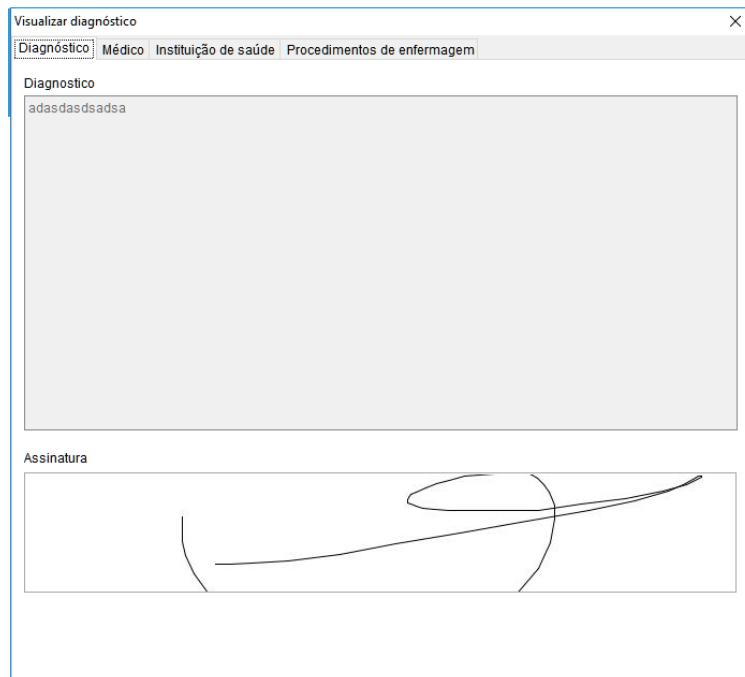
Figura 119 - Tela de diagnósticos

Médico	Registro médico	Local	Data do diagnóstico
Unimed	4545454	HOSPITAL UNIMED AMERICANA	13/11/2018 23:20:18
PHS	4545454	HOSPITAL SAMARITANO AMERICANA	17/10/2018 21:45:49
Unimed	4545454	HOSPITAL UNIMED AMERICANA	17/10/2018 08:56:44
Unimed	4545454	HOSPITAL UNIMED AMERICANA	16/10/2018 21:51:05
Unimed	4545454	HOSPITAL UNIMED AMERICANA	16/10/2018 21:04:07
PHS	4545454	HOSPITAL SAMARITANO AMERICANA	14/10/2018 18:33:36
Unimed	4545454	HOSPITAL UNIMED AMERICANA	14/10/2018 15:45:50

Fonte: Elaborado pelo autor

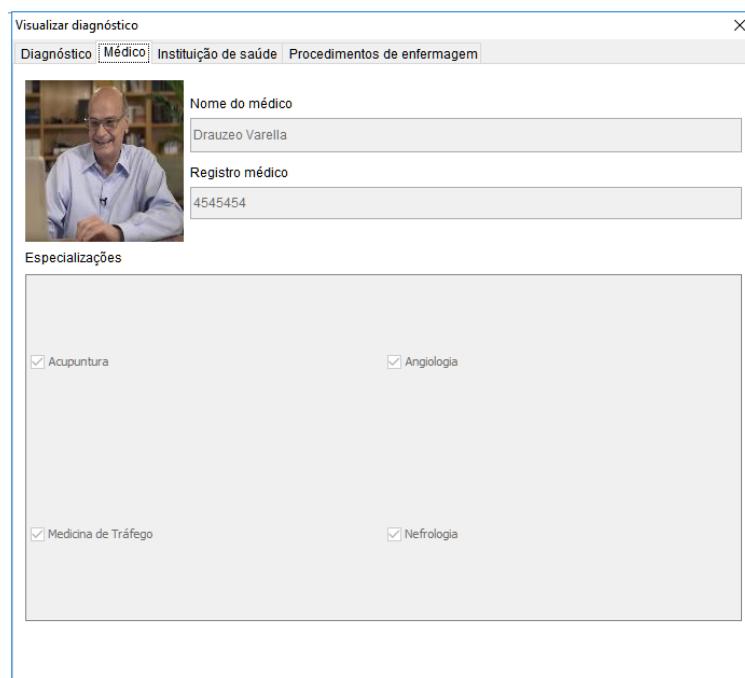
Nesta tela é possível visualizar diagnósticos já existentes, para isso é necessário que o usuário selecione um diagnóstico, assim uma nova janela se abrirá com o diagnóstico do paciente.

Figura 120 - Janela de visualização de diagnóstico



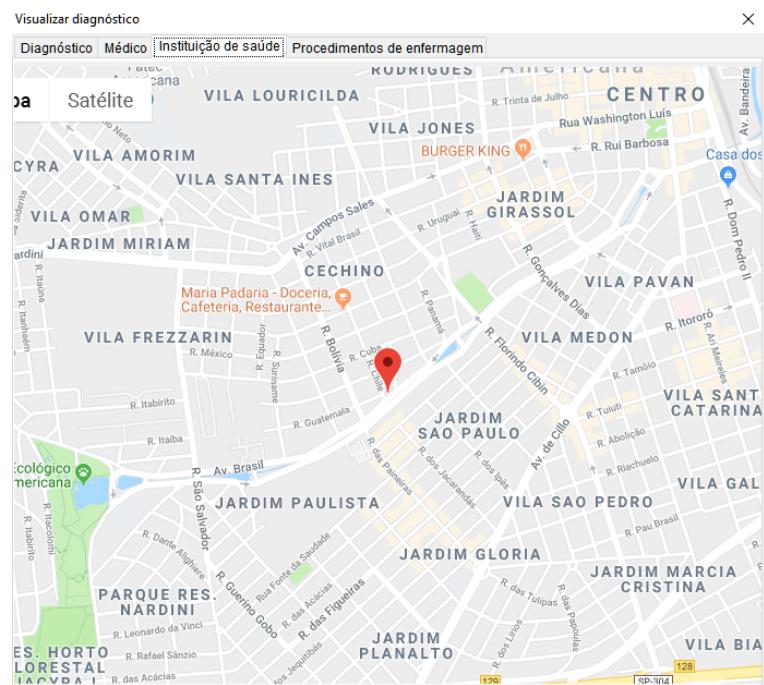
Fonte: Elaborado pelo autor

Figura 121 - Janela de informações do médico



Fonte: Elaborado pelo autor

Figura 122 - Janela da instituição de saúde



Fonte: Elaborado pelo autor

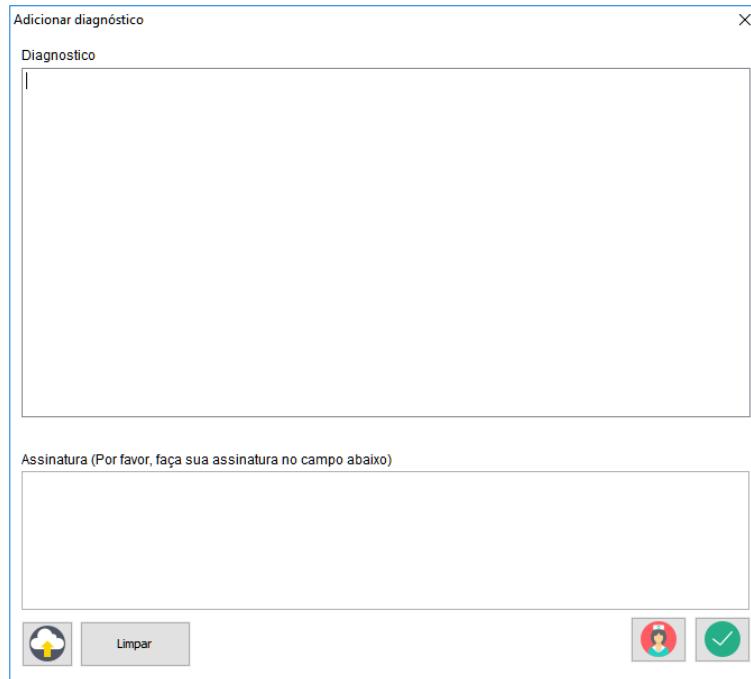
Figura 123 - Janela de procedimento de enfermagem

A interface de usuário para procedimentos de enfermagem exibe uma foto de uma enfermeira e um campo para inserir o nome da enfermeira, que está preenchido com 'Julia'. Abaixo, há uma seção para inserir procedimentos, com o texto 'guugug' digitado.

Fonte: Elaborado pelo autor

Na tela de diagnósticos também é possível fazer a adição de diagnósticos, para isto é necessário que o usuário clique em adicionar diagnóstico, assim uma nova janela se abrirá.

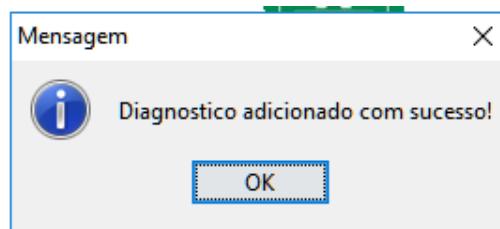
Figura 124 - Janela de adição de diagnóstico



Fonte: Elaborado pelo autor

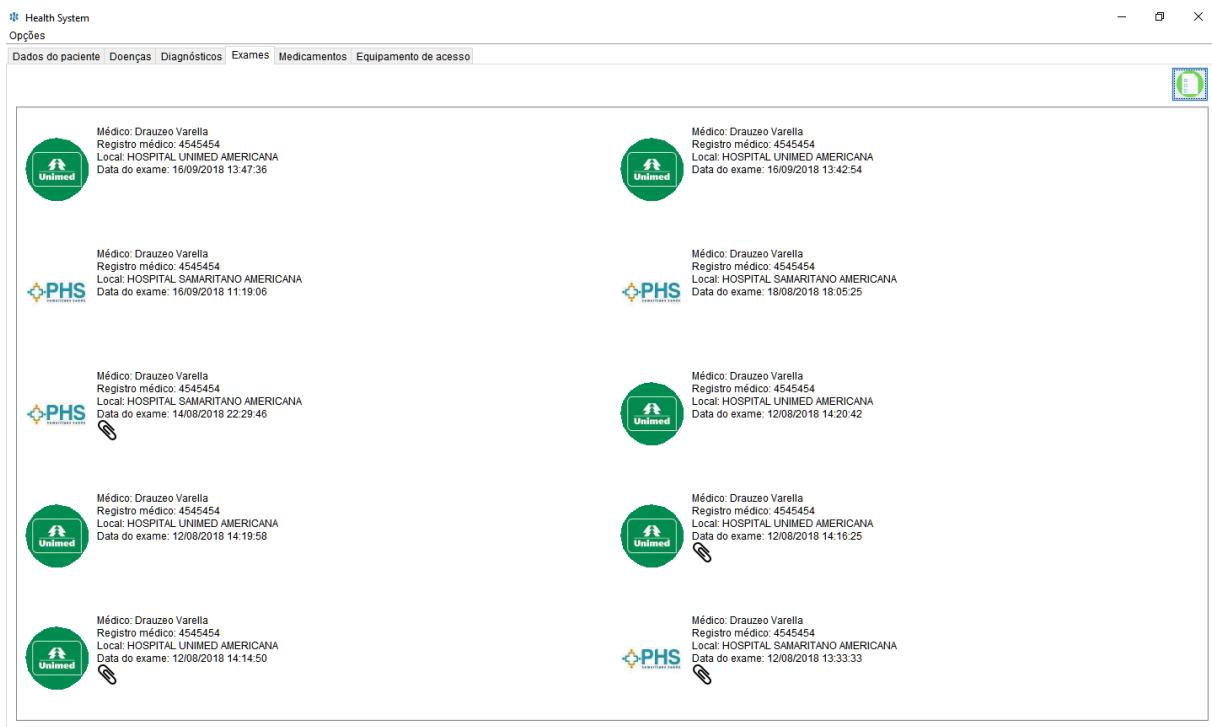
Nesta janela será solicitado que sejam preenchidos os campos diagnóstico e assinatura, e, caso seja necessário, é possível adicionar procedimentos de enfermagem também. Após o preenchimento das informações necessárias e a confirmação da ação, será exibida uma mensagem confirmando a adição do diagnóstico.

Figura 125 - Mensagem de diagnóstico adicionado com sucesso



Fonte: Elaborado pelo autor

Ao clicar na aba exames o usuário será direcionado para a tela de exames, nesta existe uma lista com todos os exames já realizados pelo paciente.

Figura 126 - Tela de exames**Fonte: Elaborado pelo autor**

Para visualizar um exame é necessário que o usuário selecione o exame que deseja visualizar, assim uma nova janela se abrirá com as informações do exame

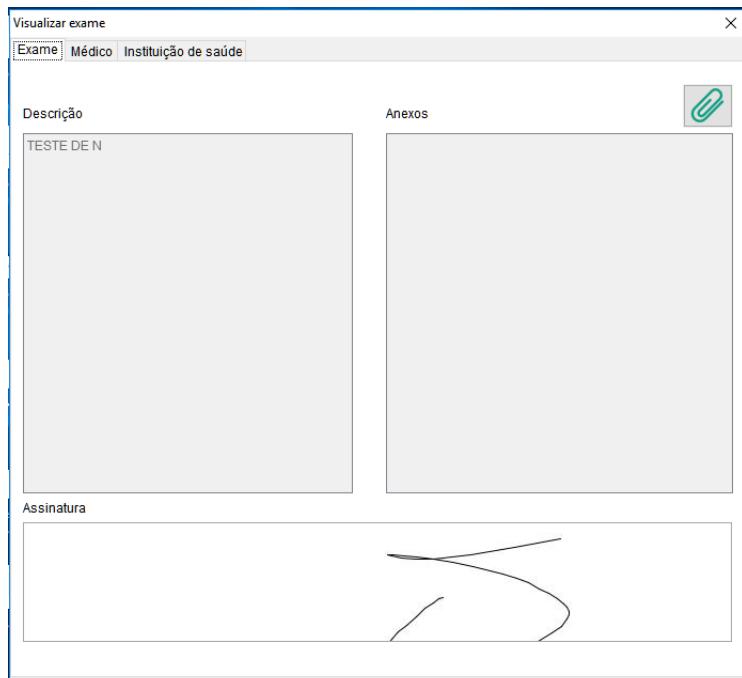
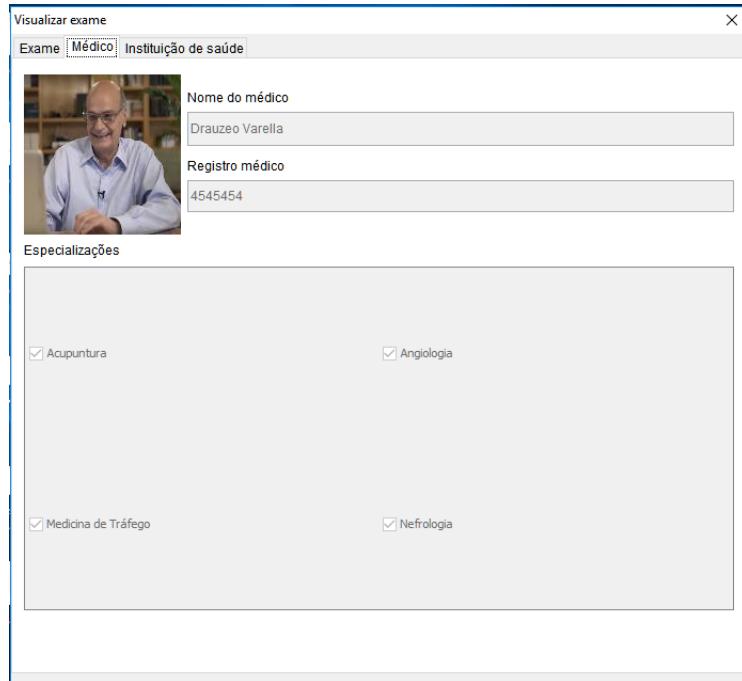
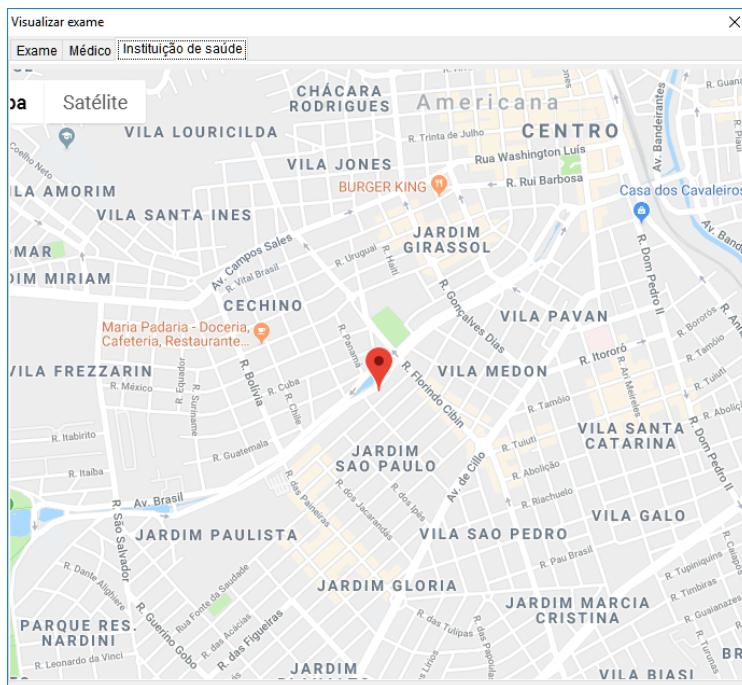
Figura 127 - Janela de visualização de exames**Fonte: Elaborado pelo autor**

Figura 128 - Janela do médico que realizou o exame



Fonte: Elaborado pelo autor

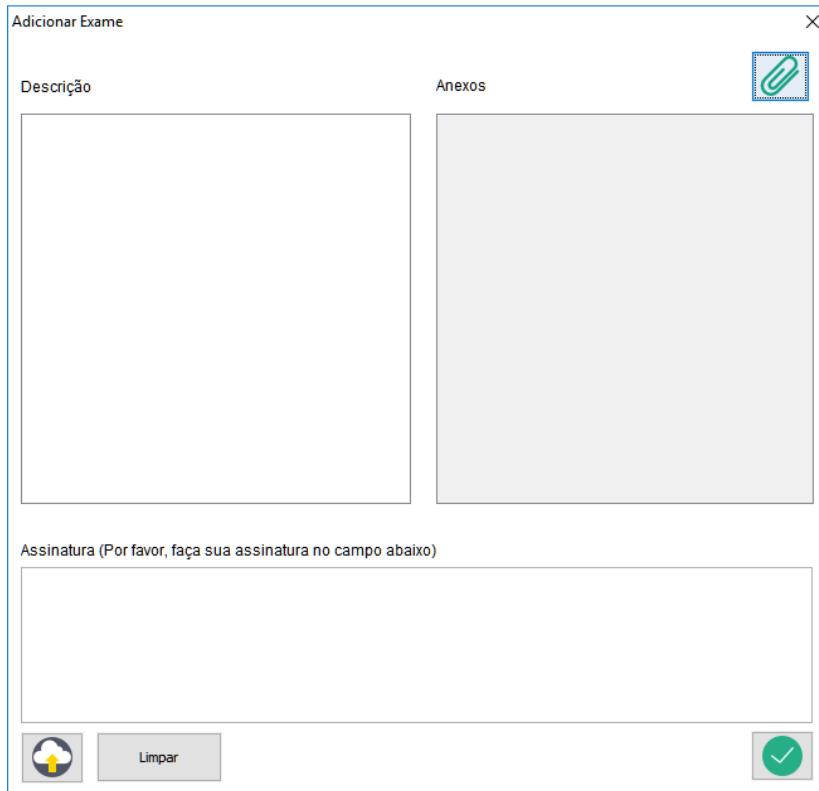
Figura 129 - Janela da instituição de saúde



Fonte: Elaborado pelo autor

Para adicionar um exame é necessário que o paciente clique em adicionar exame, assim uma nova janela se abrirá.

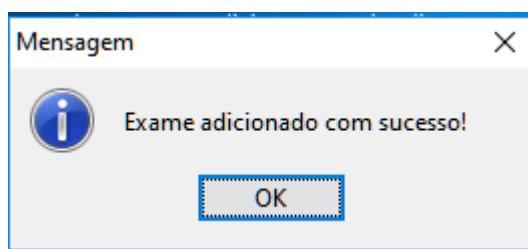
Figura 130 - Janela adicionar exames



Fonte: Elaborado pelo autor

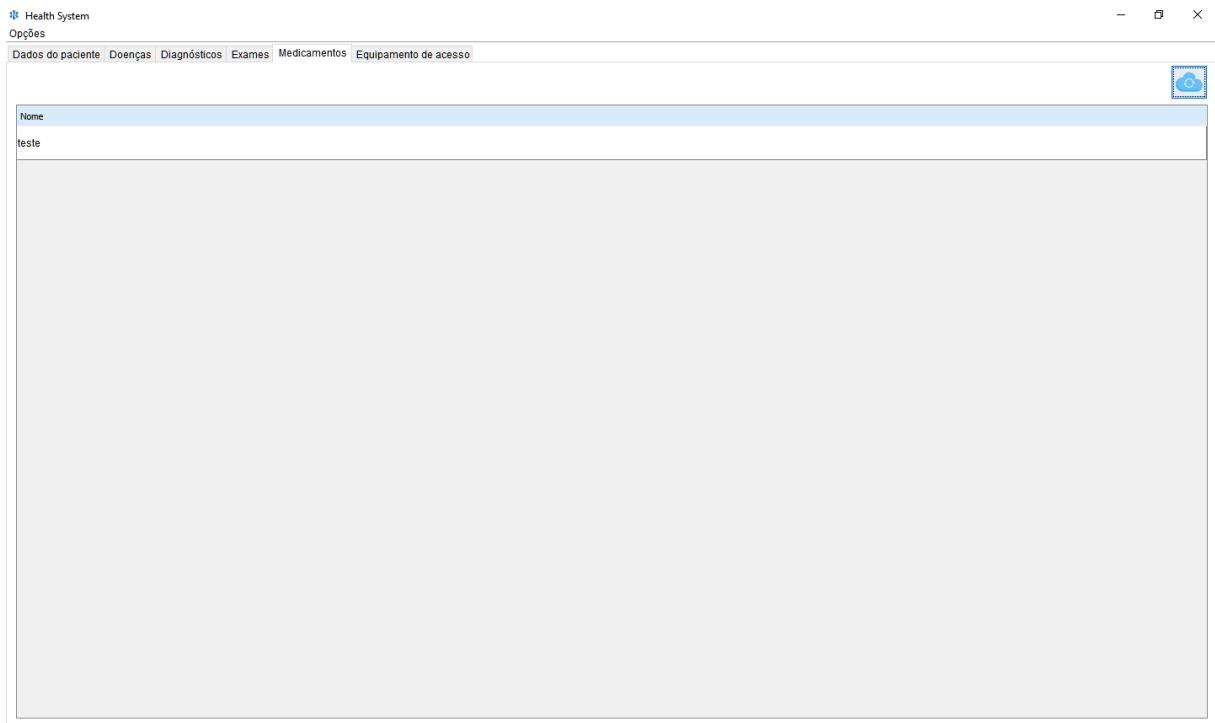
Nesta janela será necessário que o usuário adicione a descrição do exame, os anexos (caso haja algum) e a assinatura do médico, e, após realizado o preenchimento e confirmação da ação, o exame será adicionado no sistema e uma mensagem confirmando a adição do exame será exibida, o exame será adicionado à lista no sistema.

Figura 131 - Mensagem de exame adicionado com sucesso



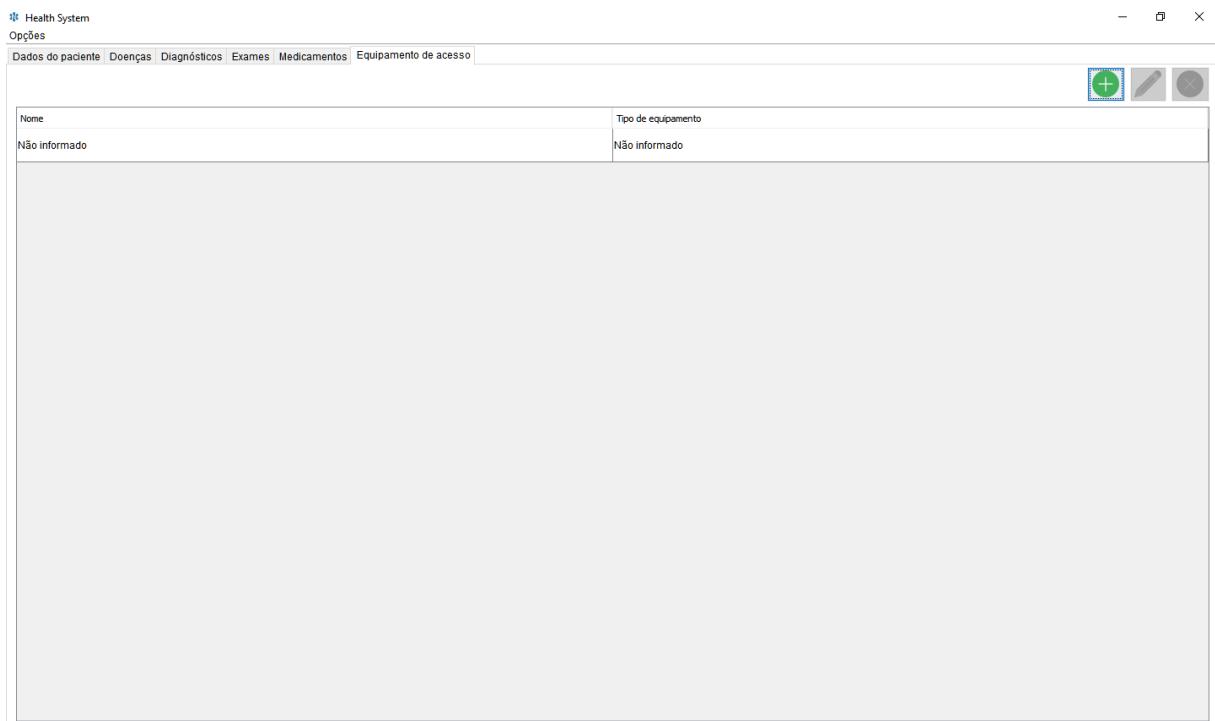
Fonte: Elaborado pelo autor

Para visualizar os medicamentos que o paciente está utilizando é necessário clicar na aba medicamentos, assim, será carregada uma lista com todos os medicamentos que paciente está utilizando.

Figura 132 - Tela de medicamentos

Fonte: Elaborado pelo autor

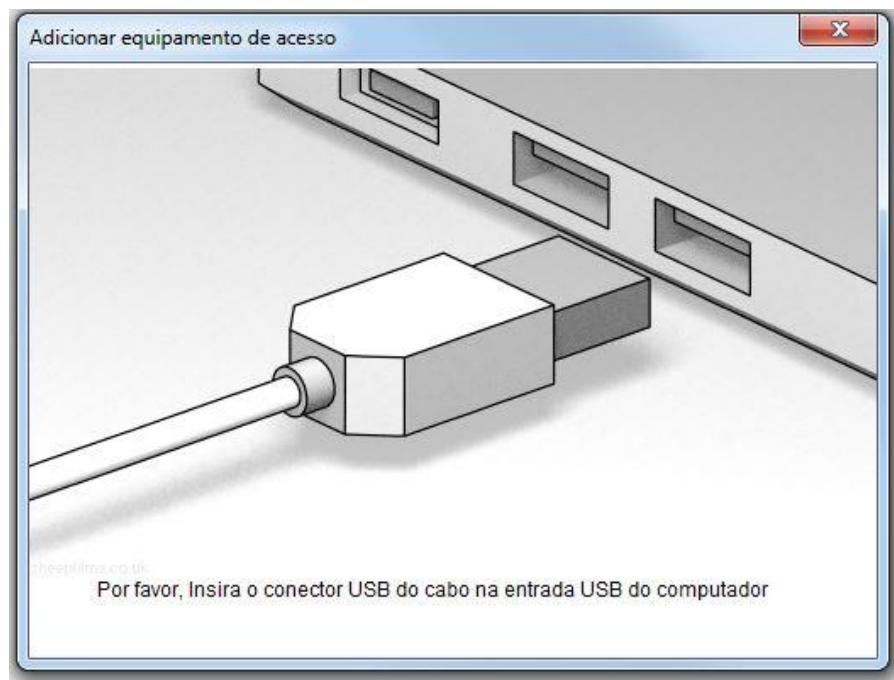
Ao clicar na aba equipamento de acesso será exibida uma lista com todos os equipamentos de acesso cadastrados.

Figura 133 - Tela de equipamentos de acesso

Fonte: Elaborado pelo autor

Para adicionar um equipamento é necessário que o usuário clique em adicionar, então será exibida uma tela solicitando que o usuário insira o equipamento de leitura (caso ele já não esteja inserido), depois será solicitado que o usuário aproxime a equipmento de acesso NFC.

Figura 134 - Janela de inserção de equipmento



Fonte: Elaborado pelo autor

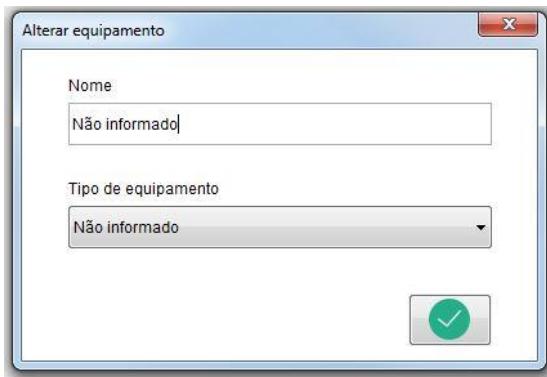
Figura 135 - Janela de solicitação do equipmento de acesso NFC



Fonte: Elaborado pelo autor

Para realizar a alteração do equipamento de acesso é necessário selecionar o equipamento e clicar em editar, uma nova janela se abrirá.

Figura 136 - Janela de edição de equipamento

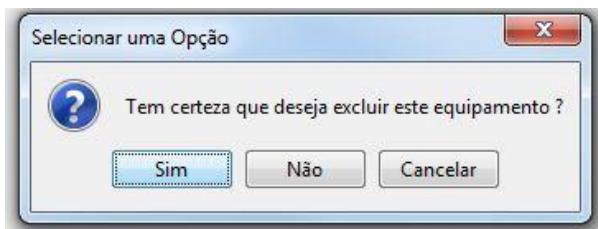


Fonte: Elaborado pelo autor

Nesta os campos nome e tipo de equipamento são editáveis. Assim que alteradas as informações são salvas no sistema e exibidas na lista de equipamentos.

Para excluir algum equipamento o procedimento é o mesmo da alteração, é necessário que o usuário selecione o equipamento que deseja excluir e clique em excluir. Uma nova janela será exibida confirmando a operação com as opções “sim” e “não”.

Figura 137 - Janela de confirmação de exclusão de equipamento de acesso



Fonte: Elaborado pelo autor

Caso o usuário clique em “sim” o equipamento será apagado do sistema, caso o usuário clique em “não” a janela será fechada e o usuário será redirecionado para a lista de equipamentos de acesso.

Voltando a lista de pacientes é possível também adicionar um paciente que ainda não possua acesso ao sistema. Para isto é preciso que o usuário clique em adicionar paciente. Assim, uma nova janela se abrirá, e nesta será solicitado o preenchimento do e-mail, nome, país e documento de identidade do paciente, e após o preenchimento e a confirmação da ação será enviado ao paciente um e-mail com uma senha provisória para acesso da conta, um código secreto de acesso e o novo paciente terá sido adicionado com sucesso.

Figura 138 - Janela de adição de paciente

Adicionar paciente

E-mail

Nome

País

Brasil

Documento de identidade

Um e-mail será enviado com sua senha temporária no fim!

Fonte: Elaborado pelo autor

Voltando para o menu de opções, selecionando minha conta é possível fazer edições no perfil do usuário.

Figura 139 - Janela de atualização de usuário médico

Atualizar usuário

Dados pessoais Dados do médico

E-mail: drauzeo@gmail.com

Tipo de usuário: 2 - Médico

Senha:

Confirmar senha:

Nome: Drauzeo Varella

País: Brasil

Documento de identidade: 775.288.360-36

CEP: 13450-144

Estado: SP

Cidade: Santa Bárbara D'Oeste

Bairro: Vila Godoy

Rua: Joaquim Felipe da Costa Machado

Número: 165

Telefone: (19) 99264-6809

Gênero: 1 - Masculino

Data de nascimento: Seg 03/05/1943

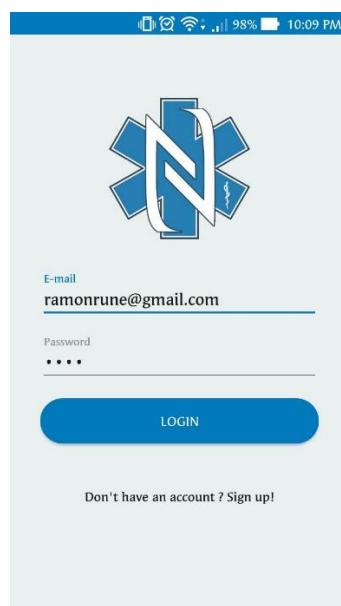
Fonte: Elaborado pelo autor

Nesta janela é possível fazer a alterações dos dados do médico, e assim que confirmados, as informações são salvas e a janela é fechada.

4.2.2 Aplicativo do paciente

O Aplicativo do paciente é um aplicativo voltado para o uso do paciente. Para poder usá-lo, primeiramente, é necessário baixá-lo na Google play e então instalar em seu dispositivo móvel. Após isso, é necessário iniciar o aplicativo e uma tela de *login* semelhante à Figura 140 aparecerá.

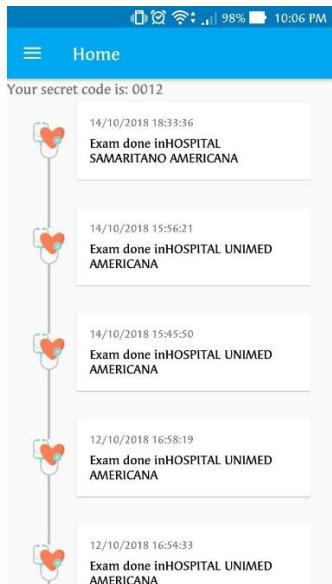
Figura 140 - Tela de autenticação



Fonte: Elaborado pelo autor

Ao realizar o *login*, o usuário será redirecionado para a tela de inicial do sistema. Nessa tela serão apresentadas todas as instituições pelo qual o paciente foi diagnosticado e os exames que realizou. No topo da imagem, é possível notar a seguinte frase: “Seu código secreto é:” O código que é mostrado logo a frente dessa frase, pode ser usado pelo(a) médico(a), para que ele possa obter suas informações de saúde.

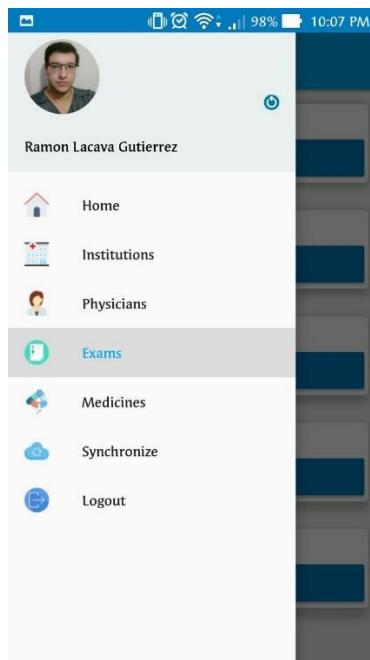
Figura 141 - Tela inicial do sistema



Fonte: Elaborado pelo autor

Caso seja clicado na barra no canto superior esquerdo, o menu será aberto.

Figura 142 - Menu



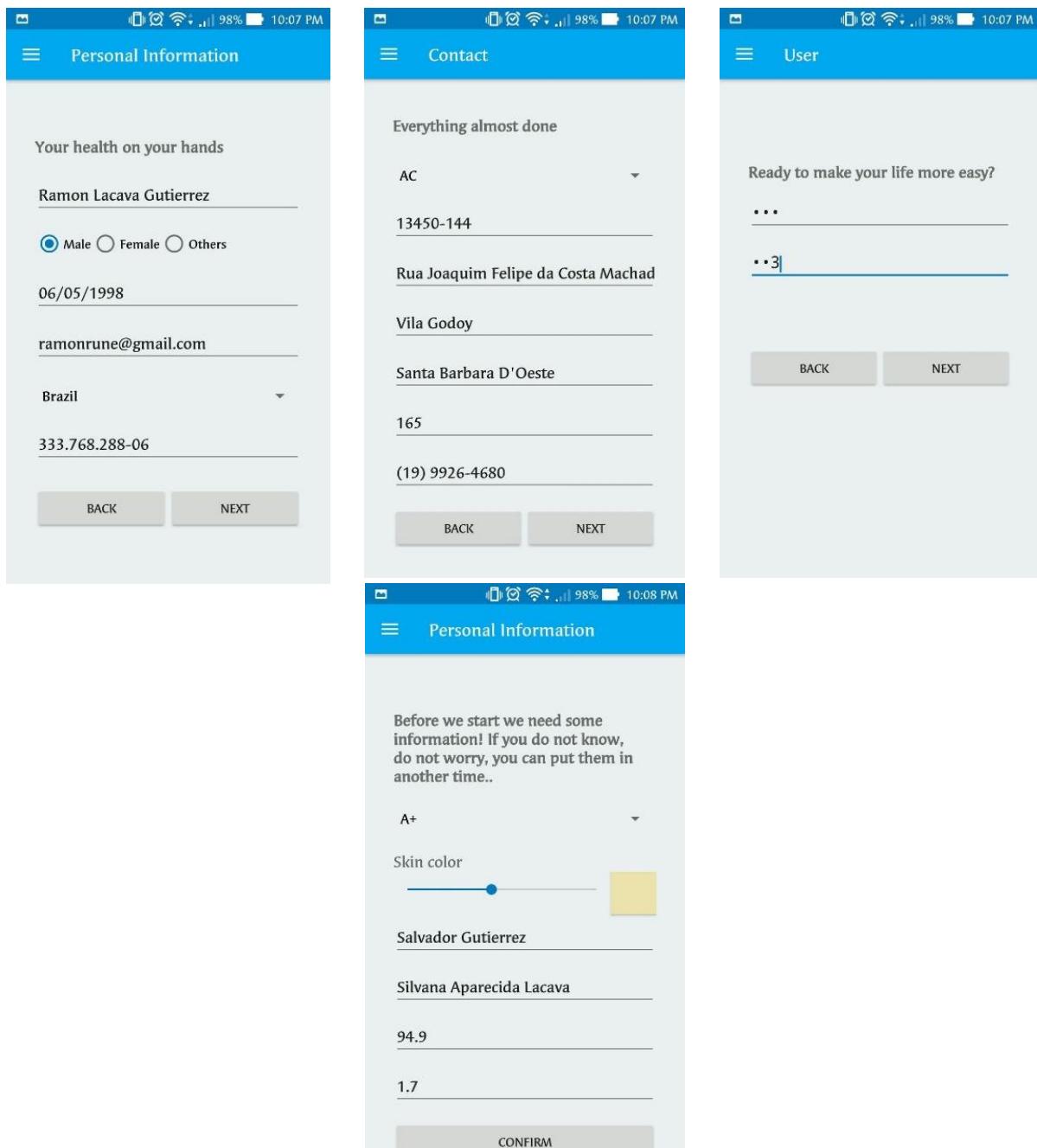
Fonte: Elaborado pelo autor

No menu serão exibidas todas as funcionalidades do aplicativo. Caso seja clicado no ícone de atualização no canto direito da foto de perfil, será aberto uma tela com o registro do paciente. No registro, é possível atualizar as informações utilizadas durante o cadastro no aplicativo.

Os passos para atualizar as informações são os mesmos para registrar uma nova conta de paciente. A última foto da Figura 143 demonstra informações

específicas, que não são obrigatorias para completar o seu registro na aplicação (é possível registrá-las em outro momento).

Figura 143 - Atualização dos dados do paciente



Fonte: Elaborado pelo autor

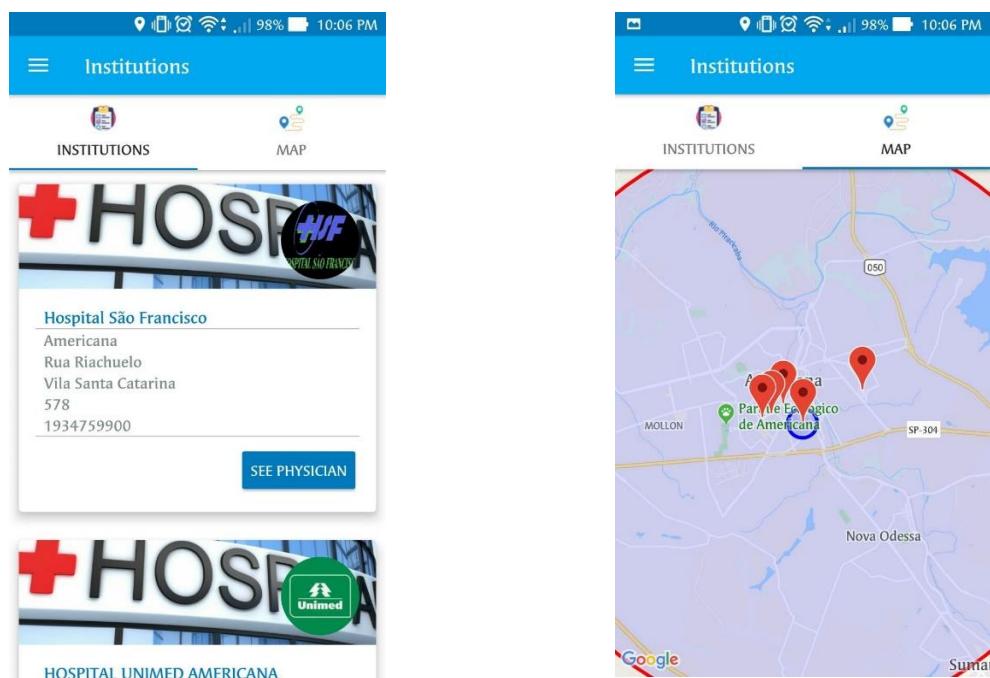
Ainda no menu, se clicar em “Instituições” irá ser mostrado as instituições que estão mais próximas. É possível vê-las em uma lista ou no mapa. O círculo azul é a mediação e o círculo maior é o hospital próximo a sua mediação (50 km). No mapa,

caso seja clicado no indicador em vermelho, é possível visualizar o nome da instituição.

Caso seja clicado em “Ver médicos”, serão apresentados todos os médicos(as) que trabalham naquela instituição.

Observação: Para utilizar todas essas funções, é necessário que o GPS esteja habilitado.

Figura 144 - Tela de instituições

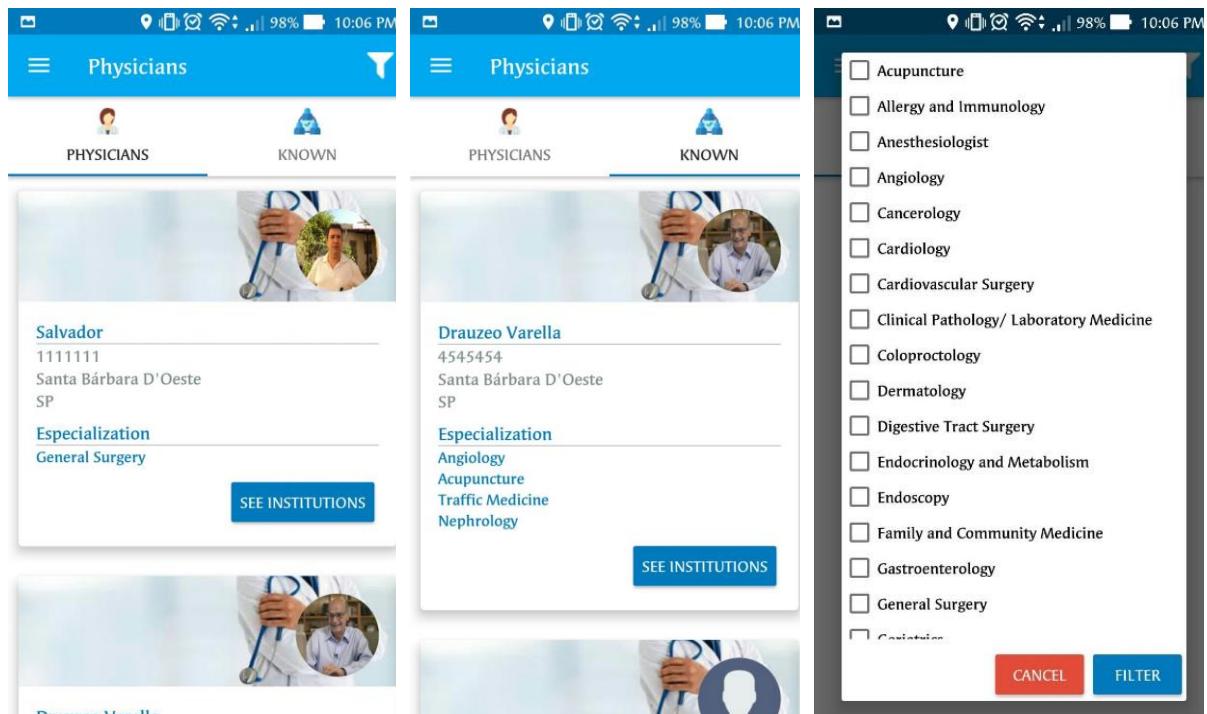


Fonte: Elaborado pelo autor

No menu, caso seja clicado em “Médicos” serão apresentados todos os médicos(as) que estão próximos e também os médicos que o paciente já conhece. Caso seja clicado no ícone de filtro, é possível filtrar médicos(as) por uma certa especialidade. Caso seja clicado em “Ver instituições” serão apresentadas as instituições que o médico(a) trabalha.

Observação: Para usar essa função é necessário que o GPS esteja habilitado em seu celular.

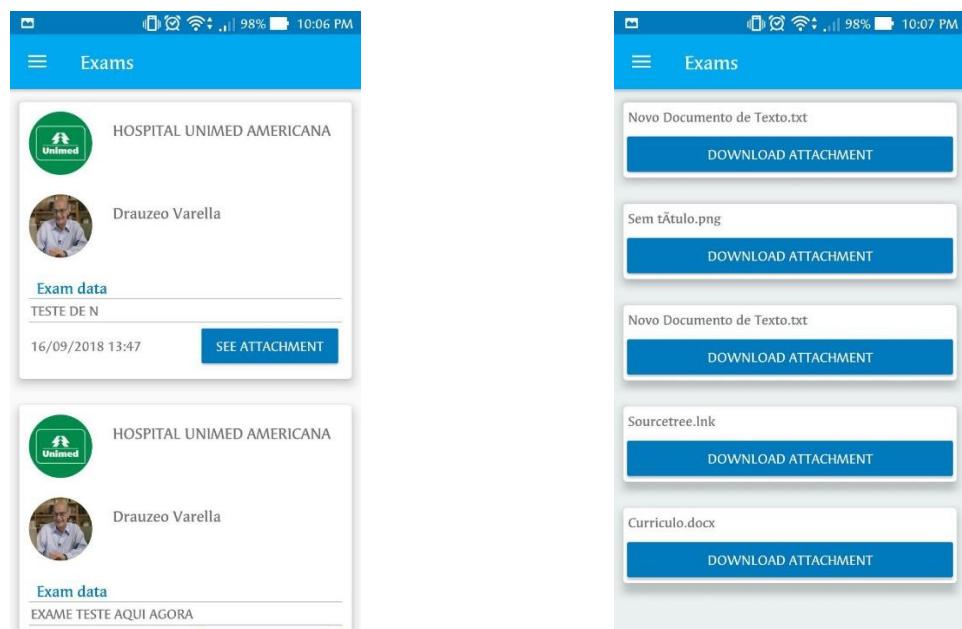
Figura 145 - Tela de médicos(as)



Fonte: Elaborado pelo autor

No menu, caso seja clicado em “Exames”, serão apresentados todos os exames que foram realizados no paciente, com suas respectivas informações, e caso seja clicado em “Visualizar anexos”, serão apresentados todos os anexos do exame, sendo possível realizar o *download*.

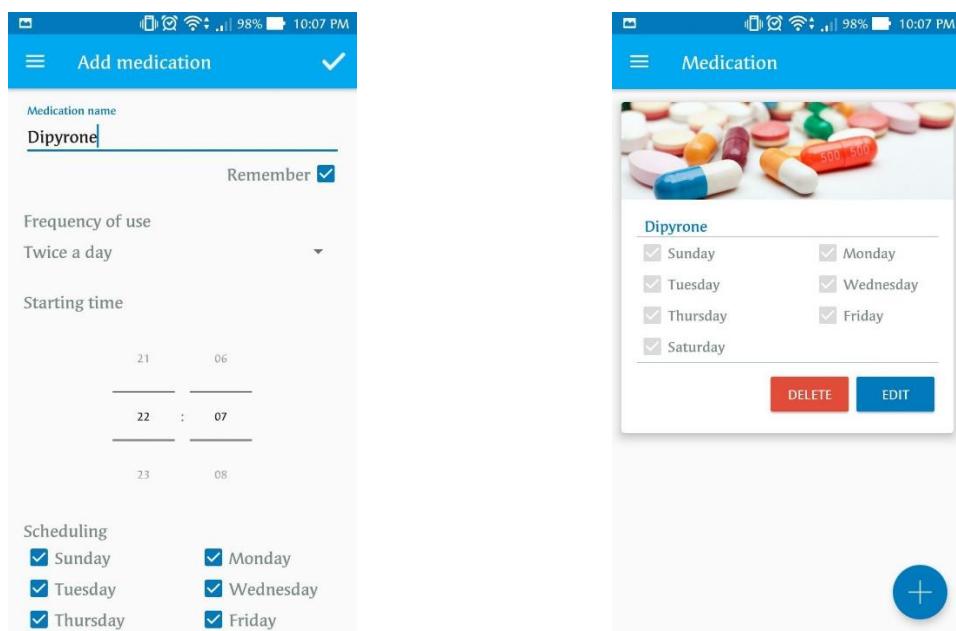
Figura 146 - Tela de exames



Fonte: Elaborado pelo autor

Caso seja clicado em “Medicamentos”, serão apresentados todos os medicamentos que o paciente registrou. Nessa tela, é possível registrar todos os medicamentos que o paciente necessita utilizar (e o dispositivo móvel irá alertar quando for necessário utilizá-lo). Para registrar um medicamento, é necessário clicar no ícone em azul de adição, no canto inferior direito. Então, uma tela de registro de medicamentos será apresentada, onde é necessário informar nome do medicamento, a frequência de uso e a hora (que é usado para alertar o paciente). Para salvar, é necessário clicar no canto superior direito (no ícone de “Ok”).

Figura 147 - Medicamentos



Fonte: Elaborado pelo autor

Após adicionar um medicamento, é possível exclui-lo ou editá-lo.

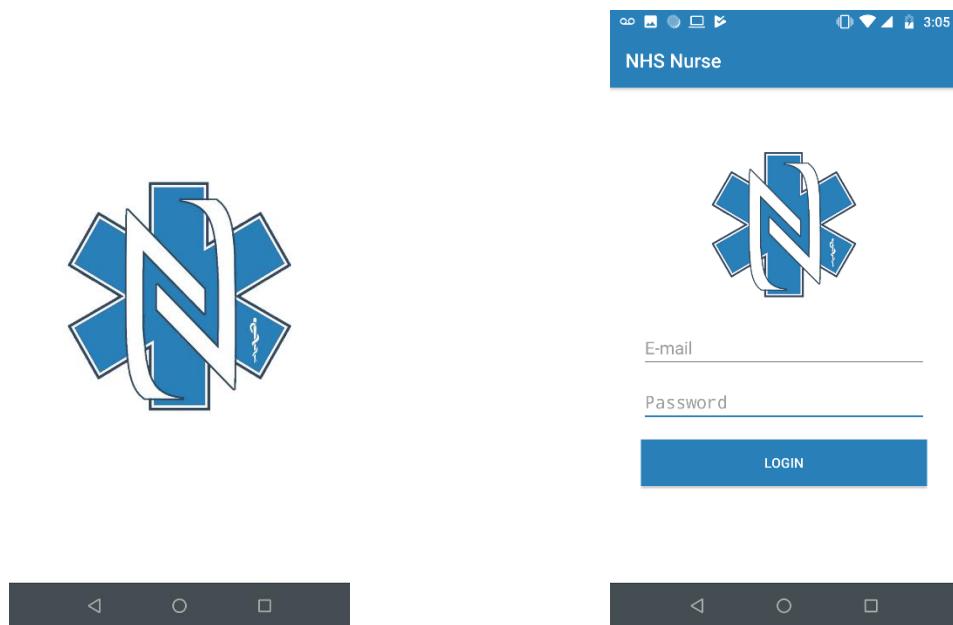
Caso seja clicado em “Sincronizar”, todas as informações do paciente serão sincronizadas com a nuvem.

Caso seja clicado em “Logout”, o paciente será desconectado do aplicativo, e será enviado para a tela de *login*.

4.2.3 Aplicativo do(a) Enfermeira

O aplicativo do(a) enfermeira é utilizado em conjunto com o sistema *desktop*. Quando um(a) médico(a) adiciona um diagnóstico para um paciente, ele pode escolher a opção de enviar o paciente para a enfermagem da instituição para que determinados procedimentos sejam realizados. Ao inicializar o sistema de enfermagem, a tela de autenticação será carregada.

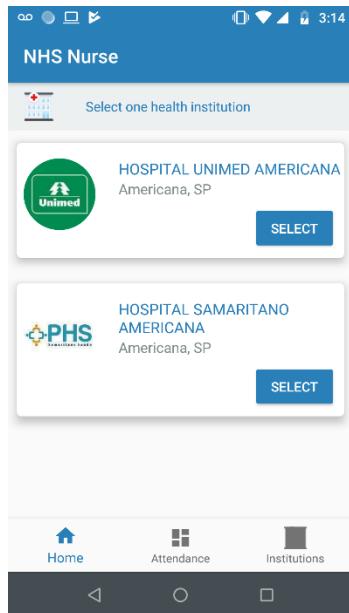
Figura 148 - Tela de carregamento e autenticação



Fonte: Elaborado pelo autor

Logo após a autenticação, o usuário será direcionado para a tela de seleção de instituições de saúde, já que um(a) enfermeiro(a) pode estar vinculado(a) a mais de uma instituição de saúde.

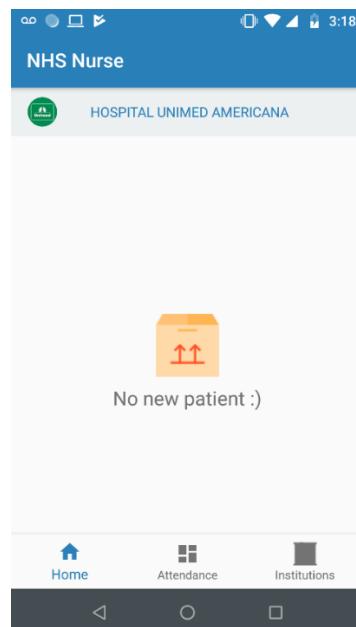
Figura 149 - Tela de seleção de instituições de saúde



Fonte: Elaborado pelo autor

Após a instituição ser selecionada, o(a) enfermeiro(a) será direcionado a tela inicial da instituição de saúde. Quando um novo paciente é direcionado a fila de procedimentos da instituição, este paciente estará na tela inicial do aplicativo. A tela de atendimento mostra todos os atendimentos em progresso. Em instituições, é possível selecionar outras instituições de saúde.

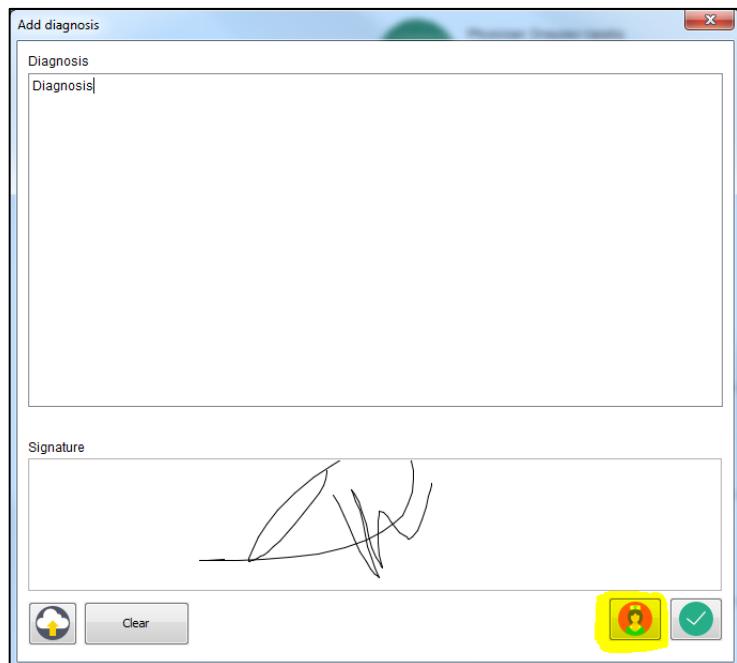
Figura 150 - Estrutura principal do aplicativo do(a) enfermeiro(a)



Fonte: Elaborado pelo autor

Para adicionar um procedimento, é necessário se autenticar no sistema desktop como um usuário médico, acessar um paciente, e no momento de adicionar um novo diagnóstico, selecionar o ícone de enfermagem.

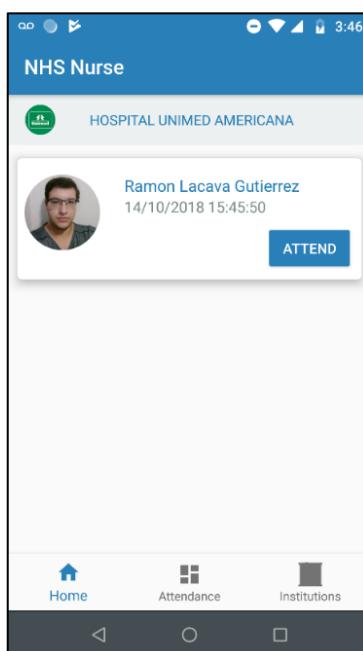
Figura 151 - Adição de diagnóstico



Fonte: Elaborado pelo autor

Assim, um novo paciente aparecerá na tela inicial do aplicativo de enfermagem.

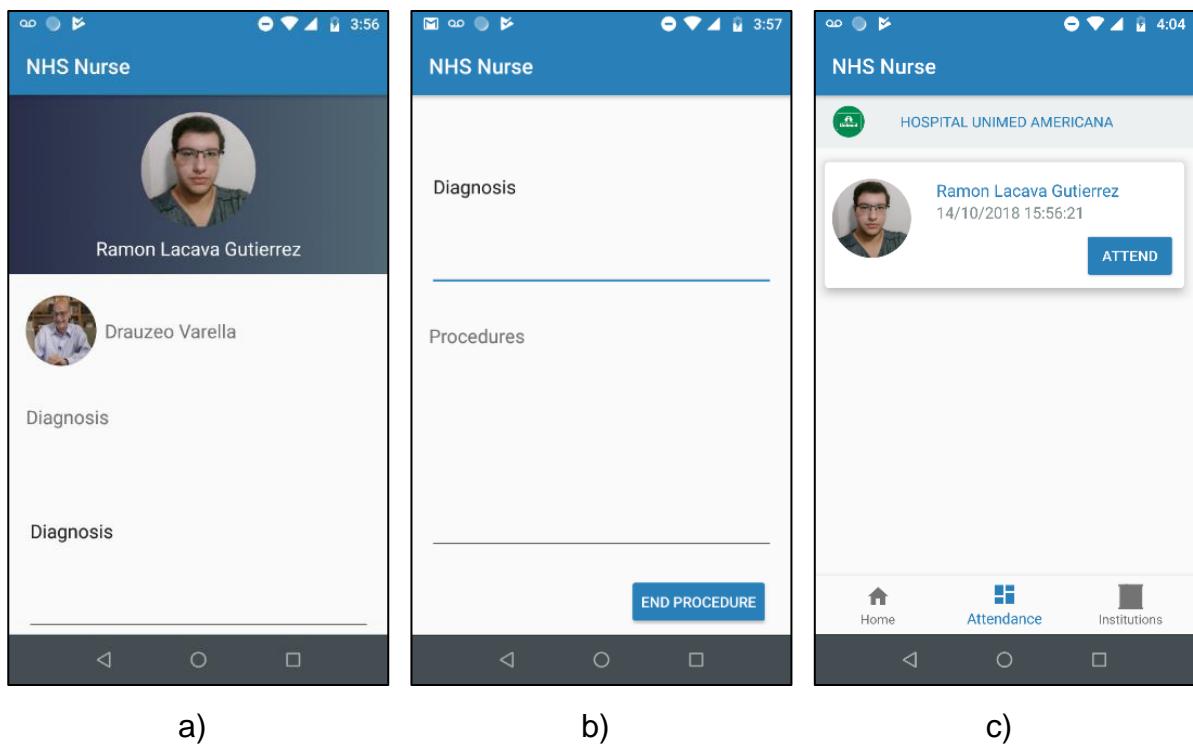
Figura 152 - Paciente na tela inicial



Fonte: Elaborado pelo autor

Caso seja clicado no botão de atender, será demonstrado uma tela com dados do(a) médico(a), do(a) paciente, o diagnóstico e os procedimentos que devem ser realizados pela enfermagem. Assim, é possível adicionar os procedimentos, ou caso seja necessário sair desta tela, todos os dados do procedimento serão salvos. Caso este segundo caso ocorra, o respectivo atendimento estará na aba de atendimento, a qual lista todos os atendimentos em progresso.

Figura 153 - Lista de atendimentos em progresso



Fonte: Elaborado pelo autor

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo final um sistema de saúde que permita que, pacientes, médicos(as), enfermeiros(as) e paramédicos(as) tenham acesso a uma plataforma que contém dados de saúde em um ambiente em nuvem, contribuindo na transparência das informações, diminuição de erros em atendimentos, melhor gestão de informações, melhoria no atendimento de atendimentos de urgência e emergência, e uma unificação de informações de saúde em uma única plataforma.

O trabalho inicialmente sofreu diversas alterações em seu planejamento, tais como modificações de requisitos, mudanças tecnológicas, alterações de como o sistema deveria funcionar, necessidade de adaptação do sistema a outros ambientes devido a parceria com a universidade da África do Sul, dentre diversos outros fatores.

A parceria com a universidade Sul Africana permitiu a obtenção de requisitos necessários para que a aplicação e o sistema desktop pudesse funcionar corretamente. Para a obtenção dos requisitos, foi realizado a divisão dos alunos africanos em três grupos, sendo cada um responsável por uma vertente da área da saúde (enfermeiro, médico, paciente).

A universidade Sul Africana forneceu um grande choque cultural durante a obtenção dos requisitos, mostrando as diferenças entre o Brasil e a África do Sul. Essas diferenças, mesmo sendo culturais, deram impactos no desenvolvimento deste trabalho, porém foram sempre resolvidas com reuniões e diálogos com os alunos Africanos, para que fosse possível sempre entendê-los(as) e assim amenizar o impacto causado.

A partir de todas estas modificações e dados coletados, foi desenvolvido um planejamento mais estratégico, eficaz e preciso, o qual foi utilizado até o final do projeto. A etapa inicial foi cumprida no momento da concretização do planejamento e da definição clara das funções que o sistema deveria realizar, juntamente a definição de como seria a metodologia da equipe, a divisão de papéis e como os processos dentro da equipe deveriam ser realizados.

Logo na sequência, se iniciou o processo de desenvolvimento, se iniciando com o sistema *desktop*, aplicativo do paciente e por fim o aplicativo do(a) enfermeiro(a).

Foram realizadas reuniões a cada quinze dias para verificar se os objetivos dados no início do processo foram cumpridos, o que poderia melhorar ou o que não funcionou, e por fim, planejar a próxima entrega. Durante o desenvolvimento, foi necessário o estudo de algumas tecnologias para melhor compreensão do funcionamento das mesmas. O desenvolvimento foi um processo de constante aprimoramento dos sistemas, tais como aprimoramento de requisitos, melhoria de funcionamento, alterações em interfaces, dentre outros.

Durante o desenvolvimento, houveram dificuldades, tanto a níveis de adaptação do sistema a novos requisitos gerados pela universidade sul africana, quanto em necessidade de um maior estudo em relação a tecnologia aplicada. Porém, todas as dificuldades foram sanadas através de reuniões, comunicações entre os membros da equipe e leitura da documentação das tecnologias.

As interfaces de usuário foram desenvolvidas de modo a serem simples e fáceis de serem utilizadas, seguindo-se padrões de posicionamento, ícones, plataformas, e se baseando na facilidade de uso.

O aplicativo cumpre o que promete, fornecendo um sistema no qual todos os usuários possuem acesso a plataforma de saúde unificada. Os aplicativos do paciente e do(a) enfermeiro(a) foram publicados na Google Play e estão em processo de análise para melhoria.

Como possíveis trabalhos futuros, pode-se apontar: Desenvolvimento do aplicativo do paramédico; Desenvolvimento de sistema de emergência; Desenvolvimento de pulseira de saúde; Sistema de detecção de doenças; Sistema de alerta de doenças em áreas geográficas.

REFERÊNCIAS

ANDROID. Disponível em: <https://www.android.com/intl/pt-BR_br>. Acesso em: 07 de mar. 2017.

Android Developers. **ARQUITETURA da plataforma.** Disponível em: <<https://developer.android.com/guide/platform/>>. Acesso em: 28 mar. 2018.

Apache Maven, **Welcome to Apache Maven.** Disponível em <<https://maven.apache.org/>>. Acesso em: 10 nov. 2018.

Apache Tomcat, **Apache Tomcat.** Disponível em <<https://tomcat.apache.org/>>. Acesso em: 10 nov. 2018.

Arduino. **Arduino 1.8.7.** Disponível em: <<https://www.arduino.cc/en/Main/Software>>. Acesso em: 15 nov. 2018.

ASTAH. **UML Features.** Disponível em: <<http://astah.net/features/uml-features>>. Acesso em: 15 nov. 2018.

BARRETT, S. F. **Arduino microcontroller processing for everyone! third edition**
Barrett. 3. ed. Wyoming: Morgan & Claypool Publishers, 2013. v. 43

CloudAMQP. **Documentation: Getting started.** Disponível em: <<https://www.cloudamqp.com/docs/index.html>>. Acesso em: 10 nov. 2018.

Conselho Federal de Medicina. **Prontuário médico.** Disponível em: <http://portal.cfm.org.br/index.php?option=com_content&id=20462:prontuario-medico/>. Acesso em: 31 mar. 2018.

Conselho Regional de Medicina do Estado de Santa Catarina. **Manual de orientação ética e disciplinar.** Disponível em: <<http://www.portalmedico.org.br/regional/crmsc/manual/parte3b.htm>>. Acesso em: 31 mar. 2018.

COSKUN, Vedat; OK, Kerem; OZDENIZCI, Busra. **Near field communication (NFC): from theory to practice.** John Wiley & Sons, 2011.

CUNHA, Alessandro. **NFC (Near Field Communication) – Aplicações e uso.** Disponível em: <<https://www.embarcados.com.br/nfc-near-field-communication/>>. Acesso em: 28 mar. 2018.

DRAGO, I. et al. Inside Dropbox: Understanding Personal Cloud Storage Services. **IMC '12 Proceedings of the 2012 ACM conference on Internet measurement conference**, p. 481–494, 2012.

FIELDING, Roy T.; TAYLOR, Richard N. **Architectural styles and the design of network-based software architectures.** Doctoral dissertation: University of California, Irvine, 2000.

FRITZING. **Get started.** Disponível em: < <http://fritzing.org/learning/get-started/>>. Acesso em: 15 nov. 2018.

GOOGLE PLAY. Disponível em: <<https://play.google.com>>. Acesso em: 22 maio 2017.

GUIDO, L. de A. Aspectos éticos da assistência de enfermagem ao cliente cirúrgico ambulatorial no centro cirúrgico e na sala de recuperação anestésica – reflexões. In: **Congresso Brasileiro de Enfermagem em Centro Cirúrgico**, 2, São Paulo, Julho 1995. *Anais*. São Paulo, p. 103-107, 1995.

KOUROU, Konstantina et all. **Machine learning applications in cancer prognosis and prediction.** Computational and Structural Biotechnology Journal, Grécia, v. 1, n. 1, p.1-10, 2014.

Medication Errors: Technical Series on Safer Primary Care. Geneva: World Health Organization; 2016. Licence: CC BY-NC-SA 3.0 IGO.

MENDIOLA, José. **Android já é o sistema operacional mais usado do mundo.** Disponível em: <https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html>. Acesso em: 28 mar. 2018.

Microsoft. **Baixar o SQL Server Management Studio (SSMS).** Disponível em: <<https://docs.microsoft.com/pt-br/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>>. Acesso em: 10 nov. 2018.

Microsoft. **Guia de introdução para desenvolvedores do Azure.** Disponível em: <<https://docs.microsoft.com/pt-br/azure/guides/developer/azure-developer-guide>>. 2017. Acesso em: 10 nov. 2018.

Netbeans. **Netbeans IDE – The Smarter and Faster Way to Code.** Disponível em <<https://netbeans.org/features/index.html?print=yes>>. 2018. Acesso em: 10 nov. 2018.

Oracle. **O que é o Java?** Disponível em: <https://www.java.com/pt_BR/about/whatis_java.jsp?bucket_value=desktop-chrome66-windows10-64bit&in_query=no>. Acesso em: 28 mar. 2018.

Portal Hospitais Brasil. **Pesquisa Global da GE Healthcare mostra importância dos avanços tecnológicos para a área da saúde.** Disponível em: <<http://portalhospitaisbrasil.com.br/pesquisa-global-da-ge-healthcare-mostra-importancia-dos-avancos-tecnologicos-para-a-area-da-saude/>>. Acesso em 30 março 2018.

PRESSMAN, R.S. **Engenharia de Software:** Uma abordagem Profissional. 7^a Edição, São Paulo, Editora: Makron Books, 2011.

Pusher. **Channels**. Disponível em: <<https://pusher.com/docs>>. Acesso em: 10 nov. 2018.

Quackit. **Introduction to SQL Server 2017**. Disponível em: <https://www.quackit.com/sql_server/sql_server_2017/tutorial/sql_server_2017_introduction.cfm>. Acesso em: 10 nov. 2018.

RabbitMQ. **Introduction**. Disponível em: <<https://www.rabbitmq.com/tutorials/tutorial-one-java.html>>. Acesso em: 10 nov. 2018.

ROCHA, Elivania Costa de Almeida. **Atuação da enfermagem em urgências e emergências**. Conteúdo Jurídico, Brasília-DF: 10 dez. 2012. Disponível em: <<http://www.conteudojuridico.com.br/?artigos&ver=2.41069&seo=1/>>. Acesso em: 31 mar. 2018.

RODRIGUEZ, Alex. RESTful Web Services: The basics. **IBM Developer works**, 2015. Disponível em <<https://www.ibm.com/developerworks/library/ws-restful/index.html>>. Acesso em 27 abr. 2018.

SCHIAVINATO, Cristiane. **Entrevista efetuada no dia 24 de março de 2018**. Americana/SP, 2018.

SCHILD'T, Herbert; MAYER, Roberto Carlos. **C completo e total**. 1997.

SCRUM. **Guia do SCRUM** : Um guia definitivo para o Scrum : As regras do jogo. 2013. 19p. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em 13 Jun. 2018.

SOMMERVILLE, I. **Engenharia de Software**. 8ª Edição. Editora: Pearson Addison-Wesley. São Paulo, 2007.

TIEGHI, Ana. **Tecnologia pode ser aliada da saúde**. Disponível em: <<http://www.usp.br/espacoaberto/?materia=tecnologia-pode-ser-aliada-da-saude/>>. Acesso em 30 março 2018.

TORRES, Victor. **NFC: Near Field Communication**. Disponível em: <https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2011_2/torres/index.php?file=kop1.php>. Acesso em: 28 mar. 2018.

UMBC. **Introduction to eclipse**. Disponível em: <<https://www.csee.umbc.edu/courses/undergraduate/341/fall08/Lectures/Eclipse/intro-to-eclipse.pdf>>. Acesso em: 10 nov. 2018.

VMEDU INC. **SCRUM Body of Knowledge (SBOK Guide)**. [s.l: s.n.].

APENDICE

APENDICE A – Equipes da universidade de Durban

APENDICE B – Dados levantados sobre algumas questões realizadas com a universidade de Durban

APENDICE C – Aplicativos na Google Play

APENDICE D – Reuniões

APENDICE A – Equipes da universidade de Durban

Tabela 77 - Equipes

Grupo 1	Natália Akina Uesugi
Mxolisi Mncube	mncube355@gmail.com
Zanele Gqojana	mgqojana@yahoo.com
Pathiswa Tolobisa	pattolobisa@gmail.com
Mfanelo Mendwana	mfanelomendwana@gmail.com
Phumza Gugwana (Team Leader)	pumzagugwana@gmail.com
Lungisa Bonga	lungisa.bonga@yahoo.com
Thandeka Mchunu	mchunuthandeka08@gmail.com
Nalini Govender	nalini.govender.dy@gmail.com
Grupo 2	Ramon Lacava Gutierrez Gonçales
Thembisa Mahlamvu	mahlamvutembisa@gmail.com
Thabani Msomi	thabanimsomi071@gmail.com
Dolly Nzoyiya	dollynz68@gmail.com
Ntuthu Sinama	nontuthuzelosinama@gmail.com
Zethu Nkabinde (Team Leader)	ze2kas@gmail.com
Nomzamo Nzama	nomzamoprudence@yahoo.com
Thabani Mhlongo	thabani.thabani.mhlongo@gmail.com
Israel Krishnee	keegan.gungarajh@za.man-mn.com
Grupo 3	Leonardo Martins de Oliveira
Lihle Dlamini (Team Leader)	lihlezzt@gmail.com
Ntombesizwe Tshingilane	ttsizwe2@gmail.com
Zama Mzobe	maxetxet@gmail.com
Thandi Ngcobo	thandy.tn54@gmail.com
Busi Shange	busishange131@gmail.com
Mpilo Zondi	mpilwenhlezondi@icloud.com
Mbalenhle Shange	zoeshange10@gmail.com
Nonto Zibula	nontozib7@gmail.com
Annes Morris	annesm213@gmail.com

Fonte: Elaborado pelo autor

APENDICE B – Dados levantados sobre algumas questões realizadas com a universidade de Durban

Questão: No Brasil há dois tipos de documentos, um para registo de pessoas físicas e outro para registro de organizações (pessoas jurídicas). Há dois tipos de registro no caso de pessoas físicas, um chamado CPF e outro chamado RG. O CPF não se repete, enquanto o RG pode se repetir. Um exemplo de CPF é 043.359.860-33. O registro de pessoa jurídica é chamado de CNPJ, e um exemplo é 94.585.081/0001-82. Quais documentos são utilizados na África do Sul para a identificação de pessoas e organizações? O número dos documentos se repete? Existe um número único para pessoas e organizações que não se repete?

Resposta: Atualmente, a África do Sul está lançando um programa universal chamado HPRS. HPRS significa Sistema de Registro de Pacientes de Saúde, que será exclusivo para cada paciente, como o número de identificação do paciente. Este programa propicia um grande salto de facilidade para todas as instituições de saúde. O sistema é novo, é um sistema on-line e não é necessária a utilização de papel. Este sistema ajudará a manter registros de saúde para todos os pacientes. No momento, temos também o TIER.NET, que é um sistema para manter todos os registros de pacientes com HIV e TB. Este sistema é menos complexo que o HPRS.

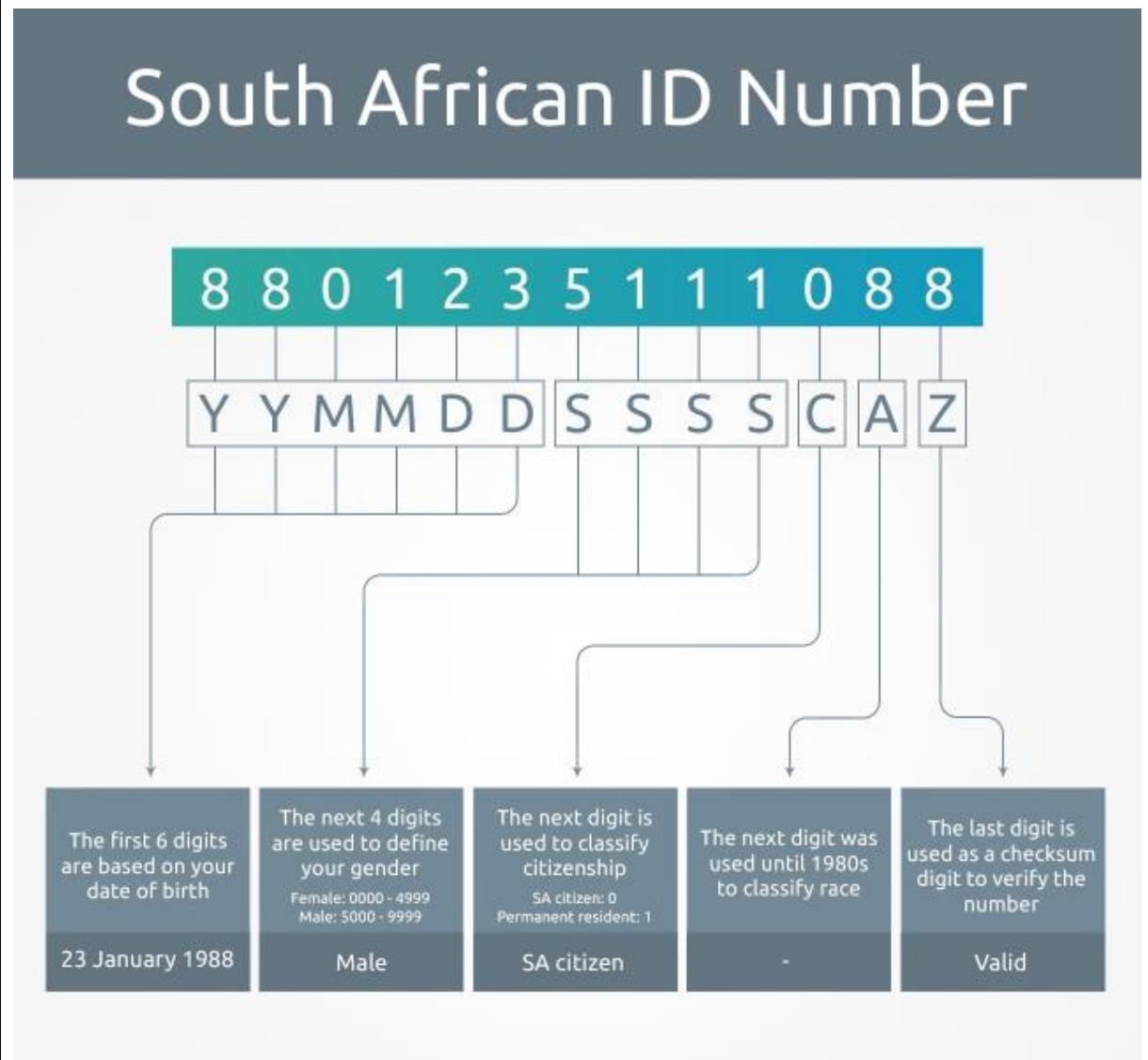
Na África do Sul, há um número de identificação relacionado ao CPF:

Número de identificação (chamado de *Identification number*)

Exemplo: 8801235111088.

Formato: YYMMDDSSSSCAZ.

Figura 154 - Explicação sobre o código de identificação sul africano



Fonte: Elaborado pelo autor

Também há um número de identificação relacionado ao CNPJ:

Registor de empresa (chamado de *company register*)

Exemplo: 1354/452648/88

Formato: XXXX/XXXXXX/XX

Questão: Nas instituições de saúde, como hospitais e clínicas, enfermeiras e médicos podem acessar a internet? Pois aqui no Brasil existem algumas limitações sobre isso. Como funciona na África do Sul?

Resposta: Aqui existem alguns locais que eles possuem acesso, e outros que não possuem, porém não é 100% aplicável.

Questão: Aqui no brasil nós possuímos um código para identificar enfermeiras, chamado de COREN. Como isso funciona na África do Sul?

Resposta: Os números que nós possuímos nós obtemos de nossos corpos de registro. Estes corpos são em sua maioria para profissionais. Enfermeiras são registradas com o no conselho Sul Africano de enfermagem (SANC), e cada enfermeira possui seu número SANC, o qual é único. O código SANC possui 8 dígitos, por exemplo, 16085961.

Questão: Quais são as divisões na área de enfermagem? (como por exemplo, enfermeira assistente, técnica em enfermagem, etc). Quais são as classificações que uma enfermeira pode possuir?

Resposta: Uma enfermeira pode ser registrada, assistente ou inscrita. As enfermeiras no hospital podem se especializar em pediatria, oftalmologia, unidade de terapia intensiva, obstetrícia avançada, cuidados com a saúde, enfermagem ocupacional, enfermagem renal, etc.

Questão: Há algum documento similar ao CRM?

Resposta: Médicos são referenciados a seus conselhos médicos. Eles também são registrados em seus respectivos conselhos médicos, e também possuem códigos de identificação únicos.

Os médicos possuem o código de prática, que é um código que possui 7 caracteres.

Questão: Quais tipos de dados são importantes em situações de urgência e emergência?

Resposta: Basicamente, sinais vitais são importantes, juntamente ao histórico médico, se o paciente utiliza alguma medicação, se possui alérgica, etc.

Questão: Qual a estrutura de um prontuário médico?

Resposta: Histórico medico, cirúrgico, familiar, diagnósticos, documentos de exames, avaliações, etc.

O prontuário na capa deve ter nome do paciente, sobrenome, idade, sexo, número de registro e alergias. O conteúdo possui as anotações, sendo necessário descrever-las de acordo com a queixa do paciente.

Questão: Quem pode escrever nos arquivos de pacientes?

Resposta: Na África do Sul todos temos direito a escrever nos arquivos de paciente.

Questão: Quais tipos de especialização um médico pode possuir?

Resposta: Algumas especializações médicas: Pediatria, medicina interna, oncologia, dermatologia, obstetrícia, etc.

Questão: Qual a linguagem padrão dos computadores e celulares na África?

Resposta: O padrão utilizado em ambos é o inglês.

Questão: Há alguma lista de medicamentos oficiais na África do Sul?

Resposta: Sim, é chamado de Lista de medicamentos sul africanos essenciais. O último ano de lançamento da mesma é de 2014, porém a mesma possui um aplicativo que trouxe algumas atualizações.

APENDICE C – Aplicativos na Google Play

Aplicativo do paciente



Aplicativo do(a) enfermeiro(a)



APENDICE D – Reuniões

Ata de reunião – 01/2018

Data: Quinta-feira, 16/08/2018

Hora: 10:30

Local: Skype

Participantes:

Leonardo Martins de Oliveira – Estudante da FATEC Americana

Natália Akina Uesugi – Estudante da FATEC Americana

Ramon Lacava Gutierrez Gonçales – Estudante da FATEC Americana

Dudu Sokhela – Professora da Universidade de Durban

Pauta: Demonstração do progresso do sistema desktop(grupo 3), demonstração do progresso aplicativo paciente (grupo 3), demonstração do progresso aplicativo médico (grupo 3), demonstração do progresso do sistema desktop(grupo 2), demonstração do progresso do sistema Android (grupo 2), demonstração do progresso do sistema Android (grupo 1), explicação do sistema por um todo (grupo 1).

Demonstração do progresso do sistema (grupo 3)

Responsável: Leonardo Martins de Oliveira

Ao iniciar a reunião, foi proposto por Leonardo começar mostrando o sistema desktop e depois o aplicativo mobile, do qual todos concordaram. Foi mostrado no sistema de desktop suas funcionalidades como ADMIN, mostrando o que ele pode fazer no sistema. Em seguida foi mostrado a interface para médico do sistema. Foi mostrado a eles a parte do leitor NFC, e perguntaram o que era e como funcionaria. Foi explicado que seria uma pulseira que seria passado em um leitor que está sobre desenvolvimento, mas entenderam bem como será o funcionamento. Leonardo solicitou sugestões, ou até algo que achem interessante ter no sistema e pediram para que fosse enviado a eles o sistema. Leonardo disse que isso ainda não seria possível, pois o mesmo está em fase de desenvolvimento e ainda falta várias funções a serem colocadas no sistema. Nesse meio tempo, solicitaram então que fosse enviado fotos do sistema para que pudessem avaliar e dar sugestões.

Foi mostrado a tela de login e as opções de menu. Em seguida foi mostrado o cadastro de paciente, onde, me informaram como é formado o número de identidade, código postal.

Foi mostrado o login de médico e suas funcionalidades e em seguida o de enfermeira, mostrando também suas funcionalidades. Foi questionado como o sistema diferencia o login e foi explicado que cada usuário tem um código e que o mesmo pode ser visto no cadastrado do usuário no sistema (que é cadastrado no sistema desktop como usuário master). Foi perguntado também se até outubro eles terão o sistema completo para uso. Foi passado que completo talvez não, mas uma boa parte já estará bem funcional, que será um aplicativo beta.

Demonstração do progresso do sistema (grupo 2)

Responsável: Ramon Lacava Gutierrez Gonçales

No início da reunião, Ramon propôs a pauta de a ser debatida, que abrangia: Sistema de saúde no computador; Sistema de Saúde em celular. Para o início da discussão, Ramon apresentou o sistema de saúde em computador, mostrando as telas de carregamento e login. Logo em seguida, foi mostrado a tela inicial de escolha de empresas e a tela de início do sistema.

Em seguida, Ramon apresentou a parte de pacientes do Sistema, juntamente ao sensor de NFC. Zethu questionou como o cartão seria distribuído, e Ramon informou que a distribuição provavelmente seria feita através do Hospital. Em seguida, na parte dos pacientes, Ramon apresentou todas as informações armazenadas no paciente. Logo após, Ramon apresentou as doenças, diagnósticos e exames. Zethu pergunta o que foi feito para lidar com a privacidade dos dados de usuário. Ramon responde que na Microsoft Azure foram configuradas algumas regras de segurança, porém esta parte ainda precisa de mais trabalho. Zethu pergunta se há algum limite de armazenamento dos anexos incluídos no exame. Ramon responde que há um limite de 5 arquivos. Zethu então pergunta, o que ocorreria caso o limite estourasse. Ramon responde que a Microsoft Azure vai aumentando o tamanho da capacidade de armazenamento, alegando uma capacidade ilimitada devido a flexibilidade. Zethu pergunta se o paciente pode editar seus dados de saúde, e Ramon responde que sim, apenas não pode ser editado diagnósticos e exames. Zethu diz que alguns problemas podem ocorrer devido a edição. Ramon responde sobre o status de verificação de

perfil, que caso esteja como verificado, o perfil foi validado por um médico, e caso não esteja validado, o perfil não foi verificado.

Ramon então começa a mostrar a parte de criação de contas do sistema, mostrando como foi modelado o cadastro com base nos requisitos coletados. Ramon faz algumas confirmações a respeito do cadastro para verificar a validade de algumas informações (cadastro de enfermeiras). Zethu confirma a validade. Ramon então começa a mostrar o sistema em celular, e Zethu pede para enviar para ela o aplicativo do celular. Ramon afirma que irá enviar. Zethu pergunta se é necessário coletar mais algum tipo de dado. Ramon responde que é necessário coletar requisitos sobre medicamentos e vacinas. Por fim, a reunião se encerrou.

Demonstração do progresso do sistema (grupo 1)

Responsável: Natália Akina Uesugi

A reunião se iniciou com a apresentação do aplicativo do paciente, mostrando todas as funções já existentes e os dados que precisariam ser fornecidos pelo paciente. Foi perguntado o que ocorreria quando o paciente fosse um bebê ou uma criança, foi respondido que nesses casos seria necessário que os pais da criança fizessem o cadastro e gerenciamento da conta.

Foi explicado como funciona o sistema NFC pois durante havia dúvidas em relação ao funcionamento.

Após estas explicações foi mostrado o protótipo do aplicativo da enfermeira e explicado que por motivos de leis brasileiras não poderíamos autorizar ao usuário enfermeira modificar o prontuário médico e apenas o histórico.

Por fim foi mostrado a versão desktop e explicado cada um dos usuários e suas funcionalidades. Deste modo a reunião se encerrou.