

UNIVERSIDADE FEDERAL DO PARANÁ

FERNANDO AUGUSTO HEEREN FALKIEWICZ

LENNON WAGNER SALLES HENK

THIAGO AGUIAR DA SILVA

SOS SOCORRISTA: UMA PLATAFORMA PARA AGILIZAR
O ATENDIMENTO DE ACIDENTADOS

CURITIBA

2013

FERNANDO AUGUSTO HEEREN FALKIEWICZ
LENNON WAGNER SALLES HENK
THIAGO AGUIAR DA SILVA

SOS SOCORRISTA: UMA PLATAFORMA PARA AGILIZAR
O ATENDIMENTO DE ACIDENTADOS

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Alessandro Brawerman.

CURITIBA

2013

TERMO DE APROVAÇÃO

FERNANDO AUGUSTO HEEREN FALKIEWICZ

LENNON WAGNER SALLES HENK

THIAGO AGUIAR DA SILVA

SOS SOCORRISTA: UMA PLATAFORMA PARA AGILIZAR O ATENDIMENTO DE ACIDENTADOS

Trabalho de Conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, pela banca examinadora:

Orientador: Prof. Dr. Alessandro Brawerman

Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná,
UFPR.

Prof. Mestre Jaime Wojciechowski

Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná,
UFPR.

Prof. Mestre Razer Anthom Nizer Rojas Montaña

Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná,
UFPR.

Curitiba, 21 de março de 2013

RESUMO

Devido à grande expansão das tecnologias móveis nos últimos anos, está cada vez mais comum o uso de smartphones para auxiliar em diversas atividades do cotidiano. Pela facilidade e praticidade destes dispositivos, torna-se quase que indispensável o uso destes aplicativos, que automatizam e tornam ágeis as atividades humanas.

A finalidade do projeto é criar uma plataforma que auxilia a equipe médica no processo inicial de atendimento a acidentados. A partir do uso de um aplicativo desenvolvido para Android, o paramédico envia informações relativas ao estado do paciente para o servidor central. Este, por sua vez, requisita informações dos hospitais e direciona a equipe de socorro ao hospital com demanda não saturada e médicos especializados disponíveis para realizar o tratamento.

Baseada em algumas características, como por exemplo, distância até o hospital, número de leitos e especialistas disponíveis, e sendo sustentada por um protocolo de comunicação médica já existente, a equipe de paramédicos descreverá algumas informações observadas do paciente ao aplicativo.

Os dados coletados são enviados a um servidor central e a partir daí, este servidor utiliza conceitos do protocolo de *Contract Network*, e dispara informações a respeito do problema a ser tratado aos diversos hospitais. Os hospitais, por sua vez, devolvem uma mensagem ao servidor com suas características. Ao receber as características dos hospitais, o servidor central define para onde o paciente deve ser levado e envia uma resposta ao aplicativo. Para finalizar o procedimento, a equipe médica só precisa dizer ao aplicativo se o paciente, cujos dados foram enviados ao servidor central, será encaminhado ao hospital indicado. Caso seja, haverá um leito reservado ao paciente no hospital.

Palavras-chave: *Contract Network*, smartfone, Android, equipe de socorro.

ABSTRACT

Due to the great expansion of mobile technologies in the last years, is increasingly common the use of smartphones to assist with several activities of daily life. Because of the ease and practicality of these devices, developers allied with professionals from diverse areas, such as engineering, have the new purpose of creating applications capable of facilitate and automate many human activities.

The aim of the project is to create a platform that assists the medical team in the initial process of care of accident victims. From the use of an application developed for Android, the paramedic send information about the patient's condition to the central server. This, in turn, requests information from the hospitals, and directs the rescue team to the hospital with unsaturated demand and specialized doctors available to perform the treatment.

Based on some features, such as distance to the hospital, number of beds and experts available, and being supported by a medical communication protocol already existing, the paramedical team will describe some of the observed information of the patient to the application.

The collected data are sent to a central server and from then on, the server uses the concepts of Contract Network Protocol, and triggers information about the problem being treated to the various hospitals. The hospitals then return a message with its features to the server. Upon receiving the characteristics of the hospitals, the central server, defines where the patient should be taken, and send a response to the application.

To complete the procedure, the medical team only needs to tell the application if the patient, whose data were sent to the central server, will be sent to the hospital indicated. In case it is, there will be a bed reserved for the patient at the hospital.

Keywords: *Contract Network*, smartphone, Android, emergency team.

LISTA DE FIGURAS

FIGURA 1 – PROTOCOLO START	18
FIGURA 2 – ALGORITMO DE GRAVIDADE	19
FIGURA 3 – ARQUITETURA DO SISTEMA	28
FIGURA 4 – GRÁFICO DE GANTT	30
FIGURA 5 – MODELO WBS	31
FIGURA 6 – WEBSERVICE	38
FIGURA 7 – TESTE DE ALTA GRAVIDADE – SERVIDOR CENTRAL	51
FIGURA 8 – TESTE DE ALTA GRAVIDADE – APLICATIVO MÓVEL	52
FIGURA 9 – TESTE DE BAIXA GRAVIDADE – SERVIDOR CENTRAL	54
FIGURA 10 – TESTE DE BAIXA GRAVIDADE – APLICATIVO MÓVEL	55
FIGURA 11 – TESTE DE BÔNUS NA DISTÂNCIA – SERVIDOR CENTRAL	57
FIGURA 12 – TESTE DE BÔNUS NA DISTÂNCIA – APLICATIVO MÓVEL	58
FIGURA 13 – TESTE DE DISTÂNCIA IDEAL – SERVIDOR CENTRAL	60
FIGURA 14 – TESTE DE DISTÂNCIA IDEAL – APLICATIVO MÓVEL	61
FIGURA 15 – TESTE DE CASO COMUM – SERVIDOR CENTRAL	63
FIGURA 16 – TESTE DE CASO COMUM – APLICATIVO MÓVEL	64
FIGURA 17 – TESTE DE HOSPITAL SEM LEITOS – SERVIDOR CENTRAL	66
FIGURA 18 – TESTE DE HOSPITAL SEM LEITOS – APLICATIVO MÓVEL	67
FIGURA 19 – TELA 01 – INÍCIO	72
FIGURA 20 – TELA 01 – INÍCIO – PRIMEIRA INICIALIZAÇÃO	73
FIGURA 21 – TELA 02 – MENU	74
FIGURA 22 – TELA 02 – MENU – ENDEREÇO DO SERVIDOR	74
FIGURA 23 – TELA 02 – MENU – PLACA DA AMBULÂNCIA	75
FIGURA 24 – TELA 03 – TIPO DE LESÃO	76
FIGURA 25 – TELA 04 – RESPIRANDO	77
FIGURA 26 – TELA 05 – FREQUÊNCIA RESPIRATÓRIA	78
FIGURA 27 – TELA 06 – ENCHIMENTO CAPILAR	79
FIGURA 28 – TELA 07 – RESPONDE COMANDOS	80
FIGURA 29 – TELA 08 – PEQUENAS LESÕES	81
FIGURA 30 – TELA 09 – ENVIAR INFORMAÇÕES – ANTES DE ENVIAR	82
FIGURA 31 – TELA 09 – ENVIAR INFORMAÇÕES – DEPOIS DE ENVIAR	83
FIGURA 32 – TELA 10 – FINALIZAR	84
FIGURA 33 – TELA 10 – FINALIZAR – DIÁLOGO FINALIZAR	85
FIGURA 34 – TELA 10 – FINALIZAR – DIÁLOGO CANCELAR	86
FIGURA 35 – DIAGRAMA DE CASOS DE USO	94
FIGURA 36 – UC – TELA 01	96
FIGURA 37 – UC – TELA 02	96
FIGURA 38 – UC – TELA 03	96
FIGURA 39 – UC – TELA 04	96
FIGURA 40 – UC – TELA 05	98

FIGURA 41 – UC – TELA 06.....	98
FIGURA 42 – UC – TELA 07.....	99
FIGURA 43 – UC – TELA 08.....	99
FIGURA 44 – UC – TELA 09.....	99
FIGURA 45 – UC – TELA 10.....	99
FIGURA 46 – UC – TELA 11.....	100
FIGURA 47 – UC – TELA 12.....	100
FIGURA 48 – UC – TELA 13.....	102
FIGURA 49 – UC – TELA 14.....	102
FIGURA 50 – UC – TELA 15.....	102
FIGURA 51 – UC – TELA 16.....	102
FIGURA 52 – UC – TELA 17.....	103
FIGURA 53 – DC – APLICATIVO MÓVEL.....	106
FIGURA 54 – DC – SERVIDOR CENTRAL.....	107
FIGURA 55 – DC – HOSPITAIS.....	108
FIGURA 56 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 1.....	110
FIGURA 57 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 2.....	111
FIGURA 58 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 3.....	112
FIGURA 59 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 4.....	113
FIGURA 60 – DS – APLICATIVO MÓVEL – CONFIGURAR.....	114
FIGURA 61 – DS – SERVIDOR CENTRAL – ATENDIMENTO.....	115
FIGURA 62 – DS – SERVIDOR CENTRAL – OUTRAS MENSAGENS.....	116
FIGURA 63 – DS – HOSPITAIS.....	117
FIGURA 64 – DIAGRAMA ENTIDADE-RELACIONAMENTO.....	118

LISTA DE SIGLAS

ADT	–	Android Development Tools (Ferramenta de Desenvolvimento Android)
API	–	Application Programming Interface (Interface de Programação de Aplicativos)
CRUD	–	Create, Read, Update and Delete (Criar, Ler, Atualizar e Remover)
DC	–	Diagrama de Classes
DS	–	Diagrama de Sequência
GPS	–	Global Positioning System (Sistema de Posicionamento Global)
IDE	–	Integrated Development Environment (Ambiente Integrado de Desenvolvimento)
IP	–	Internet Protocol (Protocolo de Internet)
JDK	–	Java Development Kit (Kit de Desenvolvimento Java)
JSON	–	JavaScript Object Notation (Notação de Objetos JavaScript)
REST	–	Representational State Transfer (Transferência de Estado Representativo)
SAMU	–	Serviço de Atendimento Móvel de Urgência
SGBD	–	Sistema Gerenciador de Banco de Dados
SIATE	–	Serviço Integrado de Atendimento ao Trauma em Emergência
START	–	Simple Triagem e Rápido Tratamento
UC	–	Use Case (Caso de Uso)
URL	–	Uniform Resource Locator (Localizador–Padrão de Recursos)
W3C	–	World Wide Web Consortium (Consórcio World Wide Web)
UML	–	Unified Modeling Language (Linguagem de Modelagem Unificada)

SUMÁRIO

RESUMO	4
ABSTRACT	5
LISTA DE FIGURAS	6
LISTA DE SIGLAS	8
1 INTRODUÇÃO	13
1.1 OBJETIVOS.....	14
1.1.1 OBJETIVO GERAL.....	14
1.1.2 OBJETIVOS ESPECÍFICOS.....	14
1.2 ORGANIZAÇÃO DO TEXTO.....	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 ATENDIMENTO DE EMERGÊNCIA.....	16
2.2 ALGORITMO PARA A DEFINIÇÃO DA GRAVIDADE DO SOCORRIDO.....	17
2.3 ANDROID.....	20
2.4 CONTRACT NETWORK PROTOCOL.....	21
2.5 INTEGRANDO SISTEMAS.....	23
2.6 GPS.....	23
2.7 ALGORITMO GENÉTICO.....	24
2.8 TRABALHOS RELACIONADOS.....	25
2.8.1 ANARCHITECTURE FOR INTEROPERABILITY AND UBIQUITY OF MEDICAL INFORMATION.....	25
2.8.2 DISTRIBUTED PROCESSING SYSTEM FOR RESTORATION OF ELECTRIC POWER DISTRIBUTION NETWORK USING TWO-LAYERED CONTRACT NET PROTOCOL.....	25
2.8.3 TRANSPORTATION SCHEDULING AND SIMULATION IN A RAILROAD SCENARIO: A MULTI-AGENT APPROACH.....	26
3 METODOLOGIA	27

3.1 VISÃO GERAL.....	27
3.2 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE	28
3.3 PLANO DE ATIVIDADES.....	30
3.4 PLANO DE RISCOS	31
3.5 RESPONSABILIDADES	32
3.5.1 ARQUITETURA	32
3.5.2 CRONOGRAMA	32
3.5.3 RISCOS.....	33
3.5.4 ESTUDO DAS TECNOLOGIAS.....	33
3.5.5 DESENVOLVIMENTO	33
3.5.6 DOCUMENTAÇÃO TÉCNICA	34
3.5.7 MONOGRAFIA	34
3.5.8 DESIGN – INTERFACE E ERGONOMIA	35
3.5.9 IMPLEMENTAÇÃO.....	35
3.6 MATERIAIS.....	36
3.6.1 ANDROID	36
3.6.2 JAVA.....	37
3.6.3 WEBSERVICES.....	37
3.6.4 BANCO DE DADOS	40
3.6.5 ALGORITMO DE FITNESS	40
3.6.6 CÁLCULO DA DISTÂNCIA.....	40
3.7 DESENVOLVIMENTO DO PROJETO.....	41
3.7.1 FUNDAMENTOS DA CLASSE HOSPITAL	42
3.7.2 MENSAGENS.....	42
3.7.3 OBTENDO INFORMAÇÕES DOS HOSPITAIS.....	45
3.7.4 DEFININDO O MELHOR HOSPITAL	45
3.8 VALIDAÇÃO	47

3.8.1 HOSPITAL 1	47
3.8.2 HOSPITAL 2	48
3.8.3 HOSPITAL 3	48
3.8.4 HOSPITAL 4	48
3.8.5 HOSPITAL 5	49
3.9 TESTES.....	49
3.9.1 TESTE DE ALTA GRAVIDADE	50
3.9.2 TESTE DE BAIXA GRAVIDADE.....	53
3.9.3 TESTE DE BÔNUS NA DISTÂNCIA.....	56
3.9.4 TESTE DE DISTÂNCIA IDEAL.....	59
3.9.5 TESTE DE CASO COMUM	62
3.9.6 TESTE DE HOSPITAL SEM LEITOS	65
3.10 CONCLUSÃO DA VALIDAÇÃO.....	68
4 APRESENTAÇÃO DO SOFTWARE	69
4.1 INSTALAÇÃO/CONFIGURAÇÃO	69
4.2 TELAS.....	71
4.2.1 TELA 01 – INÍCIO	71
4.2.2 TELA 02 – MENU	73
4.2.3 TELA 03 – TIPO DE LESÃO.....	75
4.2.4 TELA 04 – RESPIRANDO	76
4.2.5 TELA 05 – FREQUÊNCIA RESPIRATÓRIA.....	77
4.2.6 TELA 06 – ENCHIMENTO CAPILAR.....	78
4.2.7 TELA 07 – RESPONDE COMANDOS.....	79
4.2.8 TELA 08 – PEQUENAS LESÕES.....	80
4.2.9 TELA 09 – ENVIAR INFORMAÇÕES	81
4.2.10 TELA 10 – FINALIZAR.....	83
5 CONSIDERAÇÕES FINAIS.....	87

REFERÊNCIAS.....	89
APÊNDICE A – PLANO DE RISCOS.....	93
APÊNDICE B – DIAGRAMA DE CASOS DE USO.....	94
APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO.....	95
UC – CONFIGURAR APLICATIVO.....	95
UC – AVALIAR SITUAÇÃO DO PACIENTE.....	97
UC – ENVIAR INFORMAÇÕES AO SERVIDOR.....	100
UC – SELECIONAR HOSPITAL.....	103
UC – FORNECER INFORMAÇÕES DO HOSPITAL.....	104
APÊNDICE D – DIAGRAMAS DE CLASSES.....	105
DC – APLICATIVO MÓVEL.....	105
DC – SERVIDOR CENTRAL.....	107
DC – HOSPITAIS.....	108
APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA.....	109
DS – APLICATIVO MÓVEL – ATENDIMENTO.....	109
DS – APLICATIVO MÓVEL – ATENDIMENTO – 1.....	110
DS – APLICATIVO MÓVEL – ATENDIMENTO – 2.....	111
DS – APLICATIVO MÓVEL – ATENDIMENTO – 3.....	112
DS – APLICATIVO MÓVEL – ATENDIMENTO – 4.....	113
DS – APLICATIVO MÓVEL – CONFIGURAR.....	114
DS – SERVIDOR CENTRAL – ATENDIMENTO.....	115
DS – SERVIDOR CENTRAL – OUTRAS MENSAGENS.....	116
DS – HOSPITAIS.....	117
APÊNDICE F – DIAGRAMA ENTIDADE–RELACIONAMENTO.....	118
APÊNDICE G – DICIONÁRIO DE DADOS.....	119

1 INTRODUÇÃO

Existem diversos protocolos e modelos de atendimento pré-hospitalar atualmente no mundo. Cada modelo consiste em uma série de procedimentos que mantêm o paciente nas melhores condições possíveis até sua chegada ao hospital. Há uma grande necessidade que este primeiro atendimento seja eficaz, pois neste momento é feita a identificação da gravidade do socorrido, assim, é possível encaminhá-lo ao médico capacitado a tratá-lo.

A plataforma apresentada visa auxiliar nesse primeiro atendimento ao paciente. Propõe-se uma plataforma formada por um aplicativo desenvolvido para Android, um servidor central e servidores em hospitais. Com o aplicativo Android, o usuário (paramédico) fornecerá informações sobre a gravidade do paciente e, através da utilização do protocolo de *Contract Network* em conjunto com Webservices, o aplicativo se comunicará com o servidor central. O servidor central obtém informações dos servidores dos hospitais e avalia a situação atual de cada hospital, dando-lhe uma nota, definida por um algoritmo inteligente construído com base em uma função de *fitness* de Algoritmos Genéticos. Em seguida, é retornado ao aplicativo móvel o hospital o qual a vítima deve ser encaminhada.

A definição da gravidade do paciente no aplicativo móvel é baseada no protocolo START (NOVAES *et al.*; 2013), no qual os procedimentos de avaliação foram adaptados para agilmente determinar o estado do paciente. Logo após isto o aplicativo retorna qual hospital o paciente deve ser transportado, abolindo qualquer necessidade de comunicação verbal entre a equipe de paramédicos e a equipe do hospital, tornando esta comunicação automatizada.

1.1 OBJETIVOS

Esta seção apresenta os objetivos, gerais e específicos do projeto, visando estabelecer as metas finais até sua conclusão.

1.1.1 OBJETIVO GERAL

Desenvolver uma solução computacional que auxilie a equipe de paramédicos na decisão para qual hospital o socorrido deve ser levado. Esta plataforma consiste inicialmente no uso do aplicativo móvel (que utilize a plataforma Android). O paramédico fornece dados do socorrido ao aplicativo, o qual se comunica com o servidor central. Este servidor requisita e concentra os dados dos hospitais e, logo após, retorna ao aplicativo o hospital que o paciente deve ser encaminhado. Esta decisão é tomada através de um algoritmo de *fitness*, que é executado no servidor central, o qual se baseia em características como distância do hospital, número de leitos e especialistas médicos disponíveis.

1.1.2 OBJETIVOS ESPECÍFICOS

Para que o desenvolvimento da plataforma fosse completado com sucesso, uma série de passos intermediários foram realizados, sendo eles:

- Utilizar conceitos do protocolo *Contract Network*;
- Realizar a modelagem do sistema utilizando UML;
- Aprimorar os conhecimentos na utilização da plataforma Android;
- Aprimorar os conhecimentos da comunicação entre cliente servidor (criação de Webservices utilizando o servidor de aplicação Apache Tomcat e o *framework* Jersey);

- Desenvolvimento um aplicativo móvel que capture informações do paciente;
- Desenvolvimento de um servidor central que receba informações do aplicativo móvel e responda informando o melhor hospital disponível;
- Desenvolvimento de um algoritmo de *fitness* para o servidor central que dê uma nota para cada hospital com base em suas características atuais;
- Integração entre os sistemas usando Webservices.

1.2 ORGANIZAÇÃO DO TEXTO

Nos capítulos seguintes deste documento é possível compreender a organização deste trabalho.

No Capítulo 2 (Fundamentação Teórica) são apresentados conceitos referentes aos protocolos de atendimento para acidentados, bem como o as tecnologias existentes que podem servir de apoio para esses processos. No Capítulo 3 (Metodologia) é possível compreender como o sistema foi modelado, os recursos e ferramentas utilizados como, por exemplo, a UML. Também é possível compreender o processo de elaboração e desenvolvimento do sistema SOS Socorrista. Conjuntamente são apresentados o plano de atividades e o Gráfico de Gantt, as responsabilidades dos integrantes do projeto, plano de risco e os recursos de hardware e software utilizados. O Capítulo 4 (Apresentação do Software) apresenta os requisitos e instruções de instalação, as telas do aplicativo móvel e testes realizados com o sistema. Finalmente, no Capítulo 5 são apresentadas as considerações finais e descritas propostas de futuras implementações que propendem melhorias no sistema desenvolvido.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos relevantes que facilitarão um melhor entendimento do projeto pelo leitor. São apresentados também trabalhos relacionados que mostram que as tecnologias neste trabalho adotadas são vigentes e empregadas pelo mercado.

2.1 ATENDIMENTO DE EMERGÊNCIA

A fim de conhecer o procedimento no atendimento de uma emergência, foi realizada uma entrevista com médicos que atuam na área. Entendeu-se que o atendimento é efetuado em três passos:

Passo 1: Após a chegada da ambulância ao local do acidente os paramédicos realizam os procedimentos normais de primeiros socorros no paciente;

Passo 2: Ao ter a situação sob controle, entram em contato com um médico regulador, localizado em uma sede com linhas diretas de telefone com os hospitais;

Passo 3: O médico regulador faz perguntas sobre o estado do paciente da ambulância para os socorristas e com base nas informações de tipo de lesão e disponibilidade dos hospitais próximos, indica aos paramédicos o hospital em que o paciente deve ser encaminhado.

Os médicos dos hospitais que atendem emergências podem utilizar a linha direta para comunicar, por exemplo, que os médicos especialistas em certo âmbito não estarão disponíveis por determinado tempo, portanto não poderão atender pacientes com alguma lesão específica.

Todo o processo é realizado de maneira subjetiva, a informação de disponibilidade dos hospitais é apenas dada por telefone. Dada a quantidade de informações, é possível que o médico regulador confunda-as e envie um paciente para algum hospital sem condições de realizar o atendimento adequado.

Sendo assim, um sistema de informação automatizado poderia gerir todo o controle que o médico regulador faz, trazendo mais agilidade e precisão na hora de direcionar as ambulâncias com os socorridos.

2.2 ALGORITMO PARA A DEFINIÇÃO DA GRAVIDADE DO SOCORRIDO

Para que a gravidade da lesão do paciente seja estabelecida de forma objetiva, paramédicos utilizam um protocolo de triagem de acidentes com múltiplas vítimas, pela possibilidade da avaliação rápida e incisiva do paciente.

O protocolo escolhido é chamado de START, Simples Triagem e Rápido Tratamento, é o mais difundido, utilizado pelo SAMU e SIATE em casos de múltiplas vítimas (NOVAES *et al*; 2013). Pelo fato de ser focado em múltiplas vítimas a avaliação é feita rapidamente, possibilitando que seja utilizado como modo de definição da gravidade do paciente sem que o atendimento seja atrasado por causa do software.

Ao fim do algoritmo do protocolo são dados quatro níveis de prioridade:

Óbito (preto): Pacientes que não respiram, mesmo após manobras simples de abertura da via aérea;

Imediata (vermelho): Respiração presente somente após manobras de abertura da via aérea ou com frequência respiratória maior que 30 por minuto; enchimento capilar acima de 2 segundos ou que não responde a comandos;

Atrasada (amarelo): pacientes que não se enquadram nem na prioridade imediata e nem na prioridade menor;

Menor (verde): feridos que estão andando pelo local.

Pode-se chegar aos níveis seguindo o algoritmo da Figura 1.

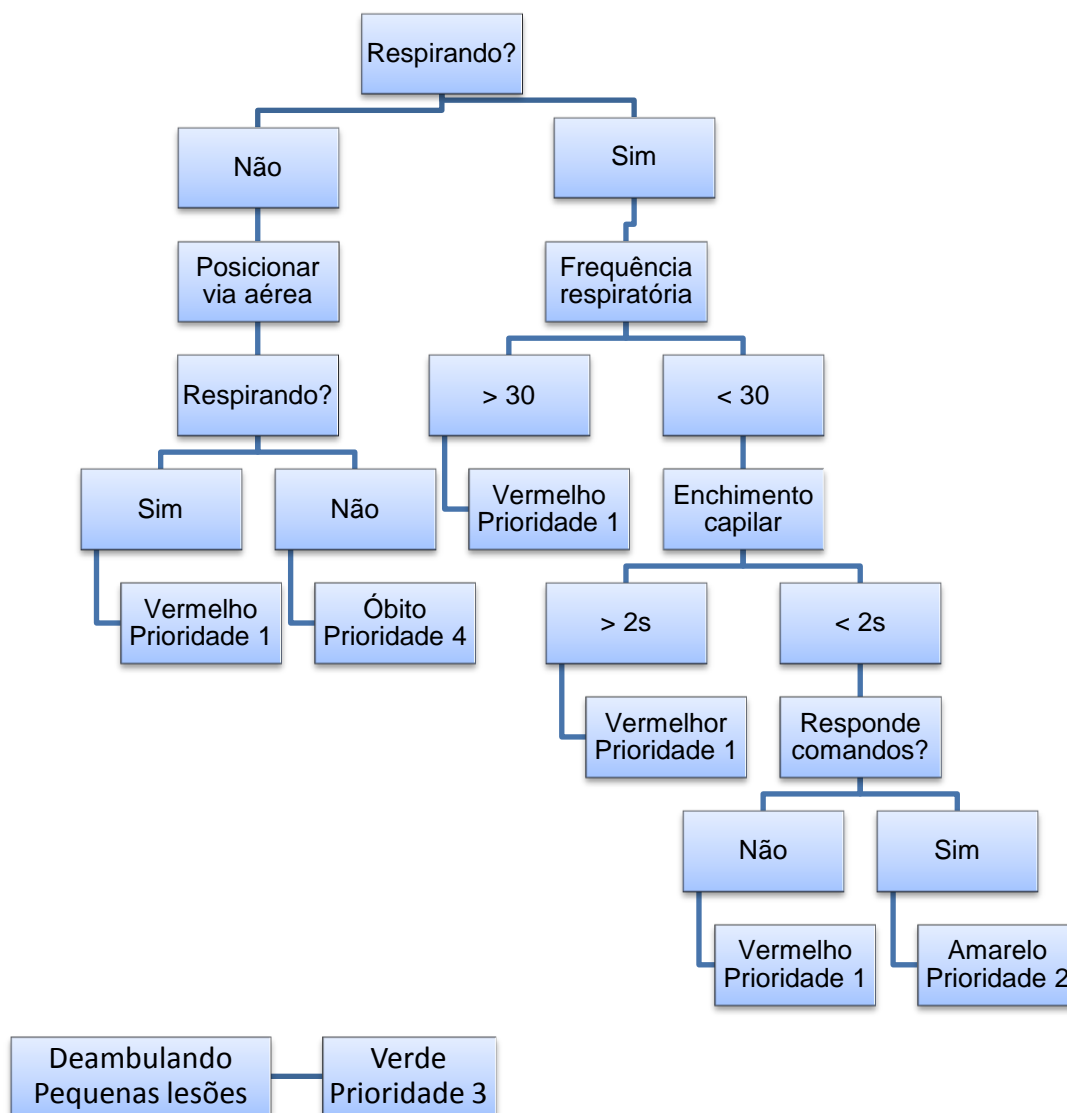


FIGURA 1 – PROTOCOLO START

De acordo com o Manual do Atendimento Pré-Hospitalar (SIATE; Corpo de Bombeiros do Paraná, 2006) a frequência respiratória é o número de ciclos respiratórios realizados por minuto. Um valor alto pode indicar falta de oxigenação. Enchimento capilar é uma técnica utilizada para avaliar o sistema circulatório nos tecidos periféricos. Faz-se uma pressão sobre as unhas ou lábios, até que fique com coloração pálida. Retira-se a pressão e mede-se o tempo em que a coloração fique rosada novamente. Se o tempo for superior a dois segundos é um sinal de que o sistema circulatório está comprometido.

O protocolo original é aplicado antes do atendimento dos pacientes, para se adequar ao projeto SOS Socorrista foi criada uma versão adaptada do algoritmo (Figura 2). Na versão adaptada, o algoritmo é executado pelos paramédicos somente após os procedimentos de primeiros socorros, portanto o caso de óbito já terá sido declarado, podendo ser excluído dos níveis de gravidade. Tem-se quatro níveis de gravidade, de 1 a 4. Quanto maior a gravidade maior é a prioridade para atendimento em um hospital de distância menor.

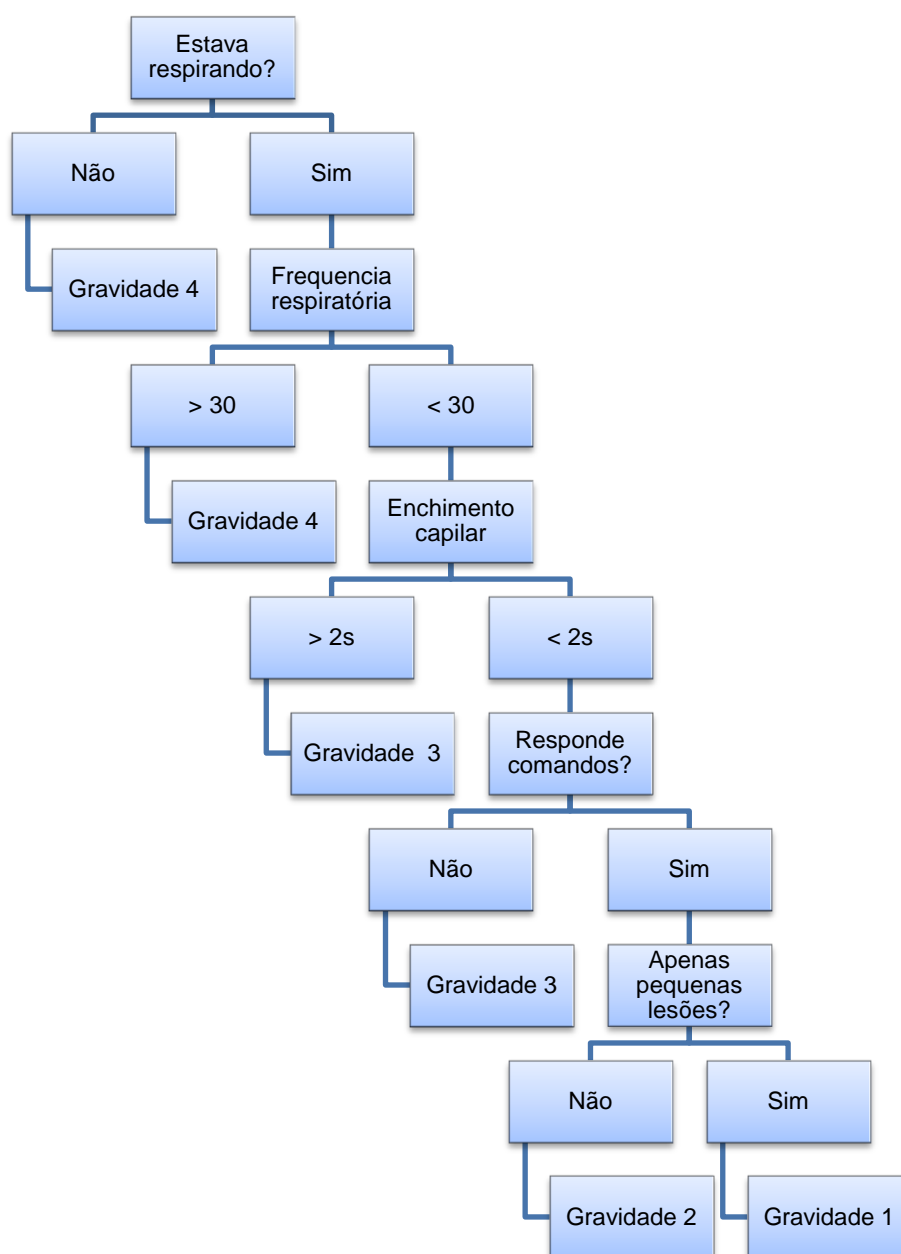


FIGURA 2 – ALGORITMO DE GRAVIDADE

2.3 ANDROID

Um aplicativo para auxiliar o atendimento de acidentados foi desenvolvido durante este projeto usando a plataforma Android. Portanto, achou-se importante a apresentação de conceitos fundamentais desta plataforma para um melhor entendimento do projeto pelo leitor.

O Android é uma plataforma de código-fonte aberto introduzido pela Google para o desenvolvimento de aplicações para dispositivos móveis como smartphones ou *tablets* baseada em um sistema operacional Linux (LECHETA, 2011). A plataforma possui parceiros de peso como a LG, Motorola, Samsung, Sony Ericsson entre outros, que fazem com que a mesma seja altamente difundida. Atualmente cerca de 1,3 milhão de aparelhos com Android são ativados por dia, segundo Eric Schmidt, presidente do Google. Segundo a Samsung “é como ter um laptop muito pequeno no bolso” (SAMSUNG; Explicação do sistema operacional Google Android, 2012).

Hoje, o sistema Android está entre os grandes serviços da Google e é o sistema mais usado em celulares, ultrapassando o iOS da Apple. Em 2012, a parcela de mercado de Android foi de 68,4% e a da Apple, de 19,4% (Pesquisa mostra supremacia do Android frente ao iOS, 2012). É possível ver essa supremacia do Android na Tabela 1.

Sistema Operacional	3º trimestre de 2012		3º trimestre de 2011		Mudança
	Volumes de Remessa	Fatia de Mercado	Volumes de Remessa	Fatia de Mercado	
Android	136.0 mi	75.0%	71.0 mi	57.5%	91.5%
iOS	26.9 mi	14.9%	17.1 mi	13.8%	57.3%
BlackBerry	7.7 mi	4.3%	11.8 mi	9.5%	-34.7%
Symbian	4.1 mi	2.3%	18.1 mi	14.6%	-77.3%
Windows Phone 7 / Windows Mobile	3.6 mi	2.0%	1.5 mi	1.2%	140.0%
Linux	2.8 mi	1.5%	4.1 mi	3.3%	-31.7%
Outros	0.0 mi	0.0%	0.1 mi	0.1%	-100.0%
Total	181.1 mi	100%	123.7 mi	100%	46.4%

TABELA 1 – SEIS MAIORES SISTEMAS OPERACIONAIS MÓVEIS.

FONTE: EXTREMETECH.COM

Além disso, uma pesquisa realizada pelo *Pew Research Center's Internet & American Life Project* revela aparelhos com Android possuem preços mais atrativos o que colabora para a difusão das tecnologias que esses smartphones oferecem (*Android Is Popular Because It's Cheap, Not Because It's Good*, 2013). Por haver inúmeros modelos de aparelhos de baixo custo com suporte ao Android torna-se viável o desenvolvimento da proposta deste trabalho.

2.4 CONTRACT NETWORK PROTOCOL

Contract Network é um protocolo desenvolvido para especificar resoluções de problemas de comunicação e controle, distribuindo tarefas por meio de negociações. Consiste em um conjunto de nós e software agente. Cada nó na rede pode ser em diferentes tempos ou tarefas, gerente ou contratante (SMITH, 1980).

Quando um nó recebe tarefas compostas ou por algum motivo não consiga resolver uma determinada tarefa, ele assume o papel de gerente quebrando a tarefa em subtarefas e então anuncia a subtarefa para a *Contract Network*. Logo em seguida surgem os possíveis contratantes e o contratante premiado recebe a tarefa (ALIBHAI, 2003).

O Contract Network Protocol (CNP) (Smith, 1980; Smith & Davis 1981; Davis & Smith 1988) foi proposto com o intuito de descentralizar a atribuição de tarefas. Sua importância reside na medida em que foi o primeiro trabalho a usar um processo de negociação envolvendo uma seleção mútua entre gestores e empreiteiros. Foi inicialmente aplicado a uma rede simulada e distribuída de sensores acústicos.

O veículo de roteamento das aplicações é desenvolvido em uma TRACONET (*TRAnspiration COoperation NET*). As negociações assíncronas e automáticas na TRACONET assemelham-se a um sistema de contratação direcionada, onde cada parte envolvida pode fazer uma licitação para cada anúncio que recebe e as propostas dos demais não são reveladas a ela (SANDHOLM, 2013).

Uma analogia simples é pensar em um sistema como o eBay, um sistema em que os compradores especificam os produtos que eles querem, bem como um

preço máximo que estão dispostos a pagar. Os softwares agentes encontram outros usuários dispostos a vender as mercadorias dentro da faixa de preço desejado. O usuário com o menor preço é selecionado para cumprir o contrato. Outras restrições podem ser aplicadas, tais como o tempo de entrega e a localização dos fornecedores (ZAFEER, 2003).

Ao utilizar este protocolo na plataforma SOS Socorrista baseou-se no modo como a informação é encapsulada, distribuída e como ela corrobora para gerar a solução. Contudo a abordagem foi feita com a centralização da decisão, ou seja, apenas um nó decide qual nó será o contratado, neste caso o servidor central.

Este protocolo de comunicação foi empregado entre os três grupos de nós no sistema. O primeiro grupo são os dispositivos móveis. Tais dispositivos são manipulados por socorristas durante o atendimento do paciente. Neste nó o protocolo carrega as informações que correspondem ao estado do socorrido, além da posição da ambulância, que por sua vez são enviadas para o nó que corresponde ao servidor central.

O segundo grupo é composto por um único nó que corresponde ao sistema que centraliza a informação, ou seja, o servidor central. Este recebe dados dos dois outros grupos de nós e decide a solução fazendo o papel de gerente.

Já o terceiro grupo corresponde aos hospitais. Nestes nós o protocolo carrega informações com as características, em tempo real, dos hospitais como o número disponível de leitos e profissionais capacitados a atender o problema. Muitos hospitais já utilizam sistemas para controlar leitos e médicos, portanto seria apenas necessária a criação de um módulo ou programa que acesse o banco de dados e crie um Webservice que atenda aos requisitos do projeto.

Em síntese, o primeiro grupo faz o papel dos nós com problemas que precisam ser resolvidos, o segundo grupo corresponde ao gerente que contratará a melhor solução para os nós do primeiro grupo e o terceiro grupo são os nós com a capacidade de solucionar o problema.

2.5 INTEGRANDO SISTEMAS

Como dito anteriormente, a plataforma completa é composta por aplicativos em dispositivos móveis, um sistema de decisão no servidor central e por sistemas de hospitais. Portanto, para que todo o processo aconteça de maneira transparente ao socorrista, é necessária a integração de todos estes sistemas.

Atualmente é muito comum a integração de sistemas de diversos tamanhos e tipos. Pode-se rapidamente integrar sistemas Web, por exemplo, o site da Amazon que possui uma integração com sistemas de empresas de cartão de crédito. Pode-se também integrar sistemas Web com aplicativos móveis, por exemplo, o aplicativo Facebook para dispositivos móveis que permite acessar um perfil via smartfone sem ser através de um navegador.

A técnica mais utilizada para a integração de sistemas é o desenvolvimento de Webservices. Segundo o W3C um Webservice (serviço web) define-se como: “um sistema de software projetado para suportar a interoperabilidade entre máquinas sobre rede” (SOARES, 2012).

No sistema proposto empregou-se Webservices para a comunicação entre os nós da Contract Network.

2.6 GPS

Sistema de posicionamento global, popularmente conhecido por GPS (*Global Positioning System*, ou do português “geoposicionamento por satélite”). É um sistema que permite obter geolocalização, em tempo real, de um dispositivo monitorado por satélites.

O rastreamento é feito por quatro dentre vinte e quatro satélites dispostos na órbita terrestre. Possui uma precisão que pode variar de centímetros para alguns metros, dependendo da necessidade e da qualidade dos aparelhos receptores (VERSIGNASSI, 2013).

Atualmente há uma quantidade significativa de dispositivos móveis com Android que dispõe de GPS. Neste projeto, o GPS foi incorporado como um dos

pontos chave para o funcionamento do sistema, pois a geolocalização da ambulância é um critério para auxiliar na escolha do melhor hospital de atendimento.

2.7 ALGORITMO GENÉTICO

A técnica de inteligência artificial baseada em algoritmos genéticos é uma proposta inspirada na teoria da evolução de Darwin e na biologia genética. Trabalha com conceitos de hereditariedade, mutação, seleção natural e recombinação. Essa técnica é utilizada para obter a melhor resposta dentro de um conjunto de possíveis soluções (LINDEN, 2008).

Possui três grupos básicos para o processo de obtenção do melhor resultado, sendo eles o conjunto das funções de avaliação (*fitness*), o conjunto dos cromossomos (problema e solução) e o algoritmo que executa as atividades de mutação, seleção e etc.

É possível aplicar a técnica em diversas modelagens de sistema, com ênfase em sistemas complexos como, por exemplo, sistemas de Data Mining, alocação de tarefas, seleção de rotas, e outros sistemas que servem para resolver problemas de otimização e aprendizado de máquina (LINDEN, 2008).

O projeto SOS Socorrista incorporou o conceito da função de *fitness*, que é responsável por dar valores para a medição da qualidade de cada hospital que concorre para servir de destino para o socorrido.

2.8 TRABALHOS RELACIONADOS

Nesta seção são apresentados trabalhos que se relacionam ao projeto SOS Socorrista por utilizar as mesmas tecnologias, ou por ter propósitos semelhantes.

2.8.1 ANARCHITECTURE FOR INTEROPERABILITY AND UBIQUITY OF MEDICAL INFORMATION

Este projeto, (PLÁCIDO *et al.*; 2011) propõe a integração entre sistemas médicos e aplicativos Android, utilizando um modelo de arquitetura orientada a serviços e webservices. O modelo visa promover a interoperabilidade entre sistemas de informação, para obtenção de informações médicas de pacientes.

O trabalho se relaciona ao SOS Socorrista pela utilização de webservices para comunicação entre diferentes sistemas para obtenção de informações médicas.

2.8.2 DISTRIBUTED PROCESSING SYSTEM FOR RESTORATION OF ELECTRIC POWER DISTRIBUTION NETWORK USING TWO-LAYERED CONTRACT NET PROTOCOL

Foi desenvolvido um sistema de restauração da distribuição de energia elétrica que se ajusta automaticamente distribuindo-a à vários geradores, através do *Contract Network Protocol* (KODAMA; HAMAGAMI, 2010).

O trabalho se relaciona ao SOS Socorrista pela utilização do *Contract Network Protocol*, para tomada de decisão em alocação de recursos.

2.8.3 TRANSPORTATION SCHEDULING AND SIMULATION IN A RAILROAD SCENARIO: A MULTI-AGENT APPROACH

Foi proposto um sistema para o planejamento e monitoramento do plano de execução em tempo real em um cenário de transporte ferroviário (FISHER; LIND, 1998).

O trabalho se relaciona ao SOS Socorrista, pois foi utilizado o *Contract Network Protocol*, para obtenção de informações necessárias para planejamento e alocação de recursos.

3 METODOLOGIA

Este capítulo apresenta toda a metodologia utilizada para o desenvolvimento da plataforma SOS Socorrista. Está dividido em subtópicos, e neles estão especificadas as tecnologias que foram usadas para produzir o código da plataforma, bem como as técnicas que serviram para gerenciar o desenvolvimento do projeto, e os testes realizados para validar o funcionamento do software.

3.1 VISÃO GERAL

O projeto tem por finalidade auxiliar o socorrista a estabelecer, de forma ágil, a gravidade do paciente e auxiliar na comunicação entre a equipe de paramédicos e os hospitais. Algumas características como, por exemplo, a distância da ambulância até o hospital, o número de leitos disponíveis e o número de profissionais especialistas em cada tipo de lesão, determinam qual hospital o paciente deve ser levado.

A equipe de paramédicos fornece algumas informações do paciente ao aplicativo. Estas informações são baseadas no modelo de atendimento pré-hospitalar START (NOVAES; ABREU; CARVALHO, 2013). Com isso, o aplicativo envia esses dados para o servidor central, este por sua vez faz requisições a todos os hospitais que atendem o tipo de lesão, e retorna à aplicação qual o melhor hospital para levar o socorrido.

O melhor hospital para que o paciente deve ser direcionado é escolhido por meio de um algoritmo baseado em uma função *fitness*. Este algoritmo atribui um peso a cada característica do hospital. Assim, o hospital que possui o conjunto de características mais adequado para atender paciente no momento do acidente será escolhido. Entretanto a decisão final de encaminhar o paciente para o hospital sugerido, pela aplicação, cabe aos paramédicos.

A Figura 3 ilustra o processo desde o momento do início do pedido por um hospital até a escolha do mesmo pelo servidor central. Note que o paramédico fornece os dados ao aplicativo, que por sua vez os envia via Internet para o servidor

central. Este se comunica com os hospitais e decide qual o mais apto a receber o paciente no momento. A comunicação entre todos os sistemas é baseada em Webservices e o *Contract Network Protocol*, são enviadas mensagens padronizadas, com uma identificação de tipo e conteúdo correspondente a cada tipo.

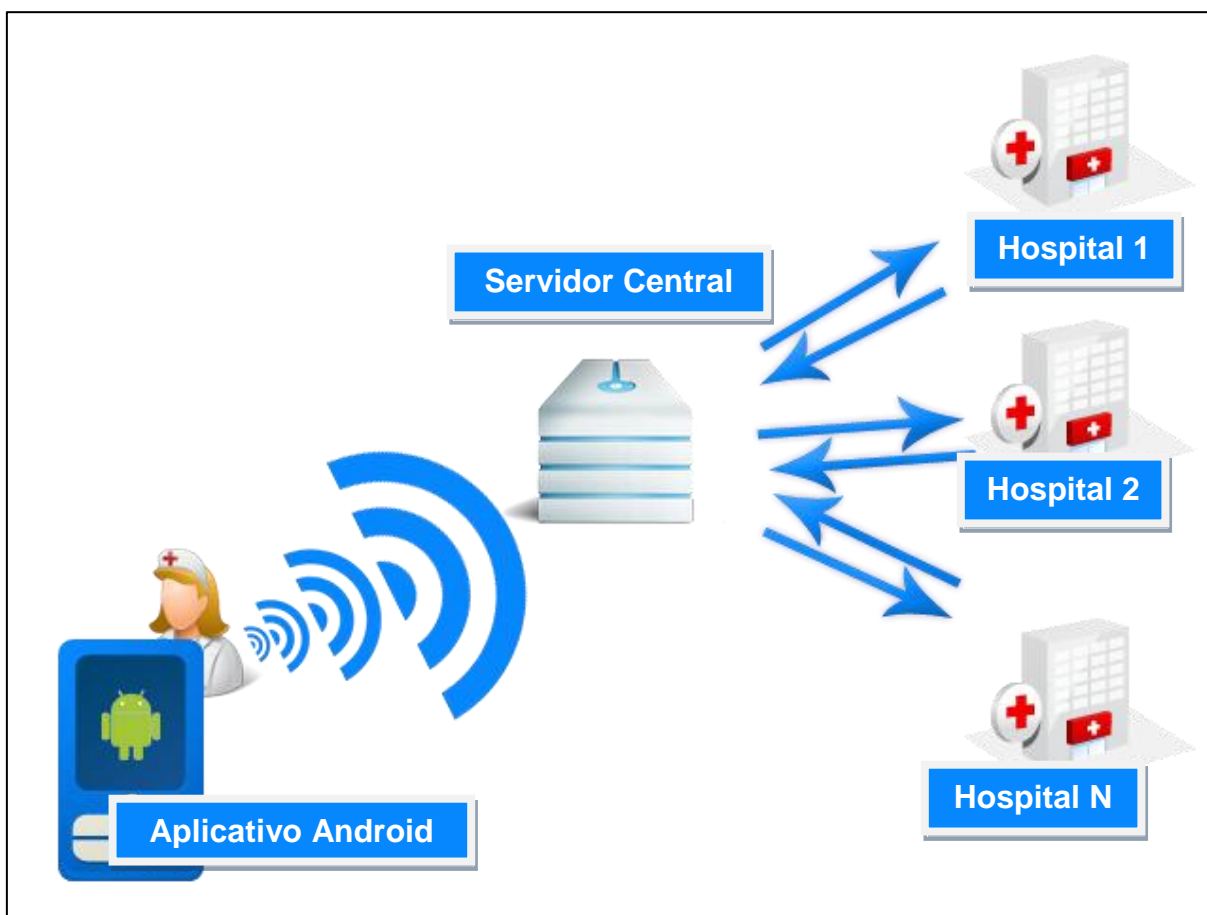


FIGURA 3 – ARQUITETURA DO SISTEMA

3.2 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

No desenvolvimento do projeto foi utilizado um modelo de desenvolvimento incremental. Foi feito um levantamento de requisitos pela equipe em conjunto com o orientador. Em seguida foram criados protótipos básicos utilizando comunicação por Webservices e operações simples como prova de conceito.

Após perceber a possibilidade real de desenvolvimento do projeto foi realizada uma entrevista com médicos do Hospital do Trabalhador para ter conhecimento do processo atualmente em vigor das equipes de paramédicos. O projeto foi desenvolvido aos poucos, aumentando sua complexidade gradativamente.

Após levantar os requisitos necessários para iniciar o desenvolvimento da plataforma, foram elaborados diagramas UML, contendo a modelagem do sistema. A documentação encontra-se em anexo. Adotou-se alguns padrões de desenvolvimento de software para aumentar a produtividade e garantir sua qualidade. O primeiro foi utilizar o padrão de desenvolvimento MVC, um padrão de projeto em Engenharia de Software. Com isso, a lógica da aplicação fica separada da interface do usuário.

A seguir será feita uma introdução sobre como é feita a comunicação entre a aplicação e os servidores:

1. **Comunicação Aplicação/Servidor:** o aplicativo Android se comunicará com um servidor central (Webservice), utilizando objetos JSON (Introdução ao JSON, JSON.ORG) para passagem de informações, através de implementações REST (RICHARDSON; RUBY, 2007). O usuário, ao responder uma série de perguntas, fornece informações relevantes sobre a gravidade do paciente. Após obter estes dados, e também as coordenadas da ambulância, as informações serão enviadas do aplicativo para o servidor central, o qual irá executar um algoritmo de fitness, e retornará ao aplicativo qual é o hospital mais indicado ao paciente.
2. **Comunicação Servidor/Servidor:** ao ser requisitado pelo aplicativo, o servidor central faz requisições aos hospitais, solicitando algumas informações (a comunicação é feita via web services, utilizando objetos JSON). O servidor central, baseado nos fatores de avaliação de seu algoritmo, dará uma pontuação aos hospitais.
3. **Comunicação Servidor/Aplicação:** após o servidor central determinar a melhor decisão, ele retornará as informações do hospital ao aplicativo. O usuário opta por confirmar a locomoção do paciente, caso decida por levá-lo, quando chegar ao hospital o usuário deve finalizar o atendimento.

3.3 PLANO DE ATIVIDADES

A Figura 4 apresenta o gráfico de Gantt do projeto, nesta pode-se encontrar cada uma das tarefas realizadas, bem como o período e os responsáveis pela mesma.

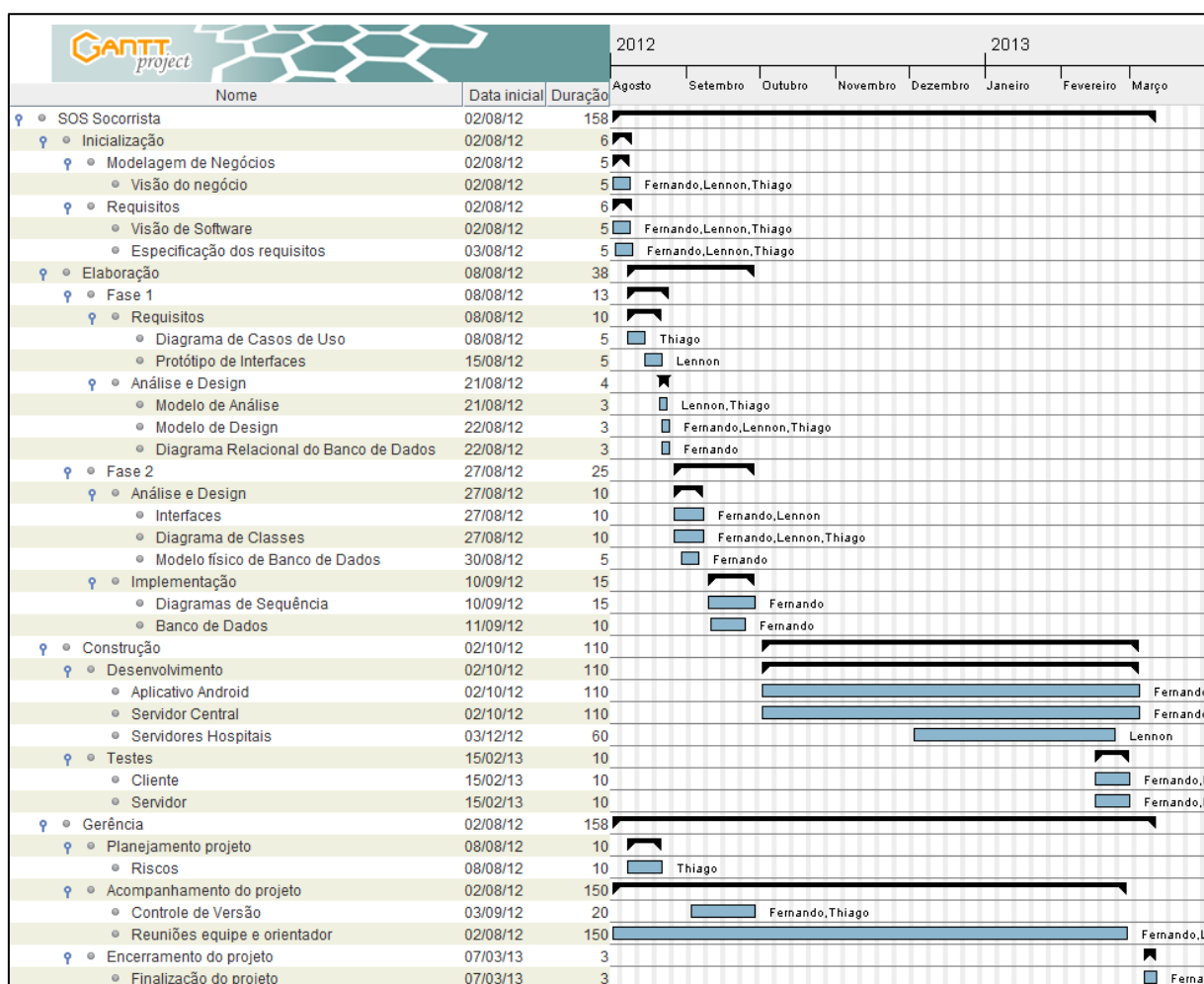


FIGURA 4 – GRÁFICO DE GANTT

A Figura 5 apresenta a estrutura do projeto desde seu início até sua conclusão. Na Inicialização do projeto foram levantados requisitos referentes ao projeto, bem como a modelagem e arquitetura do sistema. Na elaboração, foram novamente levantados os requisitos e elaborados diagramas para modelar o

sistema. Após o término das análises, foram desenvolvidos os servidores e o aplicativo Android, passando por testes e validações. Por fim a gerência está relacionada com o controle do projeto, versão dos documentos desenvolvidos, e controle dos prazos de cada etapa.

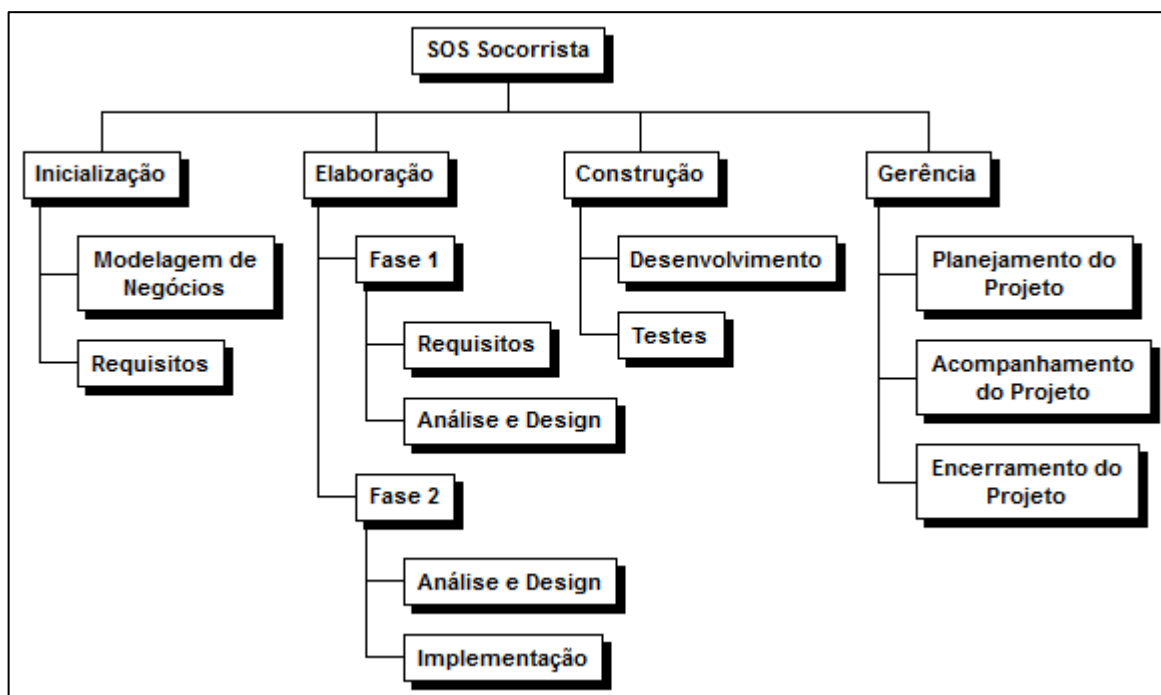


FIGURA 5 – MODELO WBS

3.4 PLANO DE RISCOS

O Plano de Riscos (APÊNDICE A) demonstra os riscos que o projeto correu, com destaque na inexperiência técnica da equipe. Com o objetivo de tornar mínimos os impactos oriundos dos riscos, foram planejadas formas de controle e monitoramento dos mesmos.

3.5 RESPONSABILIDADES

Esta seção visa explicar as tarefas designadas a cada membro da equipe, bem como salientar de forma geral o que foi feito em cada processo do desenvolvimento do projeto.

3.5.1 ARQUITETURA

A arquitetura do sistema foi planejada pelos membros da equipe, em conjunto com o orientador, caracterizando-se por dois fatores: confiabilidade e eficácia. A interação entre as partes do sistema pode ser simplificada como o exemplo a seguir: Usuário → Aplicativo → Servidor Central → Hospitais → Servidor Central → Aplicativo → Usuário.

Houve uma atenção especial com a comunicação entre os elementos do sistema, por haver uso de dados em tempo real. Isso garante a confiabilidade do aplicativo. Já a eficácia está associada com a escolha das tecnologias e padrões de desenvolvimento, garantindo que o aplicativo funcione de forma correta e sendo construído da forma mais produtiva.

Responsáveis: Todos os membros.

3.5.2 CRONOGRAMA

O cronograma foi elaborado de acordo com as experiências de cada membro do grupo, em conjunto com o orientador. Cada tarefa foi designada a um membro do grupo, estipulando-se o escopo da atividade a ser desenvolvida e o tempo a ser executada.

Responsáveis: Todos os membros.

3.5.3 RISCOS

Os riscos foram avaliados após a finalização do escopo do projeto. Os pontos avaliados como sendo propensos a riscos no desenvolvimento do projeto foram reavaliados, e estudados minuciosamente para resolução dos mesmos.

Responsáveis: Thiago Aguiar da Silva.

3.5.4 ESTUDO DAS TECNOLOGIAS

Após definir a arquitetura do sistema, houve-se a necessidade de estudar as tecnologias necessárias para a implementação do mesmo. Os estudos foram estimados no cronograma.

Responsáveis: Fernando Augusto Heeren Falkiewicz e Lennon Wagner Salles Henk.

3.5.5 DESENVOLVIMENTO

O desenvolvimento foi planejado de acordo com o cronograma de estudo das tecnologias. Após a definição das tecnologias a serem utilizadas, como também as ferramentas de desenvolvimento, definiu-se um período de tempo para que fosse realizada cada tarefa no desenvolvimento do sistema.

Responsáveis: Todos os membros.

3.5.6 DOCUMENTAÇÃO TÉCNICA

A documentação técnica foi desenvolvida para melhor entendimento de como se comporta o sistema, desde a parte funcional até os fluxos específicos de desenvolvimento. O projeto foi documentado utilizando UML 2.2. Através de seus diversos tipos de diagramas, é possível entender a comunicação dos atores do sistema, bem como suas respectivas ações. Foram desenvolvidos diagramas de classes e sequência, para visualizar as entidades envolvidas no sistema e a comunicação entre elas. E por fim, houve a modelagem do banco de dados relacional.

Responsáveis: Fernando Augusto Heeren Falkiewicz e Thiago Aguiar da Silva.

3.5.7 MONOGRAFIA

A monografia está relacionada ao domínio do caso de estudo, explicando o porquê da necessidade de se criar uma solução computacional para um determinado caso, como será aplicado na prática e quais foram as metodologias utilizadas para o desenvolvimento do projeto.

Com isso, também faz parte da documentação teórica quaisquer explicações e definições relacionadas ao projeto, como glossário, termos técnicos da tecnologia da informação e explicação de procedimentos aplicados no âmbito do caso.

Responsáveis: Todos os membros.

3.5.8 DESIGN – INTERFACE E ERGONOMIA

As interfaces da aplicação desenvolvida na plataforma Android foram elaboradas de forma que seu leiaute seja de fácil entendimento, e não contenha poluição de informações. Por se tratar de um aplicativo voltado ao ramo da saúde, e a agilidade ser necessária, as interfaces são simples e amigáveis ao usuário, facilitando seu uso. Toda informação requisitada ao usuário é feita de forma rápida e sucinta, poupando-o de procurar por instruções.

Boas práticas de ergonomia (NEIL, 2012) foram aplicadas para a elaboração das interfaces, tais como:

- Mensagens de alerta ou erro com explicações claras;
- Disposição de componentes como botões (localizados de forma que facilite a navegação);
- Legendas significativas;
- Prevenção de possíveis erros que o usuário pode vir a cometer;
- Quantidade de informação exibida ao usuário;
- Cores e fontes agradáveis.

Responsáveis: Lennon Wagner Salles Henk.

3.5.9 IMPLEMENTAÇÃO

O desenvolvimento do sistema seguiu as diretrizes da arquitetura planejada pela equipe. Os elementos do sistema foram desenvolvidos simultaneamente, tais como o aplicativo Android e os Webservices que rodam em um servidor de aplicação.

Responsáveis: Todos os membros.

3.6 MATERIAIS

Os seguintes *frameworks* e ferramentas foram utilizados no projeto: Eclipse Juno (utilizando o *plugin* ADT – *Android Development Tools*), Apache Tomcat7, MySQL, Jersey e JSON.

O software para os aparelhos Android foi desenvolvido com a linguagem de programação Java, utilizando a IDE Eclipse com o kit de desenvolvimento para Android do Google instalado. Para testes foram utilizados tanto o emulador que acompanha o kit de desenvolvimento, quanto os aparelhos celulares dos membros da equipe. Para a comunicação do software com o servidor central foi utilizado o formato JSON, que é suportado nativamente pelo Android.

O servidor central foi também desenvolvido em Java com a IDE Eclipse, tendo como base a API Jersey, focada no desenvolvimento de Webservices REST. Os cálculos com coordenadas GPS, como distância e endereço, são realizados utilizando a API da Matriz de distâncias do Google. Em testes o servidor foi executado nos computadores dos membros da equipe. O servidor utiliza banco de dados MySQL para armazenar dados necessários dos hospitais.

3.6.1 ANDROID

Para o desenvolvimento do aplicativo Android foi utilizado a IDE Eclipse com o *plugin* ADT que pode ser obtido no site *Android Developers*, do Google (ANDROID DEVELOPERS, 2013). Este *plugin* é fornecido pelo Google e amplia os recursos da IDE, permitindo criar projetos de aplicativos Android. Em conjunto com o Android SDK que fornece um emulador para simular um celular Android em um computador.

3.6.2 JAVA

A linguagem escolhida para o desenvolvimento do projeto foi o JAVA, tanto porque é uma linguagem de programação muito difundida, quanto por ser a linguagem suportada pelo Android (LECHETA, 2010).

3.6.3 WEBSERVICES

Um Webservice é baseado em tecnologias como HTTP, XML, SOAP, WSDL, SPARQL, REST, JSON e outros. Sendo que o XML, ou outra tecnologia como o JSON para codificar e decodificar os dados, e padrões como SOAP e REST podem transportá-los. Eles são a tecnologia ideal para comunicação entre sistemas, integrando sistemas de qualquer porte, sejam eles somente Web ou Web-aplicativos móvel, de maneira fácil, rápida e transparente (MACEDO, 2010).

Um exemplo do uso dessa tecnologia seria um sistema desenvolvido em JAVA em Linux se comunicando com outras plataformas com sistemas desenvolvido em PHP ou .NET .

A Figura 6 apresenta um diagrama exemplificando o uso de Webservices. O ícone da nuvem representa o Webservice que está disponível em um endereço da internet, logo que o cliente (representado pelo celular e pelo computador) requisita um serviço para o Webservice, este devolve para os clientes um objeto JSON que será exibido por um aplicativo Android, no celular, e no computador por um browser, por exemplo.

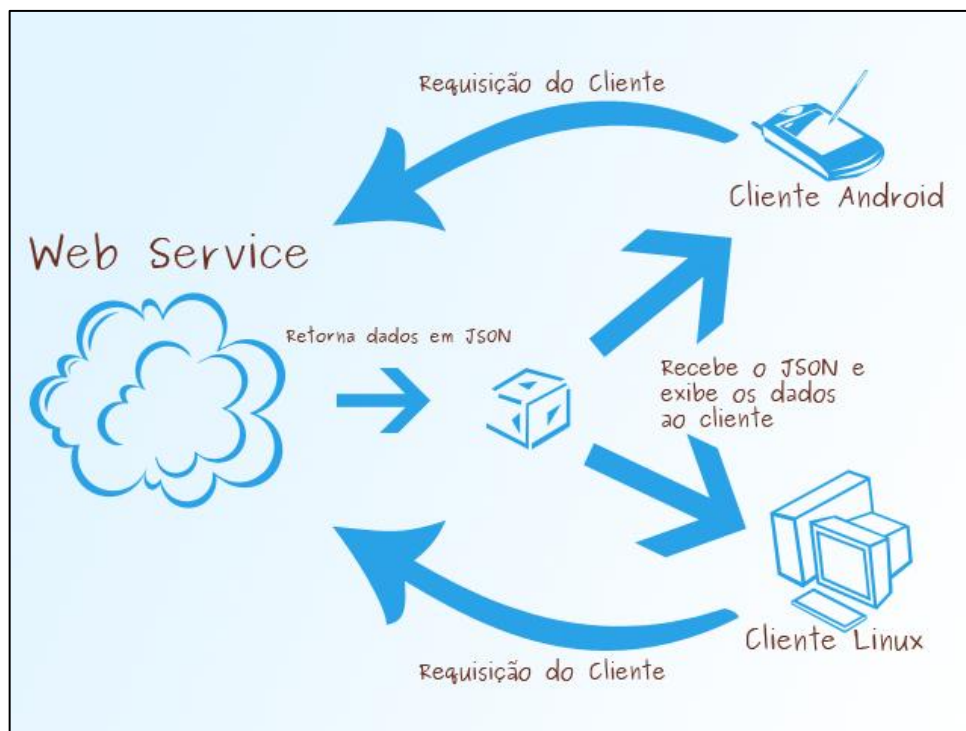


FIGURA 6 – WEBSERVICE

3.6.3.1 REST

A Transferência de Estado Representativo (*Representational State Transfer*) é um estilo de construção de aplicações Web que foi formalmente descrita por Fielding. O estilo de arquitetura REST tem sido utilizado para orientar a concepção e desenvolvimento da arquitetura moderna para a Web (FIELDING, 2000). O REST não foi uma proposta e sim uma descrição formal de como as especificações definidas por ele e Timothy Sir Berners-Lee e outros estavam sendo usadas (MCMURTRY *et tal*; 2006).

De maneira geral um servidor REST é uma técnica de engenharia de software que descreve qualquer interface web simples que utiliza tecnologias como XML, HTTP, JSON entre outras.

Geralmente um servidor REST recebe um pedido de um cliente através de um método HTTP POST, ou por uma URL que é recuperada através de um método HTTP GET.

Não há um padrão, a linguagem legível por máquina para descrever serviços REST e a segurança é geralmente limitado à utilização de HTTPS. No sistema proposto o REST será a maneira como será conduzida a comunicação os sistemas.

Jersey é um componente que implementa a API Java utilizada no desenvolvimento de Webservices REST e serve para facilitar o desenvolvimento da integração com aplicações de diversas plataformas (Java Magazine, Agosto, 2008). No projeto, o Jersey, que pode ser obtido no endereço <http://jersey.java.net>, foi utilizado em conjunto com o servidor Apache Tomcat, para facilitar a integração dos sistemas via REST.

3.6.3.2 JSON

JSON (*JavaScript Object Notation*) é um padrão simples para formato de dados, derivado da sintaxe de objetos em Javascript, contudo a estrutura do JSON tem sido utilizada em diversas linguagens como uma alternativa ao XML na transmissão de dados entre cliente e servidor (MACEDO, 2013) .

O formato JSON, que possui sua documentação disponível em <http://www.json.org>, tem a característica de ser menor que um equivalente em XML fazendo o JSON ser mais vantajoso quando precisa ser transmitido numa rede. Uma das aplicabilidades do JSON são a persistência e difusão de objetos em diversos sistemas, ou seja, é possível converter os atributos de um objeto em um JSON e transmiti-lo para outro sistema que converterá o JSON em um objeto.

Neste projeto, por meio do JSON, encapsulou-se os objetos de modo que foi possível enviar os dados de um nó ao outro na *Contract Network*.

3.6.4 BANCO DE DADOS

Uma base de dados geralmente é uma parte fundamental dentro de um sistema da informação, contudo o foco deste projeto encontra-se no algoritmo de *fitness*. Neste caso, o banco de dados não foi um ponto crítico dentro do sistema. A base de dados adotada possui apenas duas tabelas com outra tabela relacionando-as. Foi utilizado o SGBD MySQL, disponível em <http://www.mysql.com/downloads>, o banco de dados pertence ao Sistema do Servidor Central.

3.6.5 ALGORITMO DE FITNESS

O conceito de *fitness*, de Algoritmos Genéticos (LINDEN, 2008), foi utilizado no Servidor Central para calcular uma nota para cada hospital de acordo com os critérios definidos. Esta nota é utilizada, para escolher o melhor hospital para atender o paciente. O algoritmo desenvolvido e os parâmetros utilizados para escolher o hospital mais apropriado são explicados em maiores detalhes na seção de desenvolvimento do projeto.

3.6.6 CÁLCULO DA DISTÂNCIA

Para efetuar o cálculo da distância entre os hospitais e as ambulâncias foi utilizada a API da matriz de distâncias do Google (GOOGLE, 2013). Esta API é um serviço que fornece distância e tempo de deslocamento para uma matriz de origens e destinos. As informações retornadas são baseadas no trajeto recomendado entre pontos de partida e de chegada, conforme calculado pela API do *Google Maps*, e consistem em linhas que contêm valores *duration* (duração) e *distance* (distância) para cada par.

3.7 DESENVOLVIMENTO DO PROJETO

A solução computacional proposta para auxiliar a equipe de paramédicos na decisão de qual hospital encaminhar o socorrido surgiu do orientador, tornando objetivo o atendimento inicial à pessoa acidentada. A ideia proposta visa automatizar a comunicação hospitais-paramédicos.

Foram realizadas diversas reuniões entre a equipe e o orientador para definir o escopo do projeto e levantar os requisitos. Na fase inicial, a equipe entrevistou três médicos do Hospital do Trabalhador, os quais responderam um questionário previamente elaborado, para que fosse possível levantar os requisitos necessários para desenvolver a plataforma de auxílio aos paramédicos.

Após definidos os requisitos do sistema foram elaborados o WBS e o gráfico de Gantt, nos quais foram estabelecidas atividades e datas, além dos integrantes da equipe responsáveis pelo desenvolvimento delas.

Para projeto do sistema utilizou-se a linguagem de modelagem UML, que proporcionou a equipe maior produtividade no desenvolvimento do software, e possibilitou personalizações à modelagem do sistema.

Todas as atividades de análise e modelagem foram realizadas por todos os membros da equipe. Foram feitas reuniões semanais para acompanhamento do projeto, e os resultados obtidos eram analisados pelo orientador, a fim de corrigir e melhorar as tarefas executadas no projeto.

A plataforma de desenvolvimento escolhida pela equipe foi o Android, por ser o sistema operacional mais utilizado em dispositivos móveis no mundo atualmente, e proporcionar diversas vantagens sobre seus concorrentes, como diversos modelos de celulares que suportam o sistema operacional, custo acessível dos aparelhos, segurança, entre outros. Os membros da equipe, por estarem habituados com a linguagem de programação JAVA (utilizada no desenvolvimento de aplicativos para Android) puderam rapidamente aplicar os conhecimentos para desenvolver o aplicativo e os servidores, após a validação dos diagramas UML.

Quanto ao desenvolvimento do projeto, foi aplicado o padrão MVC para implementação por parte dos desenvolvedores. Cada membro da equipe ficou responsável por desenvolver determinadas funcionalidades do sistema.

Ao término das tarefas designadas a cada membro, foram feitos testes, para que se pudessem identificar possíveis falhas que afetassem a qualidade ou funcionamento do sistema, para logo então corrigi-las.

3.7.1 FUNDAMENTOS DA CLASSE HOSPITAL

Foi criada uma classe Hospital que possui atributos definindo as informações necessárias para a comunicação e identificação dos hospitais cadastrados no servidor central. A classe HospitalDAO é utilizada para obter os hospitais cadastrados no banco de dados.

Cada hospital possui um atributo do tipo *List<String>* (Lista de *strings*) chamado “lesoes”, que contém os tipos de lesão que o mesmo atende. Com esse atributo é possível que o servidor central envie requisições apenas para hospitais que atendem o tipo de lesão do paciente da ambulância, evitando o consumo desnecessário de recursos.

Os tipos de lesão estão armazenados no banco de dados na tabela especialidade, existindo uma tabela hospital_especialidade que relaciona o hospital às suas especialidades. O nome da tabela é especialidade, pois os tipos de lesão são especialidades do hospital, ou seja, os tipos de lesão que o hospital realiza tratamentos.

Também são armazenados nome, identificador único, latitude e longitude, endereço do hospital, e endereço (URL) do Webservice do hospital.

3.7.2 MENSAGENS

Todas as mensagens (objetos JSON) enviadas para o servidor central pelo aplicativo móvel possuem um campo chamado *tipoMensagem*, esse campo define o tipo da mensagem. As mensagens possíveis são: atendimento, confirmação, cancelamento, e finalização. Cada tipo de mensagem recebeu como identificador um

número inteiro para ser enviada e identificada facilmente. Após a identificação de cada mensagem é realizada a operação adequada ao tipo de mensagem.

Todas as mensagens de respostas possuem também um campo chamado *status* que pode ter os valores “sucesso” ou “falha”, esse campo é utilizado para identificar se ocorreu algum problema com a requisição.

3.7.2.1 MENSAGEM DE ATENDIMENTO

Atendimento é a mensagem enviada pelo aplicativo móvel para que seja escolhido o melhor hospital para encaminhar o paciente. Essa mensagem possui os campos: *tipoLesao*, *gravidadeLesao*, *latitude*, *longitude*, e *idAmbulancia*.

tipoLesao: é um texto contendo o tipo de lesão do paciente;

gravidadeLesao: é um número inteiro definindo a gravidade da lesão, obtida através das perguntas feitas pelo aplicativo móvel;

latitude* e *longitude: são as coordenadas em que a ambulância se encontra, obtido pelo receptor de GPS do aparelho;

idAmbulancia: é o identificador único da ambulância que está realizando o atendimento, definido como o a placa do veículo.

O servidor central também utiliza o tipo de lesão para enviar mensagens de atendimento apenas para hospitais que atendem a lesão especificada.

3.7.2.2 MENSAGEM DE CONFIRMAÇÃO

A mensagem de confirmação é enviada após a mensagem de atendimento. Essa mensagem significa que a ambulância está se dirigindo ao hospital indicado. Essa mensagem possui os campos: *idAmbulancia*, e *idHospital*.

idHospital: é o identificador único do hospital que foi selecionado no atendimento, é utilizado para que o servidor central encaminhe a mensagem para o hospital.

Ao ser recebida, o servidor central envia uma mensagem de mesmo tipo com o identificador da ambulância para o hospital selecionado. O hospital é encarregado de reservar o leito para o paciente que está sendo encaminhado.

3.7.2.3 MENSAGEM DE CANCELAMENTO

A mensagem de cancelamento é enviada após a mensagem de confirmação. Ela é disparada caso o paramédico cancele o atendimento. Essa mensagem significa que a ambulância está se dirigindo ao hospital indicado e possui os mesmos campos da mensagem de confirmação, alterando apenas seu tipo.

Ao ser recebida, o servidor central envia uma mensagem de mesmo tipo com o identificador da ambulância para o hospital selecionado. O hospital é encarregado de liberar o leito que foi reservado para o paciente.

3.7.2.4 MENSAGEM DE FINALIZAÇÃO

A mensagem de finalização é enviada após a mensagem de confirmação. Ela é disparada pelo socorrista - ao pressionar o botão "Finalizar" - para indicar que a ambulância chegou ao hospital definido. Essa mensagem possui os mesmos campos da mensagem de confirmação, alterando apenas seu tipo.

Ao ser recebida, o servidor central envia uma mensagem de mesmo tipo com o identificador da ambulância para o hospital selecionado. O intuito da mensagem é informar ao servidor do hospital da chegada do paciente.

3.7.3 OBTENDO INFORMAÇÕES DOS HOSPITAIS

Ao receber a mensagem de atendimento, o servidor central entra em contato com os hospitais que atendem o tipo de lesão do paciente para obter suas características atuais. São enviados os dados fornecidos pelo aplicativo móvel. Os campos no objeto JSON retornados pelos hospitais são: *leitos*, *qtPlantonistas*, e *qtMedicosEspecialistas*.

leitos: é a quantidade de leitos disponíveis no hospital;

qtPlantonistas: é a quantidade de médicos plantonistas disponíveis;

qtMedicosEspecialistas: é a quantidade disponível de médicos especialistas na lesão do paciente.

O servidor também utiliza a API da Matriz de distâncias do Google para obter a distância entre o aparelho móvel e cada um dos hospitais.

Todos os dados obtidos de cada hospital são definidos em objetos da classe Resposta, organizados como uma lista.

3.7.4 DEFININDO O MELHOR HOSPITAL

Após a obtenção das informações dos hospitais, cada resposta da lista de respostas é enviada para o método *selecionarMelhorHospital(List<Resposta> respostas)*, que avalia cada hospital com o método *avaliarHospital(Resposta resp)* da classe *Fitness* e retorna sua nota.

Para definir o melhor hospital foi criado um algoritmo com base no conceito de uma função de *fitness* de um algoritmo genético. A semelhança se dá pois tanto uma função de *fitness* quanto o algoritmo de seleção de hospitais definem critérios, e pesos para cada critério, e escolhem o modelo que mais se adequa ao definido como ideal para o caso avaliado.

Após a captura de todos os dados dos hospitais consultados, é feita a seleção a partir de quatro critérios: distância, quantidade de leitos disponíveis, quantidade de plantonistas, e quantidade de especialistas no tipo de lesão do

paciente. Para cada um dos critérios é dada uma nota de 0 a 100, caso tenha ocorrido algum erro na obtenção de algum dado, o critério relacionado recebe nota zero. Foi definido um valor ideal para cada critério, que o qualificaria com nota 100, caso o valor real seja melhor do que o ideal, a nota é mantida em 100. As notas obtidas são multiplicadas por um peso definido a partir de testes para que ocorra a melhor seleção possível.

Os pesos utilizados são: 2 para distância, 2 para quantidade de leitos, 3 para quantidade de plantonistas e 5 para quantidade de especialistas. Os valores ideais são: 5 km de distância, 20 leitos, 5 plantonistas e 3 especialistas.

O critério da distância é multiplicado também pela gravidade da lesão (1 a 4), desse modo quanto mais grave a lesão mais importante será a distância na seleção do melhor hospital.

A nota final do hospital é definida pelo somatório das notas de cada critério multiplicadas pelo seu peso. Ao final da avaliação de todos os hospitais é feita a seleção da maior nota.

As notas de cada critério são definidas pelos seguintes segmentos de código, retirados da classe *Fitness*:

- Distância:
`double nota = Math.max(DISTANCIA_IDEAL, distancia);`
`nota = nota / DISTANCIA_IDEAL;`
`nota = 100 / nota;`
- Leitos:
`double nota = Math.min(LEITOS_IDEAL, leitos);`
`nota = nota * (100 / LEITOS_IDEAL);`
- Plantonistas:
`double nota = Math.min(PLANTONISTAS_IDEAL, qtPlantonistas);`
`nota = nota * (100 / PLANTONISTAS_IDEAL);`
- Especialistas:
`double nota = Math.min(ESPECIALISTAS_IDEAL, qtEspecialistas);`
`nota = nota * (100 / ESPECIALISTAS_IDEAL);`

3.8 VALIDAÇÃO

Com o intuito de validar o funcionamento do algoritmo de escolha do melhor hospital, foram realizados testes utilizando quatro aparelhos celulares simulando os socorristas e dois computadores, um executando o servidor central, e um executando um servidor simulando os Webservices de cinco hospitais com características diferentes. Os fabricantes, modelos, e versões do Android eram variados entre os aparelhos celulares. Todos os componentes dos testes foram conectados à mesma rede sem fio, com conexão à Internet compartilhada. Anteriormente já tinham sido realizados testes utilizando a conexão 3G do aparelho celular com um servidor central conectado à Internet.

Uma versão de testes do aplicativo móvel foi desenvolvida com a diferença de possibilitar que as coordenadas GPS do aparelho sejam digitadas em dois campos de texto, para latitude e longitude. Após os testes em grupo terem sido executados, foram realizados testes formais, com apenas um aparelho celular, documentados neste capítulo.

Foram configurados cinco hospitais com características diferentes, suas localizações foram definidas como *shoppings* para facilitar a identificação intuitiva de sua localização e distância, também foram omitidos os endereços - na rede - de seus servidores, por não ser relevante para os resultados. O nome de cada hospital possui um número, para facilitar sua identificação. Seguem as configurações dos hospitais.

3.8.1 HOSPITAL 1

Nome: Hospital 1 - Shopping Cidade;

Latitude: -25,4952;

Longitude: -49,2884;

Endereço: Av. Mal. Floriano Peixoto, 4984 - Hauer, Curitiba - PR;

Tipos de lesão que atende: cardiológica, ortopédica, e queimadura.

3.8.2 HOSPITAL 2

Nome: Hospital 2 - Shopping Jardim das Américas;

Latitude: -25,4725;

Longitude: -49,2519;

Endereço: Av. Ns. de Lourdes, 63 - Jardim das Américas, Curitiba - PR;

Tipos de lesão que atende: cardiológica e queimadura.

3.8.3 HOSPITAL 3

Nome: Hospital 3 - Shopping Mueller;

Latitude: -25,4514;

Longitude: -49,229;

Endereço: Av. Cândido de Abreu, 127 - Centro Cívico, Curitiba - PR;

Tipos de lesão que atende: cardiológica e ortopédica.

3.8.4 HOSPITAL 4

Nome: Hospital 4 - Shopping Palladium;

Latitude: -25,4234;

Longitude: -49,2701;

Endereço: Av. Pres. Kennedy, 4121 - Portão, Curitiba - PR;

Tipos de lesão que atende: ortopédica.

3.8.5 HOSPITAL 5

Nome: Hospital 5 - Park Shopping Barigui;

Latitude: -25,4706;

Longitude: -49,2841;

Endereço: R. Luiz Felipe Lopes, 600 - Curitiba - PR;

Tipos de lesão que atende: ortopédica e queimadura.

3.9 TESTES

Utilizando a versão de testes do aplicativo, foram realizados testes alterando o tipo de lesão, gravidade, e localização. O tipo de lesão e gravidade foram definidos utilizando o procedimento normal do aplicativo, sendo apenas a localização modificada forçadamente. Há duas figuras em cada teste, uma é o resultado da saída de texto do servidor central e a outra a captura de tela do aplicativo exibindo o resultado do hospital escolhido pelo servidor central. Optou-se por não apresentar todas as telas exibidas pelo aplicativo móvel até a exibição da tela de envio de informações, pois a tela apresentada informa todas as informações necessárias para a compreensão de cada teste. Todos os testes foram realizados com apenas um aparelho executando o aplicativo móvel, para facilitar a comparação de resultados e por já terem sido realizados testes com múltiplos aparelhos. As notas de cada critério exibidas na saída do servidor não estão multiplicadas pelo seu peso, a multiplicação ocorre apenas na nota final.

3.9.1 TESTE DE ALTA GRAVIDADE

Teste realizado com o intuito de verificar a prioridade dada à distância entre a ambulância e o hospital em casos de alta gravidade.

3.9.1.1 PARÂMETROS DE ENTRADA

Tipo de lesão: Cardiológica;

Gravidade: 4 - Paciente não estava respirando no início do atendimento;

Latitude: -25,52;

Longitude: -49,27;

3.9.1.2 RESULTADOS

A Figura 7 apresenta a saída de texto do servidor central. Foi recebido um objeto JSON, identificado o tipo de mensagem de atendimento, com os parâmetros definidos acima, e foram feitas requisições de informações somente para os hospitais que atendem o tipo de lesão definido (hospitais 1, 2 e 3).

Após terem sido realizadas as requisições, o algoritmo de *fitness* forneceu a nota de cada hospital, exibindo também a nota de cada critério antes de seu peso ser aplicado. O servidor então escolheu o hospital com maior nota (hospital 1), e retornou-o como resposta para o aplicativo móvel.

```

Recebi: {"gravidadeLesao":4,"longitude":-49.27,"latitude":-25.52,"tipoMensagem":0,"tipoLesao":"Cardiologica"},
Tipo de mensagem: 0 - Atendimento
Hospital 4 - Shopping Palladium | não atende: Cardiologica
Hospital 5 - Park Shopping Barigui | não atende: Cardiologica
Hospital 3 - Shopping Mueller Distância: 12.729 Leitos: 3 Plantonistas: 2 Especialistas: 2
Nota distância: 39,280 Nota leitos: 15,000 Nota plantonistas: 40,000 Nota especialistas: 66,667
Nota final: 797,576
Hospital 2 - Shopping Jardim das Américas Distância: 7.882 Leitos: 4 Plantonistas: 3 Especialistas: 3
Nota distância: 63,436 Nota leitos: 20,000 Nota plantonistas: 60,000 Nota especialistas: 100,000
Nota final: 1227,485
Hospital 1 - Shopping Cidade Distância: 5.723 Leitos: 4 Plantonistas: 2 Especialistas: 5
Nota distância: 87,367 Nota leitos: 20,000 Nota plantonistas: 40,000 Nota especialistas: 100,000
Nota final: 1358,934
Melhor hospital: Hospital 1 - Shopping Cidade
Tempo de execução: 527 ms

```

FIGURA 7 – TESTE DE ALTA GRAVIDADE – SERVIDOR CENTRAL

A Figura 8 apresenta a tela de envio de informações - exibindo o resultado - do aplicativo móvel. O campo “Condição do paciente” exibe o valor “Paciente não estava respirando”, definido a partir de perguntas realizadas pelo aplicativo nas telas anteriores. O campo “Tipo de lesão” possui o valor “Cardiológica”, definido também nas telas anteriores. Os campos criados para a versão de testes, Latitude e Longitude, tiveram os valores -25.52 e -49.27 inseridos.

Após o envio das informações foi obtido o hospital 1 como resposta, a 5.72 km de distância.

SOS Socorrista - Testes AMB1111

Condição do paciente:
Paciente não estava respirando

Tipo de lesão:
Cardiológica

Enviar informações

Hospital: Hospital 1 - Shopping Cidade
Endereço: Av. Mal. Floriano Peixoto, 4984
- Hauer, Curitiba - PR
Distância: 5,72 km

Confirmar

Latitude:

Longitude:

Precisão do GPS: 29,00 m

FIGURA 8 – TESTE DE ALTA GRAVIDADE – APLICATIVO MÓVEL

3.9.1.3 ANÁLISE DO RESULTADO

Nota-se, como ilustrado na Figura 7, que apesar de o hospital 2 ter mais plantonistas do que o hospital 1, foi escolhido o hospital 1, como ilustrado na Figura 8, pela alta gravidade da lesão, dando-se prioridade à distância.

3.9.2 TESTE DE BAIXA GRAVIDADE

Teste realizado com o intuito de verificar a menor prioridade à distância e maior importância das melhores características dos hospitais em casos de baixa gravidade.

3.9.2.1 PARÂMETROS DE ENTRADA

Tipo de lesão: Cardiológica;

Gravidade: 1 - Paciente possui apenas pequenas lesões;

Latitude: -25,52;

Longitude: -49,27;

3.9.2.2 RESULTADOS

A Figura 9 apresenta a saída de texto do servidor central. Foi recebido um objeto JSON, identificado o tipo de mensagem de atendimento, com os parâmetros definidos acima, e foram feitas requisições de informações somente para os hospitais que atendem o tipo de lesão definido (hospitais 1, 2 e 3).

Após terem sido realizadas as requisições, o algoritmo de *fitness* forneceu a nota de cada hospital, exibindo também a nota de cada critério antes de seu peso ser aplicado. O servidor então escolheu o hospital com maior nota (hospital 2), e retornou-o como resposta para o aplicativo móvel.

```

Recebi: {"gravidadeLesao":1,"longitude":-49.27,"latitude":-25.52,"tipoMensagem":0,"tipoLesao":"Cardiologica"},
Tipo de mensagem: 0 - Atendimento
Hospital 4 - Shopping Palladium | não atende: Cardiologica
Hospital 5 - Park Shopping Barigui | não atende: Cardiologica
Hospital 2 - Shopping Jardim das Américas Distância: 7.882 Leitos: 4 Plantonistas: 3 Especialistas: 3
Nota distância: 63,436 Nota leitos: 20,000 Nota plantonistas: 60,000 Nota especialistas: 100,000
Nota final: 846,871
Hospital 3 - Shopping Mueller Distância: 12.729 Leitos: 3 Plantonistas: 2 Especialistas: 2
Nota distância: 39,280 Nota leitos: 15,000 Nota plantonistas: 40,000 Nota especialistas: 66,667
Nota final: 561,894
Hospital 1 - Shopping Cidade Distância: 5.723 Leitos: 4 Plantonistas: 2 Especialistas: 5
Nota distância: 87,367 Nota leitos: 20,000 Nota plantonistas: 40,000 Nota especialistas: 100,000
Nota final: 834,734
Melhor hospital: Hospital 2 - Shopping Jardim das Américas
Tempo de execução: 740 ms

```

FIGURA 9 – TESTE DE BAIXA GRAVIDADE – SERVIDOR CENTRAL

A Figura 10 apresenta a tela de envio de informações - exibindo o resultado - do aplicativo móvel. O campo "Condição do paciente" exibe o valor "Paciente possui apenas pequenas lesões", definido a partir de perguntas realizadas pelo aplicativo nas telas anteriores. O campo "Tipo de lesão" possui o valor "Cardiológica", definido também nas telas anteriores. Os campos criados para a versão de testes, Latitude e Longitude, tiveram os valores -25.52 e -49.27 inseridos.

Após o envio das informações foi obtido o hospital 2 como resposta, a 7,88 km de distância.

SOS Socorrista - Testes AMB1111

Condição do paciente:
Paciente possui apenas pequenas lesões

Tipo de lesão:
Cardiológica

Enviar informações

Hospital: Hospital 2 - Shopping Jardim das Américas

Endereço: Av. Ns. de Lourdes, 63 - Jardim das Américas, Curitiba - PR

Distância: 7,88 km

Confirmar

Latitude:

Longitude:

Precisão do GPS: 29,00 m

FIGURA 10 – TESTE DE BAIXA GRAVIDADE – APLICATIVO MÓVEL

3.9.2.3 ANÁLISE DO RESULTADO

Em um caso semelhante ao Teste de Alta Gravidade, mas com pouca gravidade, como ilustrado na Figura 9, foi escolhido o hospital mais bem preparado. Foi escolhido o hospital 2, como ilustrado na Figura 10, por ter mais plantonistas disponíveis, apesar de estar um pouco mais distante.

3.9.3 TESTE DE BÔNUS NA DISTÂNCIA

Teste realizado com o intuito de validar o bônus dado ao critério da distância quando a ambulância está a menos de um quilômetro de algum hospital que atende o tipo de lesão do paciente.

3.9.3.1 PARÂMETROS DE ENTRADA

Tipo de lesão: Ortopédica;

Gravidade: 2 - Paciente possui lesões significativas;

Latitude: -25,45;

Longitude: -49,23;

3.9.3.2 RESULTADOS

A Figura 11 apresenta a saída de texto do servidor central. Foi recebido um objeto JSON, identificado o tipo de mensagem de atendimento, com os parâmetros definidos acima, e foram feitas requisições de informações somente para os hospitais que atendem o tipo de lesão definido (hospitais 1, 3, 4 e 5).

Após terem sido realizadas as requisições, o algoritmo de *fitness* forneceu a nota de cada hospital, exibindo também a nota de cada critério antes de seu peso ser aplicado. O servidor então escolheu o hospital com maior nota (hospital 3), e retornou-o como resposta para o aplicativo móvel.

```

Recebi: {"gravidadeLesao":2,"longitude":-49.23,"latitude":-25.45,"tipoMensagem":0,"tipoLesao":"Ortopedica"}
Tipo de mensagem: 0 - Atendimento
Hospital 2 - Shopping Jardim das Américas | não atende: Ortopedica
Hospital 3 - Shopping Mueller Distância: 0.486 Leitos: 3 Plantonistas: 3 Especialistas: 2
Nota distância: 300,000 Nota leitos: 15,000 Nota plantonistas: 60,000 Nota especialistas: 66,667
Nota final: 1743,333
Hospital 1 - Shopping Cidade Distância: 10.937 Leitos: 4 Plantonistas: 2 Especialistas: 5
Nota distância: 45,716 Nota leitos: 20,000 Nota plantonistas: 40,000 Nota especialistas: 100,000
Nota final: 842,866
Hospital 4 - Shopping Palladium Distância: 8.367 Leitos: 9 Plantonistas: 6 Especialistas: 1
Nota distância: 59,759 Nota leitos: 45,000 Nota plantonistas: 100,000 Nota especialistas: 33,333
Nota final: 795,701
Hospital 5 - Park Shopping Barigui Distância: 10.344 Leitos: 6 Plantonistas: 3 Especialistas: 2
Nota distância: 48,337 Nota leitos: 30,000 Nota plantonistas: 60,000 Nota especialistas: 66,667
Nota final: 766,682
Melhor hospital: Hospital 3 - Shopping Mueller
Tempo de execução: 485 ms

```

FIGURA 11 – TESTE DE BÔNUS NA DISTÂNCIA – SERVIDOR CENTRAL

A Figura 12 apresenta a tela de envio de informações - exibindo o resultado - do aplicativo móvel. O campo “Condição do paciente” exibe o valor “Paciente possui lesões significativas”, definido a partir de perguntas realizadas pelo aplicativo nas telas anteriores. O campo “Tipo de lesão” possui o valor “Ortopédica”, definido também nas telas anteriores. Os campos criados para a versão de testes, Latitude e Longitude, tiveram os valores -25.45 e -49.23 inseridos.

Após o envio das informações foi obtido o hospital 3 como resposta, a 0,49 km de distância.

The screenshot shows a mobile application interface with a blue header bar containing the text 'SOS Socorrista - Testes' and 'AMB1111'. Below the header, the text 'Condição do paciente:' is followed by 'Paciente possui lesões significativas'. Underneath, 'Tipo de lesão:' is followed by 'Ortopédica'. A light blue button labeled 'Enviar informações' is positioned below this section. Further down, the text 'Hospital: Hospital 3 - Shopping Mueller' is displayed, followed by the address 'Endereço: Av. Cândido de Abreu, 127 - Centro Cívico, Curitiba - PR' and the distance 'Distância: 0,49 km'. A green button labeled 'Confirmar' is located below the distance. At the bottom, there are two input fields: 'Latitude: -25.45' and 'Longitude: -49.23'. Below these fields, the text 'Precisão do GPS: 29,00 m' is visible. The top status bar of the phone shows various icons and the time '17h54'.

FIGURA 12 – TESTE DE BÔNUS NA DISTÂNCIA – APLICATIVO MÓVEL

3.9.3.3 ANÁLISE DO RESULTADO

Como o atendimento ocorreu muito próximo ao hospital 3, foi dado um bônus à pontuação da distância, como ilustra a Figura 11, fazendo com que, mesmo estando inferior aos outros hospitais em alguns critérios, ainda assim fosse selecionado, como ilustra a Figura 12.

3.9.4 TESTE DE DISTÂNCIA IDEAL

Teste realizado com o objetivo de verificar a distância ideal, dando a mesma nota no critério da distância para hospitais com distâncias menores do que cinco quilômetros da ambulância em questão.

3.9.4.1 PARÂMETROS DE ENTRADA

Tipo de lesão: Queimadura;

Gravidade: 3 - Enchimento capilar maior que 2 segundos;

Latitude: -25,45;

Longitude: -49,27;

3.9.4.2 RESULTADOS

A Figura 13 apresenta a saída de texto do servidor central. Foi recebido um objeto JSON, identificado o tipo de mensagem de atendimento, com os parâmetros definidos acima, e foram feitas requisições de informações somente para os hospitais que atendem o tipo de lesão definido (hospitais 1, 2 e 5).

Após terem sido realizadas as requisições, o algoritmo de *fitness* forneceu a nota de cada hospital, exibindo também a nota de cada critério antes de seu peso ser aplicado. O servidor então escolheu o hospital com maior nota (hospital 2), e retornou-o como resposta para o aplicativo móvel.

```

Recebi: {"gravidadeLesao":3,"longitude":-49.27,"latitude":-25.45,"tipoMensagem":0,"tipoLesao":"Queimadura","id
Tipo de mensagem: 0 - Atendimento
Hospital 3 - Shopping Mueller | não atende: Queimadura
Hospital 4 - Shopping Palladium | não atende: Queimadura
Hospital 1 - Shopping Cidade Distância: 8.242 Leitos: 4 Plantonistas: 1 Especialistas: 5
Nota distância: 60,665 Nota leitos: 20,000 Nota plantonistas: 20,000 Nota especialistas: 100,000
Nota final: 963,989
Hospital 5 - Park Shopping Barigui Distância: 3.339 Leitos: 6 Plantonistas: 2 Especialistas: 2
Nota distância: 100,000 Nota leitos: 30,000 Nota plantonistas: 40,000 Nota especialistas: 66,667
Nota final: 1113,333
Hospital 2 - Shopping Jardim das Américas Distância: 3.643 Leitos: 4 Plantonistas: 3 Especialistas: 3
Nota distância: 100,000 Nota leitos: 20,000 Nota plantonistas: 60,000 Nota especialistas: 100,000
Nota final: 1320,000
Melhor hospital: Hospital 2 - Shopping Jardim das Américas
Tempo de execução: 474 ms

```

FIGURA 13 – TESTE DE DISTÂNCIA IDEAL – SERVIDOR CENTRAL

A Figura 14 apresenta a tela de envio de informações - exibindo o resultado - do aplicativo móvel. O campo “Condição do paciente” exibe o valor “Enchimento capilar maior que 2 segundos”, definido a partir de perguntas realizadas pelo aplicativo nas telas anteriores. O campo “Tipo de lesão” possui o valor “Queimadura”, definido também nas telas anteriores. Os campos criados para a versão de testes, Latitude e Longitude, tiveram os valores -25.45 e -49.27 inseridos.

Após o envio das informações foi obtido o hospital 2 como resposta, a 3,64 km de distância.

SOS Socorrista - Testes AMB1111

Condição do paciente:
Enchimento capilar maior que 2 segundos

Tipo de lesão:
Queimadura

Enviar informações

Hospital: Hospital 2 - Shopping Jardim das Américas

Endereço: Av. Ns. de Lourdes, 63 - Jardim das Américas, Curitiba - PR

Distância: 3,64 km

Confirmar

Latitude:

Longitude:

Precisão do GPS: 29,00 m

FIGURA 14 – TESTE DE DISTÂNCIA IDEAL – APLICATIVO MÓVEL

3.9.4.3 ANÁLISE DO RESULTADO

Dois hospitais receberam a mesma pontuação na distância, como ilustrado na Figura 13, então foi selecionado o mais bem preparado. O hospital 5 tem mais leitos, mas foi escolhido o hospital 2, como ilustrado na Figura 14, por ter mais plantonistas e médicos especialistas em queimaduras.

3.9.5 TESTE DE CASO COMUM

Teste realizado com o intuito de validar o sistema em um caso comum de atendimento, sem nenhuma característica específica de bonificação ou penalização.

3.9.5.1 PARÂMETROS DE ENTRADA

Tipo de lesão: Cardiológica;

Gravidade: 3 - Enchimento capilar maior que 2 segundos;

Latitude: -25,53;

Longitude: -49,26;

3.9.5.2 RESULTADOS

A Figura 15 apresenta a saída de texto do servidor central. Foi recebido um objeto JSON, identificado o tipo de mensagem de atendimento, com os parâmetros definidos acima, e foram feitas requisições de informações somente para os hospitais que atendem o tipo de lesão definido (hospitais 1, 2 e 3).

Após terem sido realizadas as requisições, o algoritmo de *fitness* forneceu a nota de cada hospital, exibindo também a nota de cada critério antes de seu peso ser aplicado. O servidor então escolheu o hospital com maior nota (hospital 1), e retornou-o como resposta para o aplicativo móvel.

```

Recebi: {"gravidadeLesao":3,"longitude":-49.26,"latitude":-25.53,"tipoMensagem":0,"tipoLesao":"Cardiologica"},i
Tipo de mensagem: 0 - Atendimento
Hospital 4 - Shopping Palladium | não atende: Cardiologica
Hospital 5 - Park Shopping Barigui | não atende: Cardiologica
Hospital 3 - Shopping Mueller Distância: 15.44 Leitos: 3 Plantonistas: 2 Especialistas: 2
Nota distância: 32,383 Nota leitos: 15,000 Nota plantonistas: 40,000 Nota especialistas: 66,667
Nota final: 677,634
Hospital 2 - Shopping Jardim das Américas Distância: 10.593 Leitos: 4 Plantonistas: 3 Especialistas: 3
Nota distância: 47,201 Nota leitos: 20,000 Nota plantonistas: 60,000 Nota especialistas: 100,000
Nota final: 1003,206
Hospital 1 - Shopping Cidade Distância: 6.534 Leitos: 4 Plantonistas: 2 Especialistas: 5
Nota distância: 76,523 Nota leitos: 20,000 Nota plantonistas: 40,000 Nota especialistas: 100,000
Nota final: 1119,137
Melhor hospital: Hospital 1 - Shopping Cidade
Tempo de execução: 480 ms

```

FIGURA 15 – TESTE DE CASO COMUM – SERVIDOR CENTRAL

A Figura 16 apresenta a tela de envio de informações - exibindo o resultado - do aplicativo móvel. O campo “Condição do paciente” exibe o valor “Enchimento capilar maior que 2 segundos”, definido a partir de perguntas realizadas pelo aplicativo nas telas anteriores. O campo “Tipo de lesão” possui o valor “Cardiológica”, definido também nas telas anteriores. Os campos criados para a versão de testes, Latitude e Longitude, tiveram os valores -25.53 e -49.26 inseridos.

Após o envio das informações foi obtido o hospital 1 como resposta, a 6,53 km de distância.

SOS Socorrista - Testes AMB1111

Condição do paciente:
Enchimento capilar maior que 2 segundos

Tipo de lesão:
Cardiológica

Enviar informações

Hospital: Hospital 1 - Shopping Cidade
Endereço: Av. Mal. Floriano Peixoto, 4984 - Hauer, Curitiba - PR
Distância: 6,53 km

Confirmar

Latitude:

Longitude:

Precisão do GPS: 29,00 m

FIGURA 16 – TESTE DE CASO COMUM – APLICATIVO MÓVEL

3.9.5.3 ANÁLISE DO RESULTADO

O hospital 1 foi escolhido, como ilustrado na Figura 16 por ser o melhor em quase todos os critérios, a distância foi um dos principais fatores, já que a lesão é de gravidade 3, como ilustrado na Figura 15.

3.9.6 TESTE DE HOSPITAL SEM LEITOS

Teste realizado com uma modificação para que o hospital 1 não possua leitos disponíveis, visando verificar a capacidade do sistema de escolher outro hospital que tenha características satisfatórias para o atendimento.

3.9.6.1 PARÂMETROS DE ENTRADA

Tipo de lesão: Queimadura;

Gravidade: 2 - Paciente possui lesões significativas;

Latitude: -25,52;

Longitude: -49,27;

3.9.6.2 RESULTADOS

A Figura 17 apresenta a saída de texto do servidor central. Foi recebido um objeto JSON, identificado o tipo de mensagem de atendimento, com os parâmetros definidos acima, e foram feitas requisições de informações somente para os hospitais que atendem o tipo de lesão definido (hospitais 1, 3, 4 e 5).

Após terem sido realizadas as requisições, o algoritmo de *fitness* forneceu a nota de cada hospital, exibindo também a nota de cada critério antes de seu peso ser aplicado. O servidor então escolheu o hospital com maior nota (hospital 5), e retornou-o como resposta para o aplicativo móvel.

```

Recebi: {"gravidadeLesao":2,"longitude":-49.27,"latitude":-25.52,"tipoMensagem":0,"tipoLesao":"Ortopedica
Tipo de mensagem: 0 - Atendimento
Hospital 2 - Shopping Jardim das Américas | não atende: Ortopedica
Hospital 5 - Park Shopping Barigui Distância: 8.868 Leitos: 6 Plantonistas: 3 Especialistas: 2
Nota distância: 56,382 Nota leitos: 30,000 Nota plantonistas: 60,000 Nota especialistas: 66,667
Nota final: 798,863
Hospital 1 - Shopping Cidade Distância: 5.723 Leitos: 0 Plantonistas: 2 Especialistas: 5
Nota distância: 87,367 Nota leitos: -200,000 Nota plantonistas: 40,000 Nota especialistas: 100,000
Nota final: 569,467
Hospital 3 - Shopping Mueller Distância: 12.729 Leitos: 3 Plantonistas: 3 Especialistas: 2
Nota distância: 39,280 Nota leitos: 15,000 Nota plantonistas: 60,000 Nota especialistas: 66,667
Nota final: 700,455
Hospital 4 - Shopping Palladium Distância: 18.241 Leitos: 9 Plantonistas: 6 Especialistas: 1
Nota distância: 27,411 Nota leitos: 45,000 Nota plantonistas: 100,000 Nota especialistas: 33,333
Nota final: 666,310
Melhor hospital: Hospital 5 - Park Shopping Barigui
Tempo de execução: 652 ms

```

FIGURA 17 – TESTE DE HOSPITAL SEM LEITOS – SERVIDOR CENTRAL

A Figura 18 apresenta a tela de envio de informações - exibindo o resultado - do aplicativo móvel. O campo “Condição do paciente” exibe o valor “Paciente possui lesões significativas”, definido a partir de perguntas realizadas pelo aplicativo nas telas anteriores. O campo “Tipo de lesão” possui o valor “Ortopédica”, definido também nas telas anteriores. Os campos criados para a versão de testes, Latitude e Longitude, tiveram os valores -25.52 e -49.27 inseridos.

Após o envio das informações foi obtido o hospital 5 como resposta, a 8,87 km de distância.

SOS Socorrista - Testes AMB1111

Condição do paciente:
Paciente possui lesões significativas

Tipo de lesão:
Ortopédica

Enviar informações

Hospital: Hospital 5 - Park Shopping Barigui
Endereço: R. Luiz Felipe Lopes, 600 - Curitiba - PR
Distância: 8,87 km

Confirmar

Latitude:

Longitude:

Precisão do GPS: 29,00 m

FIGURA 18 – TESTE DE HOSPITAL SEM LEITOS – APLICATIVO MÓVEL

3.9.6.3 ANÁLISE DO RESULTADO

O algoritmo identificou 0 leitos para o hospital 1 e decrementou a nota significativamente, como ilustra a Figura 17. Assim, mesmo sendo melhor em alguns critérios, não foi escolhido. Foi escolhido o hospital 5, como ilustra a Figura 18.

3.10 CONCLUSÃO DA VALIDAÇÃO

Pelos resultados obtidos conclui-se que o algoritmo de escolha do melhor hospital funciona em ocasiões diversas, sabendo sempre decidir qual é o melhor hospital em qualquer caso, levando em conta a quantidade de plantonistas e médicos especialistas disponíveis.

Desse modo, enquanto os servidores dos hospitais estiverem atualizados com o estado atual do hospital, o algoritmo será eficaz na escolha dos melhores hospitais.

Outro ponto importante é a capacidade do servidor central de não fazer requisições para hospitais que não atendem o tipo de lesão do paciente, desse modo não são gastos recursos avaliando um hospital que certamente não será capaz de alcançar a melhor pontuação.

4 APRESENTAÇÃO DO SOFTWARE

Este capítulo é responsável pela apresentação das telas e descrição de como realizar a instalação dos sistemas que compõe o SOS Socorrista. Bem como a instalação do servidor que simula o comportamento dos servidores hospitalais, para fins de teste.

4.1 INSTALAÇÃO/CONFIGURAÇÃO

Nesta seção serão explicados quais os requisitos de instalação e como instalar e configurar a plataforma SOS Socorrista a partir do CD entregue.

O CD contém uma pasta com os projetos (código-fonte) para importação no Eclipse e uma pasta com os arquivos binários exportados, assim como um arquivo de texto contendo estas mesmas instruções para instalação.

Para que seja feita a instalação do aplicativo móvel é necessário um dispositivo móvel - foram utilizados smartphones - executando o sistema operacional Android de versão 2.3 ou superior, com receptor GPS e acesso à internet. É necessário também marcar a opção que possibilite a instalação de aplicativos de origens desconhecidas, na seção Aplicativos (o nome da opção pode mudar para cada fabricante e versão do Android).

Para a instalação do servidor central e do servidor simulando os hospitais é necessário um computador com JDK (*Java Development Kit*) versão 7 ou superior, e o servidor Apache Tomcat versão 7 ou superior. É também necessário uma instalação do banco de dados MySQL para o servidor central,.

Para a importação dos projetos para o Eclipse, para o projeto Android é necessário a instalação do *plugin* ADT e o Android SDK, disponíveis em: <http://developer.android.com/sdk/index.html>, para os projetos dos servidores é necessário apenas o “*Eclipse IDE for Java EE Developers*”. O Eclipse está disponível em: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/junosr2>. Pode ser feita a importação dos projetos para o Eclipse: *File* → *Import* → “*Existing Projects into Workspace*”, clique em “*Browse...*”, navegue até a

pasta Código-fonte no CD e clique em OK. Selecione os projetos que deseja importar, marque a opção “*Copy projects into Workspace*” e clique em *Finish*.

Para instalar o aplicativo móvel, copie o arquivo SOS_Socorrista_Android.apk, na pasta Binários, para o cartão de memória do aparelho Android. Utilize algum programa gerenciador de arquivos para navegar até o arquivo copiado e execute-o, aparelho poderá exibir algumas opções, selecione a opção de instalação. O aplicativo estará instalado e pronto para configuração e utilização, seu nome é SOS Socorrista. A versão criada para testes é o arquivo SOS_Socorrista_Android_Testes.apk, siga os mesmos procedimentos.

Para instalar os servidores, simplesmente copie os arquivos SOS_Socorrista_Hospitais.war e SOS_Socorrista_Servidor_Central.war, da pasta Binários, para o diretório *webapps* da instalação do Tomcat e copie o conteúdo da pasta Dependências para o diretório “*lib*” da instalação do Tomcat. É recomendada a importação de pelo menos o projeto do servidor central para o Eclipse, pela possibilidade de alterar a configuração do banco de dados.

Caso esteja instalando o servidor central, é também necessário criar um banco de dados MySQL. Crie um banco chamado “*sossocorrista*” com *charset utf8* e *collation utf8_bin*. E execute o arquivo *sossocorrista.sql*, da pasta “Banco de dados”, no banco de dados, este arquivo criará as tabelas e irá inserir os dados utilizados nos testes realizados.

Para utilizar o aplicativo móvel ative a localização GPS do aparelho. Ao iniciar pela primeira vez será solicitado que se faça a configuração (Figura 20), selecione OK, será exibida a tela de configuração (Figura 21). Na opção “Endereço do servidor” (Figura 22) preencha com o IP e porta do servidor central, por exemplo: 192.168.1.105:8080. Na opção “Placa da ambulância” (Figura 23) preencha com alguma placa sem traços, por exemplo: AMB1111. O aplicativo móvel está pronto para ser utilizado. Para entrar na tela de configuração novamente, pressione o botão de menu do aparelho, enquanto na tela inicial (Figura 19), e selecione a opção Configurações. Para a versão de testes, siga os mesmos procedimentos.

Para o servidor central é necessário configurar a conexão com o banco de dados (por padrão, endereço “*localhost/sossocorrista*”, usuário “*root*”, e sem senha), e o endereço do servidor simulando os hospitais.

As configurações de conexão do banco de dados no servidor central podem ser alteradas no arquivo “*bancoDeDados.properties*”, no pacote

“br.ufpr.sossocorrista.servidorcentral.jdbc”. No banco de dados, altere o campo “urlWS” de cada registro da tabela hospital para que corresponda a configuração do servidor em que o servidor dos hospitais está localizado, será necessário alterar apenas o endereço IP e a porta do servidor.

4.2 TELAS

Esta seção apresenta as telas do aplicativo móvel e sua sequência de exibição. Não são apresentadas telas de outros sistemas, por não existir entrada de dados, apenas saídas textuais para fins de monitoramento de sua execução.

4.2.1 TELA 01 – INÍCIO

A tela apresentada na Figura 19 é apresentada ao iniciar o aplicativo. Possui um botão “Iniciar atendimento”, que ao ser pressionado exibe a TELA 03. A tela possui um menu - com a opção “Configurações”, que exibe o menu do aplicativo TELA 02 - que pode ser acessado pressionando o botão de menu do dispositivo móvel executando o aplicativo.

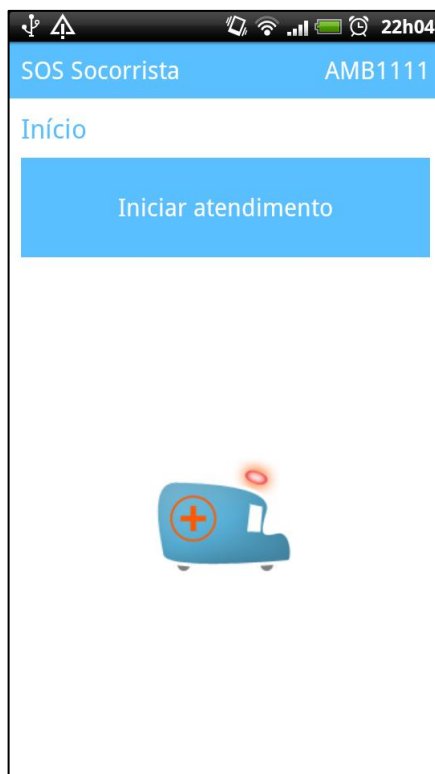


FIGURA 19 – TELA 01 – INÍCIO

Caso seja a primeira inicialização do aplicativo no aparelho, é exibido o diálogo exibido na Figura 20, ao pressionar o botão “OK”, é exibido o menu do aplicativo TELA 02.

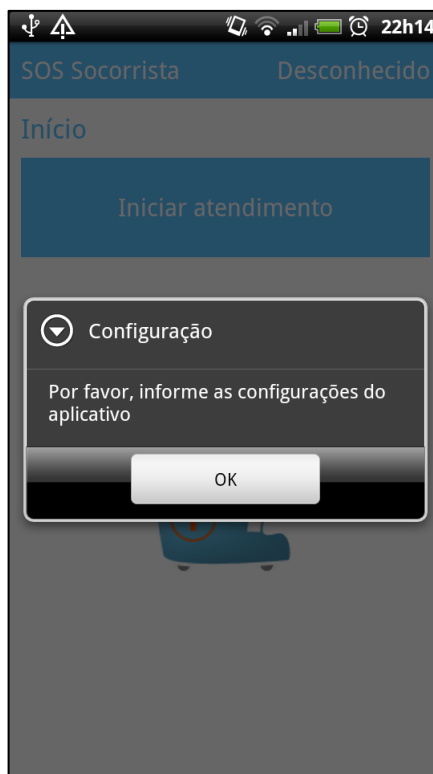


FIGURA 20 – TELA 01 – INÍCIO – PRIMEIRA INICIALIZAÇÃO

4.2.2 TELA 02 – MENU

A tela apresentada na Figura 21 é exibida ao escolher a opção “Configurações” no menu disponível na TELA 01, ou ao pressionar o botão “OK” no diálogo apresentado na TELA 01 em sua primeira inicialização. A tela possibilita a configuração do endereço IP do servidor central, e a placa da ambulância na qual o dispositivo atuará.

Ao selecionar a opção “Endereço do servidor”, é exibido o diálogo apresentado na Figura 22.

Ao selecionar a opção “Placa da ambulância”, é exibido o diálogo apresentado na Figura 23.

Ao pressionar o botão de voltar, no aparelho executando o aplicativo, a TELA 01 é exibida novamente.

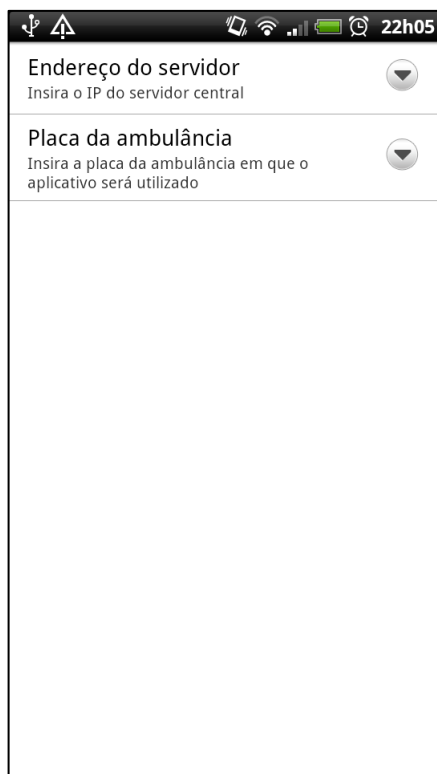


FIGURA 21 – TELA 02 – MENU

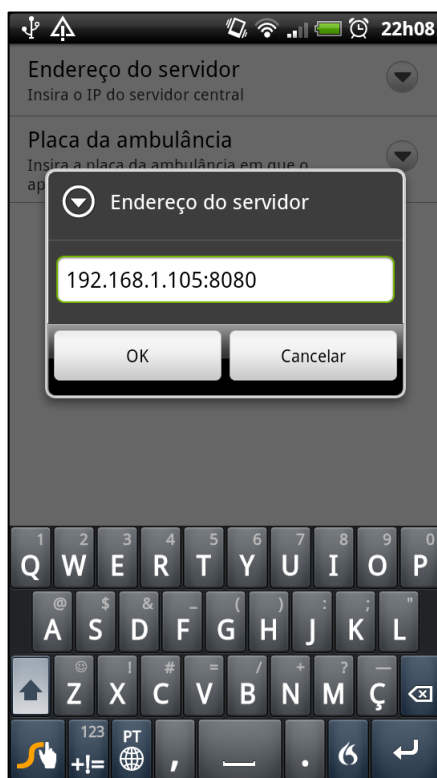


FIGURA 22 – TELA 02 – MENU – ENDEREÇO DO SERVIDOR



FIGURA 23 – TELA 02 – MENU – PLACA DA AMBULÂNCIA

4.2.3 TELA 03 – TIPO DE LESÃO

A tela apresentada na Figura 24 é exibida após a TELA 01. É exibida uma lista com tipos de lesão, ao selecionar o tipo de lesão do paciente, é exibida a TELA 04.

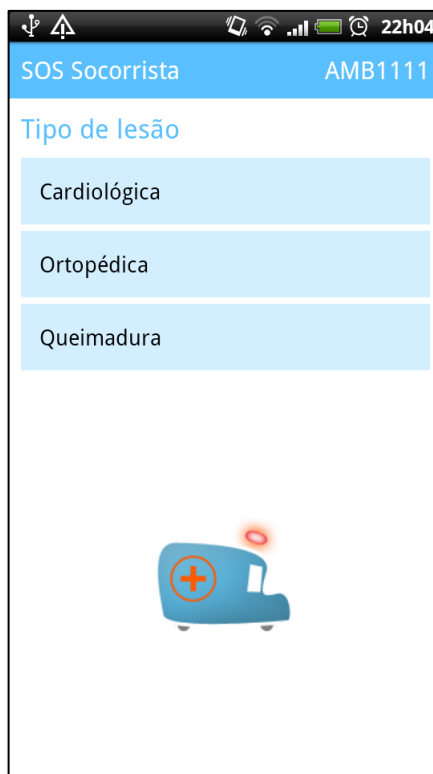


FIGURA 24 – TELA 03 – TIPO DE LESÃO

4.2.4 TELA 04 – RESPIRANDO

A tela apresentada na Figura 25 é exibida após a seleção do tipo de lesão do paciente, TELA 03. É a tela que inicia o algoritmo de definição de gravidade (Figura 2). É perguntado ao socorrista se o paciente estava respirando antes dos procedimentos de primeiros socorros.

Ao selecionar a opção “SIM” é exibida a TELA 05.

Ao selecionar a opção “NÃO” é exibida a TELA 09.



FIGURA 25 – TELA 04 – RESPIRANDO

4.2.5 TELA 05 – FREQUÊNCIA RESPIRATÓRIA

Na tela apresentada na Figura 26 é perguntado ao socorrista a frequência respiratória do paciente.

Ao selecionar a opção “Menor que 30” é exibida a TELA 06.

Ao selecionar a opção “Maior que 30” é exibida a TELA 09.



FIGURA 26 – TELA 05 – FREQUÊNCIA RESPIRATÓRIA

4.2.6 TELA 06 – ENCHIMENTO CAPILAR

Na tela apresentada na Figura 15 é perguntado ao socorrista a o tempo de enchimento capilar do paciente.

Ao selecionar a opção “Menor que 2s” é exibida a TELA 07.

Ao selecionar a opção “Maior que 2s” é exibida a TELA 09.



FIGURA 27 – TELA 06 – ENCHIMENTO CAPILAR

4.2.7 TELA 07 – RESPONDE COMANDOS

Na tela apresentada na Figura 28 é perguntado ao socorrista se o paciente responde a comandos.

Ao selecionar a opção “SIM” é exibida a TELA 08.

Ao selecionar a opção “NÃO” é exibida a TELA 09.



FIGURA 28 – TELA 07 – RESPONDE COMANDOS

4.2.8 TELA 08 – PEQUENAS LESÕES

Na tela apresentada na Figura 17 é perguntado ao socorrista se o paciente possui apenas pequenas lesões. Esta é a tela que finaliza todas as perguntas do algoritmo de definição de gravidade (Figura 2).

Ao selecionar qualquer opção é exibida a TELA 09;

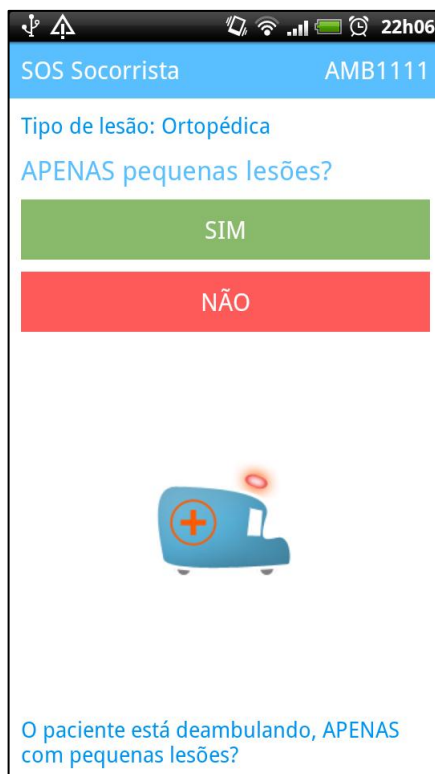


FIGURA 29 – TELA 08 – PEQUENAS LESÕES

4.2.9 TELA 09 – ENVIAR INFORMAÇÕES

Na tela apresentada na Figura 18 é exibida quando o aplicativo possui informações suficientes para definir a gravidade do paciente.

Nessa tela são exibidas as informações inseridas previamente, assim como um botão “Enviar informações”. Ao pressionar o botão, é disparada uma mensagem para o servidor central, com todas as informações necessárias para a decisão do melhor hospital a encaminhar o paciente.

The screenshot shows a mobile application interface with a black status bar at the top displaying various icons and the time 22h07. Below the status bar is a blue header bar with the text "SOS Socorrista" on the left and "AMB1111" on the right. The main content area is white and contains the following text: "Condição do paciente:" followed by "Paciente possui apenas pequenas lesões", "Tipo de lesão:" followed by "Ortopédica", a blue button labeled "Enviar informações", "Hospital:", "Endereço:", "Distância:", and a grey button labeled "Confirmar".

FIGURA 30 – TELA 09 – ENVIAR INFORMAÇÕES – ANTES DE ENVIAR

Após o envio das informações, e chegada a resposta do servidor central, a tela é atualizada para a apresentada na Figura 31. O botão “Enviar informações” é desabilitado, o botão “Continuar” é habilitado e são preenchidas as informações do hospital retornado pelo servidor central.

Ao pressionar o botão “Continuar” é exibida a TELA 10.

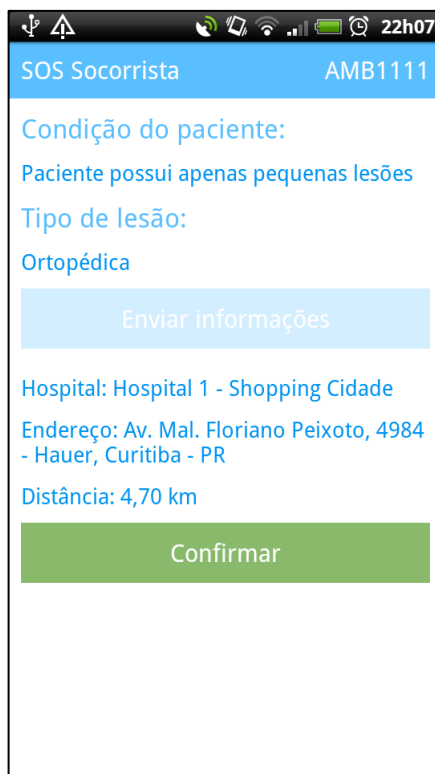


FIGURA 31 – TELA 09 – ENVIAR INFORMAÇÕES – DEPOIS DE ENVIAR

4.2.10 TELA 10 – FINALIZAR

Na tela apresentada na Figura 32 é exibida após a confirmação de encaminhamento para o hospital exibido na TELA 09.

Nessa tela são apresentadas as informações do hospital selecionado pelo servidor central, juntamente com os botões “Finalizar” e “Cancelar”.

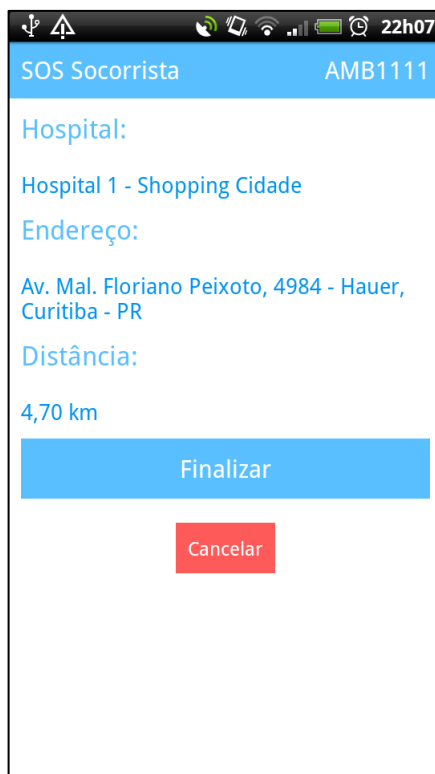


FIGURA 32 – TELA 10 – FINALIZAR

Ao pressionar o botão “Finalizar” é exibido o diálogo apresentado na Figura 33, ao selecionar a opção “Sim” é enviada uma mensagem ao servidor central comunicando que o paciente chegou ao hospital indicado. Em seguida a tela é fechada e é apresentada novamente a TELA 01, possibilitando um novo atendimento.

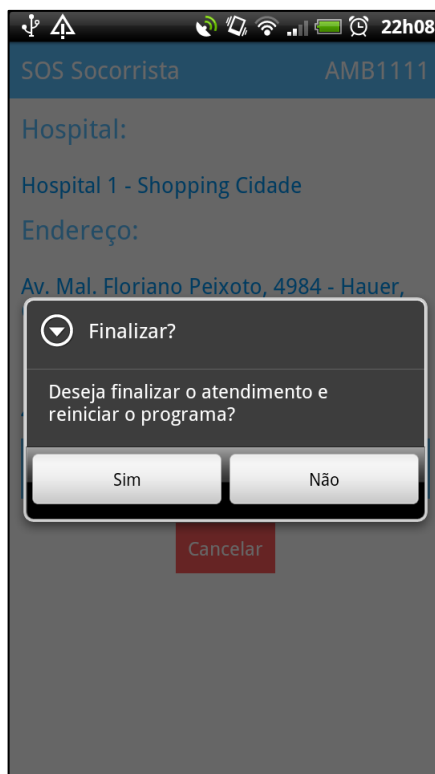


FIGURA 33 – TELA 10 – FINALIZAR – DIÁLOGO FINALIZAR

Ao pressionar o botão “Cancelar” é exibido o diálogo apresentado na Figura 33, ao selecionar a opção “Sim” é enviada uma mensagem ao servidor central comunicando o cancelando do atendimento. Em seguida a tela é fechada e é apresentada novamente a TELA 01, possibilitando um novo atendimento.



FIGURA 34 – TELA 10 – FINALIZAR – DIÁLOGO CANCELAR

5 CONSIDERAÇÕES FINAIS

Este projeto apresentou uma plataforma para otimizar o atendimento, realizado por socorristas, a acidentados.

O desenvolvimento de um algoritmo de *fitness* para a escolha do melhor hospital, para atender um paciente socorrido, foi um dos pontos positivos deste trabalho. Sendo um ponto fundamental e o mais importante para a obtenção de resultados satisfatórios.

A integração de todos os sistemas utilizando-se de tecnologias adotadas comercialmente, como REST e JSON, também é uma vantagem, já que estas tecnologias estão na vanguarda da engenharia de software. Além serem cada vez mais difundidas e utilizadas nos sistemas de informação que precisam de integração com outros sistemas. Fazendo do SOS Socorrista um sistema moderno.

Validação e testes foram realizados para comprovar a eficácia e funcionamento correto do algoritmo de *fitness*, além de verificar a comunicação entre todos os sistemas desenvolvidos.

A equipe interagiu de forma integral nos âmbitos do desenvolvimento do TCC, porém, uma dificuldade encontrada foi o atraso no cumprimento das atividades relacionadas no cronograma devido a dificuldades de comunicação com os médicos entrevistados, decidiu-se então aperfeiçoar o desenvolvimento do sistema. Foi feita uma distribuição de tarefas que correspondiam, de maneira mais adequada, com a experiência, prática, de cada membro da equipe.

A otimização do tempo despendido para a realização eficiente do TCC também agrega a lista de dificuldades, pois alguns integrantes realizam além de estágios, cursos externos e eventos.

Apesar das dificuldades, a equipe de desenvolvimento do SOS Socorrista conseguiu atingir os objetivos propostos, e desenvolver um sistema que pode, no futuro, ser utilizado no atendimento real de acidentados.

Como atividade futura para a melhoria do SOS Socorrista fica proposta a integração com um aplicativo de navegação GPS que exiba as melhores rotas disponíveis para a ambulância e exiba a localização do hospital selecionado.

Aumentar o escopo de lesões de acordo com necessidades reais e criar subgrupos para classificar as lesões como, por exemplo, citar as partes do corpo numa lesão ortopédica. Criar uma interface para realizar operações CRUD dos hospitais no sistema central.

Além de criar uma interface para monitorar as atividades do sistema em um ambiente mais visual mais adequado que não seja somente em um *console*.

REFERÊNCIAS

ALIBHAI, Zafeer, **What is Contract Net Interaction Protocol?**, IRMS Laboratory, jul.2003.

FIELDING, Roy T. **Architectural Styles and the Design of Network-based Software Architectures**. University of California, Irvine, 2000.

KODAMA, Y.; HAMAGAMI, T. **Distributed Processing System for Restoration of Electric Power Distribution Network Using Two-Layered Contract Net Protocol**. **IEEJ Transactions on Electronics, Information and Systems**, Volume 130, Issue 2, pp. 201-208, 2010. Disponível em: <<http://adsabs.harvard.edu/abs/2010ITEIS>>. Acesso em: 09/03/2013.

LECHETA, Ricardo. **Google Android**, Editora Novatec, 2010.

LIND, J.; FISCHER, K. **Transportation scheduling and simulation in a railroad scenario: a multi-agent approach**. DFKI, 1998. Disponível em: <<http://scidok.sulb.uni-saarland.de/volltexte/2011/3913/>>. Acesso em: 09/03/2013.

LINDEN, Ricardo. **Algoritmos Genéticos**. 2ª Edição, Rio de Janeiro, Editora Brasport, 2008.

MACEDO, Alexandre. **WebService Restful utilizando JSON**, Disponível em <<http://www.k19.com.br/artigos/web-service-restful-utilizando-json/>>. Acesso em: 05/02/2013

MACHADO, Jonathan. **O que é GPS?**, Disponível em <<http://www.tecmundo.com.br/conexao/215-o-que-e-gps-.htm> >. Acesso em: 05/02/2013

MCMURTRY, C; MERCURI, M; WATLING, N; **Microsoft Windows Communication Foundation: Hands-On**, Editora SAMS, 2006.

NEIL, Theresa, **Padrões de Design para Aplicativos Móveis**, Editora Novatec, 2012.

NOVAES JR, Fernando Souza, ABREU, Phillipe, CARVALHO, Fábio Henrique de, **SOS Doutor: emergências cirúrgicas em pronto-socorro**, Editora AC Farmacêutica, 2013.

PLÁCIDO, G. R; CUNHA, C. R; MORAIS, E. P; **Anarchitecture for interoperability And ubiquity of medical information. International Conference Applied Computing**. Rio de Janeiro. p. 501–506, 2011. Disponível em: <<http://hdl.handle.net/10198/6648>>. Acesso em: 09/03/2013.

RICHARDSON, Leonard; RUBY, Sam; **RESTful Serviços Web**. Editora Altabooks, 2007.

SANDHOLM, T. **An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations .Computer Science Department**, University of Massachusetts. Disponível em: <http://www.ppgia.pucpr.br/~alcides/Teaching/mestrado/FundamentosEngenhariaSoftware/artigos/sand93_1.pdf>. Acesso em: 05/02/2013.

SEBASTIAN, A. **Android now powers 75% of all smartphones sold. Are we heading towards a Google monopoly?**, Disponível em: <<http://www.extremetech.com/computing/139458-android-now-powers-75-of-all-smartphones-sold-are-we-heading-towards-a-google-monopoly>>. Acesso em: 03/12/2012.

SMITH, Reid G., **The Contract Net Protocol: High–Level Communication and Control in a Distributed Problem Solver**, IEEE transactions on computers, vol. c-29, no. 12, 1980.

SOARES, Fabrizzio. **Especialização em Desenvolvimento de Aplicações Web com Interfaces Ricas**, Instituto de Informática, UFG, 2012

VERSIGNASSI, Alexandre. **Como funciona o GPS?**, Disponível em <<http://mundoestranho.abril.com.br/materia/como-funciona-o-gps>>. Acesso em: 05/02/2013

A API da Matriz de distâncias do Google, Disponível em <<https://developers.google.com/maps/documentation/distancematrix/?hl=pt-br>>. Acesso em: 10/02/2013

Algoritmos Genéticos, Disponível em <<http://www2.icmc.usp.br/~andre/research/genetic/index.htm#apli>>. Acesso em: 05/02/2013

Android Is Popular Because It's Cheap, Not Because It's Good, Disponível em: <<http://gizmodo.com/5977625/Android-is-popular-because-its-cheap-not-because-its-good>>. Acesso em: 04/12/2012.

SAMSUNG, **Explicação do sistema operacional Google Android**, Disponível em <<http://www.samsung.com/br/article/google-android-os-explained>>. Acesso em: 03/12/2012

Google: 1.3 million Android devices activated every day, Disponível em <<http://www.h-online.com/open/news/item/Google-1-3-million-Android-devices-activated-every-day-1701513.html>>. Acesso em: 03/12/2012.

Manual do Atendimento Pré-Hospitalar, SIATE e Corpo de Bombeiros do Paraná, 2006. Disponível em: <<http://pt.scribd.com/doc/126519027/Manual-do-atendimento-pre-hospitalar-CBPR-pdf>>. Acesso em 05/02/2013.

Pesquisa mostra supremacia do Android frente ao iOS, Disponível em <<http://info.abril.com.br/noticias/tecnologia-pessoal/pesquisa-mostra-supremacia-do-android-frente-ao-ios-28012013-44.shl>>. Acesso em: 03/12/2012.

ANDROID DEVELOPERS, **Installing the Eclipse Plugin.**

Disponível em: <<http://developer.android.com/sdk/installing/installing-adt.html>>.

Acesso em: 11/06/2012.

Introdução ao JSON, Disponível em <<http://www.json.org/json-pt.html>>. Acesso em: 10/02/2013.

REST, Java Magazine , Edição 60. ago. 2008.

APÊNDICE A – PLANO DE RISCOS

PLANO DE RISCOS						
Nº	Condição	Data Limite	Consequência	Ação	Monitoramento	Probabilidade Impacto
1	Requisitos imprecisos.	01/09/2012	O sistema não atenderá aos requisitos.	Levantar novamente os dados necessários.	Prototipação do aplicativo e funcionalidades dos servidores.	Moderada Alto
2	Capacidade técnica.	Sem data.	Atraso no desenvolvimento do projeto.	Estimar tempo de estudo das tecnologias no cronograma.	Realizar reuniões e aprimorar conhecimentos técnicos.	Baixa Moderado
3	Problemas ocorridos com equipamentos.	Sem data.	Custo com novos equipamentos.	Ter cuidado com os equipamentos.	Não há.	Baixa Alto
4	Estabelecer e cumprir prazos.	15/03/2013	Atraso nas entregas.	Ser auxiliado por alguém com experiência em planejamento.	Acompanhamento, por parte do orientador, com o cumprimento dos prazos.	Moderada Alto
5	Aceitação da banca examinadora.	Sem data.	Possível cancelamento do projeto.	Apresentar os resultados semanalmente.	Reuniões semanais entre os membros e orientador.	Baixa Alto

TABELA 2 – PLANO DE RISCOS

APÊNDICE B – DIAGRAMA DE CASOS DE USO

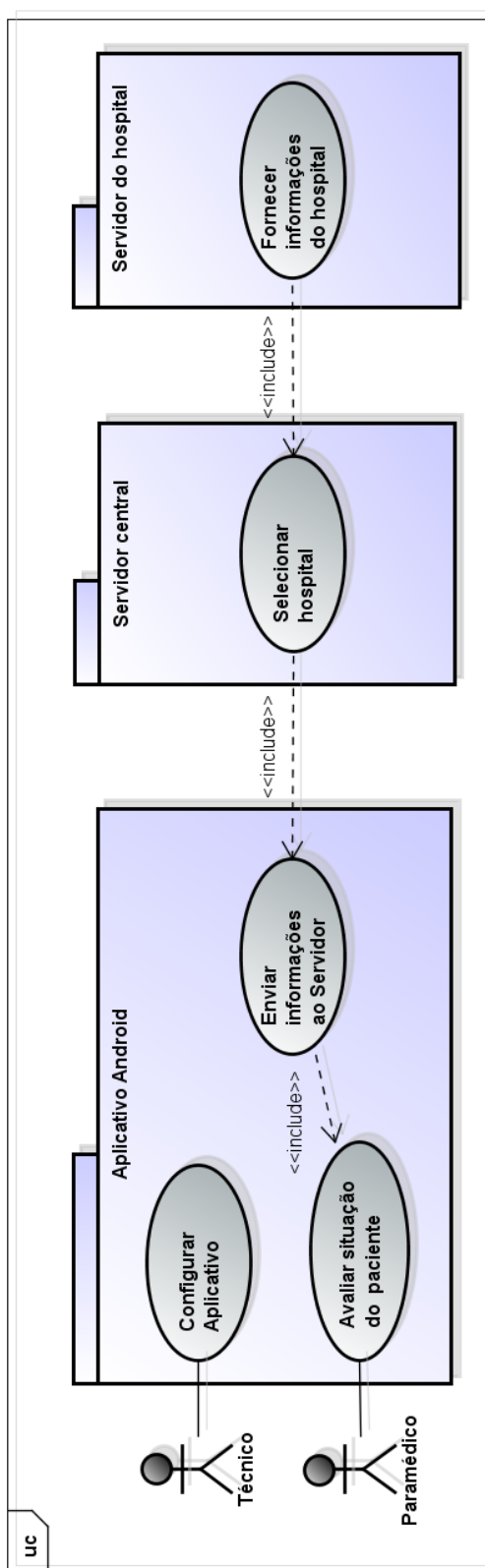


FIGURA 35 – DIAGRAMA DE CASOS DE USO

APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO

UC – CONFIGURAR APLICATIVO

Descrição sucinta

O usuário do aplicativo tem a opção de configurar o endereço IP do servidor e a placa da ambulância da qual o aplicativo está sendo usada.

Atores

1. Técnico

Fluxo Básico

1. Técnico inicia o aplicativo
2. Uma janela é aberta informando-o de que é necessária a configuração do aplicativo [Figura 36].
3. Técnico pressiona o botão “OK”.
4. Aplicativo exibe a tela de configuração do aplicativo [Figura 37].
5. Técnico pressiona a opção “Endereço do servidor”.
6. Aplicativo abre uma janela com o campo de entrada de texto “Endereço do servidor” (A1) [Figura 38].
7. Paramédico pressiona o botão “OK” (A2).
8. Aplicativo fecha a janela e salva as informações inseridas.
9. O caso de uso é encerrado.

Fluxos Alternativos

- A1. O paramédico escolhe a opção “Placa da ambulância”.
 - a. Será aberta uma janela com o campo de entrada de texto “Placa da ambulância”. [Figura 39].
- A2. Paramédico pressiona o botão “Cancelar”.
 - a. Aplicativo fecha a janela e descarta as informações inseridas.

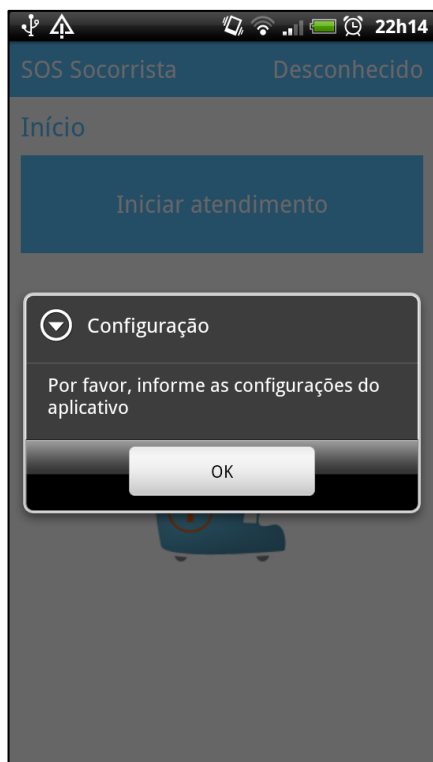


FIGURA 36 – UC – TELA 01

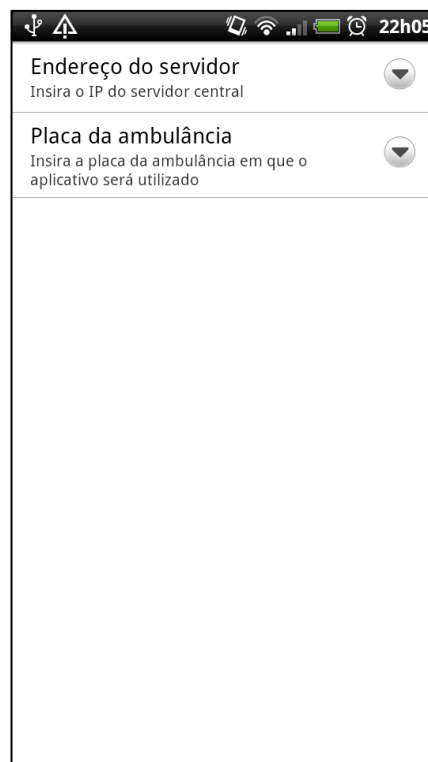


FIGURA 37 – UC – TELA 02

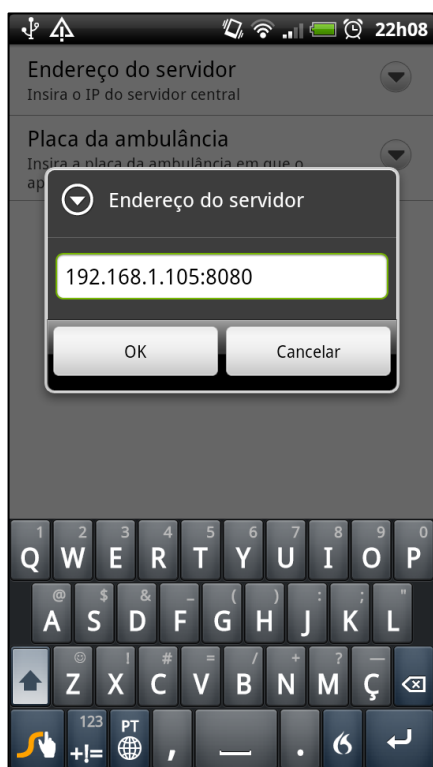


FIGURA 38 – UC – TELA 03

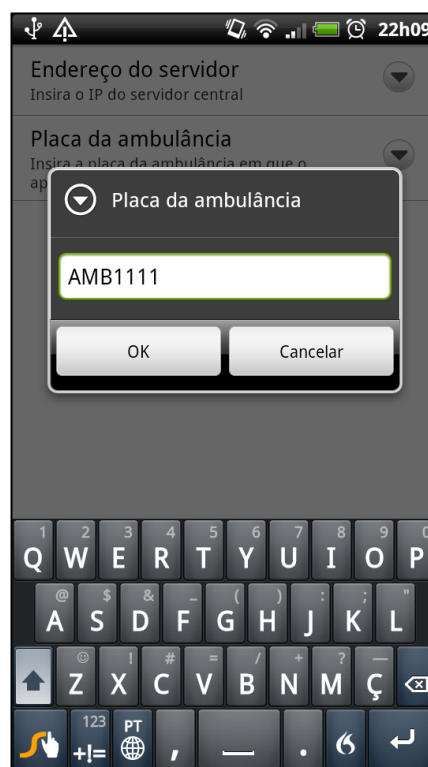


FIGURA 39 – UC – TELA 04

UC – AVALIAR SITUAÇÃO DO PACIENTE

Descrição sucinta

O paramédico responde uma série de perguntas ao aplicativo referente ao estado do paciente.

Atores

1. Paramédico

Pré-condições

O caso de uso “Configurar aplicativo” deve ter sido concluído.

Fluxo Básico

1. Paramédico inicia o aplicativo [Figura 40].
2. Paramédico pressiona o botão “Iniciar atendimento”
3. Aplicativo abre uma nova tela exibindo uma lista de lesões [Figura 41].
4. Paramédico seleciona o tipo de lesão sofrida pelo paciente.
5. Aplicativo exibe uma tela com o texto “Estava respirando?” [Figura 42].
6. Paramédico pressiona o botão “SIM” (A1).
7. Aplicativo exibe uma tela com o texto “Frequência respiratória:” [Figura 43].
8. Paramédico pressiona o botão “Menor que 30” (A2).
9. Aplicativo exibe uma tela com o texto “Enchimento capilar” [Figura 44].
10. Paramédico pressiona o botão “Menor que 2s” (A3).
11. Aplicativo exibe uma tela com o texto “Responde comandos:” [Figura 45].
12. Paramédico pressiona o botão “SIM” (A4).
13. Aplicativo exibe uma tela com o texto “APENAS pequenas lesões?” [Figura 42].
14. Paramédico pressiona o botão “SIM” (A5).
15. Aplicativo exibe a tela com o resumo dos dados do paciente [Figura 47].
16. O caso de uso é encerrado.

Fluxos Alternativos

- A1. Paramédico pressiona o botão “NÃO”.
 - a. Segue ao passo 15
- A2. Paramédico pressiona o botão “Maior que 30”.
 - a. Segue ao passo 15

- A3. Paramédico pressiona o botão “Maior que 2s”.
- a. Segue ao passo 15
- A4. Paramédico pressiona o botão “NÃO”.
- a. Segue ao passo 15
- A5. Paramédico pressiona o botão “NÃO”.
- a. Segue ao passo 15

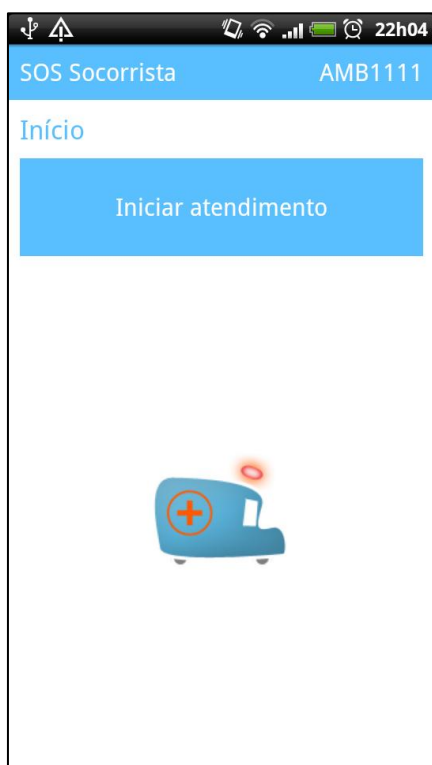


FIGURA 40 – UC – TELA 05

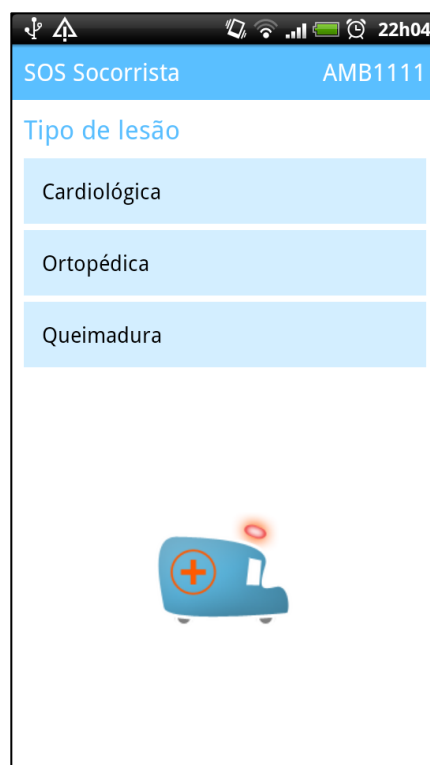


FIGURA 41 – UC – TELA 06

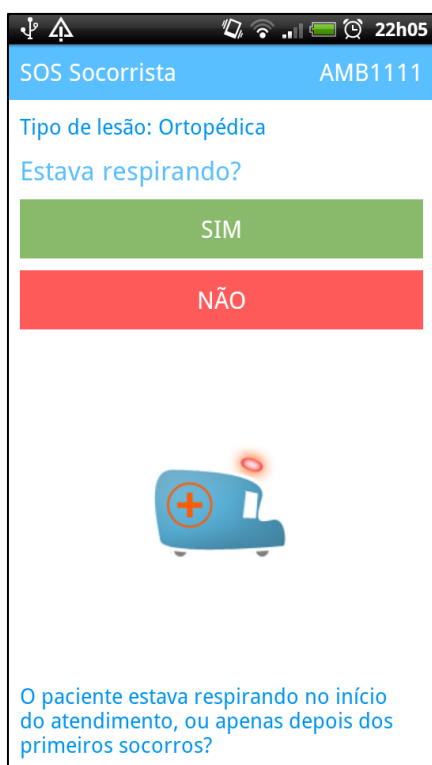


FIGURA 42 – UC – TELA 07



FIGURA 43 – UC – TELA 08

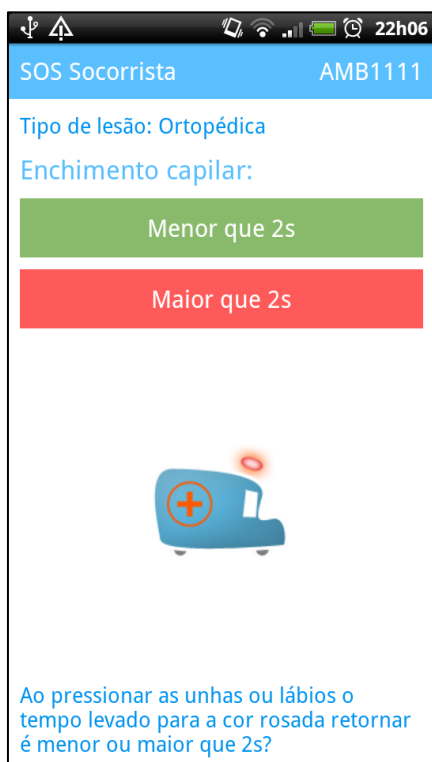


FIGURA 44 – UC – TELA 09

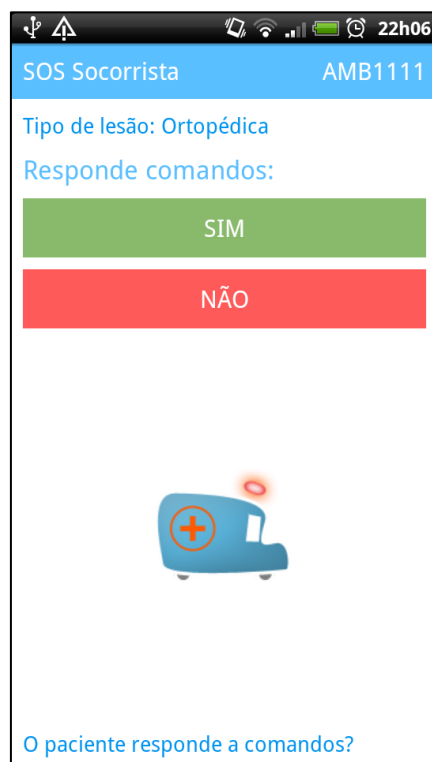


FIGURA 45 – UC – TELA 10

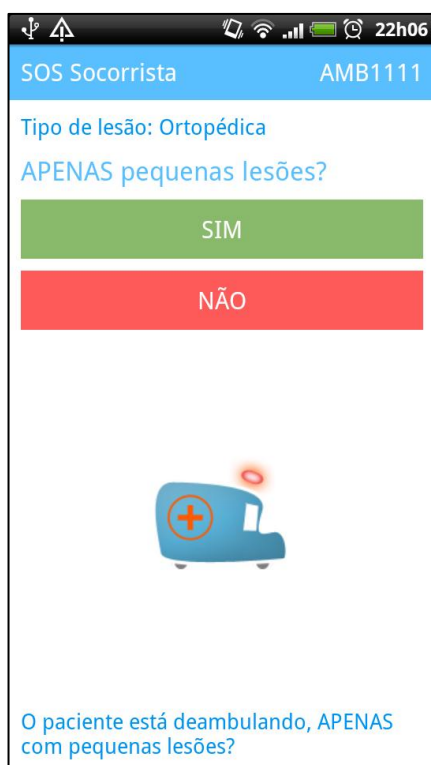


FIGURA 46 – UC – TELA 11

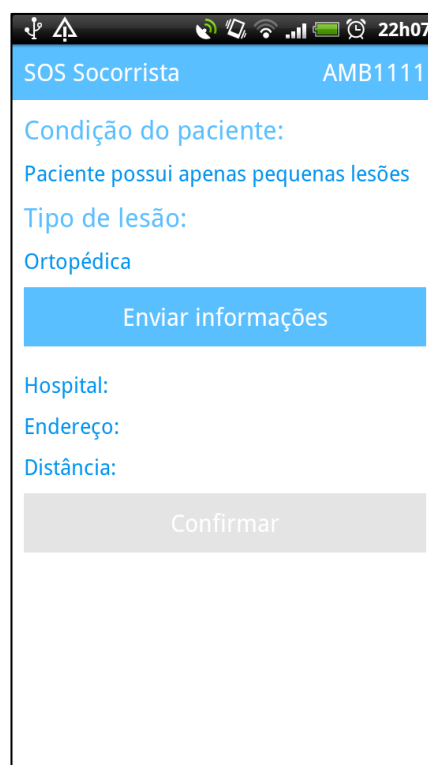


FIGURA 47 – UC – TELA 12

UC – ENVIAR INFORMAÇÕES AO SERVIDOR

Descrição sucinta

O paramédico envia os dados do paciente ao servidor central, para que este possa avaliar qual hospital melhor atende as necessidades do paciente.

Atores

1. Paramédico
2. Servidor Central

Pré-condições

O caso de uso “Avaliar situação do paciente” deve ter sido concluído para que se possa executar este caso de uso.

Fluxo Básico

1. Paramédico pressiona o botão “Enviar informações”
2. Aplicativo envia os dados do paciente ao servidor central (E1).

3. Servidor central seleciona o melhor hospital e retorna seus dados.
4. Aplicativo preenche os campos “Hospital”, “Endereço” e “Distância”, e habilita o botão “Confirmar” [Figura 49].
5. Paramédico pressiona o botão “Confirmar”.
6. Aplicativo envia mensagem de confirmação ao servidor central (E1).
7. Servidor central repassa a mensagem ao hospital (RN1).
8. Aplicativo exibe a tela de finalização [Figura 50].
9. Paramédico pressiona o botão “Finalizar” (A1).
10. Aplicativo exibe uma janela com o texto “Deseja finalizar o atendimento e reiniciar o programa?” [Figura 51].
11. Paramédico pressiona o botão “Sim”. (A2)
12. Aplicativo envia mensagem de finalização ao servidor central (E1).
13. Aplicativo é reiniciado.
14. O caso de uso é encerrado.

Fluxos Alternativos

- A1. Paramédico pressiona o botão “Cancelar”.
 - a. Aplicativo exibe uma janela com o texto “Deseja cancelar o atendimento e reiniciar o programa?” [Figura 52].
 - b. Paramédico pressiona o botão “Sim”. (A2)
 - c. Aplicativo envia mensagem de cancelamento ao servidor central (RN2) (E1).
- A2. Paramédico pressiona o botão “NÃO”.
 - a. A janela é fechada.

Regras de Negócio

RN1. O sistema do hospital reserva um leito ao paciente.

RN2. O sistema do hospital libera o leito reservado ao paciente.

Fluxos de Exceção

- E1. Erro na requisição ao servidor.
 - a. É exibida uma janela com a mensagem “Ocorreu um erro ao receber informações do servidor...” [Figura 48].
 - b. Paramédico pressiona o botão “Ok”.
 - c. O caso de uso é reiniciado.

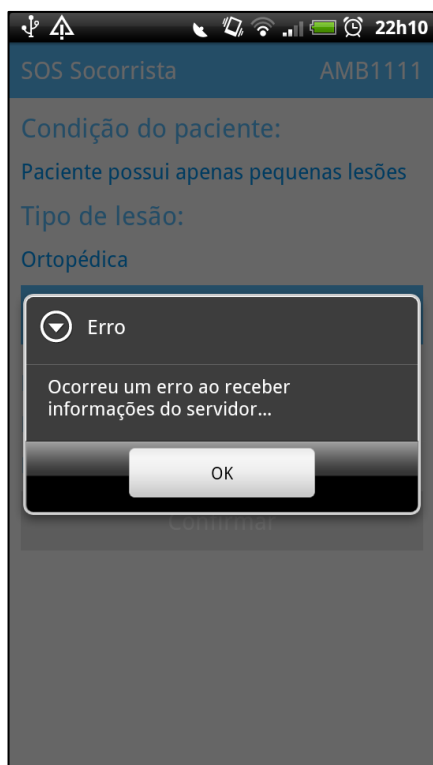


FIGURA 48 – UC – TELA 13

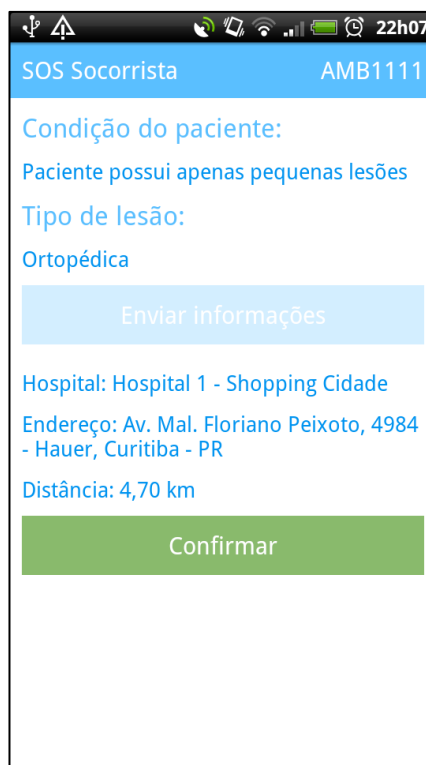


FIGURA 49 – UC – TELA 14

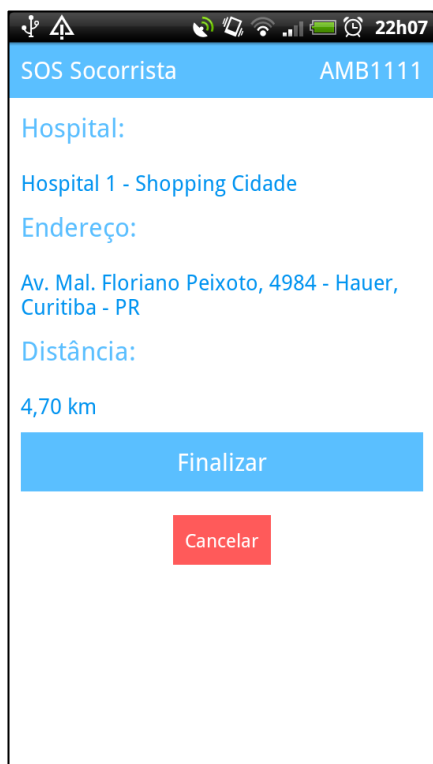


FIGURA 50 – UC – TELA 15

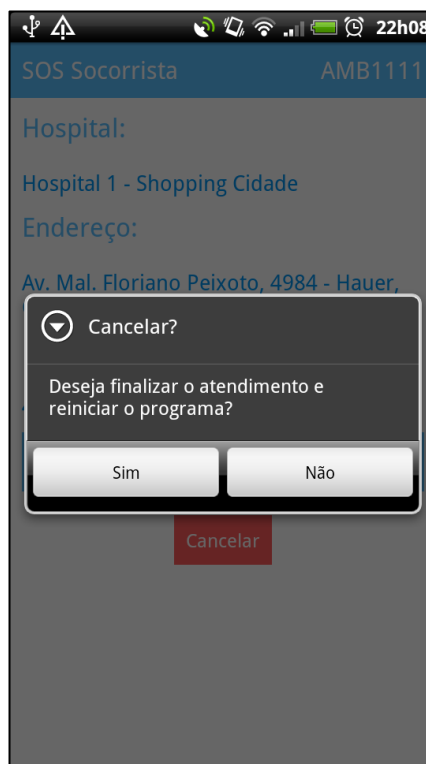


FIGURA 51 – UC – TELA 16

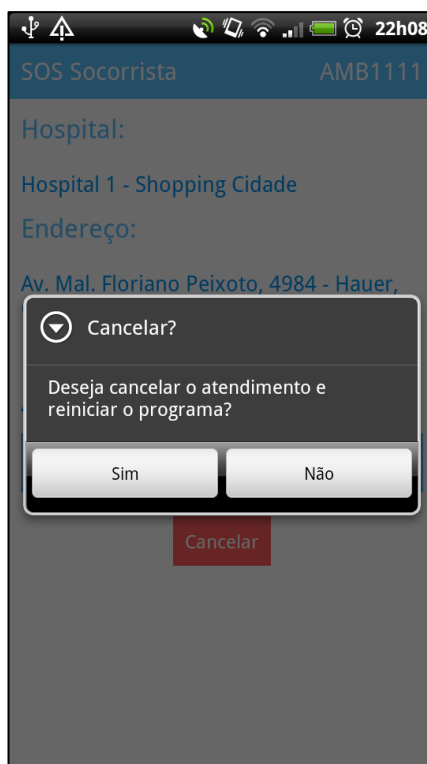


FIGURA 52 – UC – TELA 17

UC – SELECIONAR HOSPITAL

Descrição sucinta

Após o paramédico enviar os dados do paciente ao servidor central, este faz uma requisição aos servidores de cada hospital, solicitando dados para que o servidor central avalie a melhor opção.

Atores

1. Servidor Central

Pré-condições

O caso de uso “Enviar informações ao Servidor” deve estar sendo executado para que se possa executar este caso de uso.

Fluxo Básico

1. Servidor central recebe os dados do paciente através do aplicativo.
2. Servidor central requisita informações de cada servidor dos hospitais que atende ao tipo de lesão do paciente.
3. Servidor central obtém a distância entre a ambulância e cada hospital através API da Matriz de distâncias do Google.
4. Servidor central executa o algoritmo de *fitness* para cada hospital.
5. Servidor central seleciona a melhor nota.
6. Servidor central retorna o id, nome, endereço e distância do hospital selecionado ao aplicativo móvel.
7. O caso de uso é encerrado.

UC – FORNECER INFORMAÇÕES DO HOSPITAL**Descrição suscita**

O Servidor Central requisita informações ao servidor de cada hospital.

Atores

1. Servidor Central
2. Servidor Hospital

Pré-condições

O caso de uso “Selecionar hospital” deve estar sendo executado para que se possa executar este caso de uso.

Fluxo Básico

1. O servidor central requisita informações ao servidor do hospital.
2. O servidor do hospital retorna a quantidade de leitos disponíveis, quantidade de plantonistas disponíveis e a quantidade de especialistas no tipo de lesão do paciente disponíveis.
3. O caso de uso é encerrado.

APÊNDICE D – DIAGRAMAS DE CLASSES

DC – APLICATIVO MÓVEL

Como o aplicativo móvel foi desenvolvido para a plataforma Android, as classes de tela obrigatoriamente herdam da classe *Activity*, do Android. Para obter as coordenadas GPS utiliza-se a interface *LocationListener*. Foram expressos no diagrama apenas métodos da classe *Activity* e da interface *LocationListener* que foram implementados no aplicativo.

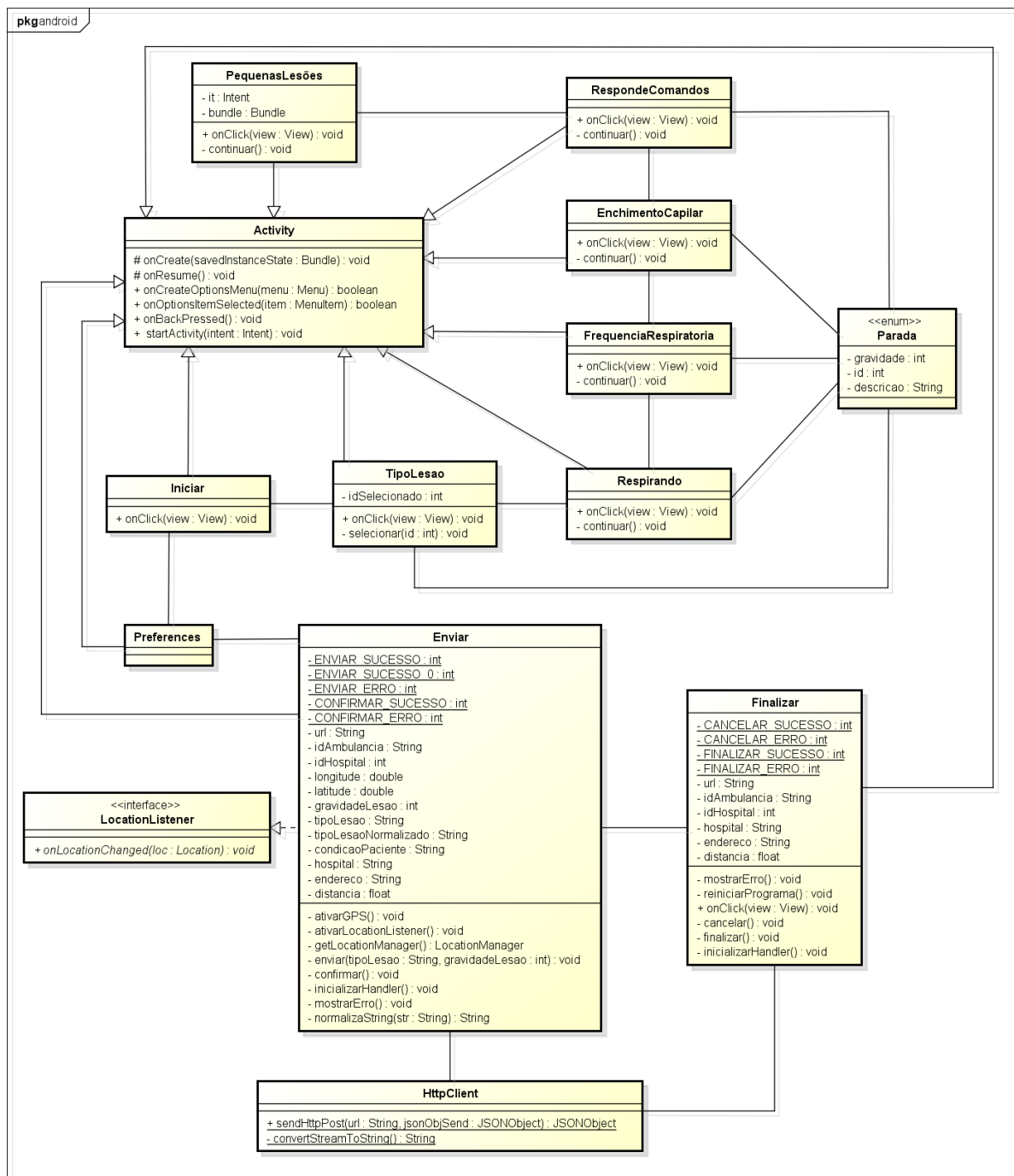


FIGURA 53 – DC – APLICATIVO MÓVEL

DC – SERVIDOR CENTRAL

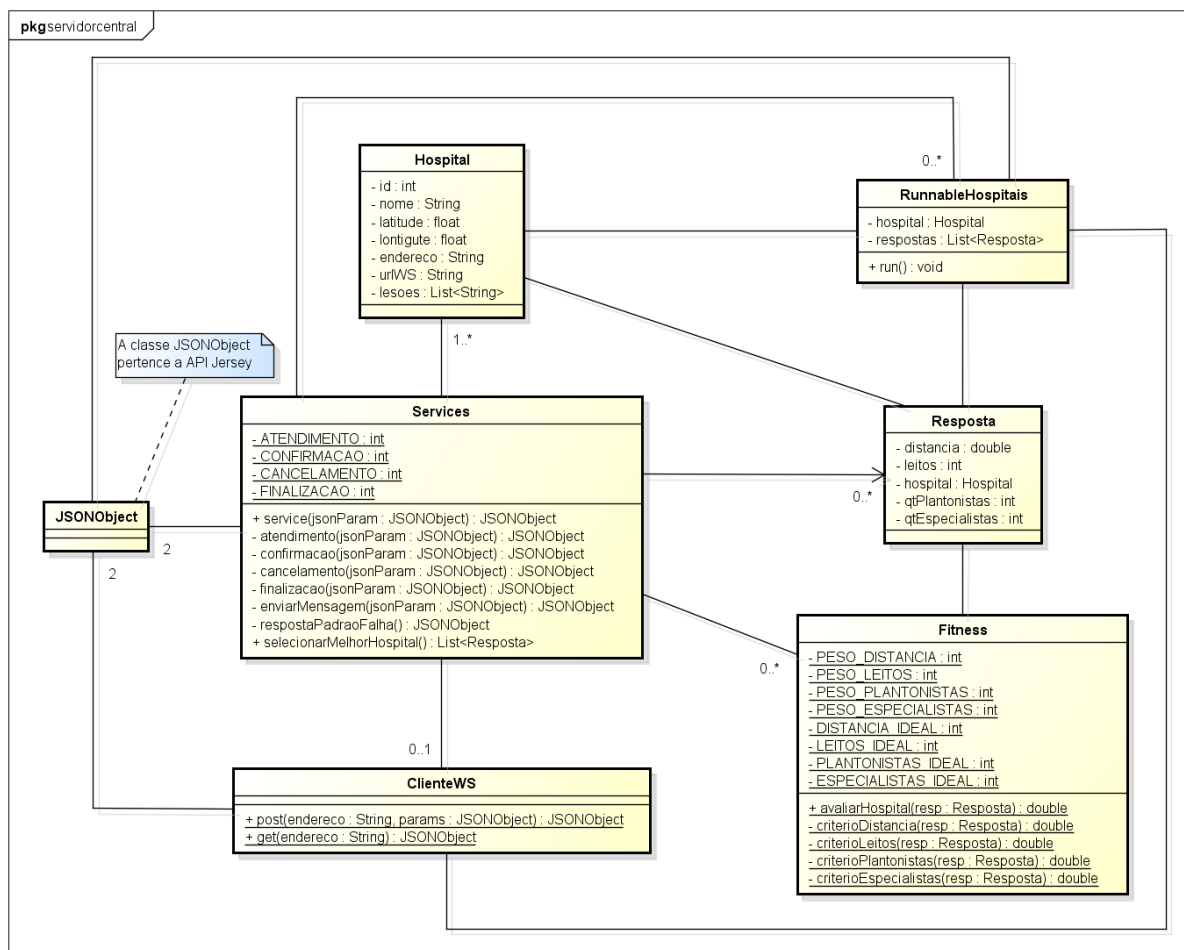


FIGURA 54 – DC – SERVIDOR CENTRAL

DC – HOSPITAIS

O diagrama seguinte é relativo ao servidor criado para simular os hospitais nos testes realizados. Este servidor não é o foco do projeto, foi criado apenas para testes. O servidor utiliza apenas uma classe com atributos *static* que representam as características dos hospitais que são alterados em tempo de execução dependendo do tipo de mensagem recebida.

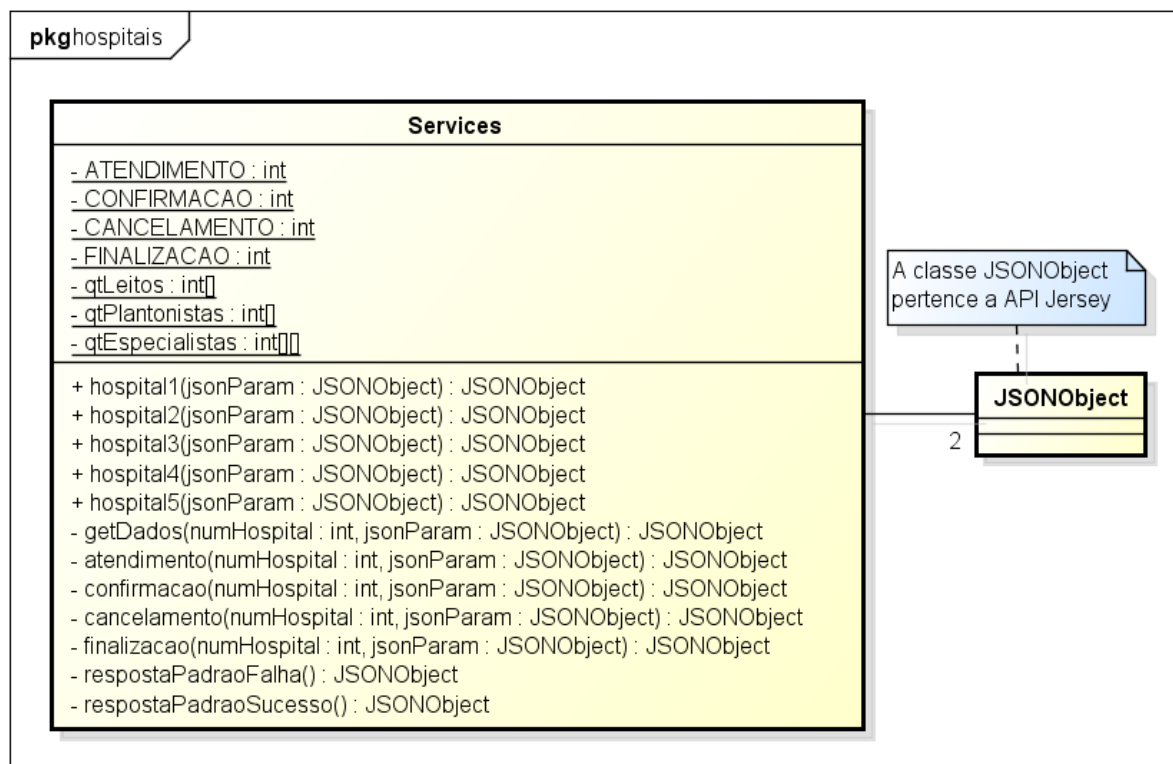


FIGURA 55 – DC – HOSPITAIS

APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA

DS – APLICATIVO MÓVEL – ATENDIMENTO

O diagrama a seguir é uma implementação do algoritmo de gravidade, juntamente com as funcionalidades do aplicativo de selecionar o tipo de lesão, e enviar as informações para selecionar o melhor hospital no servidor central. Chamadas aos métodos internos das telas foram omitidas para facilitar o entendimento, mantendo as interações do usuário.

Está dividido em quatro partes para facilitar a leitura, pois contempla uma sequência de exibição de telas com alternativas, causando uma grande quantidade de fragmentos de interação.

DS – APLICATIVO MÓVEL – ATENDIMENTO – 1

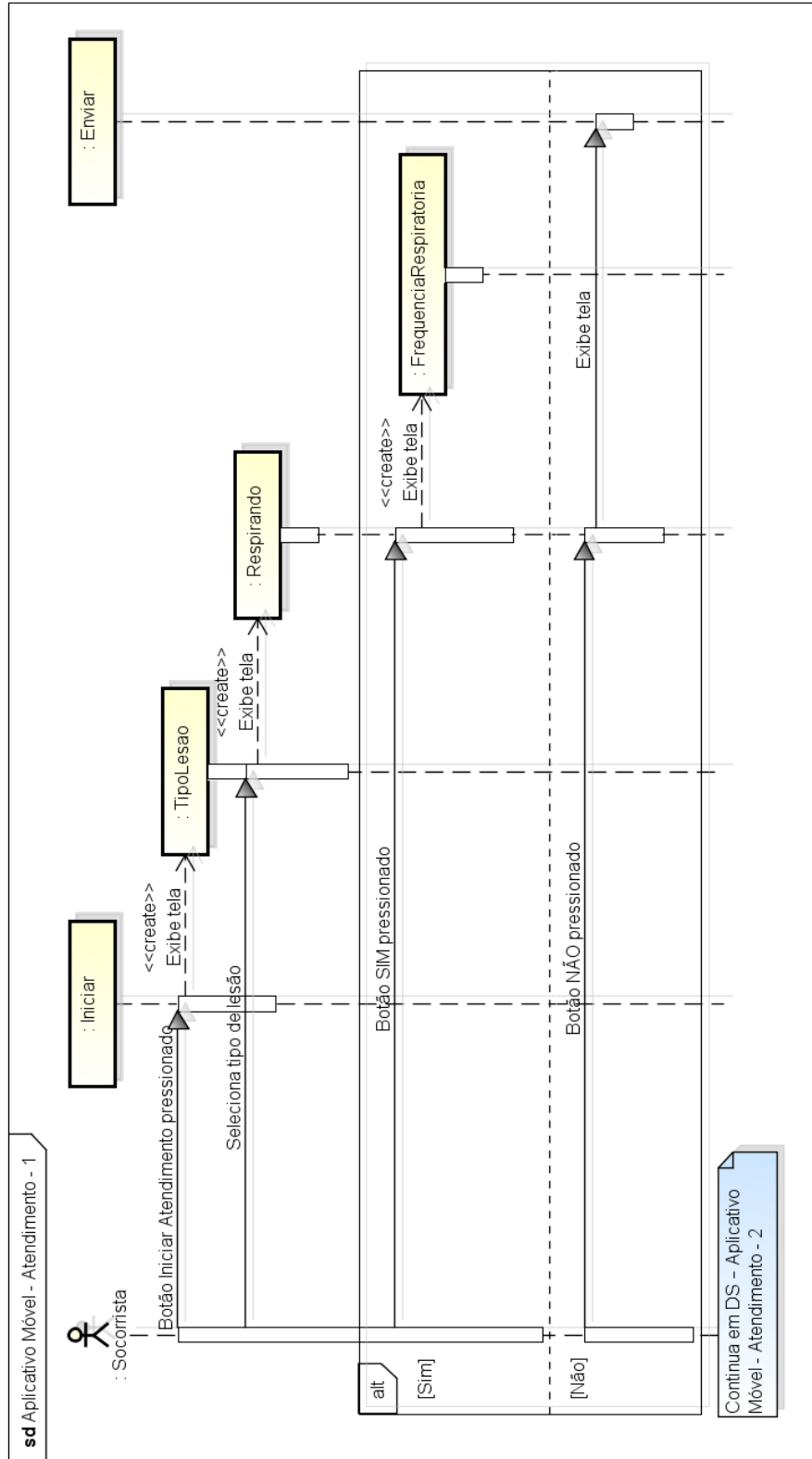


FIGURA 56 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 1

DS – APLICATIVO MÓVEL – ATENDIMENTO – 2

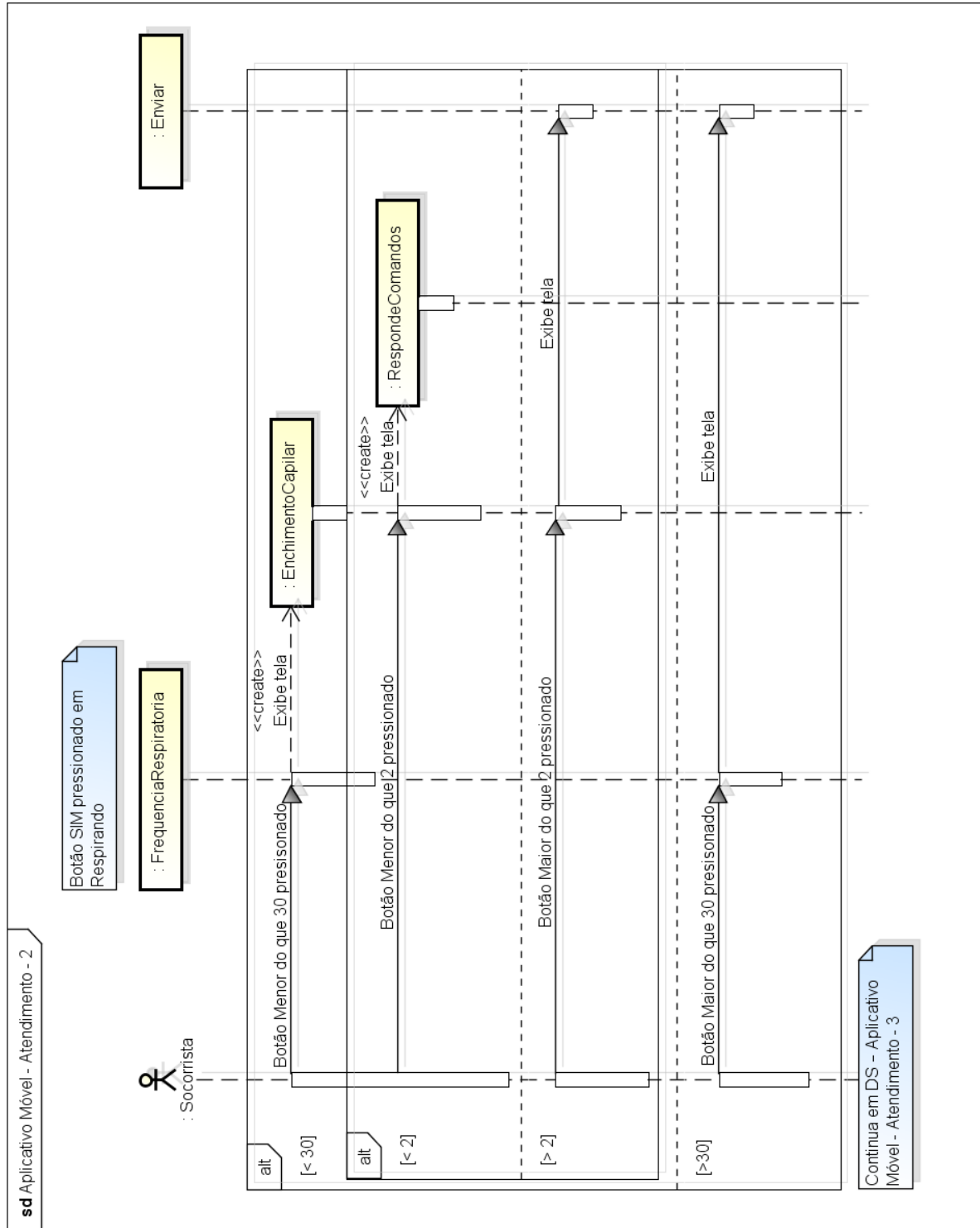


FIGURA 57 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 2

DS – APLICATIVO MÓVEL – ATENDIMENTO – 3

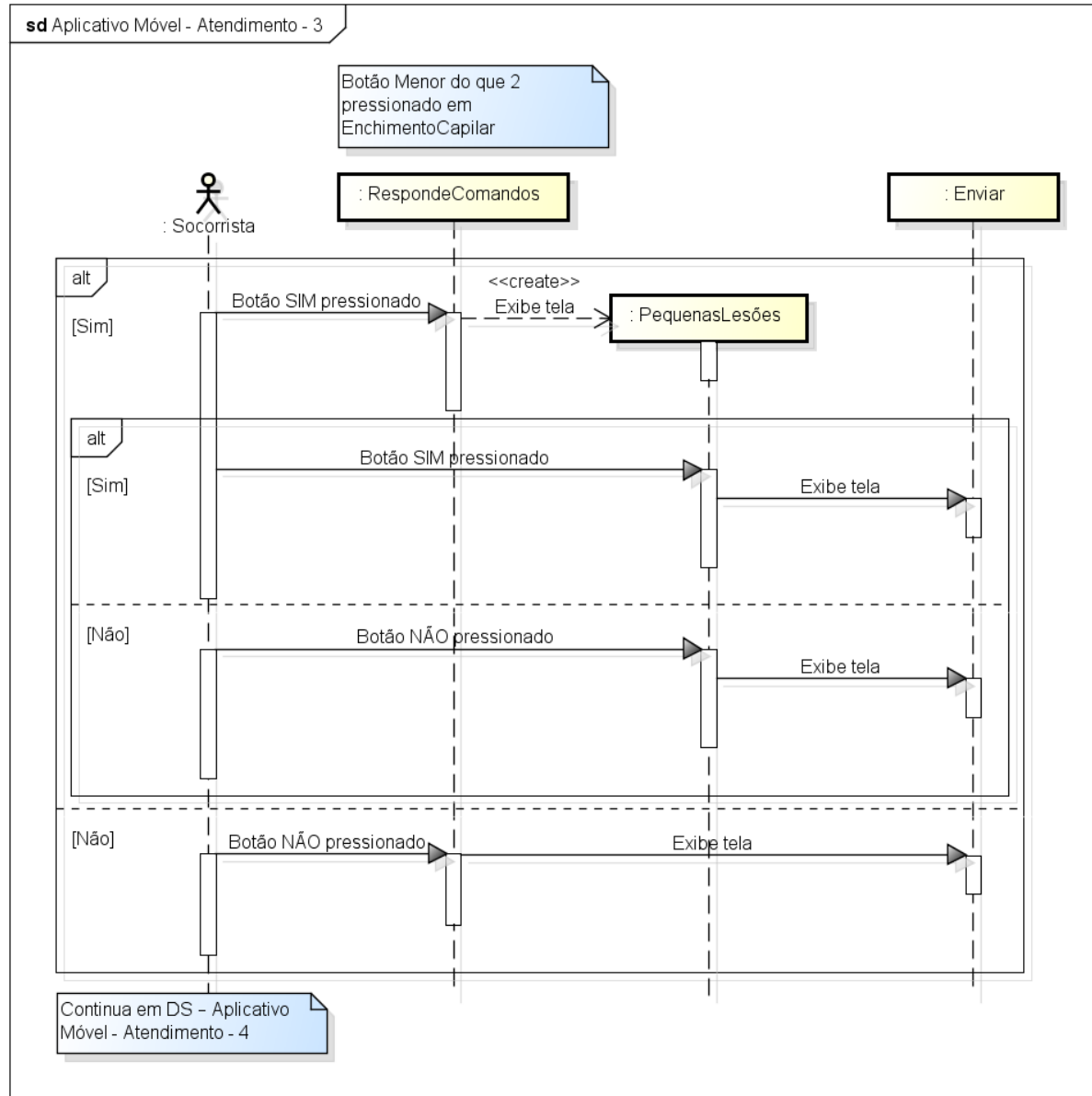


FIGURA 58 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 3

DS – APLICATIVO MÓVEL – ATENDIMENTO – 4

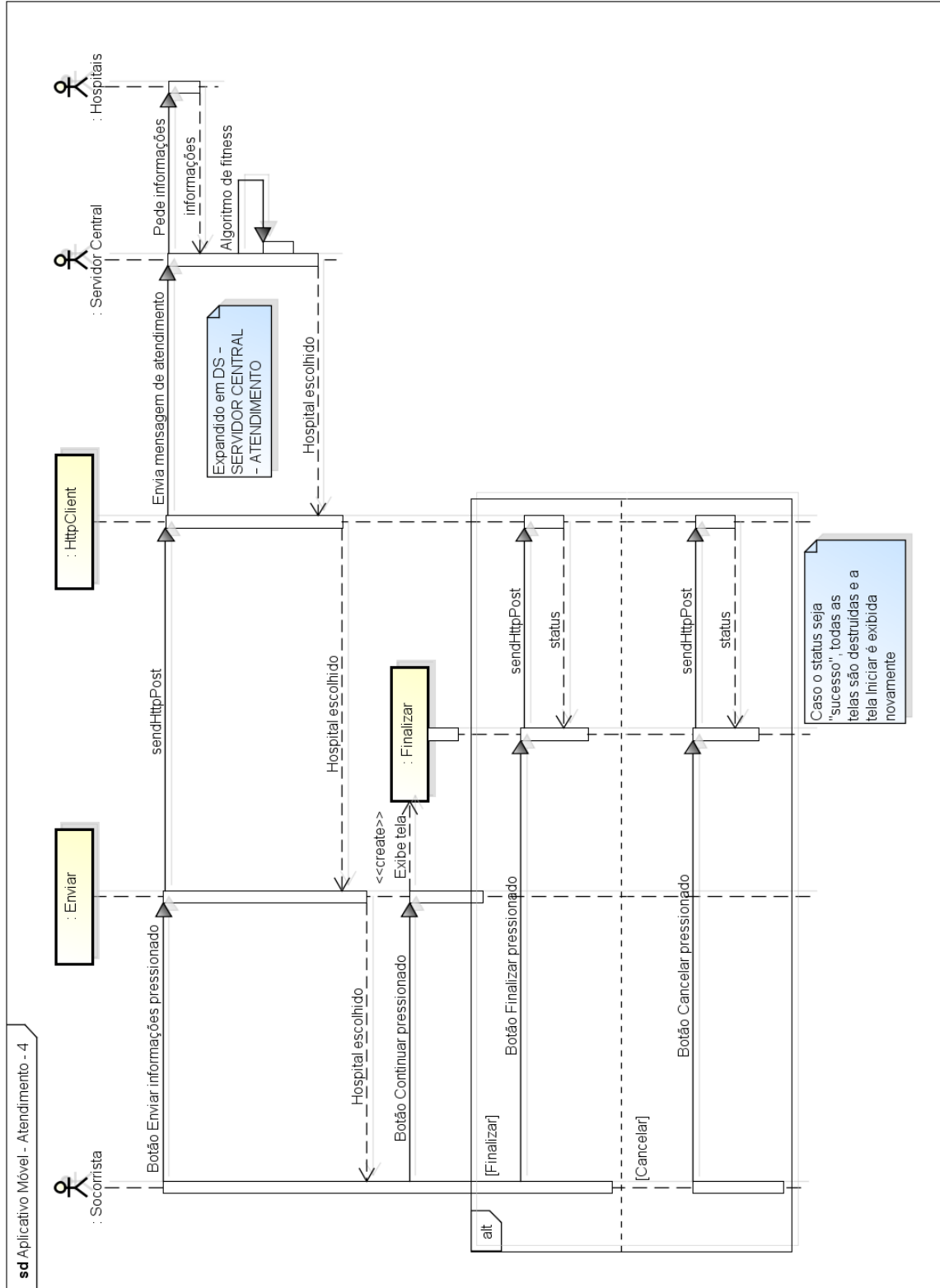


FIGURA 59 – DS – APLICATIVO MÓVEL – ATENDIMENTO – 4

DS – APLICATIVO MÓVEL – CONFIGURAR

A configuração do aplicativo ocorre na primeira inicialização do aplicativo e também quando o usuário pressiona o botão de menu do aparelho.

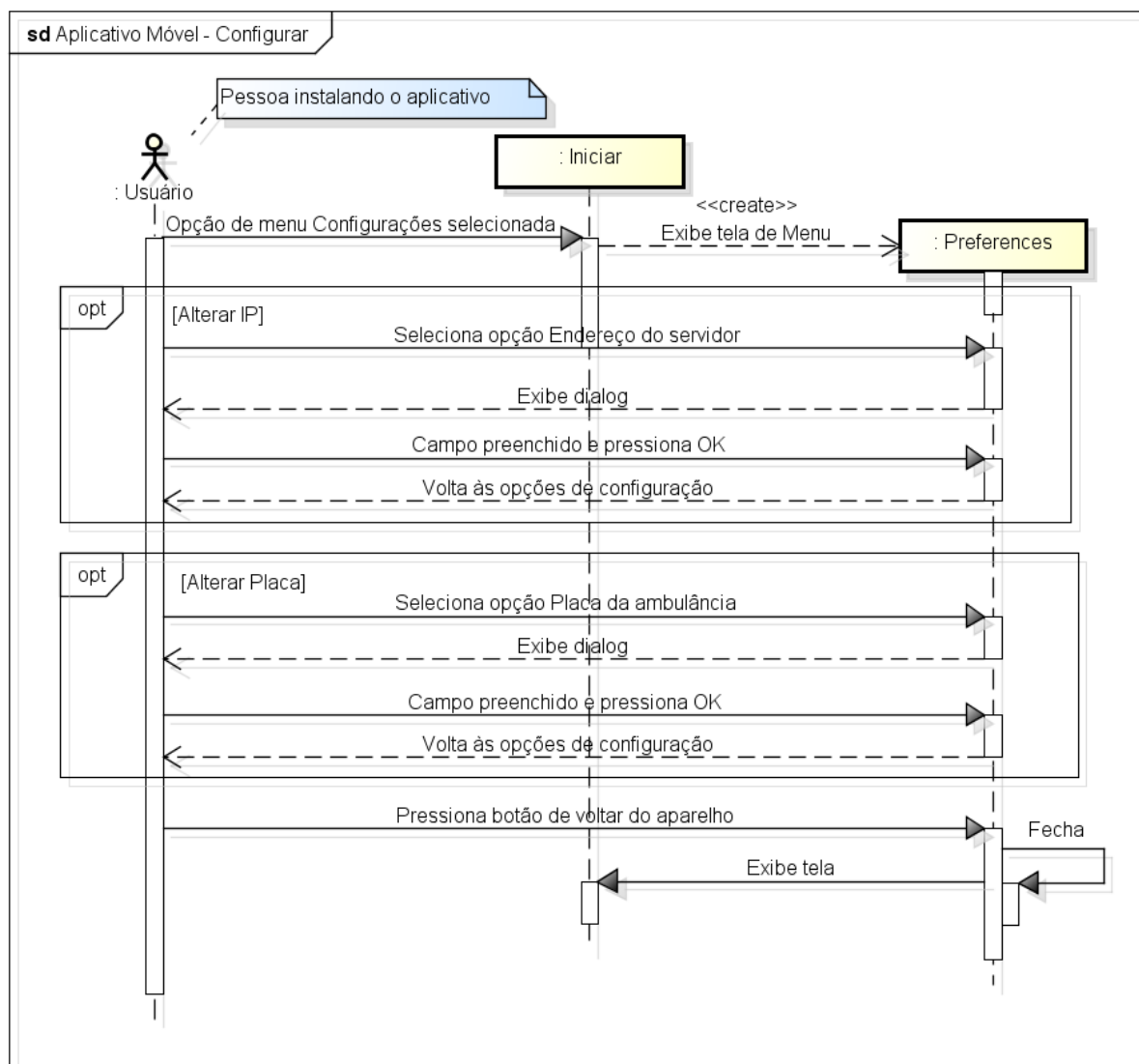


FIGURA 60 – DS – APLICATIVO MÓVEL – CONFIGURAR

DS – SERVIDOR CENTRAL – ATENDIMENTO

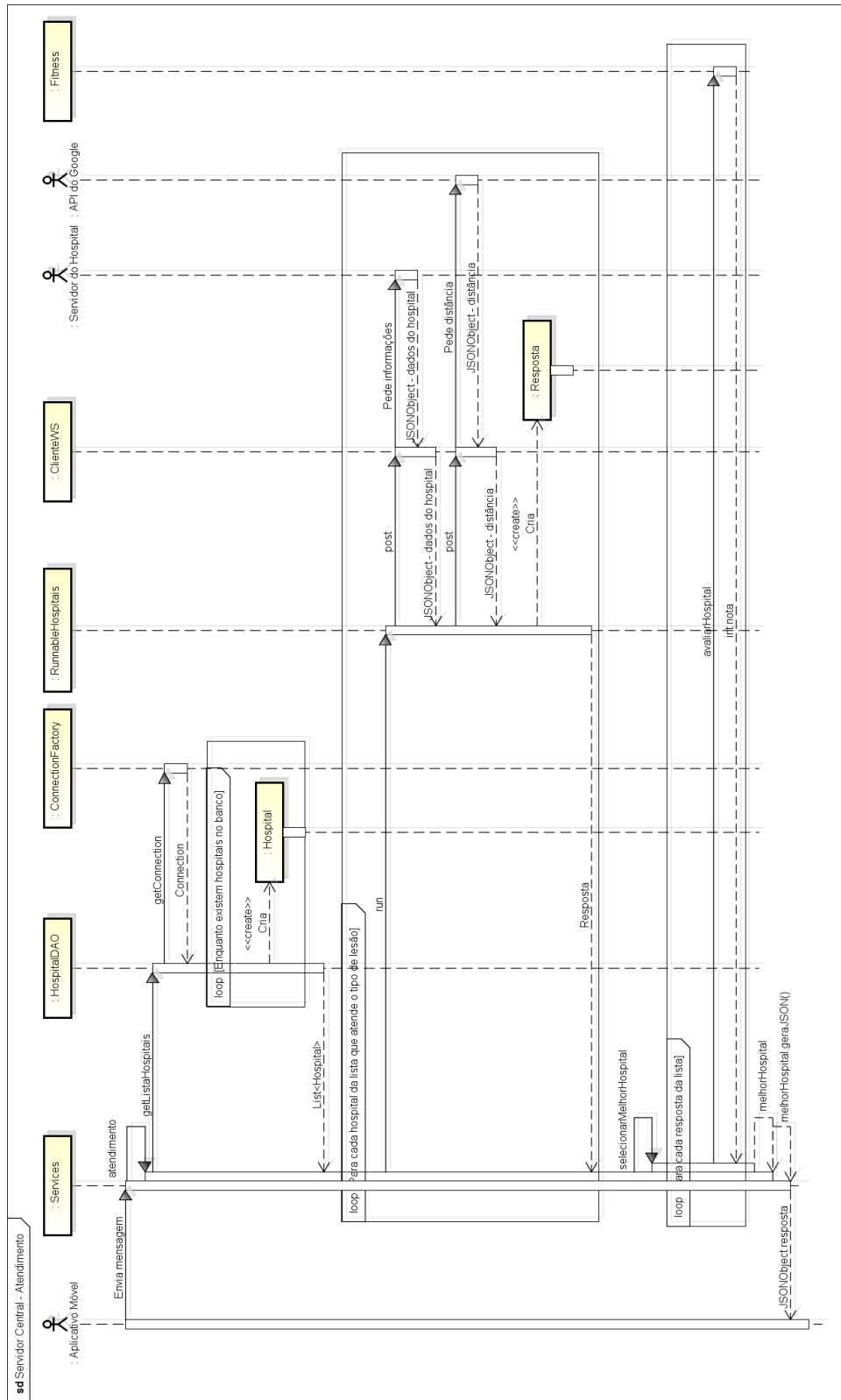


FIGURA 61 – DS – SERVIDOR CENTRAL – ATENDIMENTO

DS – SERVIDOR CENTRAL – OUTRAS MENSAGENS

O tratamento das mensagens diferentes de atendimento é o mesmo dentro do servidor central, apenas repassando-as para o hospital especificado na mensagem recebida.

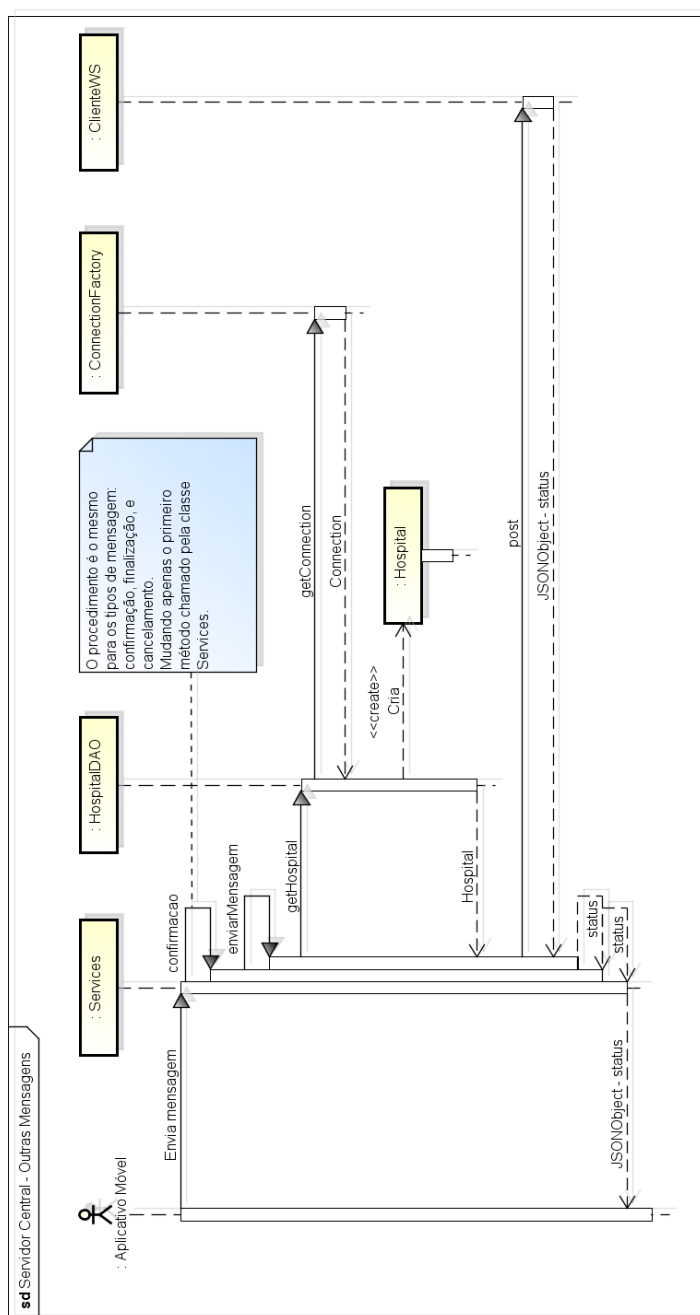


FIGURA 62 – DS – SERVIDOR CENTRAL – OUTRAS MENSAGENS

DS – HOSPITAIS

Tratando-se de um servidor criado apenas para testes, todos os tipos de mensagem são tratados na mesma classe, com suas variáveis de classe, sendo possível altera-las dinamicamente.

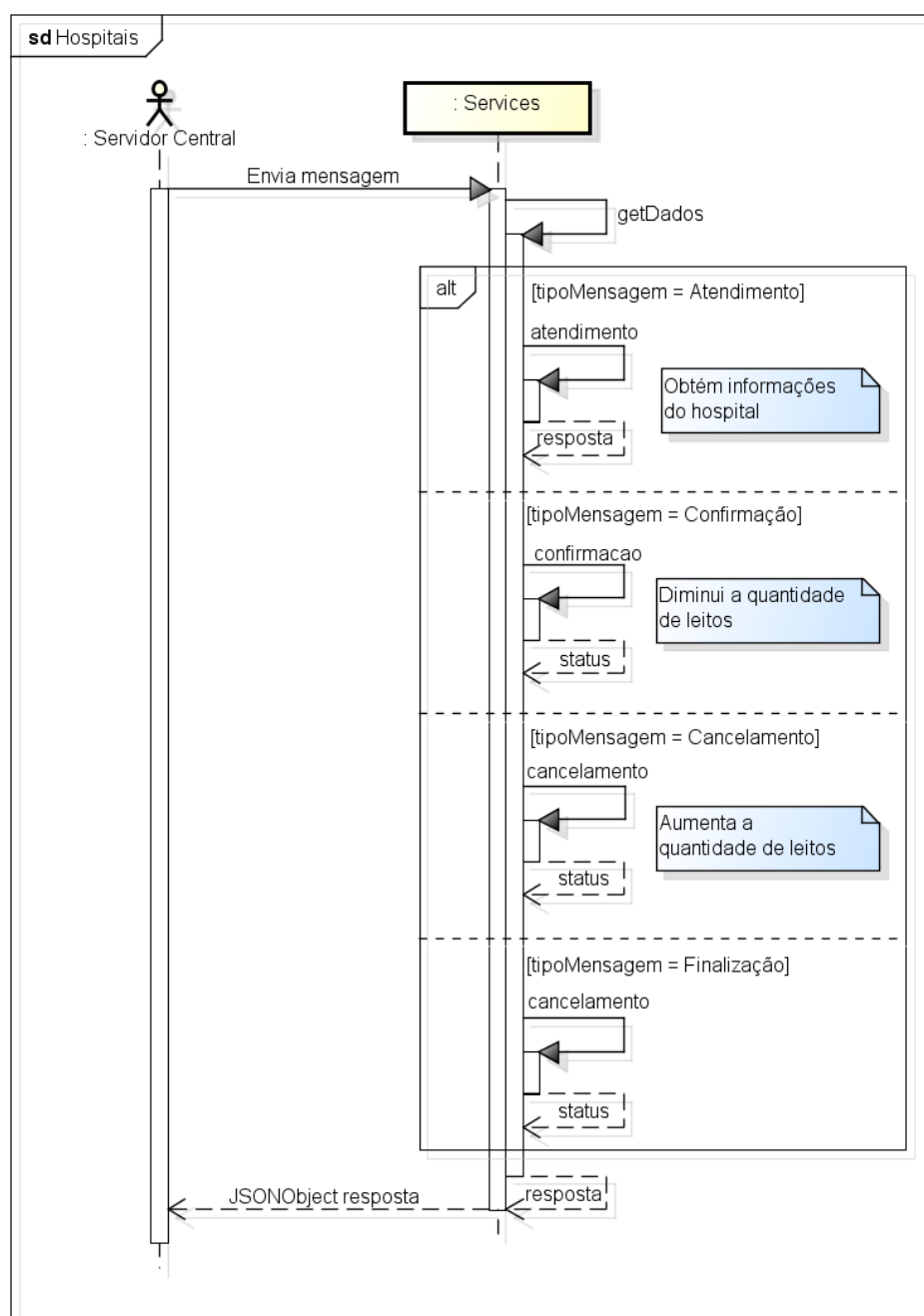


FIGURA 63 – DS – HOSPITAIS

APÊNDICE F – DIAGRAMA ENTIDADE–RELACIONAMENTO

Uma base de dados geralmente é uma parte fundamental dentro de um sistema da informação, contudo o foco deste projeto encontra-se no algoritmo de *fitness*. O banco de dados não foi um ponto crítico dentro do sistema. A base de dados adotada possui apenas duas tabelas com outra tabela relacionando-as, o banco de dados pertence ao sistema do Servidor Central.

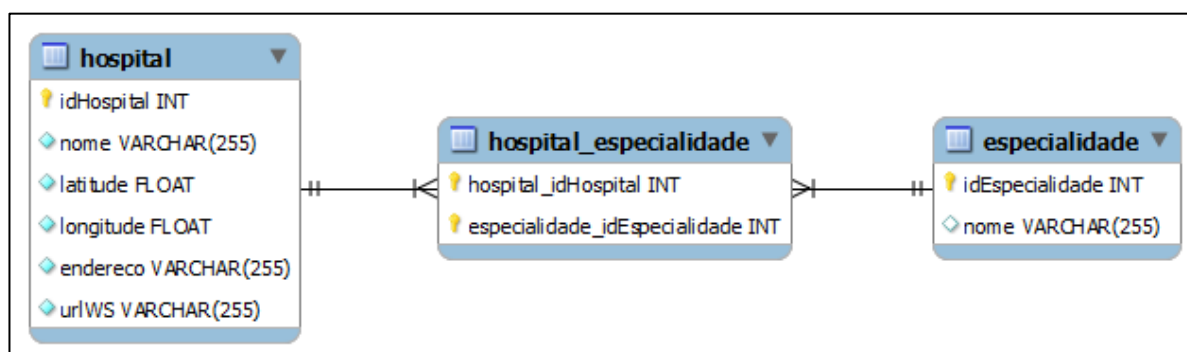


FIGURA 64 – DIAGRAMA ENTIDADE-RELACIONAMENTO

APÊNDICE G – DICIONÁRIO DE DADOS

hospital				
Campo	Descrição	Tipo	Referência	Nulo
idHospital	Código	INTEGER	PK	NOT NULL
nome	Nome do hospital	VARCHAR(255)		NOT NULL
latitude	Coordenada geográfica do hospital	FLOAT		NOT NULL
longitude	Coordenada geográfica do hospital	FLOAT		NOT NULL
endereço	Endereço físico do hospital	VARCHAR(255)		NOT NULL
urlWS	Endereço do Webservice do hospital na internet	VARCHAR(255)		NOT NULL

TABELA 3 – DICIONÁRIO DE DADOS – HOSPITAL

especialidade				
Campo	Descrição	Tipo	Referência	Nulo
idEspecialidade	Código	INTEGER	PK	NOT NULL
nome	Nome da especialidade (tipo de lesão)	VARCHAR(255)		NOT NULL

TABELA 4 – DICIONÁRIO DE DADOS – ESPECIALIDADE

hospital_especialidade				
Campo	Descrição	Tipo	Referência	Nulo
hospital_idHospital	Código do hospital	INTEGER	FK	NOT NULL
especialidade_idEspecialidade	Código da especialidade	INTEGER	FK	NOT NULL

TABELA 5 – DICIONÁRIO DE DADOS – HOSPITAL_ESPECIALIDADE