

UNIVERSIDADE FEDERAL DO PARANÁ
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Leonardo Ramón Bermúdez Alvarez
Rafael Kozar

PLATAFORMA DE TELEMEDICINA DE BAIXO CUSTO

CURITIBA, 2015

Leonardo Ramón Bermúdez Alvarez
Rafael Kozar

PLATAFORMA DE TELEMEDICINA DE BAIXO CUSTO

Trabalho de Conclusão do Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná, como requisito à obtenção do título de Tecnologia em Análise e Desenvolvimento de Sistemas Curitiba - PR.

Orientador: Prof. Dr. Luiz Antônio Pereira Neves.

CURITIBA, 2015

AGRADECIMENTOS

A Deus por ter nos dado saúde e força para superar as dificuldades.

A esta Universidade pela oportunidade de fazermos o curso.

Ao nosso Orientador/Professor Dr. Luiz Antônio Pereira Neves, pelo empenho dedicado à elaboração deste trabalho.

Aos nossos pais, esposa e filhos pelo amor, incentivo, apoio incondicional e que nos momentos de nossa ausência dedicada aos estudos, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente.

Nossos agradecimentos aos amigos, companheiros de trabalho que fizeram parte da nossa formação e que vão continuar presentes em nossas vidas com certeza.

E a todos que direta ou indiretamente fizeram parte da nossa formação, o nosso muito obrigado.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível. ”

Charles Chaplin

SUMÁRIO

1. INTRODUÇÃO	13
1.1. APRESENTAÇÃO DO TEMA.....	13
1.2. ESTRUTURA DO TRABALHO	13
1.3. OBJETIVOS	14
1.3.1 Objetivo geral	14
1.3.2 Objetivos específicos:.....	14
1.4. PROBLEMATIZAÇÃO	15
1.5. JUSTIFICATIVA	15
1.6. MOTIVAÇÃO.....	15
2. REVISÃO DE LITERATURA	16
2.1 PIA ROBOT	16
2.2 TELEMEDICINA E TELESSAÚDE	16
2.3 TELEMEDICINA NO BRASIL.....	17
2.4 EVOLUÇÃO DA TELEMEDICINA	18
3. VISÃO GERAL DO PROJETO	20
3.1 ESPECIFICAÇÕES INICIAIS	20
3.2 DEFINIÇÃO DO ESCOPO DO PROJETO	20
3.2.1 Descrição da arquitetura em nuvem.....	21
3.2.1.1 Tipo de nuvem.....	22
3.2.1.2 Provedor de nuvem	23
3.2.1.3 Sistema Operacional	23
3.2.1.4 Node.js	23
3.2.1.5 Motor V8 javascrip.....	24

3.2.1.6 Monoose.....	24
3.2.1.7 MongoDB	25
3.2.1.8 WebRTC.....	25
4. METODOLOGIA	27
4.1. MODELO INCREMENTAL	27
4.2. AMBIENTE DE HARDWARE	27
4.3. SOFTWARES UTILIZADOS.....	28
4.4 WBS	28
4.5 GANTT	29
4.6 PLANO DE COMUNICAÇÃO.....	29
4.7 PLANO DE RISCOS	30
4.8 DIAGRAMA DE CASOS DE USO	31
4.9 DESCRIÇÃO DE CASOS DE USO.....	32
4.10 DIAGRAMA DE CLASSES.....	39
5. APRESENTAÇÃO DO SOFTWARE	42
5.1 Tela de login.....	42
5.2 Tela inicial do administrador.....	43
5.3 Tela de lista de pacientes cadastrados	43
5.4 Tela de cadastro de pacientes	44
5.5 Tela atender paciente.....	45
5.6 Tela cadastro robô	46
5.7 Tela listar robôs.....	47
5.8 Tela cadastro usuário do sistema.....	48
5.9 Tela listar usuário	49
5.10 Tela lista de Atendimentos	50

5.11 Tela atendimento.....	51
5.12 Tela históricos	52
5.13 Tela histórico	52
6. CONCLUSÃO	54
6.1. PEQUENO ESCOPO DO PROJETO	54
6.2. OBJETIVOS	54
6.4. RELEVÂNCIA E CONTRIBUIÇÃO DO PROJETO.....	54
REFERÊNCIAS.....	56
APÊNDICES	59

LISTA DE FIGURAS

Figura 1 – Arquitetura de serviços em nuvem	22
Figura 2 - Digital Ocean.....	23
Figura 3 – Arquitetura típica de aplicação que utiliza WebRTC.....	26
Figura 4 – Diagrama WBS.....	29
Figura 5 – Diagrama de Gantt.....	29
Figura 6 – Diagrama de casos de uso.....	32
Figura 7 – Diagrama de classes.....	40
Figura 8 – Tela de login.....	41
Figura 9 – Tela Inicial do Administrador.....	42
Figura 10 – Tela Lista de pacientes cadastrados.....	43
Figura 11 – Tela de cadastro do paciente.....	44
Figura 12 – Tela Atender paciente.....	45
Figura 13 – Tela Cadastro do Robô.....	46
Figura 14 – Tela Lista de Robôs.....	47
Figura 15 – Tela Cadastro do Usuário.....	48
Figura 16 – Tela Lista de Usuários.....	49
Figura 17 – Tela Lista Atendimentos.....	50
Figura 18 – Tela Atendimento.....	50
Figura 19 – Tela lista de Históricos.....	51
Figura 20 – Tela Histórico.....	52
Figura 21 – Diagrama Cadastrar Atendente.....	60

Figura 22 – Diagrama Cadastrar Atendimento.....	61
Figura 23 – Diagrama Cadastrar Paciente.....	61
Figura 24 – Diagrama Cadastrar Robô.....	62
Figura 25 – Diagrama Editar Paciente.....	63
Figura 25 – Diagrama Editar Atendente.....	64
Figura 27 – Diagrama Excluir Atendente.....	64
Figura 28 – Diagrama Editar Robô.....	65
Figura 29 – Diagrama Excluir Robô.....	66
Figura 30 – Diagrama Excluir Paciente.....	67
Figura 31 – Diagrama Movimentar Robô.....	67
Figura 32 – Diagrama Realizar atendimento com Vídeo.....	68
Figura 33 – Diagrama Receber Atendimento.....	69

LISTA DE QUADROS

Tabela 1 – Plano de comunicação.....	30
Tabela 2 – Plano de riscos.....	30
Tabela 3 – Caso de uso realizar atendimento com vídeo.....	32
Tabela 4 – Caso de uso movimentar robô.....	33
Tabela 5 – Caso de uso cadastrar atendimento.....	34
Tabela 6 – Caso de uso cadastrar paciente.....	35
Tabela 7 – Caso de uso cadastrar usuário.....	35
Tabela 8 – Caso de uso cadastrar robô.....	36
Tabela 9 – Caso de uso editar paciente.....	37
Tabela 10 – Caso de uso editar robô.....	37
Tabela 11 – Caso de uso editar atendente.....	38
Tabela 12 – Caso de uso excluir paciente.....	39
Tabela 13 – Caso de uso excluir robo.....	39
Tabela 14 – Caso de uso excluir atendente.....	40

LISTA DE SIGLAS

ARDUINO – Plataforma micro processada de prototipagem embarcada de código aberto, baseada em hardware e software

BD – Banco de Dados

GEPTA – Grupo de Estudos e Pesquisas em Tecnologia Aplicada

IaaS – Infra-estrutura como Serviço

NIST – National Institute of Standards and Technology

ODM - Object Data Mapping

OMS - Organização Mundial da Saúde

OPENCV – Open Source Computer Vision Library

OPENSOURCE – Projeto de código aberto

ORM - Object Relational Mapping

PaaS – Plataforma como Serviço

SaaS – Software como Serviço

SLA – Services Level Agreement

TI – Tecnologia da Informação

UUID – Identificadores Únicos Universais

W3C – World Wide Web Consortium

WEB – Rede de Alcance Mundial ou WWW

OMS – Organização Mundial da Saúde

RESUMO

Este trabalho apresenta uma solução computacional que visa colaborar com médicos e profissionais da saúde no atendimento a pacientes localizados remotamente.

O sistema proposto denominado de “Telemedicina de baixo custo”, tem exatamente o objetivo de fornecer aos profissionais da saúde uma ferramenta de atendimento remoto aos pacientes através de dispositivos computacionais de uso comum conectados à internet (Telemedicina), através de uma plataforma computacional de código aberto e sobre a qual não incida nenhum tipo de pagamento de licenciamento.

A definição pelo desenvolvimento desta ferramenta, teve como base a situação de carência de profissionais médicos que acomete à sociedade brasileira.

Para desenvolver esta plataforma foram utilizados valiosos conhecimentos e conceitos apreendidos durante o curso, também foram pesquisadas e implementadas novas tecnologias que se mostraram necessárias para atingir os propósitos de funcionamento em nuvem, performance e escalabilidade.

Embora existam no mercado outras ferramentas que proporcionam as mesmas funcionalidades, as mais famosas são as dos hospitais Albert Einstein e Sírio Libanês em São Paulo, por diversos motivos estas são de difícil acesso.

Palavras Chave: Telemedicina, Telessaúde, e-saúde, Robótica, Visão Computacional, Atendimento Remoto

ABSTRACT

This paper presents a computational solution that aims to collaborate with doctors and health professionals in remotely located patient care.

The system, called "low-cost telemedicine", has the purpose to provide a remote service tool for health professionals through general-purpose computer connected to the internet (Telemedicine). For this, it is computing open source platform on which does not fall any licensing payment would be used.

The development of this tool has been inspired on the shortage of medical professionals that affects Brazilian society.

In order to develop this platform, valuable knowledge and concepts learned during the course are been used and new technologies have been developed and implemented in order to achieve the cloud computing, performance and scalability requirements.

There are other tools on the market that provide the same functionality, the most famous being the ones used by Albert Einstein and Sírío Libanês hospitals, in São Paulo. However, for various reasons, those are not widely available to cover demand.

Keyword: Telemedicine, Telehealth, e-health, Robotics, computer vision, Remote service

1. INTRODUÇÃO

O sistema de saúde brasileiro sofre com a falta de médicos e profissionais da saúde, tendo inclusive que importar estes profissionais. No entanto o Brasil possui 400 mil médicos em atividade, correspondendo a 5 médicos para 1 mil habitantes, enquanto a OMS recomenda um mínimo de 3,5 médicos para esta mesma quantidade de habitantes (COISI, 2015).

Vemos então que o problema não é a falta de médicos e sim a distribuição, poucos aceitam se instalarem nas periferias ou cidades do interior, sendo diversos os motivos para este desequilíbrio, tais como distância, falta de segurança, falta de infraestrutura médica, salários baixos.

Ao tempo que faltam médicos, a telemedicina que foi uma promessa durante muitos anos, possui hoje todos os recursos técnicos necessários para permitir atendimentos médicos à distância e mitigar o problema da escassez de médicos.

1.1. APRESENTAÇÃO DO TEMA

O tema deste trabalho é o desenvolvimento de uma plataforma de software em nuvem que permita um completo atendimento médico ou acompanhamento de tratamento à distância. A solução tem como atores o atendente e o paciente.

1.2. ESTRUTURA DO TRABALHO

A organização deste documento está feita nos seguintes capítulos:

Capítulo 1: Definições dos objetivos gerais e específicos, problematização justificativa e motivação do projeto;

Capítulo 2: Revisão de literatura;

Capítulo 3: Visão geral do projeto;

Capítulo 4: Metodologia;

Capítulo 5: Visão geral do projeto.

1.3. OBJETIVOS

1.3.1. Objetivo geral

O objetivo de deste trabalho é desenvolver uma plataforma de software de conexão ponto a ponto, a qual permita que profissionais da saúde, tais como médicos, psicólogos, fisioterapeutas, assistentes sociais e outros, tenham a possibilidade de atender remotamente os pacientes em tratamento de saúde.

1.3.2. Objetivos específicos

Para alcançar os objetivos gerais descritos acima, tem-se os seguintes objetivos específicos:

- Implantar um ambiente de computação na nuvem:
 - Definir um servidor de baixo custo e suficiente poder computacional;
 - Escolher sistema operacional apropriado;
 - Instalar compiladores das linguagens que envolvem o projeto;
 - Implantar a biblioteca OpenCV encarregada das funções de Visão Computacional.

- Estruturar uma arquitetura de software que comporte o processamento em
 - Desenvolver um módulo que permita conexão remota ponto a ponto;
 - Desenvolver rotinas que permitam a comunicação entre o paciente e o atendente, com o registro de informações resultante;
 - Programar uma camada de apresentação para interação com o usuário e com o atendente;
 - Programar aplicativo Android que permita interagir com o paciente e também enviar comandos de direção, que em trabalhos futuros

serão implementados no aplicativo para enviar os mesmos, para o robô.

1.4. PROBLEMATIZAÇÃO

O Brasil vive um momento de escassez de profissionais no setor médico, tendo inclusive que importar estes profissionais de outros países para tentar suprir a demanda interna de atendimentos (MAIS MEDICOS, 2015).

Normalmente os atendimentos são feitos mediante o deslocamento do paciente até o profissional da saúde, porém dependendo da condição de saúde e mobilidade do paciente pode ser necessário que o médico vá até o paciente, tornando o atendimento ainda mais custoso do que costuma ser normalmente e afetando ainda mais a disponibilidade destes profissionais

1.5. JUSTIFICATIVA

Ao mesmo tempo que as aplicações tradicionais, que funcionam em ambiente desktop como o sistema operacional Windows, devem seguir paulatinamente o caminho da migração para a arquitetura em nuvem, em virtude do estabelecimento e prevalência da nova arquitetura. Este modelo cria também um ambiente ideal para o processamento de aplicações que exigem alta performance e rápida escalabilidade, características excepcionais para atender as demandas técnicas da Telemedicina.

1.6. MOTIVAÇÃO

- Dominar o funcionamento da arquitetura de processamento em nuvem;
- Criar uma arquitetura computacional em nuvem própria para o software de telemedicina e o processamento de Visão Computacional;
- Migrar as rotinas de controle do projeto acadêmico Pia Robot (PIA ROBOT, 2011) para o processamento em nuvem.

2. REVISÃO DE LITERATURA

2.1 Pia Robot

O projeto Pia Robot foi concebido no ano de 2013 orientado pelo Professor Dr Luiz Antônio Pereira Neves, com acadêmicos do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da UFPR que teve como intuito o desenvolvimento de um robô para auxiliar nos cuidados de criança e idosos em suas residências, a longa distância e também de um simples sistema Web para ajudar no gerenciamento de usuários e controle do robô.

Este projeto foi construindo com Arduino, plataforma eletrônica *Open source* de prototipação baseadas em hardware e softwares flexíveis, utilizando a linguagem Java para aplicação na Web no projeto e a linguagem C, na programação embarcada no Arduino.

2.2 Telemedicina e Telessaúde

De acordo com a OMS, telemedicina compreende “a oferta de serviços usando tecnologias de informação e de comunicação para o a oferta de serviços ligados aos cuidados com a saúde, nos casos em que a distância é um fator crítico; tais serviços são prestados por profissionais da área da saúde intercâmbio de informações válidas para diagnósticos, prevenção e tratamento de doenças e a contínua educação de prestadores de serviços em saúde, assim como para fins de pesquisas e avaliações”. Além da Medicina, a Telessaúde envolve o apoio a profissionais de diversas áreas da saúde, com equipe qualificada de teleconsultoria, que disponibiliza a melhor evidência científica para tomada de decisão (MONTEIRO, 2015).

As experiências exitosas e que têm tido sustentabilidade financeira no Brasil, frequentemente, estão associadas à parceria de sua realização com Programas de Residência em Medicina de Família e Comunidade (Programa de Pós-Graduação Médica), que oferece suporte com formação e experiência em Atenção Primária à Saúde, além de muitos especialistas focais (MONTEIRO, 2015).

Na área da regulação, a Telessaúde ajuda a diminuir o número de encaminhamentos da atenção primária em saúde para outras especialidades. Por exemplo, no Rio Grande do Sul, o Telessaúde solucionou 69% das dúvidas dos médicos de família, portanto reduzindo os custos associados a esses encaminhamentos desnecessários (MONTEIRO, 2015).

2.3 Telemedicina no Brasil

Segundo o Professor Emérito György Miklós Böhm (USP, 2013), da Faculdade de Medicina da USP, em 1985, surge a Disciplina de Informática Médica da Faculdade de Medicina da USP. De 1994 a 2000, as ações foram isoladas, sem constituição. Não havia empenho do governo. Não havia a percepção da revolução digital. Neste período (2000) o foco era centrado em redes e salas de teleconferência. Iniciou-se a teleducação na área médica e de saúde pública, as teleconferências médicas com o exterior, em geral para pacientes da rede privada, a tele eletrocardiografia, por fax e depois pela Internet em setores públicos e privados. Minas Gerais, Pernambuco, Santa Catarina e São Paulo foram os primeiros estados brasileiros a se engajarem com telemedicina.

De 2003 a 2013, surgiram redes e salas de teleconferência em todas as partes do país. A teleducação existia na maioria dos estados, não como disciplina. Eram frequentes as teleconferências médicas dentro e fora do país. Cresceu a teleassistência em todo o Brasil, porém não atendeu as necessidades do país (USP, 2013).

Atualmente, conforme Chao Lung Wen (WEN, 2013), professor e coordenador geral da Faculdade de Medicina de São Paulo, existem muitas definições para a Telemedicina, podendo variar segundo suas aplicações e características, e com o surgimento e incorporação de novas tecnologias. Definir as áreas de performance da Telemedicina é tão complexo quanto determinar todas as áreas que a informática pode ser aplicada. Entretanto, constituiu-se algumas características básicas da Telemedicina, entre elas, podemos citar:

- Distância física entre comunidades: as que precisam e a que provêm o serviço médico;

- Utilização da tecnologia para conseguir a assistência, em substituição a presença física;
- Disponibilidade de uma junta médica e de profissionais de saúde para proporcionar o serviço;
- Disponibilidade de profissionais das áreas de tecnologia responsáveis pelo desenvolvimento e manutenção de infraestrutura de Telemedicina;
- Estruturação de segurança, qualidade e sigilo dos dados e serviços oferecidos através da Telemedicina.

Tendo por base estas características, pode-se dizer que a Telemedicina não é uma atividade unicamente médica, mas sim o resultado da união de profissionais de saúde e de tecnologia, aperfeiçoando uma importante sinergia para o desenvolvimento de atividades que visam a promover a saúde.

Nesta última década, a Telemedicina deu um grande salto, devido as melhorias das tecnologias de eletrônica, informática e telecomunicações. Com isso obteve-se um progresso que colaborou para que a Telemedicina alcançasse maior qualidade funcional e com concomitante redução de custos (WEN, 2013).

2.4 Evolução da Telemedicina

Nos últimos anos, com o crescimento da globalização e a consumerização da tecnologia da informação exigem um avanço de produtividade no mundo dos negócios, incluindo as áreas de saúde e bem-estar à toda comunidade. Com isso, enquanto as organizações são desafiadas com a demanda por tecnologia como um serviço expressivo dentro da empresa, no setor de saúde não é diferente. Em nossa atual realidade o acesso a qualquer aplicativo, a qualquer hora, momento e lugar, a partir do dispositivo escolhido, também traz avanços para o setor da saúde. Isso denota que as pessoas exigirão o uso dessas tecnologias avançadas em clínicas, hospitais, enfim, em ambientes de saúde e até em suas casas.

Hoje em dia já temos diversos casos de sucesso no Brasil. Um deles é como a tecnologia está aprimorando a qualidade de vida das pessoas com a precaução e controle de doenças.

A tecnologia efetivamente reduz a distância e acelera o acesso à saúde de pacientes em regiões mais afastadas dos centros urbanos.

Na realidade, a principal finalidade de um projeto de Telemedicina e participação clínica avançada é aperfeiçoar a condição da assistência médica pelo acesso ao atendimento remoto especializado.

Pode-se citar como exemplo precário, o sistema de cobertura universal em saúde, o SUS – Sistema Único de Saúde, onde inclui aproximadamente 6.500 hospitais, sendo a maior parte na zona urbana, com uma baixa ou inexistente infraestrutura para acolher um setor de tecnologia. Há clínicas de saúde que nem sempre estão integradas à rede de hospitais públicos. Com esse cenário, vale ressaltar que, muitas pessoas com problemas de saúde precisam de atendimento especializado, porém moram distantes dos grandes centros urbanos.

Dispositivos móveis de vídeo com qualidade de som e imagem em alta definição para telemedicina foram desenvolvidos para uso de médicos, contendo consulta remota de pacientes, o cuidado virtual e a educação médica. Este tipo de recurso possibilita o compartilhamento de conteúdo, gravação, firewall e capacidade de gestão. E em alguns casos de consultas que necessitam também os resultados de exames dos pacientes, são feitas pelo equipamento de telediagnósticos.

Exemplo de projeto piloto do programa de Telemedicina do Hospital Israelita Albert Einstein, que usa tecnologia de vídeo para telemedicina da Cisco. Essa tecnologia de ponta viabiliza a comunicação constante entre as equipes da UTI e Pronto-Socorro do Hospital Municipal Dr. Moysés Deutsch (SP) e da Central de Telemedicina do Hospital Israelita Albert Einstein (SP). A finalidade é proporcionar assistência à pacientes graves, no caso de ausência de especialistas na unidade pública ou da necessidade de uma segunda opinião para avaliação de um diagnóstico mais preciso e completo (BARCELLOS, 2014).

3. VISÃO GERAL DO PROJETO

A solução de atendimento médico remoto que é objeto deste trabalho, root denominada “Telemedicina de baixo custo”, teve como escolha principal a utilização da arquitetura denominada de computação em nuvem para o desenvolvimento da plataforma que compõe a solução.

Escolhemos o provedor Digital Ocean por atender os requisitos de custo, performance e escalabilidade, necessários para viabilizar o projeto.

No servidor, foi instalado um sistema operacional Linux Ubuntu 14.04.1 LTS e também o OpenCV, biblioteca *Open Source* desenvolvida em C++ usada no processamento de imagens, que permite o processamento de rotinas de Visão Computacional neste servidor.

Como o objetivo original do projeto era transferir para o processamento em nuvem do projeto Pia Robot, que já funcionava em desktop, optamos por utilizar um conjunto de linguagens e bibliotecas que atendessem todas as necessidades que o novo projeto exigia. As rotinas do Pia Robot estavam escritas em C++ e precisávamos um servidor web que tivesse alta performance, juntando as partes chegamos à conclusão que o servidor Web Node.js seria a melhor opção em termos de performance e que sua biblioteca de funções, denominada V8, é ideal para executar as rotinas C++ legadas do projeto Pia Robot.

3.1 ESPECIFICAÇÕES INICIAIS

Nesta seção são apresentados o escopo do projeto e também a definição de cada módulo da plataforma de Telemedicina.

3.2 DEFINIÇÃO DO ESCOPO DO PROJETO

Faz parte do projeto o desenvolvimento de um software que permita ao médico o atendimento remoto dos seus pacientes, que forneça recursos para

registrar todo o atendimento e permita agregar anotações ou pareceres pertinentes ao atendimento.

3.2.1 Descrição da arquitetura em Nuvem

A computação em nuvem é um modelo de computação que permite ao usuário final conectado à internet utilizar um poder computacional sem precedentes a partir de qualquer lugar, bastando para isso apenas ter um terminal conectado à “nuvem” e utilizar a interface padrão de navegação do seu browser favorito.

O laboratório de tecnologia da informação do *National Institute of Standards and Technology* (NIST) define computação em nuvem como um modelo do tipo “pague pelo uso” para possibilitar acesso de rede disponível, conveniente e sobre demanda a um pool compartilhado de recursos computacionais configuráveis (por exemplo, servidores, armazenamento, redes, aplicações, serviços) que podem ser rapidamente adquiridos e liberados com o mínimo esforço gerencial ou de interação de provedor de serviços (NIST, 2012).

A Computação em Nuvem distribui os recursos na forma de serviços, sendo que de acordo com a modalidade de serviço oferecido podem ser caracterizados como:

- **Infrastructure as a Service (IaaS):** É a camada dos recursos físicos, tais como servidores, armazenamento de dados (*storage*), equipamentos de rede e comunicação e outros afins;
- **Platform as a Service (PaaS):** Compreende a plataforma das aplicações, usando serviços de desenvolvimento e de execução, além de serviços de banco de dados;
- **Software as a Service (SaaS):** Trata-se da disponibilização dos serviços de software para o usuário final, ou seja, é uma forma de distribuir e comercializar o software.

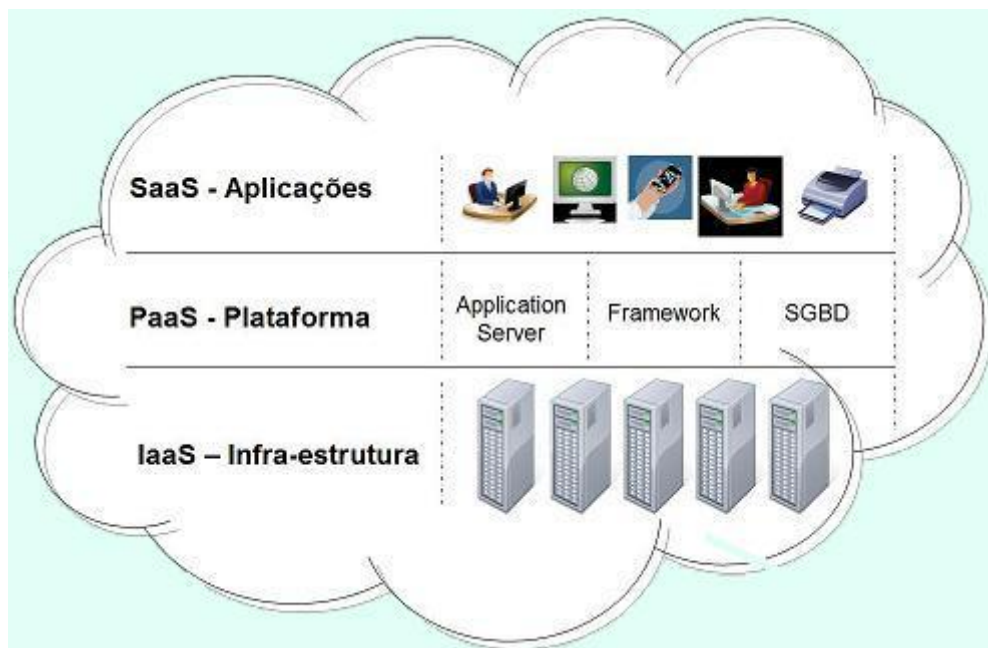


Figura 1 – Arquitetura de serviços em nuvem

3.2.1.1 Tipo de nuvem

- a) **Nuvem Pública** – A nuvem pública é o modelo padrão de computação em nuvem, no qual os serviços são oferecidos de forma aberta a todos os usuários, seja de forma gratuita ou pagando pelo uso conforme contratado.
- b) **Nuvem Privada;**
A infraestrutura de nuvem privada é voltada exclusivamente para uma única organização, podendo estar hospedada internamente ou externamente, mas em geral é construída sobre um data-center privado e fica protegida atrás de um firewall com acesso restrito aos funcionários da organização e parceiros de negócios.
- c) **Nuvem Comunitária;**
É uma infraestrutura de nuvem compartilhada por várias organizações que partilham interesses e responsabilidade comum. Pode ser administrada pelas próprias organizações ou por um terceiro e também pode estar hospedada interna ou externamente.
- d) **Nuvem Híbrida.**
Neste caso temos uma composição de duas ou mais nuvens, sejam elas privadas, públicas ou comunitárias.

3.2.1.2 Provedor de nuvem

O provedor Digital Ocean foi escolhido por tratar-se de um provedor de hospedagem em nuvem simples, que atende desenvolvedores em qualquer escala, uma vez que até para os planos mais básicos (US\$5 por mês) disponibiliza discos SSD, que proporcionam rápido acesso para utilização de aplicativos e sites. Desta forma, este ambiente oferece as melhores características para o processamento de Visão Computacional de forma confiável.

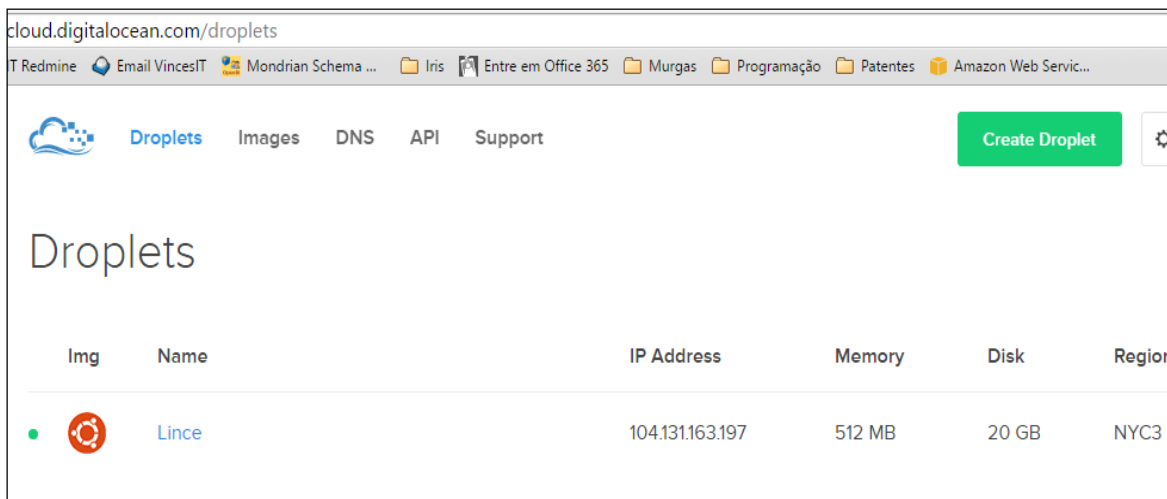


Figura 2 – Console Digital Ocean

3.2.1.3 Sistema Operacional

Este projeto utiliza o Linux Ubuntu versão 14.04 por tratar-se de um sistema operacional *Open Source*, de alta confiabilidade e estabilidade, portanto totalmente apropriado para este projeto.

3.2.1.4 Node.js

Node.js é uma plataforma de desenvolvimento Web, construída sobre o motor V8 JavaScript para facilmente construir aplicações de rede rápidas e escaláveis, utiliza um modelo de I/O direcionado a evento não bloqueante que o

torna leve e eficiente, ideal para aplicações em tempo real com troca intensa de dados através de dispositivos distribuídos (NODE, 2015).

Com Node.js podemos criar uma variedade de aplicações *Web* utilizando apenas código em JavaScript do lado do servidor.

Portanto de forma diferente de outras plataformas como .NET, Java, Ruby ou Python em que os processos são enfileirados e executados e paralisam o processamento enquanto utilizam um I/O no servidor (*blocking-thread*), no Node.js não existe prioridade nem bloqueios, todas as *threads* são executadas na medida que vão chegando (*single thread*), desta forma todo o poder de processamento do servidor é disponibilizado o tempo todo, permitindo dezenas de milhares de conexões simultâneas (IMASTERS, 2015).

3.2.1.5 Motor V8 JavaScript

V8 é o interpretador de *JavaScript Open Source* implementado pelo Google em C++ para a utilização no navegador Google Chrome e posteriormente adotado pela comunidade de desenvolvedores mundial.

Ele permite executar código *JavaScript* no lado do servidor e confere ótimo desempenho ao Node.js que utiliza esta linguagem como base.

Pelo fato do V8 ter sido escrito em C++, ele consegue acelerar o desempenho de uma aplicação compilando o código *JavaScript* para o formato nativo de máquina antes de executá-lo, permitindo que rode a velocidade de um código binário compilado (V8, 2015).

Esta tecnologia foi escolhida para fazer parte da plataforma pelo fato do V8 ser escrito em C++, uma vez que existem rotinas no projeto Pia Robot para controlar os robôs, bem como rotinas de processamento de imagem; ambas escritas na linguagem C++. Portanto, o V8 apresenta portabilidade para executar programações em C++.

3.2.1.6 Mongoos

Mongoose é uma biblioteca do Node.js que permite o acesso a dados no banco de dados Mongo

O Mongoose funciona com base em esquemas para modelar os dados da aplicação, fornece um mapeamento de objetos do MongoDB similar ao ORM ou

ODM, isso significa que o Mongoose traduz os dados do banco de dados para objetos JavaScript para que possam ser utilizados por sua aplicação. Ele possui sistema de conversão de tipos, validação, criação de consultas e *hooks* para lógica de negócios. (MONGOOSE, 2015).

3.2.1.7 MongoDB

O MongoDB é um banco de dados OPENSOURCE orientado a documentos, ele faz parte da nova geração de BD chamados de NoSQL, denominados desta forma para serem diferenciados dos bancos de dados relacionais que utilizam a sintaxe SQL e trabalham de forma bidimensional, armazenando registros em linhas e colunas.

No MongoDB ao invés de registros trabalhar com documentos que são autocontidos e auto descritivos utilizando a sintaxe JSON.

Bancos de Dados Orientados a Documentos tem como característica conter todas as informações importantes em um único documento, ser livre de esquemas, possuir identificadores únicos universais - UUID, possibilitar a consulta de documentos através de métodos avançados de agrupamento e filtragem, técnica conhecida como MapReduce, permitindo também melhorar a redundância e eliminar inconsistências (MONGODB, 2015).

3.2.1.8 WebRTC

WebRTC é uma API elaborada pelo W3C sob a modalidade OPENSOURCE, que permite a comunicação de áudio e vídeo de forma extremamente simplificada, sem a necessidade de instalar plug-ins ou bibliotecas adicionais. Desta forma permite aos desenvolvedores criar sistemas de comunicação em tempo real – RTC, utiliza tecnologias web padrão tais como HTML e JavaScript, funciona perfeitamente em todos os navegadores de última geração (WEBRTC, 2015).

Na implementação feita neste projeto ambos clientes (paciente e atendente) registram seu IP no servidor, para que a partir da identificação do IP qualquer uma das partes possa iniciar uma chamada, estabelecendo uma

conexão direta com o outro (par a par) e transferir dados de vídeo e voz (streaming) sem passar pelo servidor.

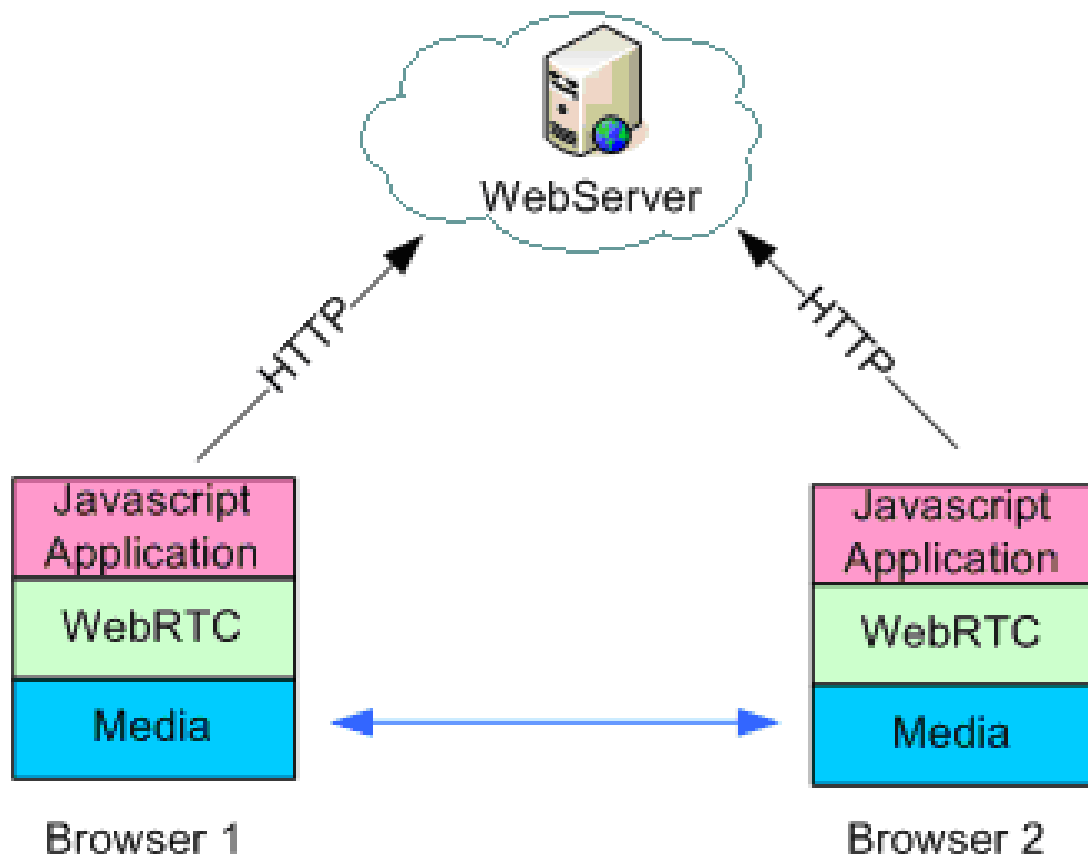


Figura 3 – Arquitetura típica de aplicação que utiliza WebRTC

4. METODOLOGIA

Neste capítulo é apresentada as metodologias usadas no desenvolvimento do sistema. Serão representados o modelo incremental, ambiente de hardware, a linguagem de programação utilizada, a modelagem de dados, diagramas de caso de uso, de classes e de sequência e como também o plano de atividades.

4.1. MODELO INCREMENTAL

O modelo incremental foi a melhor abordagem escolhida pelo fato de nossos requisitos mudarem bastante, conforme o desenrolar do projeto.

4.2. AMBIENTE DE HARDWARE

O hardware utilizado para o desenvolvimento do projeto foram dois notebooks portáteis e um tablet Samsung Galaxy Tab 3 .

- Notebook Acer:
 - Processador: Intel(R) Core(TM) i5-3230M CPU @2.60GHz
 - Memória (RAM): 8,00 GB
 - Disco Rígido: 500 GB
 - Sistema Operacional: Windows 10 Pro

- Notebook Samsung ATIV Book 2
 - Processador: Intel(R) Core(TM) i5- 4210U CPU @2.70GHz
 - Memória (RAM): 8,00 GB
 - Disco Rígido: 1 T
 - Sistema Operacional: Windows 10 Pro

- Tablet: Samsung Galaxy Tab 3
- Processador: 2x ARM Cortex-A9 @ 1,20 GHz
- Memória (RAM): 832 MG
- Sistema Operacional: Android

- Nuvem
 - Processador: Intel(R) Xeon(R) CPU E5-2630L v2 @ 2.40GHz
 - Memória: 512MB
 - Disco: 20 GB.

4.3 SOFTWARES UTILIZADOS

- Visual Studio 2013: Software usa para programação do sistema
- Putty: Software utilizado para acesso remoto para nuvem;
- Winscp: Software utilizado para transferência de arquivos;
- SmartGit: Software utilizado para o versionamento do código;
- Google Chrome: navegador para testes de interface.
- Astah Community: Usado na criação de diagramas da documentação.
- WBS Schedule Pro: Ferramenta usada na geração da estrutura analítica do projeto
- Pacote Microsoft Office: Usado na geração de documento de textos.

4.4 WBS

O WBS (Work Breakdown Structure) é modelo de divisão das tarefas do projeto, no qual se busca identificar visualmente as atividades a serem executadas para se concluir o projeto. Apresentado na Figura 4.

Para o sucesso do projeto há necessidade que haja sinergia entre os integrantes, do qual só pode ser alcançado com a efetiva e boa comunicação entres os mesmo, de forma a providenciar uma ligação harmoniosa para que cada um consiga saber o que a outra pessoa está fazendo e de qual será sua parte na contribuição no desenvolvimento, deste modo busca não perder tempo em discussões desnecessárias. Segue abaixo o plano de comunicação.

TABELA 1 – PLANO DE COMUNICAÇÃO

Grupo Interessado	Método/Meio	Objetivo	Frequência
Equipe do Projeto	Reunião entre a equipe e orientador	Andamento do Projeto	Quinzenal
Equipe do Projeto	Reuniões e telefonemas	Acompanhamento das tarefas	Quando necessário
Equipe do Projeto	Reunião online ou presencial	Desenvolvimento do software	Quando necessário

4.7 PLANO DE RISCOS

Tem por finalidade identificar possíveis problemas que podem ocorrer durante o desenvolvimento do projeto, de forma a identificar ações preventivas e corretivas para que os riscos não afetem o produto final desejado.

O risco é classificado entre uma nota de 1 a 9, através dos indicadores probabilidade e impacto, como mostra a Tabela 2.

TABELA 2 – PLANO DE RISCOS

Nº	Condição	Consequência	Ação	Probabilidade	Impacto	Classificação
1	Falta de conhecimento nas	Atraso na entrega do produto final.	Estudar as tecnologias utilizadas no	Alto	Alto	7

	tecnologias usadas		desenvolvimento do projeto			
2	Mudanças no escopo do projeto	Atraso nas atividades do cronograma	Ações que utilizem menos recursos de tempo	Muito Alto	Moderado	7
3	Falha de comunicação	Atraso nas atividades do cronograma	Comprometimento com reuniões entre integrantes e orientador	Alto	Moderado	6
4	Defeito nos hardwares e softwares utilizados	Atraso nas atividades do cronograma	Procurar pro hardware alternativos	Muito Alto	Muito Baixo	5

4.8 DIAGRAMA DE CASOS DE USO

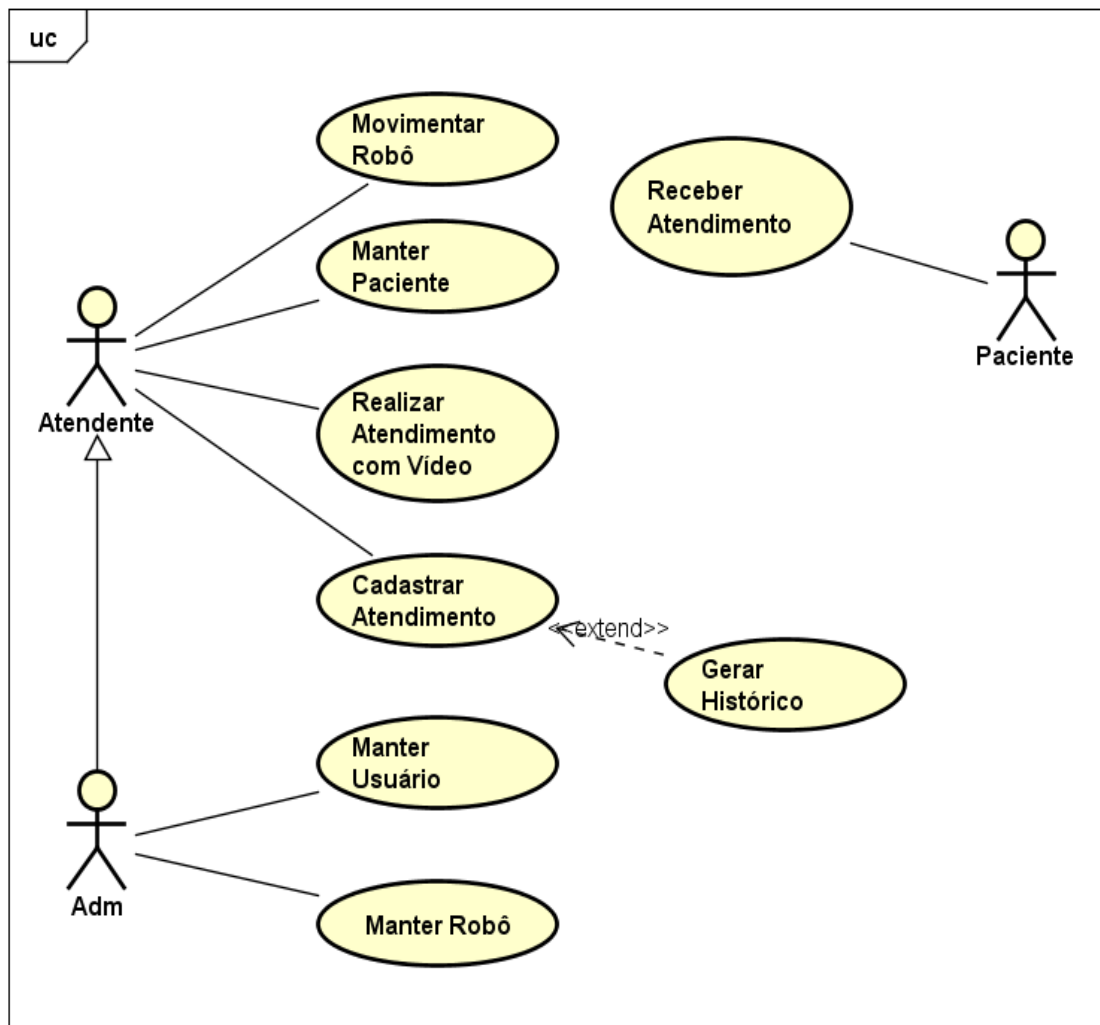
O diagrama de caso de uso (figura 5), apresenta de forma resumida os módulos integrantes que a solução implementa, além da interação que ocorre dos usuários com o sistema.

São três usuários ao todo, que serão explicados a seguir:

O atendente tem à disposição as funções de movimentar Robô (comandos de direção que são enviados para o servidor e de lá para o Tablet e que podem ser reenviados via Bluetooth para um dispositivo pareado), manter paciente (cadastro, deleção e edição do paciente), realizar atendimento com vídeo e cadastrar atendimento, que pode gerar um histórico de acesso.

O administrador herda todas funções do atendente e contém acesso privilegiado dos módulos manter usuário (cadastro, deleção e edição do atendente), e manter Robô (cadastro, deleção e edição do robô).

O paciente tem módulo Receber Atendimento a sua disposição.



powered by Astah

Figura 6 – Diagrama de casos de uso

4.9 DESCRIÇÃO DOS CASOS DE USO

TABELA 3 – CASO DE USO REALIZAR ATENDIMENTO COM VÍDEO

Caso de Uso: Realizar atendimento com Vídeo	
Descrição:	Este caso de uso descreve como o atendente pode realizar um atendimento com vídeo.
Pré-condições:	Atendente precisa estar logado no sistema. Paciente precisa estar com o aplicativo Android rodando.

Pós-condições:	<i>Stream</i> de vídeo entre o paciente e o atendente.
Ator Envolvido:	Atendente
Fluxo de eventos Principais:	
1	O atendente escolhe determinado paciente e clica em acessar;
2	Clica no botão ligar;
3	O sistema recebe e envia uma <i>stream</i> de vídeo para o tablet do paciente;
Fluxo de Exceção:	
E1	Problema com API de <i>stream</i> de vídeo.
Caso de Uso:	Movimentar Robô
Descrição:	Este caso uso descreve como o atendente pode enviar comandos de movimentação.
Pré-condições:	Atendente precisa estar logado no sistema. Pacienta precisa estar com o aplicativo Android rodando.
Pós-condições:	Um comando de movimentação é recebido pelo aplicativo Android.
Ator Envolvido:	Atendente.

TABELA 4 – CASO DE USO: MOVIMENTAR ROBÔ

Caso de Uso: Movimentar Robô	
Descrição:	Este caso uso descreve como o atendente pode enviar comandos de movimentação.
Pré-condições:	Atendente precisa estar logado no sistema. Pacienta precisa estar com o aplicativo Android rodando.
Pós-condições:	Um comando de movimentação é recebido pelo aplicativo Android.
Ator Envolvido:	Atendente.
Fluxo de eventos Principais:	
1	O atendente escolhe determinado paciente e clica em acessar;
2	Clica na setas de direção e envia as direções para o sistema;

3	O sistema recebe essas direções;
4	Re envia-os para o tablet do paciente;
Fluxo de Exceção:	
E1	Problema com sockets no envio da web para o sistema ou do sistema para Android.
Caso de Uso:	Receber Atendimento
Descrição:	Este caso uso descreve como o paciente recebe o atendimento.
Pré-condições:	Atendente precisa estar logado no sistema. Pacienta precisa estar com o aplicativo Android rodando.
Pós-condições:	<i>Stream</i> de vídeo entre paciente e atendente.
Ator Envolvido:	Atendente.

TABELA 5 – CASO DE USO CADASTRAR ATENDIMENTO

Caso de Uso: Cadastrar Atendimento	
Descrição:	Este caso uso descreve como o atendente pode cadastrar um atendimento.
Pré-condições:	Atendente precisa estar logado no sistema e ter no mínimo um paciente cadastrado.
Pós-condições:	Um cadastrado é realizado.
Ator Envolvido:	Atendente.
Fluxo de eventos Principais:	
1	O atendente escolhe determinado paciente e clica em acessar;
2	Seleciona o tipo de atendimento;
3	Registra o atendimento, no campo específico.
4	Clica em salvar;
5	O sistema salvo o atendimento;
6	Retorna para lista de pacientes.

TABELA 6 – CASO DE USO CADASTRAR PACIENTE

Caso de Uso: Cadastrar Paciente	
Descrição:	Este caso uso descreve como o atendente cadastra um paciente no sistema.
Pré-condições:	Atendente precisa estar logado no sistema.
Pós-condições:	Insere um novo paciente no sistema.
Ator Envolvido:	Atendente.
Fluxo de eventos Principais:	
1	O atendente preenche os campos com os dados do novo paciente;
2	Clica em salvar;
3	O sistema valida os dados;
4	O sistema retorna para lista de pacientes com uma mensagem de confirmação do cadastro.
Fluxo de Exceção:	
E1	Campos de preenchimento obrigatório não preenchidos.

TABELA 7 – CASO DE USO CADASTRAR USUÁRIO

Caso de Uso: Cadastrar Usuário	
Descrição:	Este caso uso descreve como o administrador cadastra um atendente no sistema.
Pré-condições:	Administrador precisa estar logado no sistema.
Pós-condições:	Insere um novo atendente ou administrador no sistema.
Ator Envolvido:	Administrador.
Fluxo de eventos Principais:	
1	O administrador preenche os campos com os dados do novo atendente;
2	O sistema valida o e-mail preenchido;

3	Clica em salvar;
4	O sistema válida os dados;
5	O sistema retorna para lista de atendente com uma mensagem de confirmação do cadastro.
Fluxo de Exceção:	
E1	Campos de preenchimento obrigatório não preenchidos.
E2	O mesmo e-mail já estar sendo usado por outro atendente.

TABELA 8 – CASO DE USO CADASTRAR ROBÔ

Caso de Uso: Cadastrar Robô	
Descrição:	Este caso uso descreve como o administrador cadastra um robô no sistema.
Pré-condições:	Administrador precisa estar logado no sistema.
Pós-condições:	Insere um novo robô no sistema.
Ator Envolvido:	Administrador.
Fluxo de eventos Principais:	
1	O sistema válida o preenchimento do campo nome;
2	O administrador preenche campos com os dados do novo robô;
3	Clica em salvar;
4	O sistema válida os dados;
5	O sistema retorna para lista de robôs com uma mensagem de confirmação do cadastro.
Fluxo de Exceção:	
E1	Campos de preenchimento obrigatório não preenchidos.
E2	O nome do robô já estar sendo usado por outro robô.

TABELA 9 – CASO DE USO EDITAR PACIENTE

Caso de Uso: Editar Paciente	
Descrição:	Este caso de uso descreve como o Atendente edita um paciente.
Pré-condições:	Atendente estar logado e ter no mínimo um paciente já cadastro.
Pós-condições:	Os dados de um paciente são atualizados.
Ator Envolvido:	Atendente;
Fluxo de eventos Principais:	
1	Fluxo de eventos Principais:
2	O atendente escolhe um paciente e clica em editar;
3	O atendente preenche os outros campos;
4	Clica em salvar;
5	O sistema válida os dados;
6	O sistema retorna para lista de pacientes com uma mensagem de confirmação da edição.

TABELA 10 – CASO DE USO EDITAR ROBÔ

Caso de Uso: Editar Robô	
Descrição:	Este caso de uso descreve como o administrador edita um robô.
Pré-condições:	administrador precisa estar logado e ter no mínimo um robô já cadastro.
Pós-condições:	Os dados de um robô são atualizados.
Ator Envolvido:	Administrador;
Fluxo de eventos Principais:	
1	O administrador escolhe um robô e clica em editar;
2	É redirecionado para tela de edição do robô;
3	O sistema valida o preenchimento do campo nome do robô;
4	O administrador termina de preenche os outros campos;
5	Clica em salvar;
6	O sistema válida os dados;
7	O sistema retorna para lista de robôs com uma mensagem de confirmação da edição;
Fluxo de Exceção:	

E1	Campos de preenchimento obrigatório não preenchidos.
E2	O nome do robô já estar sendo usado por outro robô.
Caso de Uso: Excluir Paciente	
Descrição: Este caso de uso descreve como o atendente exclui um paciente.	

TABELA 11 – CASO DE USO EDITAR ATENDENTE

Caso de Uso: Editar Atendente	
Descrição: Este caso de uso descreve como o administrador edita um atendente.	
Pré-condições:	Administrador precisa estar logado e ter no mínimo um atendente já cadastro.
Pós-condições:	Os dados de um atendente são atualizados, o mesmo pode se tornar também administrador do sistema.
Ator Envolvido:	Administrador;
Fluxo de eventos Principais:	
1	O administrador escolhe um atendente e clica em editar;
2	É redirecionado para tela de edição do atendente;
3	O sistema valida o preenchimento do campo e-mail;
4	O administrador termina de preenche os outros campos;
5	Clica em salvar;
6	O sistema válida os dados;
7	O sistema retorna para lista de atendentes com uma mensagem de confirmação da edição;
Fluxo de Exceção:	
E1.	Campos de preenchimento obrigatório não preenchidos.
E2.	E-mail já estar sendo usado por outro atendente.
Pré-condições:	Atendente precisa estar logado e ter no mínimo um paciente já cadastro.

Pós-condições:	Um paciente é apagado do sistema.
Ator Envolvido:	Administrador;

TABELA 12 – CASO DE USO EXCLUIR PACIENTE

Caso de Uso: Excluir Paciente	
Descrição:	Este caso de uso descreve como o atendente exclui um paciente.
Pré-condições:	Atendente precisa estar logado e ter no mínimo um paciente já cadastro.
Pós-condições:	Um paciente é apagado do sistema.
Ator Envolvido:	Administrador;
Fluxo de eventos Principais:	
1	O atendente escolhe um paciente e clica em excluir;
2	O sistema deleta as referências deste paciente;
3	Retorna para lista de pacientes com uma mensagem de confirmação da exclusão;

TABELA 13 – CASO DE USO EXCLUIR ROBÔ

Caso de Uso: Excluir Robô

Descrição:	Este caso de uso descreve como o administrador exclui um robô.
Pré-condições:	Administrador precisa estar logado e ter no mínimo um robô cadastro.
Pós-condições:	Um robô é apagado do sistema.
Ator Envolvido:	Administrador;
Fluxo de eventos Principais:	
1	O administrador escolhe um robô e clica em excluir;
2	O sistema deleta as referências deste robô;
3	Retorna para lista de robôs com uma mensagem de confirmação da exclusão;

TABELA 14 – CASO DE USO EXCLUIR ATENDENTE

Caso de Uso: Excluir Atendente	
Descrição:	Este caso de uso descreve como o administrador exclui um atendente.
Pré-condições:	Administrador precisa estar logado e ter no mínimo um atendente cadastro.
Pós-condições:	Um atendente é apagado do sistema.
Ator Envolvido:	Administrador;
Fluxo de eventos Principais:	
1	O administrador escolhe um atendente e clica em excluir;
2	O sistema deleta as referências deste atendente;
3	Retorna para lista de atendentes com uma mensagem de confirmação da exclusão;

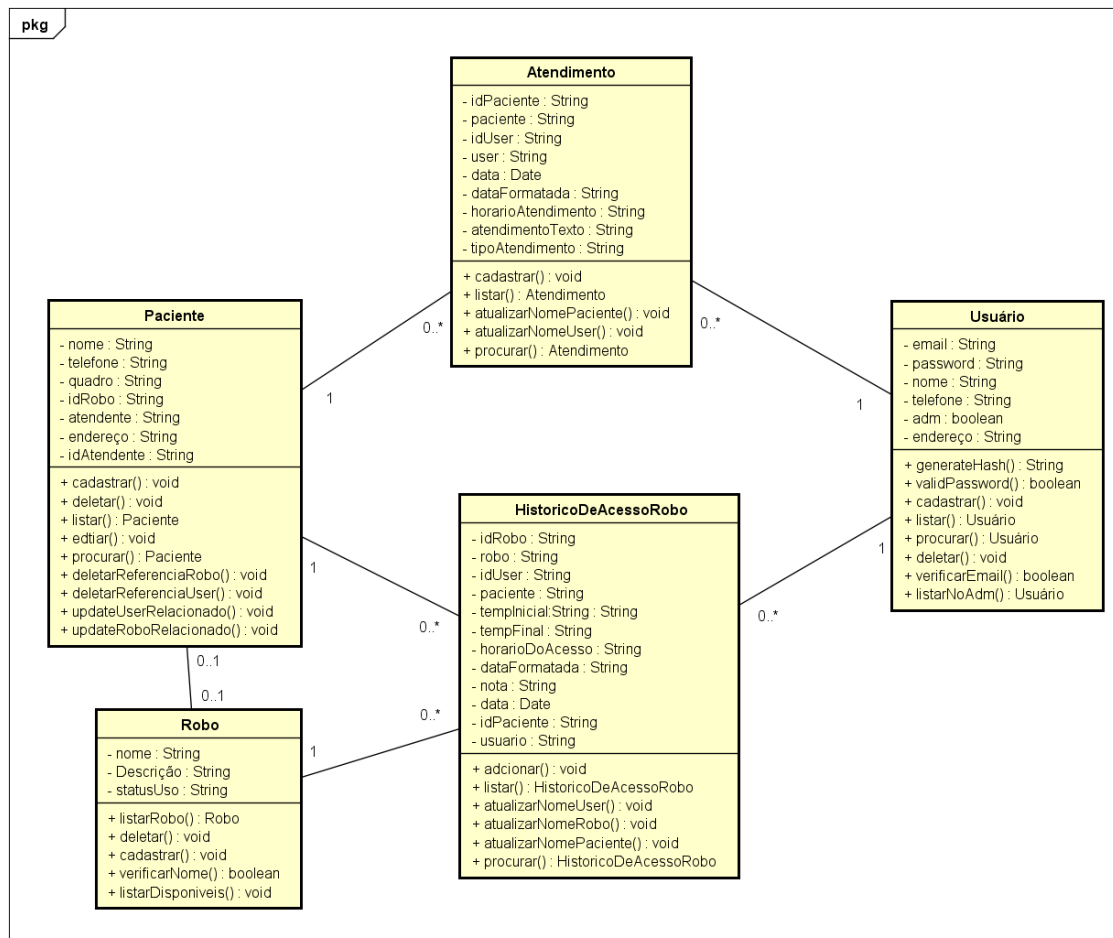
4.10 DIAGRAMA DE CLASSES

Nesta seção é apresentado o diagrama de classes (figura 7), que contém no total cinco classes.

A classe Paciente contém referências das classes Usuário e Robô, estas duas classes quando deletadas ou atualizadas de modo geral tem método que deletam ou atualizam suas referências na classe Paciente.

A classe Atendimento contém referências das classes Usuário e Paciente.

A classe HistoricoDeAcesso contém referências das classes Pacientes, Usuário e Robo.



powered by Astah

Figura 7 – Diagrama de classes

5 APRESENTAÇÃO DO SOFTWARE

Nesta seção será apresentado o software e suas telas, demonstrando suas características e descrevendo suas funcionalidades e a forma de como usar suas aplicações.

5.1 Tela de login

O atendente ou administrador preenche seu login e senha e clica no botão de login (logo abaixo do campo *password*), caso os dados estejam errados uma mensagem de erro é retornada, se não o usuário é redirecionado para sua tela inicial, como mostra a Figura 8.

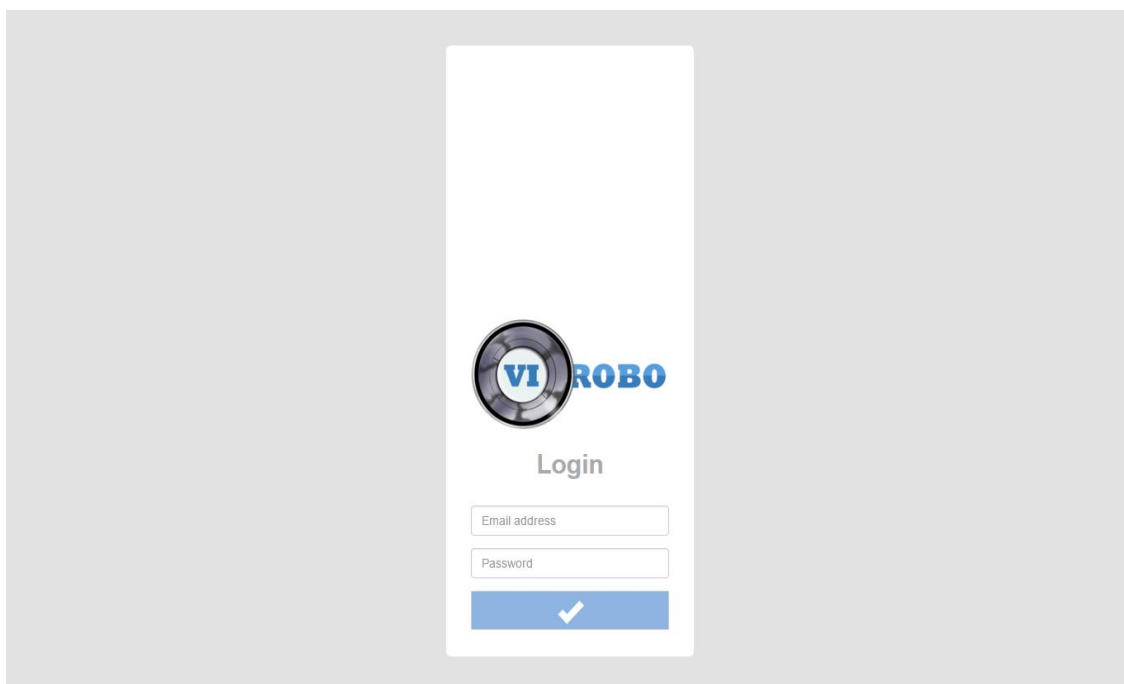


Figura 8 – Tela de Login

5.2 Tela Inicial administrador

Depois de logado o administrador é redirecionado para sua tela inicial, no qual é a apresenta todas as funções, em um menu de botões com desenho, que o sistemas contém (ver Figura 9).

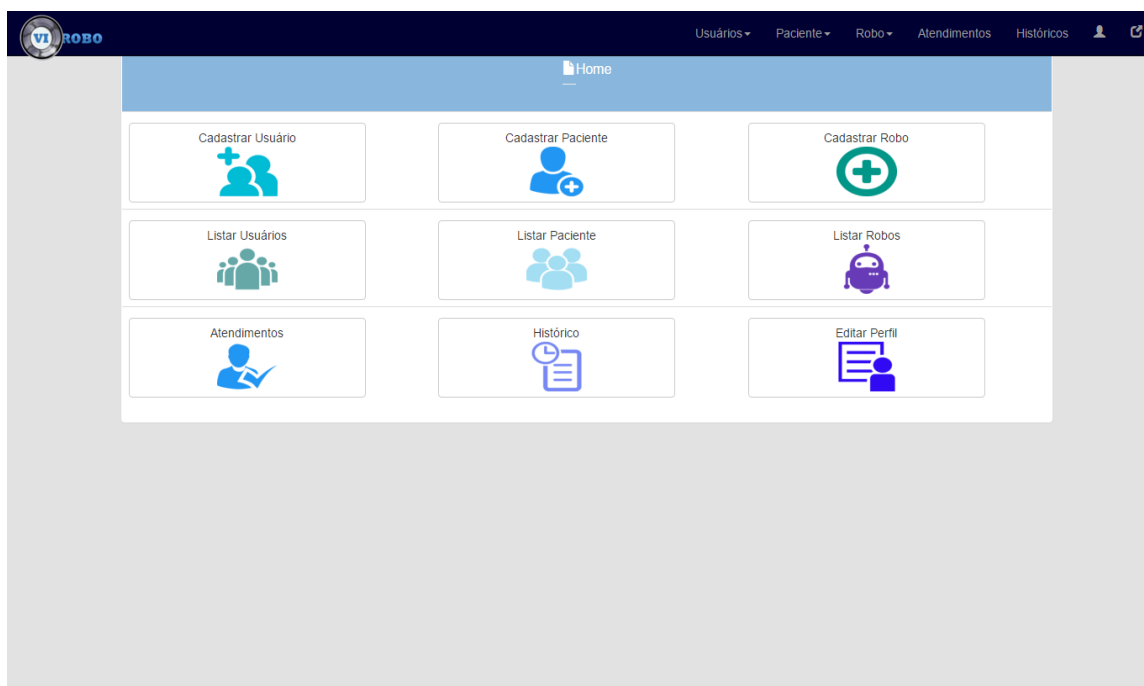
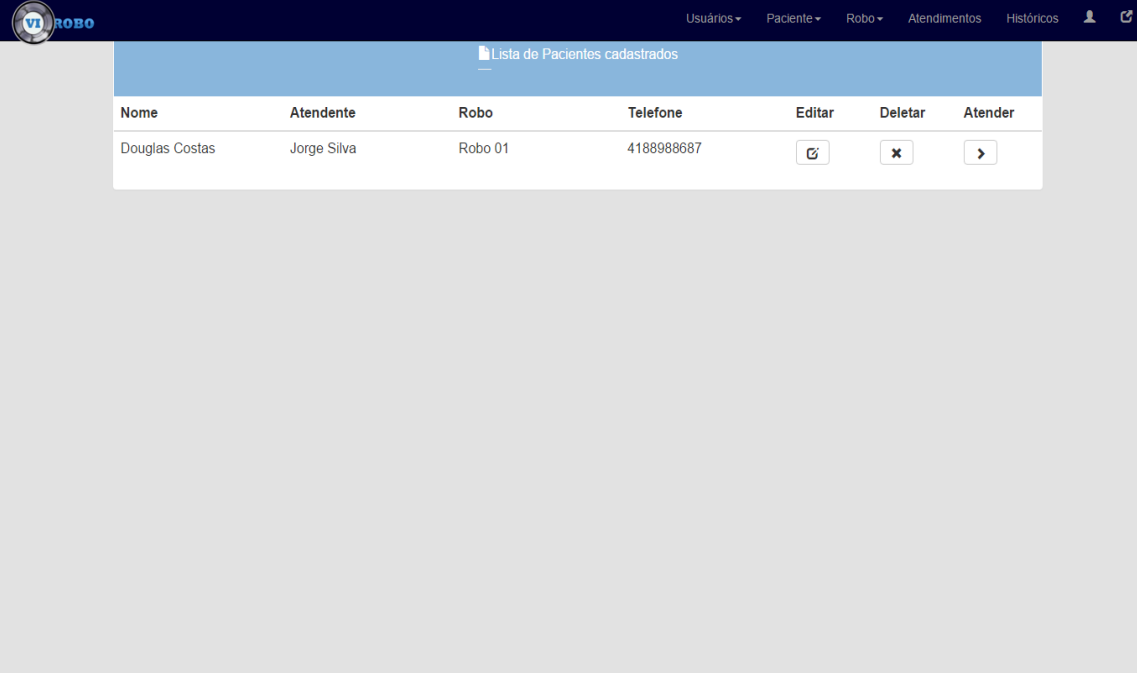


Figura 9 – Tela Inicial do Administrador

5.3 Tela de Lista de Pacientes cadastrados

Esta tela apresenta todos os pacientes cadastrados tanto pelos administradores, quanto dos atendentes, além de trazer informações como nome do robô e atendente relacionado. Cada paciente listado possui os botões editar, deletar e acessar

Se o administrador clicar no botão deletar, uma mensagem de confirmação é mandada para a tela, caso seja confirmado o paciente em questão é excluído do sistema e uma mensagem de confirmação da exclusão é mostrado no canto superior da tela (em cima da lista), como mostra a Figura 10.






Nome	Atendente	Robo	Telefone	Editar	Deletar	Atender
Douglas Costas	Jorge Silva	Robo 01	4188988687			

Figura 10 – Tela Lista de pacientes cadastrados

5.4 Tela de Cadastro do Pacientes

Na tela da Figura 11, o administrador pode cadastrar um paciente. Nela é onde o administrador preenche os dados e escolhe o atendente e o robô específico (ser houver para isso robô e paciente cadastrados) para o mesmo. Esta mesma tela também é usada para exibir e alterar (editar) os dados do paciente.

The screenshot shows a web interface for patient registration. At the top, there is a dark blue navigation bar with the 'VI ROBO' logo on the left and menu items: 'Usuários', 'Paciente', 'Robo', 'Atendimentos', 'Históricos', and a user profile icon. Below the navigation bar is a light blue header with the title 'Cadastro do Paciente'. The main form area is white and contains the following elements:

- Nome do Paciente:** A text input field with the placeholder 'Nome do Paciente'.
- Número de telefone:** A text input field with the placeholder 'Telefone'.
- Robo do Paciente:** A dropdown menu currently showing 'Robo 8'.
- Atendente do Paciente:** A dropdown menu currently showing 'manuela'.
- Quadro Clínico:** A large, empty text area for clinical notes.
- Salvar:** A blue button located at the bottom center of the form.

Figura 11 – Tela de cadastro do paciente

5.5 Tela Atender Paciente

Na Figura 12, o administrador tem a opção de poder cadastrar uma nota do estado do paciente, para isso ele pode ver o paciente através da função do botão ligar, que depois clicado inicializa uma vídeo conferência com o mesmo, porém para isso que ele esteja com seu aplicativo aberto. O botão “Tela inteira Remota” disponibiliza o vídeo do paciente em *full screen*, já o “Tela inteira Local” disponibiliza o vídeo em local em *full screen*.

As setas enviam comandos de direção para o servidor, que reenviam para o Android do paciente.

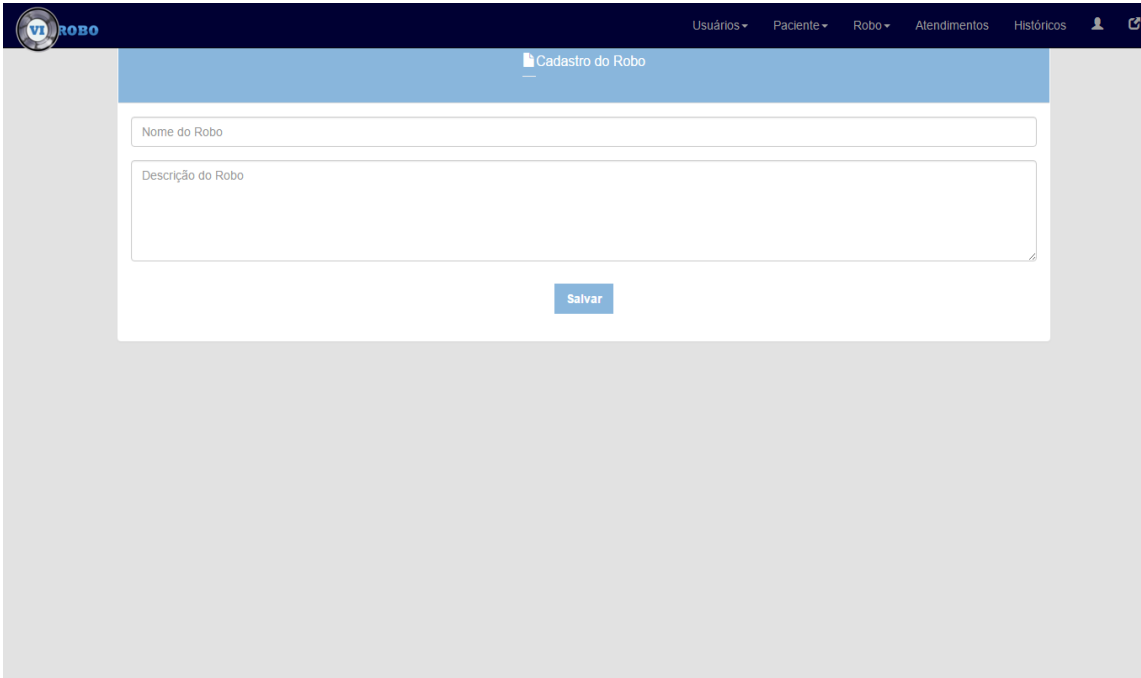
Toda vez que esta tela é acessada é gerado um log de quanto tempo o usuário ficou na tela, em que horário e data ele acessou, além de poder classificar uma nota de como estava a qualidade da vídeo conferência.

The screenshot shows a web application interface for patient care. At the top, there is a dark navigation bar with the 'ROBO' logo on the left and links for 'Paciente', 'Atendimentos', and 'Histórico' on the right. Below this is a white main area with a blue header bar containing the word 'Paciente'. The central part of the screen features three blue buttons: 'Tela Inteira Remota', 'Tela Inteira Local', and 'Ligar'. To the left of these buttons is a directional pad icon. Below the buttons is a dropdown menu with 'Excelente' selected. A blue bar with the title 'Cadastrar Atendimento' is positioned below the dropdown. Underneath this bar is another dropdown menu with 'Relatório' selected. A large text area contains the text 'Paciente esta bem'. At the bottom of the text area is a blue 'Salvar' button.

Figura 12 – Tela Atender paciente

5.6 Tela Cadastro Robô

A tela da Figura 13, o administrador pode cadastrar um robô. Nela é onde o administrador preenche os dados. Esta mesma tela também é usada para exibir e alterar (editar) os dados do robô.



The image shows a web application interface for robot registration. At the top, there is a dark blue navigation bar with the logo 'VI ROBO' on the left and several menu items: 'Usuários', 'Paciente', 'Robo', 'Atendimentos', and 'Históricos'. Below the navigation bar, the page title 'Cadastro do Robo' is displayed in a light blue header. The main content area is a white form with two input fields: 'Nome do Robo' and 'Descrição do Robo'. A blue 'Salvar' button is located at the bottom center of the form.

Figura 13 – Tela Cadastro do Robô

5.7 Tela Listar Robôs

A tela da Figura 14 apresenta todos os robôs cadastrados pelos administradores. Cada robô listado possui os botões editar e deletar.

Se o administrador clicar no botão deletar, uma mensagem de confirmação é mandada para a tela, caso seja confirmado o robô em questão é excluído do sistema e uma mensagem de confirmação da exclusão é mostrado no canto superior da tela (em cima da lista).

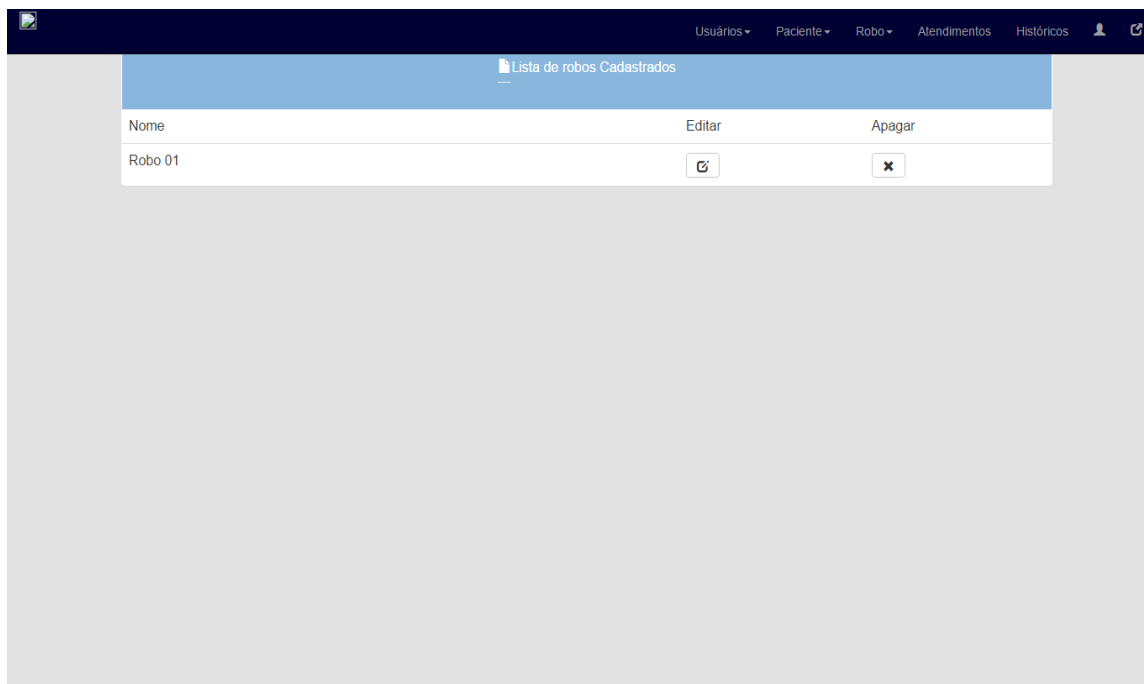


Figura 14 – Tela Lista de Robôs

5.8 Tela Cadastro Usuário do Sistema

Nesta tela (ver Figura 15) o administrador pode cadastrar um usuário e pode especificar se ele será um administrador ou não, através do *checkbox*, nela também irá escolher à especialização do profissional da saúde e colocar seu e-mail (no qual deve ser único no sistema, senão um alerta é gerado avisando que tal e-mail já está em uso), senha, nome, telefone e área de especialização do profissional.

Esta mesma tela também é usada para exibir e alterar (editar) os dados do Atendente.

The screenshot shows a web application interface for user registration. The header is dark blue with the 'VI ROBO' logo on the left and navigation links for 'Usuários', 'Paciente', 'Robo', 'Atendimentos', and 'Históricos' on the right. The main content area is white and contains a form titled 'Cadastro do Usuário'. The form has the following fields: 'E-mail' (with a sub-label 'Email de Login'), 'Password' (with a sub-label 'Senha'), 'Nome' (with a sub-label 'Nome do Usuário'), and 'Telefone'. There is also a dropdown menu for 'Especialização' currently set to 'Psicólogo'. A checkbox labeled 'administrador' is present below the dropdown. A blue 'Salvar' button is at the bottom of the form.

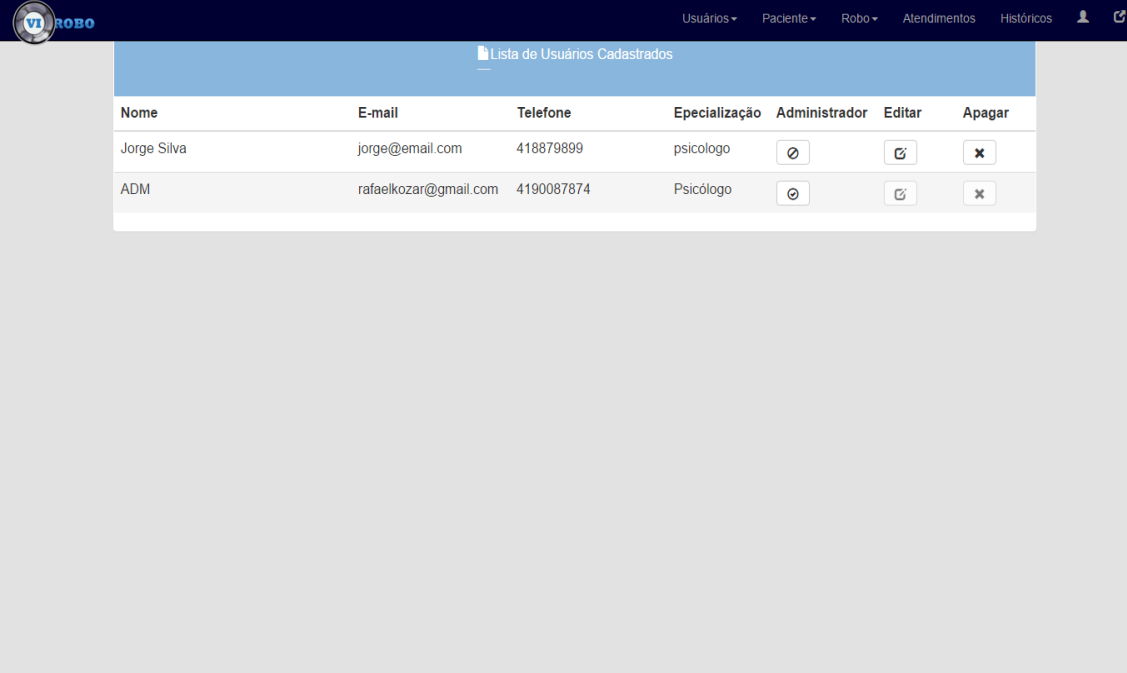
Figura 15 – Tela Cadastro do Usuário

5.9 Tela Listar Usuário

A tela da Figura 16 apresenta todos os usuários cadastrados, além de trazer informações como nome, e-mail, telefone e especialização. Cada usuário listado possui os botões editar, deletar e administrador, que indica se a pessoa é administradora ou não do sistema.

Se o administrador clicar no botão deletar, uma mensagem de confirmação é mandada para a tela, caso seja confirmado o paciente em questão é excluído do sistema e uma mensagem de confirmação da exclusão é mostrado no canto superior da tela (em cima da lista).

Os botões editar e apagar ficam desativados caso a pessoa em questão seja administradora do sistema.

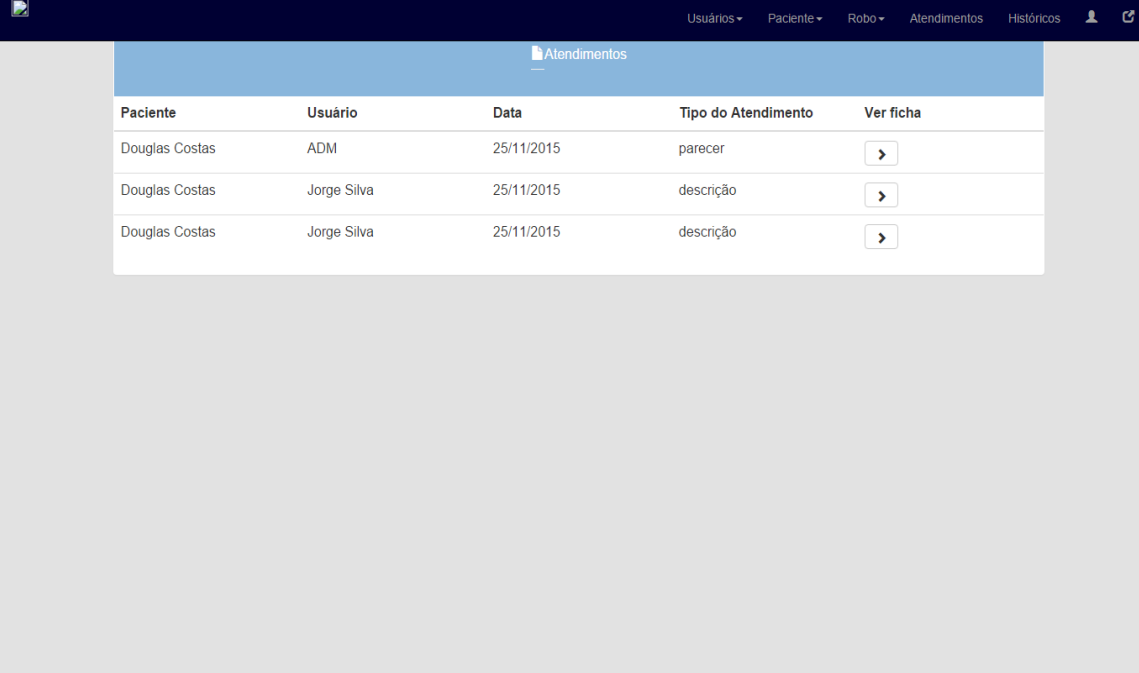


Nome	E-mail	Telefone	Especialização	Administrador	Editar	Apagar
Jorge Silva	jorge@email.com	418879899	psicologo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ADM	rafaelkozar@gmail.com	4190087874	Psicólogo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 16 – Tela Lista de Usuários

5.10 Tela Lista de Atendimentos

A tela da Figura 17 o administrador pode visualizar todos seus atendimento realizados por todos os usuários, além de pode visualizar informações como, paciente atendido, usuário, data e tipo de atendimento.

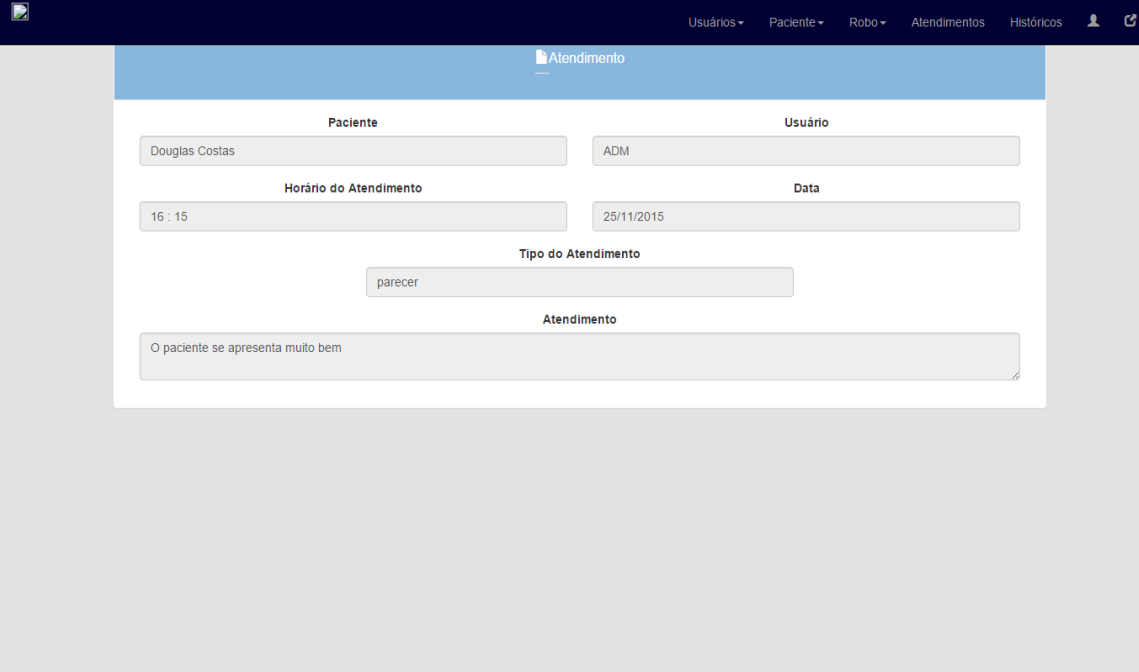


Paciente	Usuário	Data	Tipo do Atendimento	Ver ficha
Douglas Costas	ADM	25/11/2015	parecer	>
Douglas Costas	Jorge Silva	25/11/2015	descrição	>
Douglas Costas	Jorge Silva	25/11/2015	descrição	>

Figura 17 – Tela Lista Atendimentos

5.11 Tela Atendimento

A tela da Figura 18 o atendimento pode ser visualizado com maiores informações sobre o atendimento realizado.

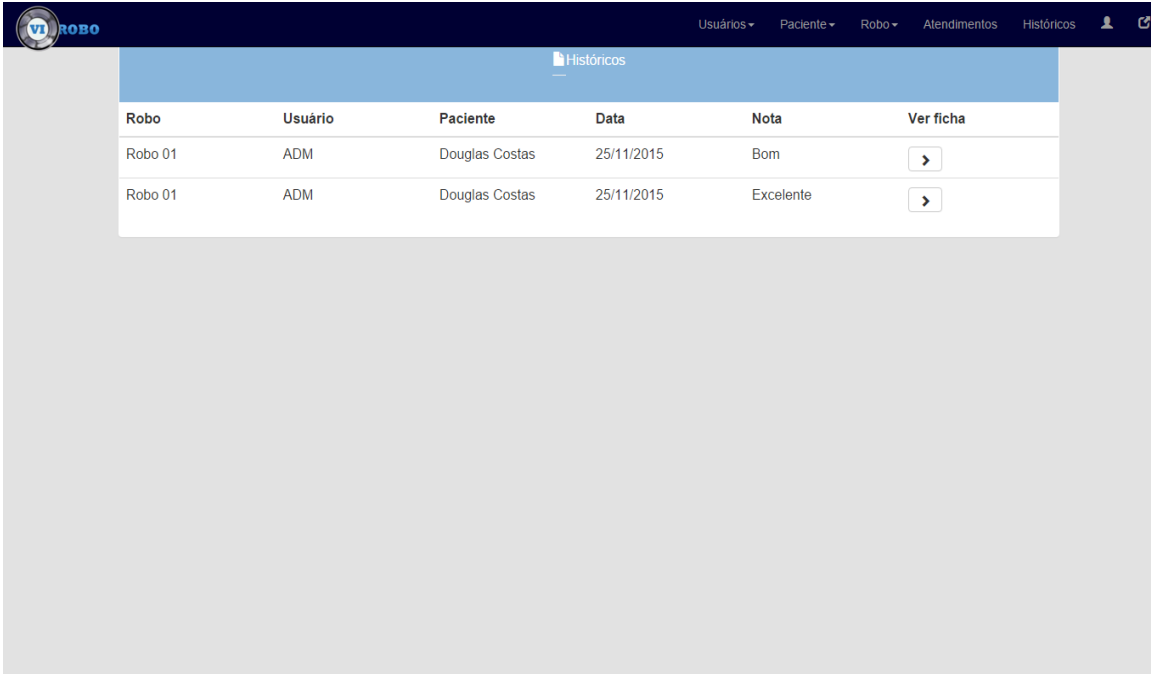


Paciente Douglas Costas	Usuário ADM
Horário do Atendimento 16:15	Data 25/11/2015
Tipo do Atendimento parecer	
Atendimento O paciente se apresenta muito bem	

Figura 18 – Tela Atendimento

5.12 Tela Históricos

Nesta tela da figura 19 o administrado pode verificar todos os históricos gerados pelo sistema e mais algumas informações.



Robo	Usuário	Paciente	Data	Nota	Ver ficha
Robo 01	ADM	Douglas Costas	25/11/2015	Bom	>
Robo 01	ADM	Douglas Costas	25/11/2015	Excelente	>

Figura 19 – Tela lista de Históricos

5.13 Tela Histórico

Na tela figura 20 as informações de um histórico são apresentadas de forma mais detalhada que na tela de lista de atendimentos.

Esses dados da tela são gerados a todo momento que um atendente acessa um paciente para realizar uma vídeo conferência ou cadastrar um atendimento.

The screenshot shows a web application interface with a dark blue header. The header contains the logo 'VI ROBO' on the left and navigation links: 'Usuários', 'Paciente', 'Robo', 'Atendimentos', 'Históricos', and a user profile icon. The main content area is titled 'Histórico' and contains a form with the following fields:

Paciente	Usuário
Douglas Costas	ADM
Robo	Data
Robo 01	25/11/2015
Duração	Nº do acesso
00:00:16.4	1
Nota	
Bom	

Figura 20 – Tela Histórico

6 CONCLUSÃO

6.1 Pequeno escopo do projeto

Este projeto de “Telemedicina de baixo custo”, tem como objetivo principal o atendimento médico a pacientes de forma remota, através de um software que permita a comunicação simultânea entre atendente e paciente, além do registro do atendimento. Ainda possui como objetivos adicionais, a disponibilização de uma biblioteca de visão computacional para processamento em nuvem e a possibilidade de comandar robôs para acompanhamento de pacientes que demandem um acompanhamento diferenciado.

A solução aqui mostrada trata de uma plataforma de software em nuvem, que utiliza linguagens e recursos de última geração para atender os objetivos propostos.

6.2 Objetivos

Conforme apresentado, a plataforma de software em nuvem para Telemedicina que foi desenvolvida, objeto deste trabalho, contém todas as funcionalidades necessárias para atender os objetivos previamente definidos, consideramos que estes foram atingidos plenamente.

6.3 Relevância e contribuição do projeto

Quando no início deste trabalho, os objetivos e prioridades eram diferentes, houve fases tais como:

- “Construção da plataforma de visão computacional na nuvem”;
- Migração do projeto pré-existente “Pia Robô” para a nuvem;
- Estruturação de uma plataforma de processamento de Visão Computacional em nuvem, com alta escalabilidade.

Estes itens deixaram de serem objetivos, embora foram implementados e assim enriqueceram a “Plataforma de Telemedicina de Baixo Custo”,

contribuindo com diferentes funcionalidades que hoje estão sob a mesma plataforma de nuvem.

Apesar de não ter resultado numa efetiva utilização na “Plataforma de Telemedicina de baixo custo”, a biblioteca OPENCV foi portada para a plataforma de nuvem e está disponível de forma totalmente funcional para futuros trabalhos, fundamentalmente para o grupo de pesquisa GEPTA da UFPR que poderá utiliza-lo em seus projetos futuros.

Outro ponto importante para salientar sobre o projeto, foi a substituição dos softwares de mercado ManyCam e IP Webcam, utilizados no início do projetos para fazer *streaming* de vídeo entre o Tablet para PC, e que hoje não são mais necessários uma vez que a plataforma implementa esta funcionalidade de forma nativa, tornando a comunicação mais rápida. Por outro lado temos que deixar registrado que a solução em alguns *smartphones* pode apresentar variações na visualização, provavelmente ocasionado por diferenças de tamanho de tela e resolução. Este comportamento, ainda precisa ser resolvido.

Por último e de fundamental importância, acreditamos verdadeiramente que a contribuição desta pesquisa possa ser introduzida no mercado e venha contribuir no atendimento remoto de pacientes, fundamentalmente em locais com escassez de profissionais da saúde, conforme relatado neste trabalho.

REFERÊNCIAS

BARCELLOS, Marco Canatech: A evolução da telemedicina e as novas ferramentas de colaboração clínica Disponível em: <<http://canaltech.com.br/coluna/saude/A-evolucao-da-Telemedicina-e-as-novas-ferramentas-de-colaboracao-clinica/>>

Acesso em 10 de Outubro de 2015.

COISI, Juliana Folha de São Paulo: Brasil aumenta o número de médicos mas distribuição continua desigual Seminários Folha da Saúde do Brasil, São Paulo, Disponível em <<http://www1.folha.uol.com.br/seminariosfolha/2015/05/1628533-brasil-aumenta-numero-de-medicos-mas-concentracao-continua.shtml>>

Acesso em 10 de Outubro de 2015.

GUEDES, G. **UML 2 Guia Prático**. São Paulo: Novatec, 2007.

PEREIRA, CAIO RIBEIRO **Node.js: Aplicações web real-time com Node.js**. São Paulo: Casa do Código, 2014.

DOUGLAS, ERIC **JavaScript Furtivo**. São Paulo: Leanpub, 2014

SILVA, MAURICIO SAMY **JavaScript: Guia do programador** São Paulo: Novatec, 2010

GONZALEZ, Rafael C., WOODS, Richard E. **Processamento de imagens digitais** São Paulo: Blucher, 2000

PRESSMAN, R. S. **Engenharia de Software**. Rio de Janeiro: McGraw-Hill, 2006

MONTEIRO, Alexandra **A história da telessaúde: da cidade para o estado do Rio de Janeiro**. Rio de Janeiro: EdUERJ, 2015

MAIS MEDICOS Mais médicos para o Brasil, mais saúde para você, Brasília, Disponível em < <http://www.maismedicos.gov.br/conheca-programa>> Acesso em 10 de Outubro de 2015.

NODE, O que é node.js? Disponível em <<http://nodebr.com/pagina7/>> Acesso em 12 de Outubro de 2015

IMASTERS, Como o Node funciona, Disponível em <<http://imasters.com.br/artigo/22016/javascript/o-que-exatamente-e-o-nodejs/>> Acesso em 12 de Outubro de 2015

V8, Motor V8 Javascript, Disponível em <<https://code.google.com/p/v8/>> Acesso em 12 de Outubro de 2015

MONGOOSE, Introdução ao Mongoose Disponível em < <http://nodebr.com/nodejs-e-mongodb-introducao-ao-mongoose>> Acesso em 12 de Outubro de 2015

MONGODB, Introdução ao Mongodb Disponível em < <http://www.devmedia.com.br/introducao-ao-mongodb/30792>> Acesso em 13 de Outubro de 2015

USP, Telemedicina no Brasil. São Paulo: Usp, 2013 Disponível em < <http://telemedicina.fm.usp.br/portal/telemedicina-no-brasil/>> Acesso em 18 de Outubro de 2015

WEBRTC, Webrtc Disponível em < <http://www.webrtc.org>> Acesso em 13 de Outubro de 2015

WEN, Chao L. Telemedicina e a Telessauda. São Paulo, 2013 Disponível em < <http://avancasaudebrasil.org.br/chaowen/artigos/telemedicina.aspx>> Acesso em 18 de Outubro de 2015

SANTOS, A.D; MIRANDA, C.A; GRONOVICZ, J; SILVA, J.R. **Pia Robot**
Concepção de um robô embarcado usando web e arduino. Trabalho de
Graduação (Tecnologia em Análise e Desenvolvimento de Sistemas) – Setor de
Educação Profissional e Tecnológica, Universidade Federal do Paraná, Curitiba,
2013

APÊNDICES

APÊNDICE A – DIAGRAMA CADASTRAR ATENDENTE..... 60

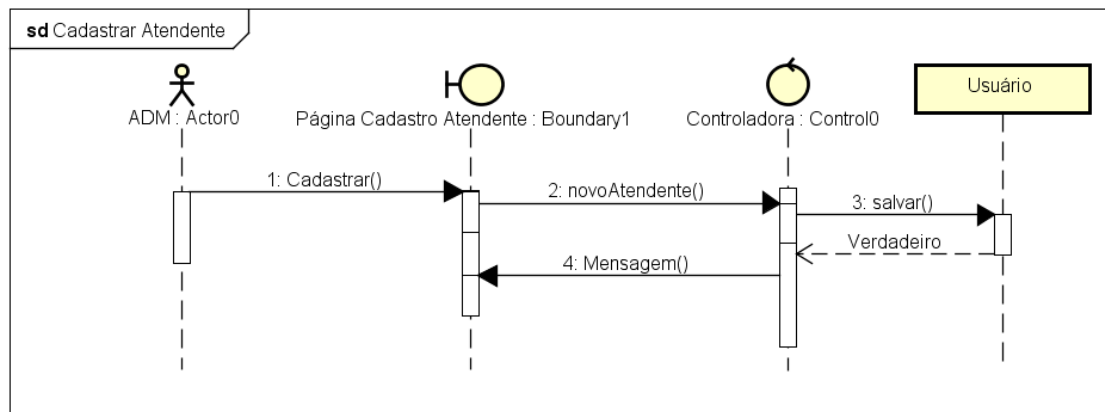
APÊNDICE B – DIAGRAMA CADASTRAR ATENDIMENTO..... 60

APÊNDICE C – DIAGRAMA CADASTRAR PACIENTE.....	61
APÊNDICE D – DIAGRAMA CADASTRAR ROBÔ.....	61
APÊNDICE E – DIAGRAMA EDITAR PACIENTE.....	62
APÊNDICE F – DIAGRAMA EDITAR ATENDENTE.....	63
APÊNDICE G – DIAGRAMA EDITAR ROBO.....	65
APÊNDICE H – DIAGRAMA EXCLUIR ATENDENTE.....	65
APÊNDICE I – DIAGRAMA EXCLUIR ROBO.....	66
APÊNDICE J – DIAGRAMA EXCLUIR PACIENTE.....	67
APÊNDICE K – DIAGRAMA MOVIMENTAR ROBÔ.....	67
APÊNDICE L – DIAGRAMA REALIZAR ATENDIMENTO COM VÍDEO.....	68
APÊNDICE M – DIAGRAMA RECEBER ATENDIMENTO.....	69

APÊNDICE A – DIAGRAMA CADASTRAR ATENDENTE

O administrador clica no botão cadastrar (ver figura 21) e então é enviado um evento para Controladora que irá usar a função salvar da classe usuário e

depois irá retorna uma mensagem para tela e assim um novo usuário será salvo no banco.

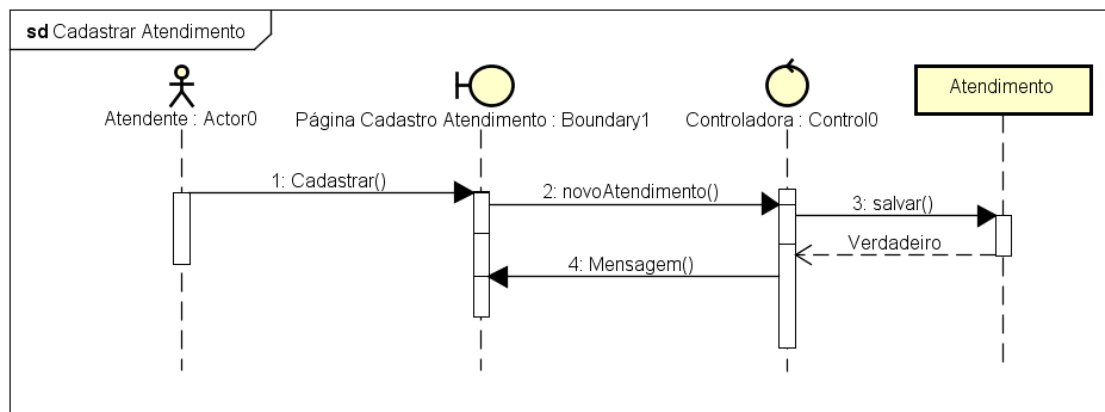


powered by Astah

Figura 21 – Diagrama Cadastrar Atendente

APÊNDICE B – DIAGRAMA CADASTRAR ATENDIMENTO

O administrador ou atendente clica no botão cadastrar (ver figura 22) e então é enviado um evento para Controladora que irá usar a função salvar da classe Atendimento e depois irá retorna uma mensagem para tela e assim um novo atendimento será salvo no banco.

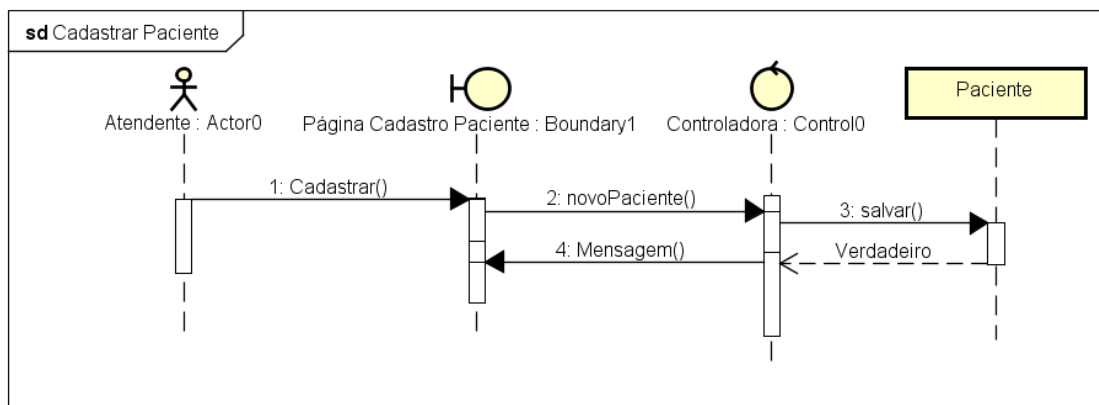


powered by Astah

Figura 22 – Diagrama Cadastrar Atendimento

APÊNDICE C – DIAGRAMA CADASTRAR PACIENTE

O administrador ou atendente clica no botão cadastrar (ver figura 23) e então é enviado um evento para Controladora que irá usar a função salvar da classe Paciente e depois irá retorna uma mensagem para tela e assim um novo paciente será salvo no banco.

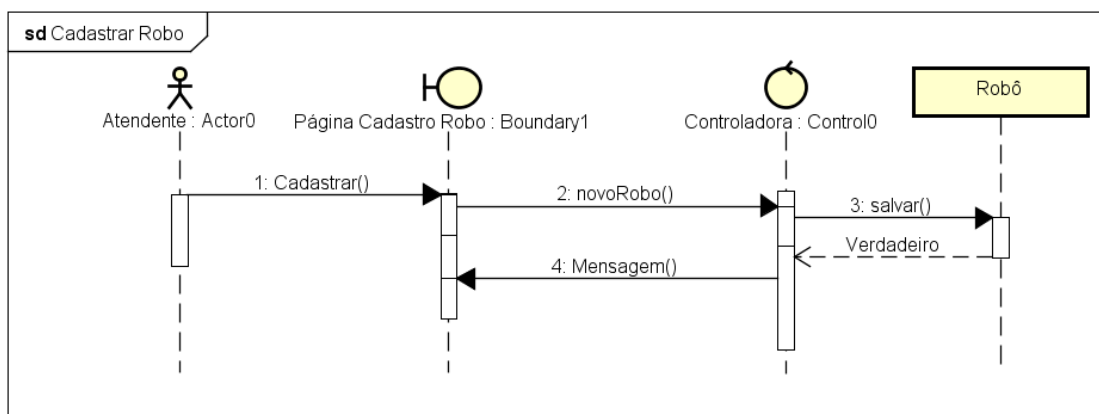


powered by Astah

Figura 23 – Diagrama Cadastrar Paciente

APÊNDICE D – DIAGRAMA CADASTRAR ROBÔ

O administrador ou atendente clica no botão cadastrar (ver figura 24) e então é enviado um evento para Controladora que irá usar a função salvar da classe Robô e depois irá retorna uma mensagem para tela e assim um novo robô será salvo no banco.



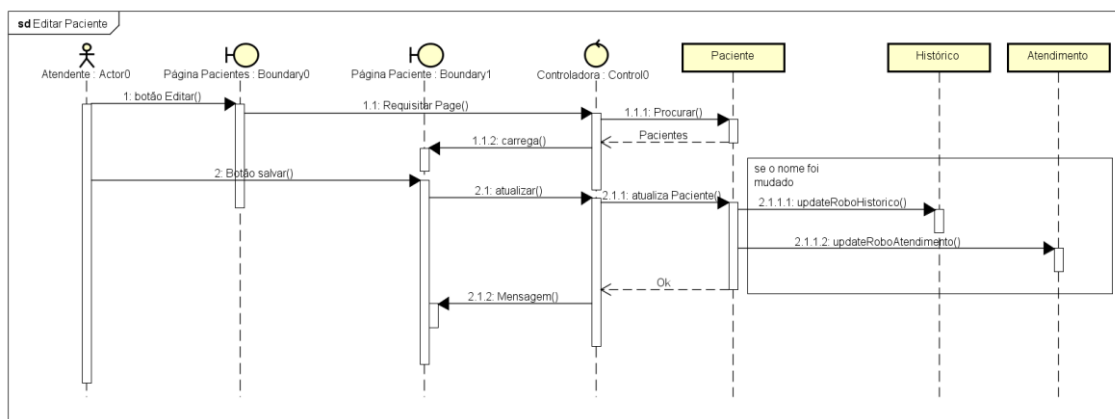
powered by Astah

Figura 24 – Diagrama Cadastrar Robô

APÊNDICE E – DIAGRAMA EDITAR PACIENTE

O atendente escolhe um paciente da lista e clica no botão editar (ver figura 25), é então requisitada a página para controladora que procura o paciente específico e retorna página carregada com os dados.

Depois de modificar os dados o atendente clica no botão salvar, um evento é enviado para a controladora que aciona a função atualizar paciente na classe paciente, ela em seguida dispara funções `updateRoboHistorico` e `updateRoboAtendimento` respectivamente para as classes Histórico e Atendimento. Depois de o paciente já estar atualizado ela retorna um ok para controladora que retorna uma mensagem para tela.



powered by Astah

Figura 25 – Diagrama Editar Paciente

APÊNDICE F – DIAGRAMA EDITAR ATENDENTE

O administrador escolhe um atendente da lista e clica no botão editar (ver figura 26), é então requisitada a página para controladora que procura o atendente específico e retorna página carregada com os dados.

Depois de modificar os dados o administrador clica no botão salvar, um evento é enviado para a controladora que aciona a função atualizar atendente na classe atendente, em seguida a classe Atendente dispara função

atualizarNomeUser para as classes Histórico, Atendimento e Paciente. Depois de o atendente já estar atualizado a classe Atendente retorna um ok para controladora que retorna uma mensagem para tela.

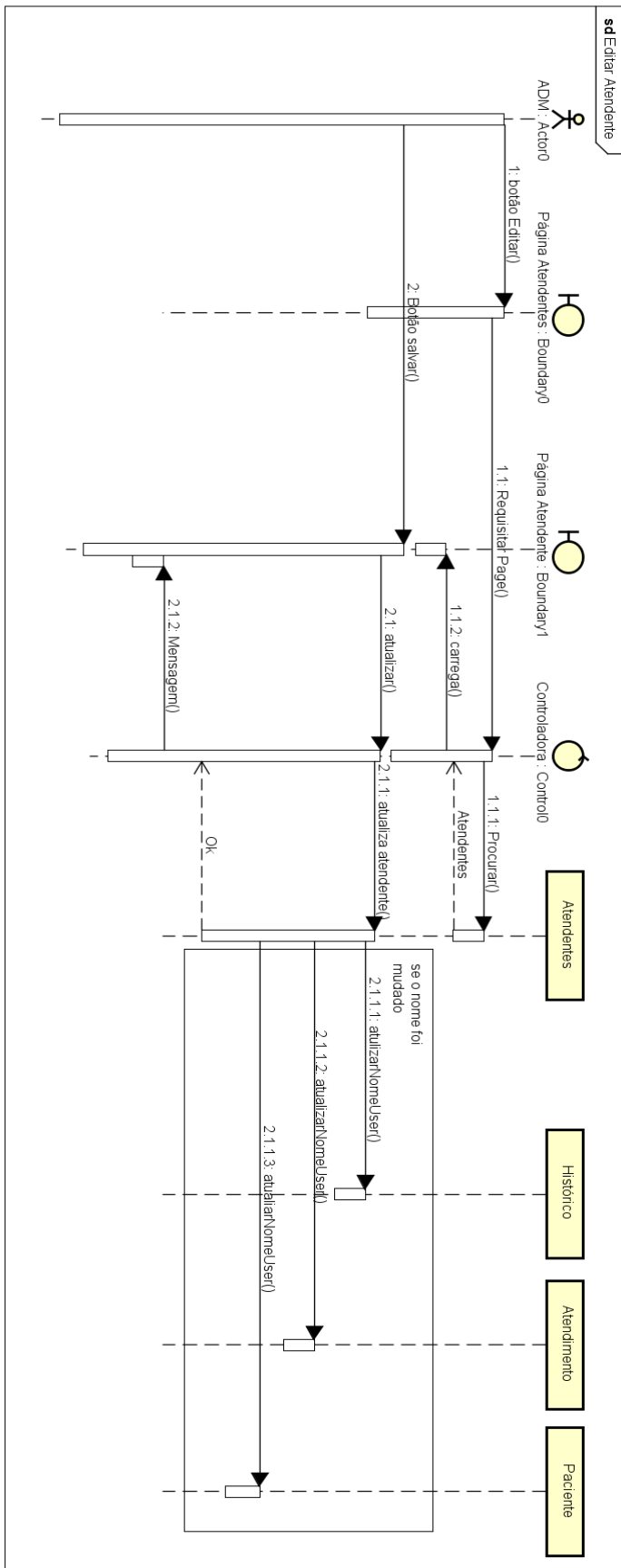
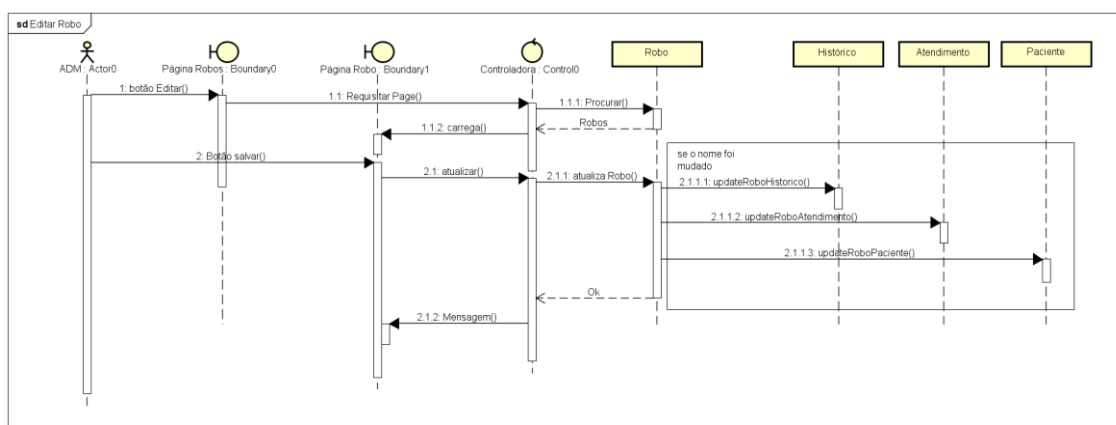


Figura 26 – Diagrama Editar Atendente

APÊNDICE G – DIAGRAMA EDITAR ROBÔ

O administrador escolhe um robô da lista e clica no botão editar (ver figura 27), é então requisitada a página para controladora que procura o robô específico e retorna página carregada com os dados.

Depois de modificar os dados o administrador clica no botão salvar, um evento é enviado para a controladora que aciona a função atualizar robô na classe Robô, em seguida a classe Robô dispara as funções updateRoboHistorico, updateRoboAtendimento, updateRoboPaciente para as classes Histórico, Atendimento e Paciente respectivamente. Depois de o robô já estar atualizado a classe retorna um ok para controladora que retorna uma mensagem para tela.



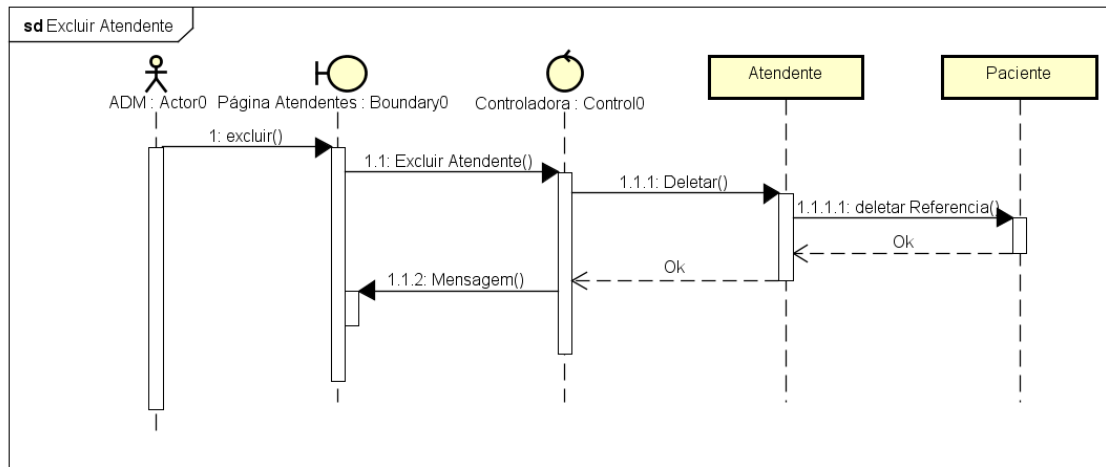
powered by Astah

Figura 27 – Diagrama Editar Robô

APÊNDICE H – DIAGRAMA EXCLUIR ATENDENTE

O administrador depois escolher um atendente que deseja deletar (ver figura 28), clica no botão excluir e um evento é enviado para controladora, que requisita a função deletar da classe Atendente e respectivamente a mesmo solicita a função deletarReferencia da classe Paciente, depois as duas retornam

uma mensagem. Assim o atendente é deletado e a controladora envia uma mensagem para a tela.

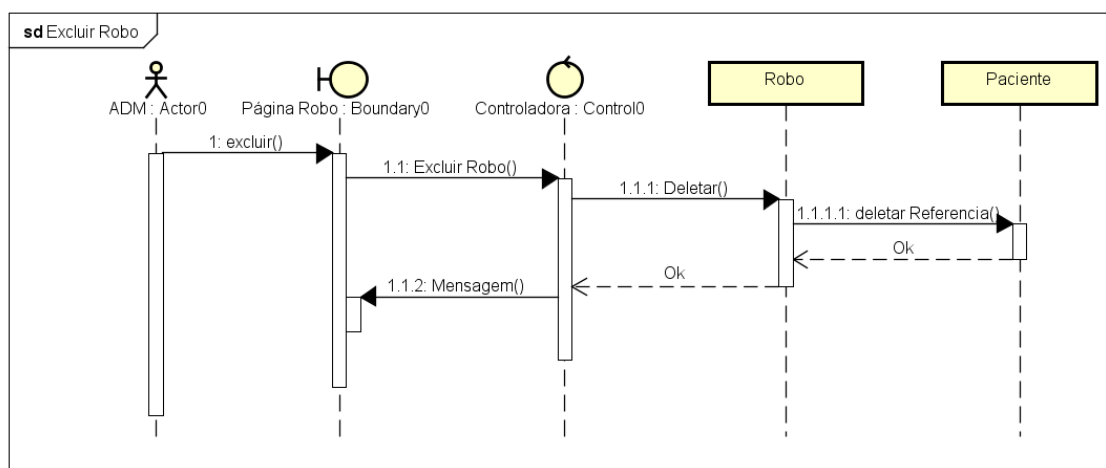


powered by Astah

Figura 28 – Diagrama Excluir Atendente

APÊNDICE I – DIAGRAMA EXCLUIR ROBÔ

O administrador depois escolher um robô que deseja deletar (ver figura 29), clica no botão excluir e um evento é enviado para controladora, que requisita a função deletar da classe Robô e respectivamente a mesmo solicita a função deletar da classe Paciente, depois as duas retornam uma mensagem. Assim o Robô é deletado e a controladora envia uma mensagem para a tela.

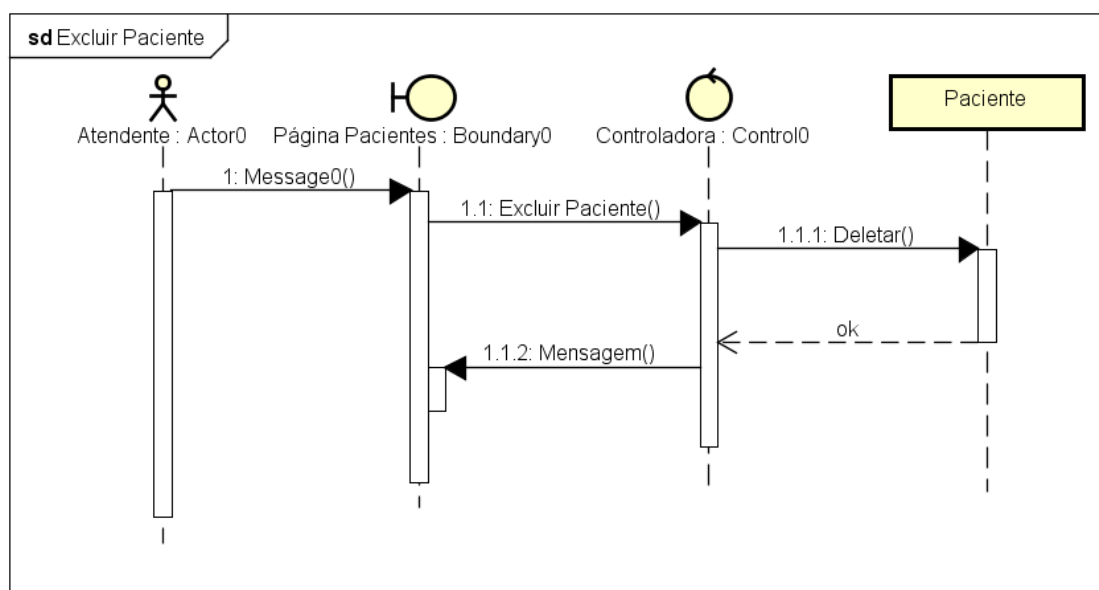


powered by Astah

Figura 29 – Diagrama Excluir Robô

APÊNDICE J – DIAGRAMA EXCLUIR PACIENTE

O administrador depois escolher um paciente que deseja deletar (ver figura 30), clica no botão excluir e um evento é enviado para controladora, que requisita a função deletar da classe Paciente, depois a classe retornam uma mensagem. Assim o paciente é deletado e a controladora envia uma mensagem para a tela.

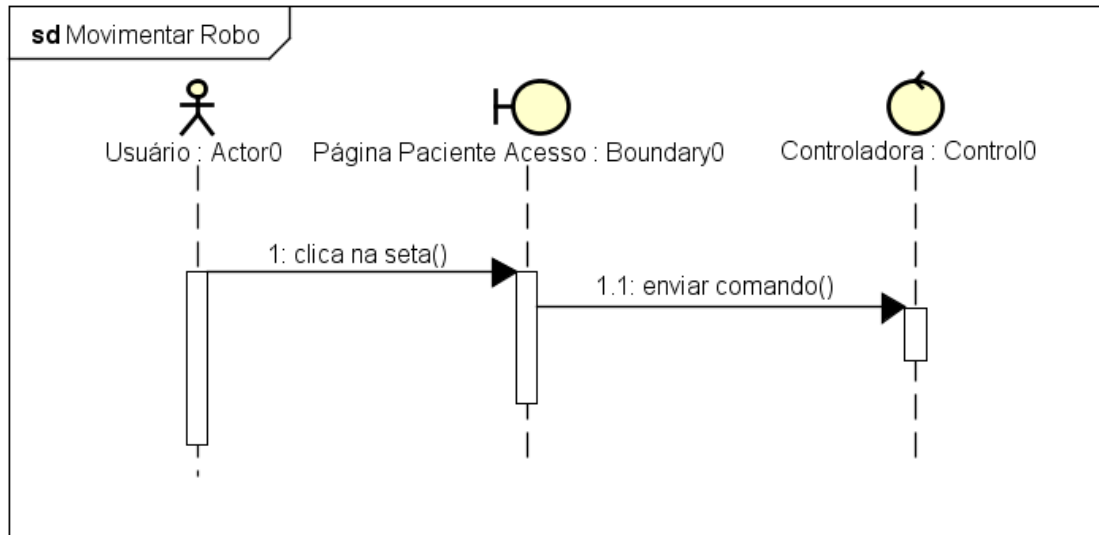


powered by Astah

Figura 30 – Diagrama Excluir Paciente

APÊNDICE K – DIAGRAMA MOVIMENTAR ROBÔ

O usuário clica na seta e o comando de direção é enviado para controladora (ver figura 31).

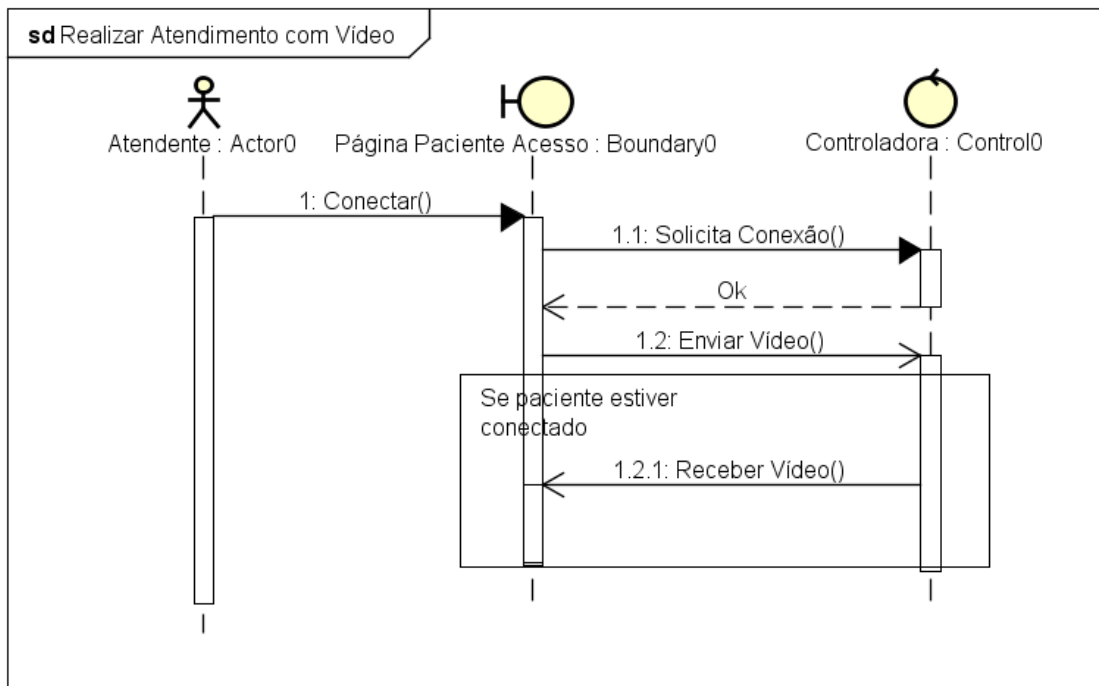


powered by Astah

Figura 31 – Diagrama Movimentar Robô

APÊNDICE L – DIAGRAMA REALIZAR ATENDIMENTO COM VÍDEO

O atendente clica no botão conectar da tela que solicita conexão (ver figura 32). Se tudo estiver ok na página começa enviar o vídeo e também receber o vídeo do paciente.

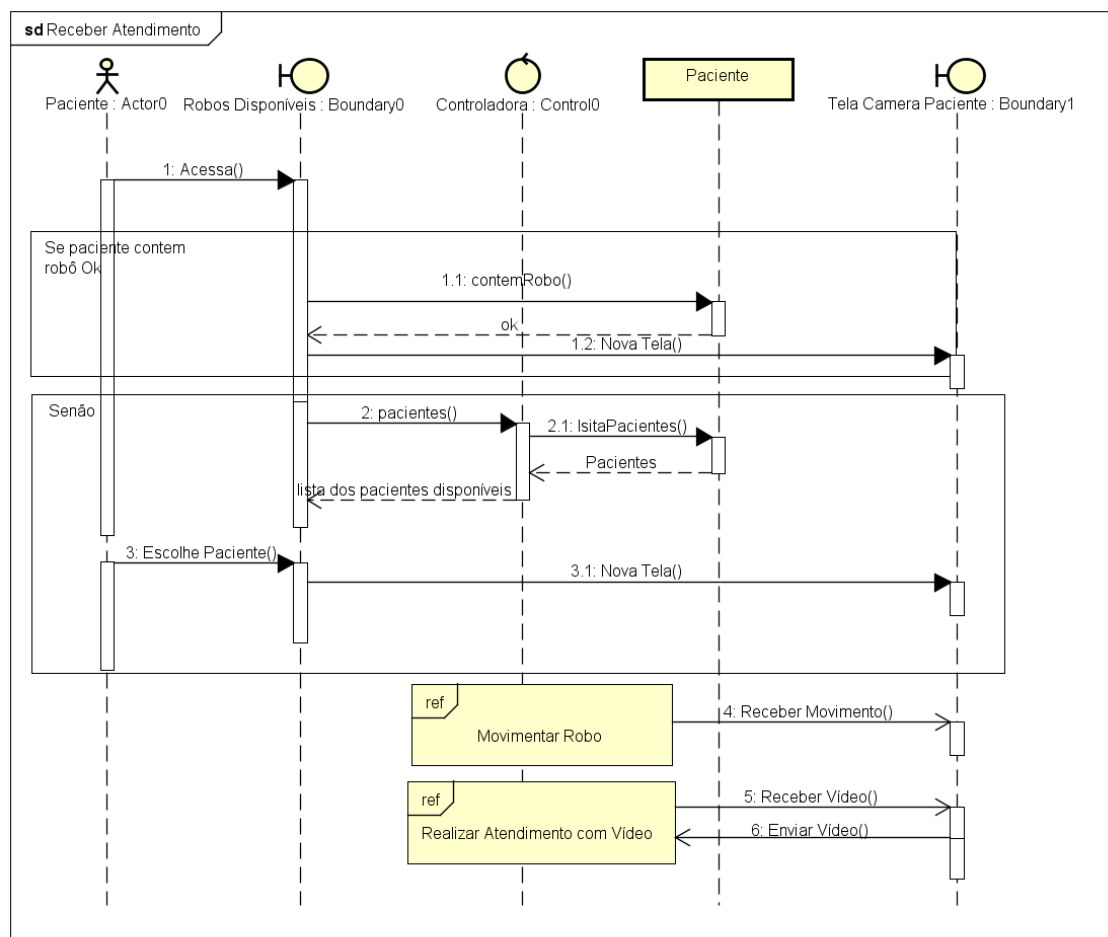


powered by Astah

Figura 32 – Diagrama Realizar atendimento com Vídeo

APÊNDICE M – DIAGRAMA RECEBER ATENDIMENTO

O paciente acessa a página (ver figura 33), uma lista de pacientes é solicitada para a classe Paciente. O paciente escolhe quem ele é. Uma nova tela é aberta, que agora recebe vídeo e movimentos e também.



powered by Astah

Figura 33 – Diagrama Receber Atendimento