

ARQUITETURA MIMO PARA PULSEIRAS INTELIGENTES COM
APLICAÇÃO MÓVEL PARA MONITORAMENTO DE BAIXO CUSTO
DE PACIENTES NO PÓS-CIRÚRGICO

Juliano Barbosa Prettz

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ARQUITETURA MIMO PARA PULSEIRAS INTELIGENTES COM
APLICAÇÃO MÓVEL PARA MONITORAMENTO DE BAIXO CUSTO
DE PACIENTES NO PÓS-CIRÚRGICO**

JULIANO BARBOSA PRETTZ

**ORIENTADOR: JOÃO PAULO CARVALHO LUSTOSA DA COSTA
COORIENTADOR: RICARDO KEHRLE MIRANDA**

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO:

BRASÍLIA / DF, 06 DE JULHO DE 2017.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**ARQUITETURA MIMO PARA PULSEIRAS INTELIGENTES COM APLICAÇÃO
MÓVEL PARA MONITORAMENTO DE BAIXO CUSTO DE PACIENTES NO PÓS-
CIRÚRGICO**

JULIANO BARBOSA PRETTZ

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA
FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
ENGENHARIA ELÉTRICA.**

APROVADA POR:

Prof. João Paulo Carvalho Lustosa da Costa
(Orientador)

Prof. Georges Daniel Amvame Nze
(Examinador Interno)

Prof. Tales Heimfarth
(Examinador Externo)

BRASÍLIA/DF, 06 DE JULHO DE 2017

Ficha catalográfica

Barbosa Prettz, Juliano

ARQUITETURA MIMO PARA PULSEIRAS INTELIGENTES COM APLICAÇÃO MÓVEL PARA MONITORAMENTO DE BAIXO CUSTO DE PACIENTES NO PÓS-CIRÚRGICO / Juliano Barbosa Prettz; orientador JOÃO PAULO CARVALHO LUSTOSA DA COSTA; coorientador RICARDO KEHRLE MIRANDA. -- Brasília, 2017. XX p.

Dissertação (Mestrado - Mestrado em Engenharia Elétrica) -- Universidade de Brasília, 2017.

1. Pulseiras inteligentes. 2. Concentrador de sinais.
 3. Desenvolvimento de aplicação. 4. Arquitetura de comunicação MIMO.
- I. CARVALHO LUSTOSA DA COSTA, JOÃO PAULO, orient. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

CESSÃO DE DIREITOS

AUTOR: JULIANO BARBOSA PRETTZ

TÍTULO:

GRAU: Mestre ANO: 2017

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

JULIANO BARBOSA PRETTZ

CLN 212, Bloco B – Apartamento 208 – Asa Norte / Brasília - DF

CEP: 70864-520

AGRADECIMENTOS

Aos meus pais Maristela Barbosa Prettz e Carlos Alberto Brum Prettz, e aos meus irmãos Cassio Barbosa Prettz e Bruno Barbosa Prettz pelo amor incondicional, carinho e força em todos os momentos.

Ao meu professor e orientador João Paulo Carvalho Lustosa da Costa, pela parceria, paciência e confiança na minha caminhada dentro da Universidade de Brasília da qual muito me orgulho.

Ao amigo e coorientador Ricardo K. Miranda, pelo auxílio, parceria e orientação na pesquisa e nos artigos.

A fundação Lemann pela aposta e confiança na pesquisa apoiando financeiramente para que a solução fosse desenvolvida.

A minha namorada Thaise P. Brião pela compreensão, carinho e amor.

Ao professor Edison P. de Freitas pelo auxílio nos trabalhos publicados e pela parceria ao longo dos anos.

Aos alunos e amigos João Alvim, Gabriel Pinheiro, Matheus Baffuto, Jorge Mendes e Vitor Bordini pela caminhada e auxílio na pesquisa como um todo.

Aos amigos e colegas Luciano Almeida e Mateus Zanatta, pelas inúmeras trocas de conhecimento e ideias que vieram agregar ao trabalho.

RESUMO

ARQUITETURA MIMO PARA PULSEIRAS INTELIGENTES COM APLICAÇÃO MÓVEL PARA MONITORAMENTO DE BAIXO CUSTO DE PACIENTES NO PÓS-CIRÚRGICO

Autor: Juliano Barbosa Prettz

ORIENTADOR: João Paulo Carvalho Lustosa da Costa

COORIENTADOR: Ricardo Kehrle Miranda

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 06 de julho de 2017.

O constante acompanhamento de pacientes pós-cirúrgicos é crucial para uma recuperação bem-sucedida. Muitos pacientes ficam internados em hospitais por longos períodos após as cirurgias, implicando tanto em uma recuperação mais lenta e menos efetiva do paciente, como também em custos mais elevados para as instituições. Durante o período de observação, o corpo médico faz a coleta manual de informações dos pacientes, inserindo estas informações no sistema de informação médico. Neste sentido, a utilização de tecnologias vestíveis de saúde, do inglês *wearable health technologies*, apresenta um papel importante tanto no contexto de permitir uma reocupação mais independente destes espaços com constante acompanhamento, como também para redução dos custos em tratamento pós-cirúrgicos. Uma das tecnologias vestíveis de saúde amplamente disponível no mercado é a das pulseiras inteligentes, do inglês *smartbands*. Tais pulseiras são capazes de medir batimentos cardíacos, a quantidade de passos, localização do usuário e elevação de degraus. Cada pulseira inteligente do mercado possui um aplicativo para *smartphones*, *tablets* e acesso *web*, que faz com que o usuário armazene e gere gráficos com os dados de monitoramento em períodos desejados. Neste contexto, este trabalho propõe um sistema de monitoramento eficiente e de baixo custo, baseado em pulseiras inteligentes comerciais. Este sistema visa o monitoramento de pacientes, uma vez que as pulseiras inteligentes disponíveis são restritas a conexões *Single Input Single Output* (SISO). Propõe-se uma nova arquitetura, com um concentrador de sinal que permite que as pulseiras inteligentes possam realizar conexões *Multiple Input Multiple Output* (MIMO). Como resultado obtém-se uma arquitetura MIMO para pulseiras inteligentes apoiadas por uma aplicação móvel proposta voltada para pacientes.

Palavras-chave: Pulseiras inteligentes, Concentrador de sinais, Desenvolvimento de aplicação móvel, Arquitetura de comunicação MIMO.

ABSTRACT

MIMO ARCHITECTURE FOR SMARTBANDS WITH MOBILE APPLICATION FOR LOW COST MONITORING OF POST-SURGICAL PATIENTS

Autor: Juliano Barbosa Prettz

ORIENTADOR: João Paulo Carvalho Lustosa da Costa

COORIENTADOR: Ricardo Kehrle Miranda

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 06 de julho de 2017.

The constant monitoring of post-surgical patients is crucial to a successful recovery. Several patients are hospitalized for long periods after surgery, resulting in both slower and less effective recovery of patients, as well as higher costs for hospitals. During observation period, the medical staff manually provides vital information from patients by inserting this information into the medical information system. In this sense, the use of health wearable technologies plays an important role, both in the sense of allowing a more independent reoccupation of hospital's rooms and constant monitoring, as well as to reduce costs of post-surgical treatment. One of the most widely available health-care technologies in the market are the smartbands. Such smartbands are capable of measuring heart rate, number of steps, user location and elevation. Every smartband in the market has an application for smartphones, tablets and web, which allows the user to store and generate charts with the collected monitoring data at desired periods. This work proposes an efficient and low-cost monitoring system based on commercial smartbands. This monitoring system aims to monitor patients in hospital post-operative rooms. Since the available smartbands are restricted to Single Input Single Output (SISO) connections, which allow the user to monitor only their own health status, it is proposed to extend the functionality of smartbands. Proposing a new architecture with a signal concentrator that allows smartbands to make Multiple input multiple output (MIMO) connections. As a result, a MIMO architecture for smartbands supported the develop of a mobile patient-oriented application is obtained.

Keywords: Smartbands, Signal Concentrator, Mobile application development, MIMO communication architecture

SUMÁRIO

1 – INTRODUÇÃO.....	1
1.1 - OBJETIVOS.....	3
1.1.1 – Objetivo Gerais.....	3
1.1.2 – Objetivos Especificos.....	3
1.2 - ORGANIZAÇÃO DO TRABALHO.....	3
2 – ESTADO DA ARTE DE SOLUÇÕES DE MONITORAMENTO MÉDICO.....	5
3 – PROPOSTA DE ARQUITETURA PARA SOLUÇÃO DE BAIXO CUSTO.....	13
3.1 – PROPOSTA DE ARQUITETURA DE COMUNICAÇÃO MIMO EFICIENTE E DE BAIXO CUSTO PARA PULSEIRAS INTELIGENTES APLICADAS AO ATENDIMENTO AO PACIENTE NO PÓS-OPERATÓRIO.....	14
3.1.1. - Arquitetura MIMO para Pulseiras Inteligentes Comerciais.....	14
3.1.2. - Validação Experimental.....	18
3.2. - SUBSTITUIÇÃO DO CONCENTRADOR POR UM SMARTPHONE.....	20
3.2.1. - Concentrador de Sinais Utilizando Raspberry PI.....	21
3.2.2. - Concentrador de Sinais Utilizando um Smartphone.....	22
4 – DESENVOLVIMENTO DE APLICATIVO PARA A ARQUITETURA MIMO PROPOSTA.....	24
4.1- ESTRUTURA DE FUNCIONAMENTO DA APLICAÇÃO PROPOSTA.....	27
4.2. - APLICAÇÃO MÓVEL.....	30
5 - CONCLUSÕES	44
5.1 - TRABALHOS FUTUROS	45
REFERÊNCIAS BIBLIOGRÁFICAS.....	46
APÊNDICE I.....	52
APÊNDICE II.....	70

LISTA DE TABELAS

Tabela 1 – Comparativo das soluções investigadas no estado da arte.....	10
Tabela 2 – Comparativo das soluções investigadas no estado da arte que possuem pulseiras inteligentes.....	11
Tabela 3 – Taxa de transferência de dados.....	20
Tabela 4 – Percentual de processamento computacional do concentrador de sinal variando o número.....	21
Tabela 5 – Comparativo entre os modelos utilizados de pulseiras inteligentes.....	27

LISTA DE FIGURAS

Figura 1 – Arquitetura nativa SISO das pulseiras inteligentes.....	5
Figura 2 – Arquitetura MIMO utilizando o concentrador de sinais.....	16
Figura 3 – Arquitetura MIMO com concentradores de sinal P para qualquer quantidade M de pulseiras inteligentes.....	18
Figura 4 – Arquitetura desenvolvida para funcionamento do concentrador de sinais.....	23
Figura 5 – Modelos de pulseiras inteligentes utilizados nesta dissertação.....	28
Figura 6 – Diagrama da estrutura da base de dados.....	30
Figura 7 – Diagrama simplificado das telas da aplicação.....	32
Figura 8 – Diagrama de Caso de Uso modelo da aplicação.....	34
Figura 9 – Tela de login da aplicação.....	35
Figura 10 – Feed inicial da aplicação.....	36
Figura 11 – Lista de pacientes na aplicação.....	37
Figura 12 – Cadastro de Paciente.....	38
Figura 13 – Perfil do Paciente.....	39
Figura 14 – Gráficos gerados pela aplicação através dos dados da pulseira inteligente.....	40
Figura 15 – Lembrete do paciente na aplicação.....	41
Figura 16 – Anexo de pulseira inteligente a paciente.....	41
Figura 17 – Equipes da aplicação.....	42
Figura 18 – Perfil Médico.....	43
Figura 19 – Configuração da Aplicação.....	44
Figura 20 – Login da Aplicação.....	53
Figura 21 – Primeiro cadastro na aplicação.....	54
Figura 22 – Recuperação de senha.....	55
Figura 23 – Feed de Atualizações.....	56
Figura 24 – Pacientes.....	57
Figura 25 – Cadastrar paciente.....	58
Figura 26 – Grupo de Pacientes.....	59
Figura 27 – Perfil do Paciente.....	61
Figura 28 – Dados de Saúde.....	62
Figura 29 – Edição de dados de paciente.....	63
Figura 30 – Lembretes.....	64

Figura 31 – Compartilhar com equipe.....	65
Figura 32 – Configurações do Paciente.....	66
Figura 33 – Perfil médico.....	67
Figura 34 – Edição de dados do perfil médico.....	68
Figura 35 – Equipes.....	69
Figura 36 – Configurações.....	70
Figura 37 – Adicionar pulseiras.....	71

ACRÔNIMOS

AFP	<i>Adaptive Frequency Hopping</i>
API	<i>Application Programming Interface</i>
APM	Associação Paulista e Medicina
BLE	<i>Bluetooth Low Energy</i>
CEP	Processamento de Eventos Complexos
CFM	Conselho Federal de Medicina
CPU	<i>Central Processing Unit</i>
DF	Distrito Federal
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
IHC	Interface Humano-Computador
MIMO	<i>Multiple Input Multiple Output</i>
MISO	<i>Multiple Input Single Output</i>
OMS	Organização Mundial da Saúde
SDK	<i>Android Software Development Kit</i>
SISO	<i>Single Input Single Output</i>
SQL	<i>Structure Query Language</i>
UnB	Universidade de Brasília
ROM	<i>Read Only Memory</i>
XP	<i>Extreme Programming</i>

1. INTRODUÇÃO

Atualmente as estatísticas mostram que mais de 44% dos países membros da Organização Mundial da Saúde (OMS) relatam ter menos de 1 médico por 1000 habitantes [1]. O pós-operatório tem uma grande contribuição na ocupação dos leitos de um hospital e do número limitado de médicos. Tradicionalmente, o estado de cada paciente do pós-operatório é observado pessoalmente pela equipe médica em horários diferentes. No entanto, devido à quantidade limitada de médicos e enfermeiros em um hospital, os sinais vitais somente são medidos em determinados períodos de tempo e, eventualmente, são inseridos manualmente no sistema de informações médicas. Além do atraso causado pela observação de pacientes múltiplos por um grupo médico limitado, esse procedimento manual é propenso a falhas devido à perda de informação ou mesmo à inserção errônea de dados [2, 3].

Uma pesquisa realizada pelo Instituto Data-Folha [4], por meio de uma solicitação do Conselho Federal de Medicina (CFM) e da Associação Paulista de Medicina (APM) mostra que os serviços públicos e privados de saúde no Brasil são considerados como regulares, ruins ou péssimos por 93% da população. Isto indica que a saúde precisa de soluções inovadoras, que auxiliem e melhorem o acompanhamento dos pacientes.

Pensando em otimizar o acompanhamento e o atendimento de pacientes, direcionado para pacientes em estado pós-cirúrgico, pois é onde os enfermos mais necessitam de um acompanhamento melhor em sua recuperação, propôs-se a utilização de tecnologias vestíveis de saúde, do inglês *wearable health technologies*, que apresentam um papel importante no acompanhamento de sinais provindos de seus usuários, permitindo uma reocupação mais independente e ágil dos leitos à disposição em um hospital ou clinica, aliada ao constante acompanhamento do paciente como, também, à redução dos custos em tratamentos pós-cirúrgicos. Desta forma, optou-se pelo uso de pulseiras inteligentes, do inglês *smartbands*.

Para este trabalho foram escolhidas, em um primeiro momento, o uso de pulseiras inteligentes comerciais da marca Fitbit [5]. Esta marca foi escolhida por possuir uma interface de programação

de aplicação, do inglês *Application Programming Interface (API)* aberta, permitindo realizar o desenvolvimento utilizando a mesma. Uma API é um conjunto de funções, comandos, protocolos e objetos que facilitam o desenvolvimento de aplicações por parte dos programadores, fazendo com que os mesmos possam criar e/ou interagir com um sistema externo. A API normalmente fornece diversos comandos para executar operações comuns, permitindo que o desenvolvedor não tenha que escrever seu código do zero [6].

Além disso, a aplicação móvel proposta permite a redução dos custos, melhorando a comunicação entre paciente e médico, além de fornecer uma ferramenta de apoio à saúde com possibilidade de impacto positivo na área médica. Também é proposta a expansão de arquitetura de conexão atual das pulseiras inteligentes para sua utilização no aplicativo móvel.

Atualmente, os dispositivos comerciais de baixo custo, como pulseiras inteligentes, estão sendo amplamente utilizados no monitoramento de sinais de seus usuários em tempo real por profissionais de autorizados ou pelo próprio usuário, porém sua direção ainda é muito guida para o publico atlético *fitness*. No entanto, as marcas comerciais atuais possuem uma arquitetura de comunicação de Uma Entrada Uma Saída, do inglês *Single Input Single Output (SISO)*, ou seja, classificando a arquitetura como 1:1, que conecta o dispositivo portátil ao smartphone pessoal do paciente, usando um padrão *Bluetooth IEEE 802.15* [5,7,8,9,10]. Portanto, devido à arquitetura de comunicação SISO, diversas pulseiras inteligentes comerciais não são, a princípio, adequadas para aplicações médicas, uma vez que não é possível compartilhar as informações com outros dispositivos.

Neste contexto, esta dissertação propõe o desenvolvimento de uma aplicação para acompanhamento de pacientes no pós-cirúrgico, transformando uma arquitetura nativa SISO em uma arquitetura de Múltiplas Entradas Múltiplas Saídas, do inglês *Multiple Input Multiple Output (MIMO)*, eficiente e de baixo custo, para o uso de pulseiras inteligentes comerciais no apoio aos pacientes no pós-cirúrgico em seu acompanhamento.

1.1 OBJETIVOS

1.1.1 Objetivos Gerais

O objetivo geral desta dissertação de mestrado é o de apresentar o desenvolvimento de uma arquitetura MIMO eficiente e de baixo custo, utilizando pulseiras inteligentes para o acompanhamento de pacientes no pós-cirúrgico, também é apresentado o desenvolvimento da aplicação móvel para *smartphones*.

1.1.2 Objetivos Específicos

Os objetivos específicos desta dissertação estão destacados abaixo:

- Desenvolver uma solução que possibilite melhorar a qualidade do atendimento médico em hospitais e clínicas;
- Otimizar e expandir uma arquitetura SISO para uma arquitetura MIMO;
- Desenvolvimento da aplicação móvel apoiando a utilização das pulseiras inteligentes.

1.2 ORGANIZAÇÃO DO TRABALHO

Tendo determinado a finalidade do estudo e as definições que servem de base para a pesquisa, os capítulos seguintes focam na revisão da literatura, descrição do procedimento de investigação e os resultados da pesquisa, e finalizam nas conclusões e recomendações para um estudo mais aprofundado. Para um melhor entendimento, este trabalho é organizado em capítulos na seguinte sequência.

O Capítulo 2 traz uma revisão da literatura onde estão explicitados conceitos importantes relacionados às áreas abordadas na pesquisa. São mostradas as principais soluções no mercado, no Brasil e no mundo, destacando a motivação para o desenvolvimento e uso de dispositivos vestíveis como pulseiras inteligentes.

No Capítulo 3, é descrita a proposta para arquiteturas de baixo custo. São apresentadas soluções de baixo custo, utilizando microcomputadores como *Raspberry Pi*, até mesmo celulares, e as diferenças entre ambas tecnologias.

Já no Capítulo 4, detalha-se a proposta de aplicativo móvel para variadas pulseiras inteligentes com um concentrador de sinais.

Por fim no Capítulo 5 são apresentadas as conclusões do trabalho desenvolvido e as sugestões para trabalhos futuros.

2. ESTADO DA ARTE DE SOLUÇÕES DE MONITORAMENTO MÉDICO

Neste capítulo, apresenta-se o estado da arte envolvendo soluções para pulseiras inteligentes comerciais e monitoramento médico. Os maiores exemplos atualmente são os dispositivos vestíveis, como pulseiras inteligentes que medem diversos sinais do corpo humano através de sensores e realizam o processamento de tais sinais para o usuário possa verificar as informações. Um ponto importante que atinge os usuários dessas tecnologias é o uso de tecnologia móvel. Atualmente, todos os dias são lançados novos aparelhos, tendo o foco quase que totalmente para novos aplicativos, sendo desenvolvidos para o monitoramento do usuário com arquiteturas SISO como [5,7,8,9,10,20,25,26,27].

Os atuais sistemas de monitoramento de sinais do corpo humano, desenvolvidos com base em dispositivos vestíveis, funcionam de acordo com o esquema básico mostrado na Figura 1, em que: os dados adquiridos pela pulseira são transmitidos para o *smartphone* do paciente; em seguida, o *smartphone* retransmite os dados para o servidor de nuvem - de modo que o servidor da API, representado pelo servidor de nuvem, possa processar os dados e devolvê-los ao *smartphone*; os dados recebidos pelo *smartphone* a partir da API são exibidos somente no *smartphone* do usuário. Conforme ilustrado na Figura 1, a arquitetura do estado da arte é composta pela pulseira inteligente e pelo *smartphone* que está ligado à pulseira inteligente, isto é, uma arquitetura de comunicação SISO [11].

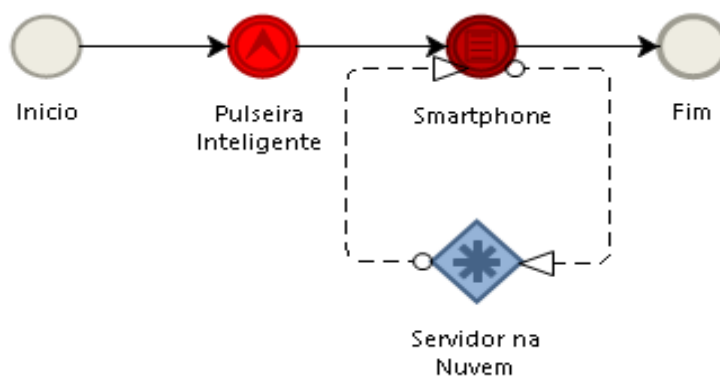


Figura 1 – Arquitetura nativa SISO das pulseiras inteligentes [11]

Em sua maioria, a arquitetura de funcionamento das pulseiras inteligentes dificulta hospitais e clínicas a utilizar as soluções do mercado, uma vez que os dados não podem ser compartilhados. Logo com o advento de novas tecnologia é considerado um grande desafio de adaptação e conhecimento das novas ferramentas para qualquer sistema de saúde. A popularização de diversas tecnologias como monitoramento de corpo humano atinge pacientes e profissionais da área da saúde.

Em [12], os autores descrevem um modelo de software que utiliza informações de pressão arterial, batimentos cardíacos e peso para analisar e cruzar tais informações, a fim de verificar riscos aos pacientes. Ainda em [12] é exemplificado, de modo simples, como funciona a arquitetura proposta em seu trabalho, em função do monitoramento dos sinais analisado. Um ponto interessante em [12] é que os dados clínicos de pacientes acabam sendo definidos utilizando informações inseridas manualmente, ou seja, as informações não são inseridas de forma automática. Entretanto, este artigo destaca uma melhoria no serviço de monitoramento e de saúde dos pacientes, analisando medições, por meio de perguntas feitas ao paciente, podendo ser realizado via telefone ou presencialmente, onde o médico responsável insere as informações no sistema.

Em [13] é apresentada uma proposta de modelo para auxiliar o monitoramento de pacientes remotamente, focando sua utilização ao Processamento de Eventos Complexos, do inglês, *Complex Event Processing (CEP)* em parte, outra etapa utiliza a ciência da situação para o desenvolvimento do processo. É visto que a maior contribuição do trabalho é na apresentação de um mecanismo de auxílio à equipe médica a compreender diversas situações, interpretando eventos e realizando uma análise inteligente desses dados. Ao decorrer do trabalho os autores dissertam sobre a técnica CEP e como ela tem sido amplamente utilizada na detecção de eventos em tempo real. Os autores também reconhecem que quando se obtém grande número de eventos, se tem a necessidade da utilização do tratamento de falhas durante sua utilização.

Em [14] os autores apresentam um modelo de software para monitoramento de pacientes, onde diversos sinais são adquiridos como: pressão arterial, medidas de peso e frequência cardíaca, que são captados por diversos médicos por meio de sensores ligados ao corpo do paciente. Os autores deram

o nome de WANDA ao sistema, que tem como objetivos melhorar o monitoramento diário de um paciente, auxiliar na tomada de decisão médica e fornecer uma solução que se agregue ao ambiente médico.

Em [15] o autor propõe a elaboração e o desenvolvimento de uma aplicação para dispositivos móveis para a prevenção de diabetes. Os autores propõem a avaliação de pontos, por meio de teste de monofilamento, analisando 8 pontos da região plantar, ou seja, são avaliados 8 pontos distintos da parte de baixo do pé do paciente. No artigo ainda é exemplificada a utilização da aplicação móvel em seu trabalho. Um ponto que vale ressaltar é que todas as informações são inseridas de modo manual e, em seus trabalhos futuros, fica aberta a questão de captar esses sinais utilizando dispositivos vestíveis.

Em [16], a empresa Afectiva exhibe um produto chamado *Q Sensor* que, além de medir temperatura e movimento, também é capaz de medir estímulos emocionais dos usuários. O produto possui conexão *Bluetooth 2.0*, seu sensor transmite informações sobre o humor do seu usuário. Além disso é possível compartilhar as informações em suas redes sociais e para seu *smartphone*. Também é possível enviar essas informações para seu médico, caso o mesmo tenha a aplicação em seu *smartphone*. Não possui API aberta e impossibilita o desenvolvimento.

A solução chamada CPRGlove, proposta em [17], é um dispositivo em forma de luva que auxilia o salvamento de vidas, ou seja, o dispositivo auxilia o usuário a realizar massagem cardíaca de modo correto. Este procedimento, chamado de Ressurreição Cardiopulmonar, é um procedimento bem conhecido dos médicos e enfermeiros. Porém, se realizado de modo incorreto, pode acarretar na perda de várias vidas. Na solução proposta em forma de luva existe uma coleta de dados como ângulo e força, por meio de sensores acoplados na luva. Posteriormente esses dados são enviados para uma aplicação que é executada apenas em computadores via sincronização de dados. Tais dados serão analisados por profissionais da saúde.

A empresa Motorola, em 2012, lançou seu primeiro dispositivo vestível. A solução, chamada

MOTOACTV [18], foi o primeiro dispositivo da empresa voltado para a captação de sinais de seus usuários. O produto foi projetado para auxiliar atletas a monitorar seus sinais vitais. O dispositivo grava dados do treino, como calorias queimadas, monitor de frequência cardíaca. Também contém um sistema de posicionamento global, do inglês, *Global Positioning System (GPS)* para mapear o caminho percorrido pelo usuário. Operado totalmente via toque, é resistente a suor, chuva e se adapta à iluminação do ambiente. O produto é muito grande, o que acarretou os seus usuários utilizarem ele no braço e não no punho como um relógio inteligente ou pulseira inteligente.

A empresa Fitbit [5] surgiu como uma *startup* no ano de 2007, apostando no mercado de dispositivos vestíveis, que na época era pouco explorado comercialmente. Seus dispositivos são capazes de monitorar dados diários do usuário, como a contagem de passos, distância percorrida, tempo de descanso, calorias, elevação, batimentos cardíacos e até mesmo o sono. Um destaque é sua bateria que pode durar até 10 dias, dependendo do modelo escolhido. Sua arquitetura permite o desenvolvimento de aplicações utilizando uma API própria [19] de desenvolvimento.

A empresa Sony [20] desenvolveu uma pulseira inteligente para uso restrito de usuário do sistema operacional *Android 4.4* [21]. A pulseira inteligente tem a capacidade de monitorar o sono, possui despertador e pedômetro que serve para contar os passos de seus usuários. É uma pulseira que não teve muito êxito no mercado, além de possuir uma plataforma fechada para desenvolvimento.

A *Apple* [22] lançou uma pulseira inteligente de fácil utilização, tendo como característica o minimalismo. A pulseira inteligente controla a dieta dos usuários, por meio de informações inseridas manualmente, por meio da aplicação proprietária da pulseira. Também é possível ativar alerta de ociosidade, despertador. A bateria dura por até 7 dias, por não haver visor, o que reduz o gasto de energia. Um diferencial é que a mesma se integra a outros aplicativos como o *MyFitnessPal* [23], que é uma aplicação para atletas e o *Runkeeper* [24], que tem o mesmo objetivo.

A empresa *Samsung* [25] lançou sua pulseira inteligente chamada *Samsung Gear S*. Tal pulseira possui funções como controle de passos e batimentos cardíacos. É uma das poucas pulseiras

inteligentes que não tem seu escopo desenvolvido para o meio *fitness* e aposta mais no social (não entendi) do seu usuário. Tendo entrada para *chip*, permite chamadas telefônicas e notificações. Sua interface de programação é fechada, inviabilizando o desenvolvimento de aplicações para o dispositivo.

A Garmin [26], empresa conhecida por desenvolver produtos para a linha *fitness*, lançou sua pulseira inteligente, a Vivofit, possuindo monitor cardíaco, de gasto calórico e de metas de exercício físico. Um grande problema é que sua pulseira inteligente não possui GPS e seu aplicativo proprietário é confuso, fazendo com que a Garmin tivesse que relançar a aplicação para um modo computador de mesa, assim, deixando mais explícito ao usuário como funciona sua aplicação. Vale ressaltar que sua bateria dura por 1 ano sem precisar de recarga. Sua arquitetura é fechada.

A empresa Do Bem [27], conhecida por sua linha de alimentos, como sucos dos mais diversos sabores e barras de cereal, lançou seu primeiro dispositivo em forma de pulseira inteligente, que possui funções como monitoramento do sono, alerta de ociosidade, calorias, distância e passos. É uma pulseira de baixo custo, porém sua arquitetura é fechada, não sendo encontrado nenhum tipo de documentação para a mesma e é compatível apenas com o sistema operacional iOS [28].

Em [29], o autor descreveu seu modelo para a captação de sinais dos pacientes, utilizando equipamentos médicos portáteis, com foco em equipamento de eletrocardiograma, peso corporal, pressão arterial e medidas em geral, sendo processadas utilizando um servidor remoto. O modelo apresentado pelo autor permite alterações de ordem na visualização dos dados e processamento, de acordo com o nível de interação do paciente que realiza uma autoavaliação.

A pulseira inteligente *Embrace+* [30] é uma pulseira desenvolvida com escopo diferente de todas as outras tratadas neste trabalho. Sua função é produzir estímulos visuais como o nível de bateria, mensagens, *e-mail* e atualizações de redes sociais, provindas do *smartphone* do seu usuário. O usuário pode configurar a cor que deseja para representar uma determinada identificação de sinal provinda do *smartphone*.

Um projeto de sucesso no Distrito Federal (DF) é o *EasyGlic* [31], um bracelete inteligente flexível, que quando utilizado pelo usuário é capaz de detectar estados de hipoglicemia. O bracelete monitora a temperatura e umidade do corpo, com o objetivo de detectar quaisquer alterações compatíveis com um estado hiperglicêmico. O dispositivo trabalha com uma aplicação proprietária e tem um sistema eletrônico embarcado, que emite diversos alertas com vibração e luzes de *led*. Um detalhe muito importante é que a criação deste produto aconteceu na Universidade de Brasília (UnB). A Tabela 1 apresenta um resumo das soluções investigadas no estado da arte, uma vez que a lista de trabalhos é muito vasta.

Tabela 1 – Comparativo das soluções investigadas no estado da arte.

Tema / Dispositivo	Objetivo	Pulseira inteligente	Inserção de dados
Gerenciamento e cuidado de falhas cardíacas utilizando um sistema de telemedicina <i>online</i> [12].	Dados inseridos de modo manual; Guardar e gerenciar dados cardíacos; Sistema online.	Não	Manual
Mod. de acompanhamento de pacientes remotamente [13].	Acompanhamento de pacientes remotamente; Fluxo em tempo real.	Não	Manual
Aplicação WANDA [14].	Otimiza o monitoramento de pacientes; Auxilia a tomada de decisão médica.	Não	Automatizada
Aplicação móvel para prevenção de diabetes [15].	Avaliação de pontos por meio de teste de monofilamento; Aplicação móvel.	Não	Manual

CPRGlove [17]	Luva inteligente; Auxílio para realizar massagem cardíaca.	Não	Automatizada
---------------	---	-----	--------------

Tabela 2 – Comparativo das soluções investigadas no estado da arte que possuem pulseiras inteligentes

Tema / Dispositivo	Objetivo	Pulseira inteligente	Inserção de dados
Embrace+ [30]	Produz estímulos visuais; Nível de bateria; Mensagens, E-mail e atualizações de redes sociais.	Sim	Automatizado
EasyGlic [31]	Detecta estados de hipoglicemia; Monitora a temperatura e umidade do corpo.	Sim	Automatizado
Affectiva Qsensor [16].	Sensor para medir estímulos emocionais; Sensor de temperatura e movimentos.	Sim	Automatizado
MOTOACTV [18]	Monitora calorias queimadas; Frequência cardíaca; Possui GPS.	Sim	Automatizado
Fitbit [5]	Monitora contagem de passos; Distância percorrida; Tempo de descanso; Calorias; Elevação; Batimentos cardíacos e sono.	Sim	Automatizado

Sony [20]	Monitora o sono; Passos; Possui despertador.	Sim	Automatizado
Apple Jawbone [22]	Controle da dieta; Informações inseridas manualmente; Alerta de ociosidade.	Sim	Automatizado
Samsung Gear S [25]	Entrada para chip; Chamadas telefônicas e notificações.	Sim	Automatizado
Vivofit [26]	Possui monitor cardíaco; Gasto calórico; Bateria dura por 1 ano.	Sim	Automatizado
Do bem [27]	Monitora o sono; Alerta de ociosidade; Calorias; Distância e passos.	Sim	Automatizado

É possível notar que existem diversas soluções que não utilizam pulseiras inteligentes como exemplificado na Tabela 1. Na Tabela 2 pode-se ver as aplicações que possuem pulseiras inteligentes para monitoramento de sinais que são provenientes do corpo humano. Entretanto, comercialmente e cientificamente não existe a junção dessas duas áreas como propõe-se nesta dissertação. É possível notar que existem lacunas de pesquisa e contribuição científica, como o monitoramento de pacientes por diversos médicos, assunto tratado nesta dissertação.

Ao realizar o estudo sobre os trabalhos destacados no estado da arte, foi possível analisar diferentes abordagens, conceitos e metodologias utilizadas com vários escopos, como o meio *fitness* e o monitoramento de pacientes utilizando tecnologia vestível. Com tal estudo buscou-se imprimir muitas

características no modelo proposto nesta dissertação e preencher diversas lacunas existentes. Uma dessas lacunas é o desenvolvimento de uma aplicação para monitoramento médico utilizando pulseiras inteligentes e de baixo custo.

3. PROPOSTA DE ARQUITETURAS PARA SOLUÇÃO DE BAIXO CUSTO

A arquitetura das pulseiras inteligentes comerciais utilizada para o desenvolvimento deste trabalho é classificada como SISO. Muitas delas têm suas interfaces de programação fechadas, ou seja, não permitem o desenvolvimento de soluções por meio de sua plataforma. Logo com a sua plataforma fechada fica impossível o desenvolvimento de alguma solução utilizando o dispositivo vestível. O dispositivo utilizado neste trabalho [5] possui API própria, porém aberta, permitindo a troca de informações.

Em sua totalidade, as pulseiras inteligentes utilizam conexão sem fio. As pulseiras inteligentes investigadas e com maior número de vendas no mercado utilizam a tecnologia *Bluetooth* de Baixa Potência, do inglês, *Bluetooth Low-Energy (BLE)* [32], para se comunicar com o dispositivo que recebe os dados provindos do dispositivo vestível. O BLE fornece um consumo de energia muito baixo e sua cooperação com diversos dispositivos como telefones, *tablets*, relógios inteligentes e pulseiras inteligentes é muito recorrente por se adaptar muito bem ao ambiente. O BLE normalmente é utilizado em um modelo cliente-servidor, adquirindo um papel central para a ferramenta, ou seja, um cliente que necessita de dados se conecta ou acessa um ou diversos servidores que possuem dados para captação e transmissão [33].

Considerando otimizar a arquitetura nativa das pulseiras inteligentes, foi necessário o desenvolvimento de um concentrador de sinais, para que o mesmo possa suportar diversas pulseiras inteligentes. Assim, de uma arquitetura SISO, a plataforma poderia se transformar em uma arquitetura de múltiplas entradas única saída, do inglês *Multiple Input Single Output (MISO)* e também em uma arquitetura MIMO.

Com a evolução da arquitetura nativa das pulseiras inteligentes foi possível a confecção de uma aplicação móvel, que capta esses dados e os transfere para a aplicação móvel do *smartphone*. A aplicação desenvolvida vai monitorar diversos pacientes em ambiente hospitalar, facilitando e melhorando o serviço médico em hospitais e clínicas.

Apoiados na arquitetura otimizada, na seção 3.1 é apresentada a proposta de arquitetura de comunicação MIMO, eficiente e de baixo custo para pulseiras inteligentes, aplicadas ao atendimento ao paciente no pós-operatório. Em seguida, na seção 3.2 é apresentado o caso da substituição do *Raspberry Pi* como concentrador de sinais por um telefone celular e sua validação experimental.

3.1. PROPOSTA DE ARQUITETURA DE COMUNICAÇÃO MIMO EFICIENTE E DE BAIXO CUSTO PARA PULSEIRAS INTELIGENTES APLICADAS AO ATENDIMENTO AO PACIENTE NO PÓS-OPERATÓRIO

A arquitetura proposta é composta por um concentrador de sinais que recebe dados de M pulseiras inteligentes ao mesmo tempo e transmite-os à nuvem, permitindo seu acesso por meio de uma aplicação móvel somente por pessoal médico autorizado.

A arquitetura proposta foi validada conectando-se oito pulseiras inteligentes ao mesmo tempo ao concentrador de sinais. Uma vez que a arquitetura permite a inclusão de múltiplos concentradores, esta aplicação pode atender a qualquer número de pacientes.

3.1.1. Arquitetura MIMO para Pulseiras Inteligentes Comerciais

Nesta seção é apresentada uma arquitetura MIMO para compartilhar os dados e informações de M pulseiras inteligentes com N smartphones.

Para otimizar a arquitetura nativa das pulseiras inteligentes e conseguir expandir o acesso aos dados provindos da mesma, foi necessária a implementação de um concentrador de sinais. Neste caso, o concentrador de sinais foi elaborado utilizando um microcomputador *Raspberry Pi*. Para este trabalho utilizou-se o *Raspberry Pi Modelo B* [54].

Para que fosse possível a expansão da arquitetura, utilizou-se o concentrador de sinais para parear as pulseiras inteligentes. O concentrador de sinais serve como eixo principal para expansão da arquitetura. De acordo com a Figura 2, M pulseiras inteligentes com arquitetura SISO se conectam ao concentrador de sinais; os dados recebidos de M pulseiras inteligentes são encaminhadas para o servidor da API e os dados são passados diretamente para os *smartphones* N predefinidos, usando a aplicação móvel desenvolvida.

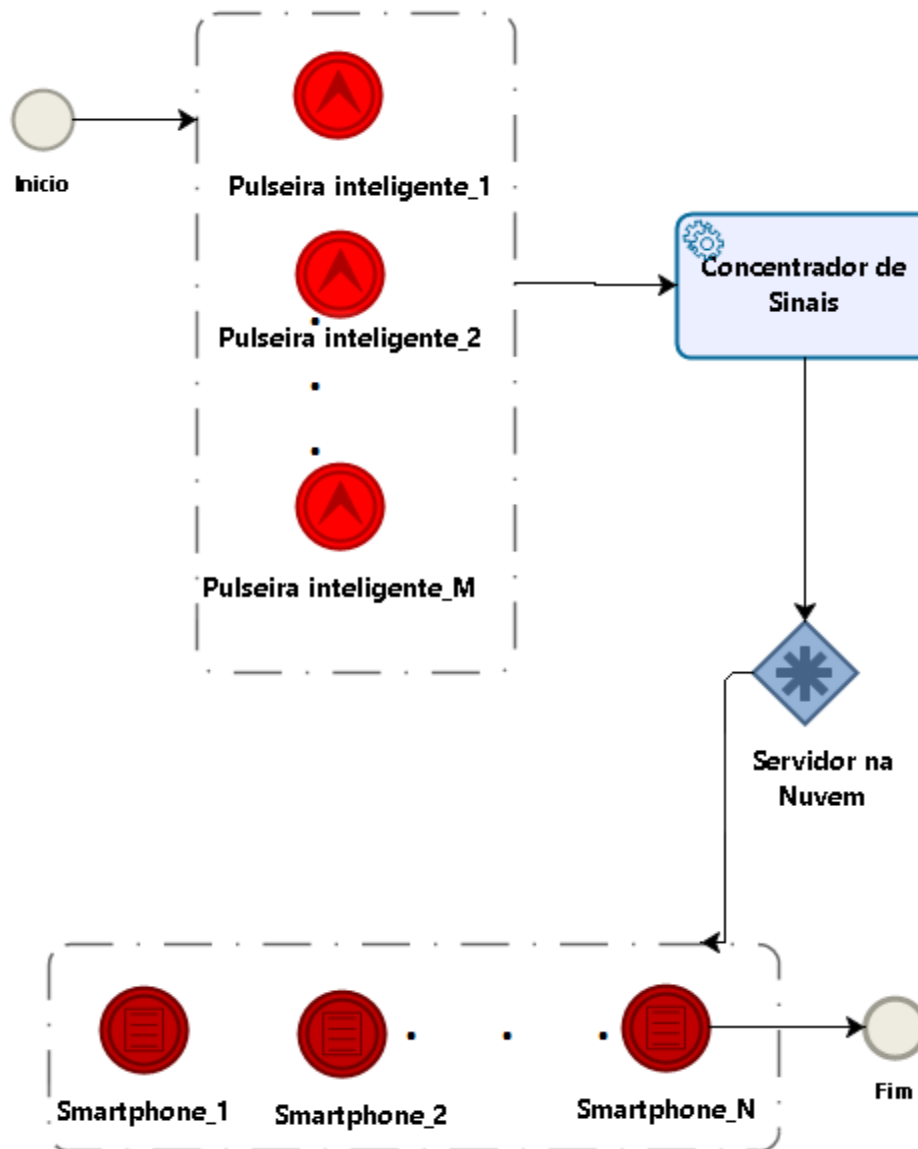


Figura 2 – Arquitetura MIMO utilizando o concentrador de sinais [11]

No aplicativo móvel multiplataforma proposto para *smartphones* são apresentados os dados captados pelas pulseiras inteligentes. Neste sentido, cada pulseira inteligente possui um perfil em sua API proprietário. Logo, esses dados são captados, armazenados e gerenciados pelo aplicativo móvel por meio padrão utilizando a API própria. Deve-se observar que o número máximo de conexões *Bluetooth* não simultâneas é 248 [34, 35], ou seja, pareadas, mas sem troca de informações e requisições ao servidor. Ainda conforme exemplificado na Figura 2, o servidor em nuvem recebe os dados enviados pela API a partir das pulseiras inteligentes e executa o processamento interno dessas informações, ou seja, o tráfego dessas informações até os *smartphones*. Os *smartphones* recebem os dados relativos aos pacientes no aplicativo instalado em seus dispositivos.

Esse tratamento é realizado pela própria API e foi desenvolvido utilizando o *framework Node.js* para facilitar sua usabilidade na aplicação. *Node.js* [36] é uma plataforma construída e apoiada ao *JavaScript* do *Google*. O *Node.js* utiliza um modelo de entrada e saída de dados não-bloqueado por eventos, tornando a plataforma leve e eficiente. A plataforma *Node.js* [36] contém um ecossistema de bibliotecas de código aberto, considerada uma das maiores do mundo. Utilizando *Node.js*, a plataforma fica altamente escalável, descartando as necessidades e complexidades de um sistema *multithreading*. O *Node.js* é muito utilizado quando existe um grande tráfego de dados, como é o caso deste trabalho, que envolve o monitoramento de pacientes por meio da aplicação móvel, que deverá receber uma grande carga de dados dos pacientes em tempo real, uma vez que deve manter os sinais capturados em constante atualização.

Para realizar uma conexão M:N, ou seja, M(pulseiras):N(smartphones) com diversos concentradores de sinal P , onde o conjunto de pulseiras inteligentes M_p estão conectadas ao concentrador de p -ésimo sinal, o total de pulseiras inteligentes é dado por:

$$M = \sum_{p=1}^P M_p \quad (1)$$

Portanto, para garantir a integridade das informações trocadas foram utilizadas técnicas de adaptação

de contexto associativo, ou seja, a informação usa as APIs das pulseiras inteligentes nativas, além da API desenvolvida para a aplicação, garantindo assim o gerenciamento completo de informações. A Figura 3 ilustra uma arquitetura onde temos diversos concentradores de sinal trabalhando, validando uma arquitetura M: N com concentradores de sinal P .

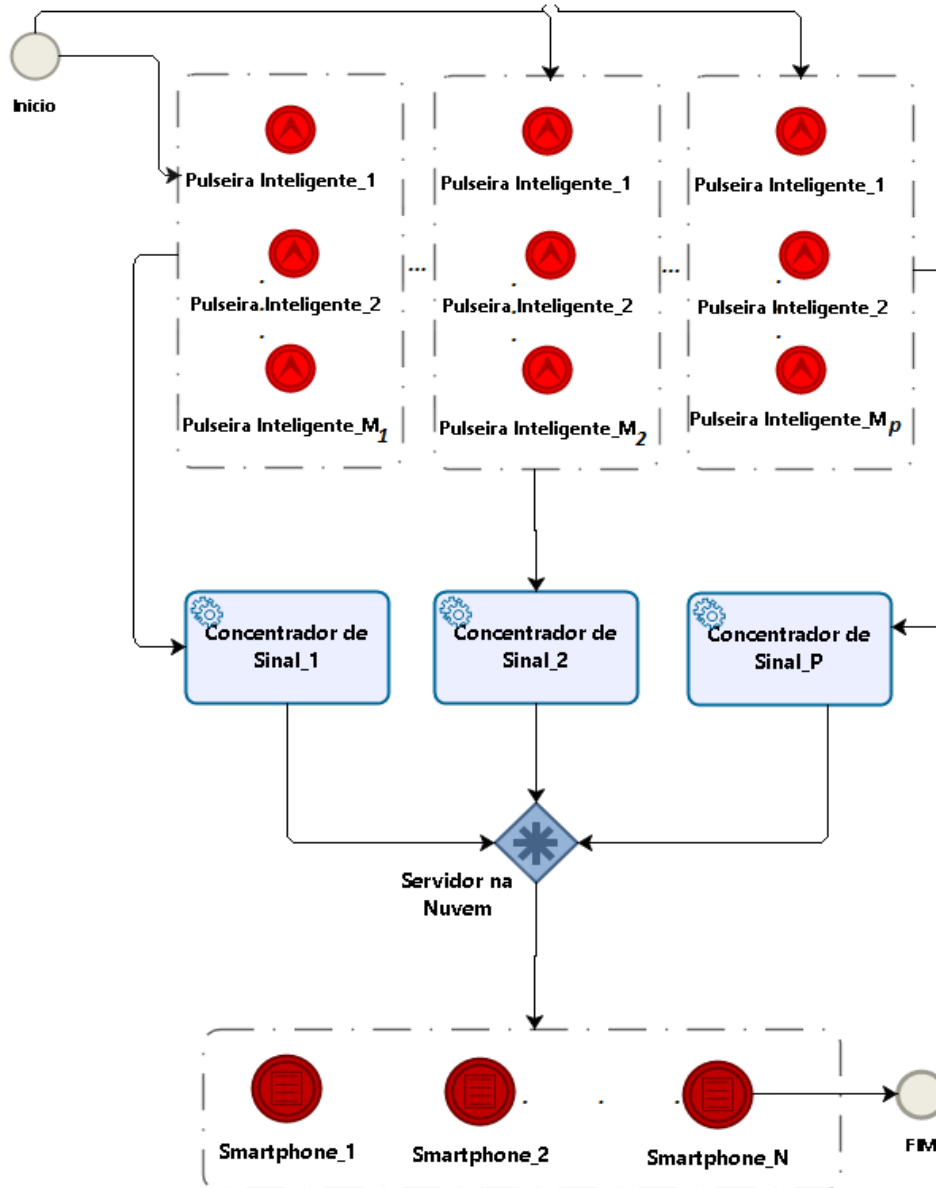


Figura 3 – Arquitetura MIMO com concentradores de sinal P para qualquer quantidade M de pulseiras inteligentes [11].

A Arquitetura ilustrada pela Figura 3 [11], exemplifica o caso de diversos concentradores de sinal trabalhando em pontos do hospital e com vários celulares. O Fluxo é de constante troca de informações, conforme mostra a Figura 3 [11], as pulseiras inteligentes são pareadas pelos concentradores de sinal que realiza a chamada para API proprietária dos dispositivos inteligentes, em seguida a API própria criada para a aplicação tratada neste trabalho busca as informações do servidor proprietário da pulseira, enviando os dados para os *smartphones* dos usuários.

3.1.2. Validação Experimental

Nesta seção, apresenta-se a validação experimental da arquitetura MIMO. A configuração experimental é dada por 6 pulseiras inteligentes com tecnologia BLE, um microcomputador *Raspberry Pi* Modelo B com módulo nativo *Bluetooth* e *Wireless* incorporado no microcomputador e vários *smartphones* com Sistema Operacional *Android* 5.0 ou superior.

A plataforma *Raspberry Pi* 3 Modelo B foi escolhida devido à sua portabilidade, baixo custo e suporte oferecido a uma grande quantidade de dispositivos. Devido à pequena quantidade de *hardware* disponível no laboratório da UnB, validou-se apenas a arquitetura apresentada na Figura 2. A arquitetura apresentada na Figura 3 deverá ser validada em um trabalho futuro. As pulseiras inteligentes usadas transmitem as seguintes informações ao concentrador de sinal: frequência cardíaca, número de passos, nível de elevação, número de calorias e detecção de sono.

Nos experimentos foi adotado o Salto Adaptativo de Frequência, do inglês *Adaptive Frequency Hopping (AFP)*, que tem a função de readaptar o mapa das frequências disponíveis, excluindo as rotas já ocupadas por outros sinais, como o sem fio. As transmissões não simultâneas das pulseiras inteligentes permitem o uso da capacidade máxima de 248 conexões *Bluetooth*, ou seja, pareadas, mas sem realizar requisições ao servidor proprietário.

Na Tabela 3, a primeira experiência observa a taxa de transferência de dados na saída do concentrador de sinal, variando o número de pulseiras inteligentes conectadas, isto é o número de pulseiras inteligentes. Conforme é mostrado na Tabela 3, de uma a seis pulseiras inteligentes conectadas, a taxa de transferência de dados varia de 392 *bits por segundo (bps)* a 1.32 *kilobits por segundo (kbps)*. Deve-se observar que há um aumento médio na taxa de dados em 200 bps para a inclusão de uma pulseira inteligente.

Tabela 3 – Taxa de transferência de dados

#Pulseiras Inteligentes	Taxa de Transferência de Dados
1	392 bps
2	527 bps
3	714 bps
4	915 bps
5	1.19 kbps
6	1.32 kbps

Em termos de aplicação em um cenário real, cada concentrador de sinal deve atender um número de pacientes que pode variar entre quatro e oito pacientes, segundo informações repassadas pelo médico conselheiro que apoia a equipe. Além disso, de acordo com experiências realizadas nesta pesquisa, o alcance das conexões *Bluetooth* tem um máximo de 10 metros, levando em conta todos os diferenciais do ambiente. O alcance pode ser otimizado uma vez que não há obstáculos para o sinal ser distribuído, porém, tal fator é bem diferente das alas hospitalares que apresentam um ambiente com diversos obstáculos como paredes, outros dispositivos emitindo sinais e o grande número de pessoas se movimentando no local.

Como o protocolo *Bluetooth* opera na faixa de 2,4 GHz, e sabendo que o índice de modulação para o protocolo está sempre entre os valores 0,28 e 0,35 [37, 38], obtem-se a margem onde a intensidade do sinal está concentrada no espectro de frequência dos sinais transmitidos.

Nesse sentido, como as taxas são sujeitas a mudanças, o regime de transmissão está sujeito a um aumento no volume de informações não transmitidas. Com a finalidade de homogeneidade da pesquisa, todas as pulseiras inteligentes foram sincronizadas com o servidor antes de serem testadas.

Para que não ocorra uma troca de dados com informações que não são interessantes aos testes de tráfego de informação.

Na Tabela 4, são apresentados os resultados do uso médio da CPU do concentrador de sinal, aumentando o número de pulseiras inteligentes conectadas. Durante o emparelhamento das pulseiras inteligentes com o concentrador de sinal, é necessário um processamento computacional alto. No entanto, após um tempo de pareamento das pulseiras, o processamento computacional é drasticamente reduzido. Nota-se que, para seis pulseiras inteligentes, menos de 20% da capacidade computacional do *Raspberry Pi* é utilizada.

Tabela 4 – Percentual de processamento computacional do concentrador de sinal variando o número de pulseiras inteligentes

#Pulseiras Inteligentes	Uso da CPU durante o pareamento	Uso da CPU após o pareamento
1	6 %	1 %
2	11 %	4 %
3	13 %	4 %
4	14 %	4 %
5	16 %	5 %
6	16 %	5 %

Logo, a arquitetura de comunicação MIMO eficiente e de baixo custo para as pulseiras inteligentes aplicadas aos cuidados médicos, foi considerada uma arquitetura que conecte qualquer quantidade N de pulseiras inteligentes em uma ala médica tradicional. A arquitetura proposta é estendida no caso de qualquer quantidade M de smartphones por meio da incorporação de concentradores de sinal P . Realizando assim uma arquitetura de múltiplos concentradores distribuindo os sinais das pulseiras para a aplicação móvel.

3.2 SUBSTITUIÇÃO DO CONCENTRADOR POR UM SMARTPHONE

Levando em conta toda evolução tratada na seção 3.1, esta seção explora o tratamento da substituição do microcomputador *Raspberry Pi* por um *smartphone* com conexão *Bluetooth*.

A arquitetura nativa das pulseiras inteligentes utilizadas para captação dos sinais dos pacientes é SISO, ou seja, permite apenas uma conexão, apenas para 1 *smartphone*, caracterizando 1:1.

A solução encontrada foi a de desenvolver um concentrador de sinais que se conecta ao servidor para transmissão dos dados de N pulseiras conectadas ao dispositivo. Escolheu-se a utilização do *Raspberry Pi* como primeira opção e testou-se a utilização de um *smartphone* para substituição do mesmo. As seções 3.2.1 e 3.2.2 investigam os casos da utilização do *Raspberry Pi* e do *smartphone* bem como a possível troca desses dispositivos.

3.2.1. Concentrador de Sinais utilizando Raspberry Pi

Conforme introduzido na seção 3.2, a primeira opção considerada foi a utilização de algum microcomputador viável e de baixo custo no mercado, neste caso foi optado pelo *Raspberry Pi*. O mesmo tem baixo custo e o tamanho de um cartão de crédito, o que facilita sua utilização em lugares com pouco espaço, além de ser muito discreto. Nele, podem ser conectados diversos dispositivos, utilizando a tecnologia *Bluetooth* e *Wi-Fi*. O *Raspberry Pi* foi criado com o objetivo de ser utilizado em projetos educacionais para adultos e crianças, focados na área de computação e assuntos relacionados ao meio tecnológico. Atualmente, existem variados modelos [54], que variam sua arquitetura e processamento, porém, esta dissertação traz detalhes e informações apenas do modelo utilizado no desenvolvimento deste trabalho. A escolha do microcomputador *Raspberry Pi* foi devido ao seu tamanho reduzido, baixo consumo energético, vasta comunidade e documentação disponíveis em diferentes meios.

Após a configuração do microcomputador *Raspberry Pi*, durante a configuração foi instalado no mesmo o sistema operacional oficial, o *Raspbian OS*. A instalação do sistema ocorreu de modo que

ele ficasse operacional para sua utilização. Em seguida instalou-se um aplicativo chamado *Galileo* [39], que é um utilitário de código aberto desenvolvido em linguagem de programação *Python*, para sincronizar de forma segura a pulseira inteligente. Neste sentido, conectou-se M pulseiras inteligentes com o concentrador de sinais utilizando o *Galileo*, colaborou-se com o código inserindo uma pausa assim que as pulseiras são sincronizadas. Na aplicação móvel tem-se o resgate dos dados por meio do ID do usuário, com sincronismo de 1 em 1 minuto que é o tempo mínimo de resgate. É possível ver a arquitetura elaborada para o funcionamento do concentrador de sinais na Figura 4.

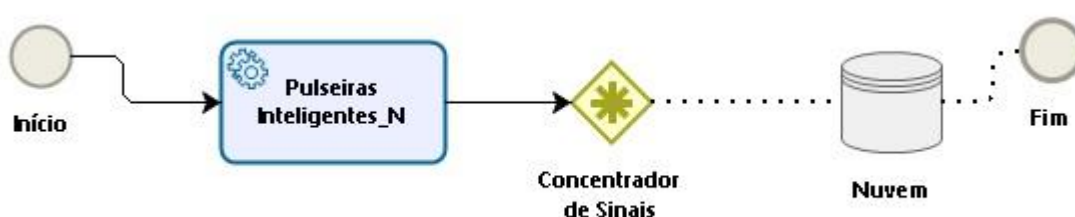


Figura 4: Arquitetura desenvolvida para funcionamento do concentrador de sinais

Atualmente o concentrador é usado com o microcomputador *Raspberry Pi*, por sua segurança, usabilidade, além do baixo custo.

3.2.2. Concentrador de Sinais utilizando um smartphone

Após o desenvolvimento do concentrador de sinais utilizando um microcomputador *Raspberry Pi* ter sido realizado com sucesso, optou-se por tentar realizar o experimento com um *smartphone*, uma vez o dispositivo processa um sistema operacional comum e de código aberto, como o *Android* [40]. O caso é a utilização de um celular, com o sistema operacional *Android*, classificado como um *smartphone*, por ser um telefone inteligente, tendo por objetivo o de simular a mesma dinâmica que tem-se com o concentrador de sinais utilizando o *Raspberry Pi* com um celular.

Com a conexão *Bluetooth* foi possível realizar o pareamento de diversas pulseiras inteligentes com o *smartphone*, porém, o celular não suporta uma rotina de conexão para armazenagem e mudança dos

dados captados pelas pulseiras, servindo apenas mesmo para parear o processo. Entretanto apenas parear as pulseiras faz com que as informações mantidas nas pulseiras fiquem apenas alocadas em uma determinada área o que não facilita sua utilização em ambientes médicos.

Para que a proposta de tornar um *smartphone* um concentrador de sinais capaz de captar sinais verificou-se que seria necessário trocar a ROM do *Android*. Logo, com outra versão de *Android* sendo executada foi possível tentar novamente a rotina, que acabou não se portando como deveria.

A arquitetura presente em *smartphones* permite a manutenção e alteração de várias características, porém, ainda não serão apresentadas soluções para este problema, uma vez que a utilização do *Raspberry Pi*, como concentrador de sinais tem menor custo e segurança para utilização, além de estar estabilizado, pretende-se assim, a transformação e investigação deste caso. Também pretende-se realizar a troca do concentrador de sinais por um *Raspberry Pi Zero*, pelo seu baixo custo e grande usabilidade. Ambos serão trabalhos futuros.

4. DESENVOLVIMENTO DE APLICATIVO PARA A ARQUITETURA MIMO PROPOSTA

Este capítulo apresenta informações referentes ao desenvolvimento de um aplicativo para variadas pulseiras utilizando o concentrador de sinais na arquitetura M: N ou 1: N.

Segundo [41], o *software* é um dos mais importantes expoentes da tecnologia no cenário atual. Nos últimos anos, os *softwares* evoluíram de ferramentas de análise de informações para ferramentas de resolução de problemas complexos e/ou banais. Ainda segundo [41], existem alguns itens que devem ser levados em conta para o desenvolvimento de uma aplicação, tais como usabilidade, confiabilidade, funcionalidade, manutenibilidade e portabilidade. Por meio desses itens é possível desenvolver um aplicativo abrangendo estes aspectos. Para exemplificar melhor o significado desses itens, empregaram-se as definições de [41]:

Usabilidade: conjunto de características que ilustra a utilização da aplicação, bem como a validação de cada uso por diversos usuários;

Confiabilidade: conjunto de características que ilustra a capacidade da aplicação de manter o nível sob variadas condições por um determinado tempo;

Funcionalidade: conjunto de características que ilustra as funções e suas propriedades na aplicação;

Manutenibilidade: conjunto de características que ilustra a facilidade de se realizar modificações na aplicação;

Portabilidade: conjunto de características que exemplificam a mobilidade da aplicação em diversos ambientes.

Levando-se em conta as definições citados acima a aplicação teve seu desenvolvimento iniciado. Neste sentido todos os itens foram aplicados ao desenvolvimento da aplicação móvel para se manter

melhor utilizada por parte de seus usuários e facilitando o desenvolvedor através da manutenibilidade em sua estrutura.

O primeiro passo para o início do desenvolvimento da aplicação apresentada nesta dissertação foi o levantamento de requisitos e modelagem da aplicação. Por fim ocorreu a escolha da plataforma de desenvolvimento que a aplicação iria ser implementada. Atualmente existem diversos meios para desenvolvimento de aplicações móveis. Estes são intrinsicamente associados aos sistemas operacionais dos dispositivos e equipamentos utilizados. Levando-se em conta que, de cada 10 *smartphones* vendidos, 9 possuem o Sistema Operacional (SO) *Android* [21], a aplicação foi desenvolvida para ser executada neste SO. Além disso, outros fatores implicaram na escolha da plataforma para o desenvolvimento do projeto, tais como:

- O ambiente de desenvolvimento é compatível com a maioria das plataformas atuais;
- A comunidade de desenvolvimento é uma das maiores do mundo;
- É possível realizar associações com diversas IDEs de uso generalizado.

Neste sentido, o custo para o desenvolvimento torna-se muito menor, tanto de *hardware* como de *software*. O desenvolvimento direcionado para outras plataformas como a iOS [28] da Apple [28] é necessário equipamento com um custo muito elevado para os atuais padrões de pesquisa. Para o desenvolvimento do aplicativo diversos *frameworks* de desenvolvimento para área médica foram estudados, tais como *Monaca UI* [43], *PhoneGap* [45], *Meteor* [49], *Ionic* [50], *Bootstrap* [51], *Angular.js* [52] e *jQuery* [53].

Para desenvolver uma aplicação com o objetivo desta dissertação, foi necessária a utilização de diversas bibliotecas, fornecidas pelo Kit de desenvolvimento de *Software Android*, do inglês *Android Software Development Kit* (Android SDK) [42] e pelo *framework* *Monaca UI* [43] que realiza o desenvolvimento de aplicações híbridas, utilizando HTML5 [44] juntamente com o *PhoneGap/Cordova* [45,46], de modo simples e fácil. A comunidade de desenvolvimento é muito ampla o que facilita a implementação da solução.

Um grande desafio foi a articulação da arquitetura SISO nativa em pulseiras inteligentes comerciais, como as utilizadas na aplicação desenvolvida. Para o desenvolvimento do trabalho foram utilizadas pulseiras inteligentes da marca Fitbit [4]. Em primeiro momento utilizou-se o modelo *Fitbit Charge HR* e, posteriormente, o modelo *Fitbit Charge 2*. A Tabela 5 mostra um comparativo entre os dois modelos de pulseiras inteligentes utilizadas.

Tabela 5 – Comparativo entre os modelos utilizados de pulseiras inteligentes

Sinais	Fitbit Charge HR	Fitbit Charge 2
Frequência cardíaca	X	X
Pedômetro	X	X
Elevação	X	X
Calorias	X	X
Distância	X	X
Bateria superior a 5 dias	X	X
<i>Bluetooth Low-Energy</i>	X	X

Os dois modelos de pulseiras inteligentes utilizados no processo de desenvolvimento são praticamente iguais, como mostra a Tabela 5, o que difere os modelos é o *design*, conforme mostra a Figura 5. A escolha dessas pulseiras se deu por meio de estudos das diferentes plataformas encontradas no mercado de pulseiras inteligentes, onde a *Fitbit* se mostrou a mais coerente com a ideia desta dissertação, além de contar com uma API própria, o que auxilia o desenvolvimento de aplicações. Para aplicação apresentada nesta dissertação foi desenvolvida uma API própria, para busca de informações dos servidores da FitBit e retorno destas informações para o aplicativo móvel desenvolvido.



Figura 5 – Modelos de pulseiras inteligentes utilizados nesta dissertação. [5]

Em comparação com outras pulseiras inteligentes do mercado [7, 9, 10, 20, 25, 26, 27, 31] a *Fitbit* [5] se mostrou o melhor dispositivo para desenvolvimento, uma vez que possui uma boa documentação e uma ótima API [19] para utilização. Os outros dispositivos não apresentam nenhuma documentação para desenvolvimento de soluções.

A elaboração do protótipo, após inúmeras reuniões e interações com o médico colaborador da pesquisa, foi decidido e levantado os atributos necessários para que a aplicação desenvolvida tivesse êxito. A metodologia escolhida para o desenvolvimento da aplicação foi o *eXtreme Programming* (XP). O XP facilita o desenvolvimento quando é implementado por meio de reuniões e validações durante diversas iterações.

Utilizando sinais captados por pulseiras inteligentes foi possível analisar e desenvolver uma aplicação que englobasse a ideia inicial. Além disso, foi possível analisar o aplicativo desenvolvido com situações novas e diversas, proporcionando uma troca dinâmica de informações e articulação de estudos para o desenvolvimento da aplicação móvel e, principalmente, na captação de questões físicas do usuário e do paciente. A estrutura de funcionamento da aplicação móvel proposta é detalhada na seção 4.1 e a aplicação móvel na seção 4.2.

4.1. ESTRUTURA E FUNCIONAMENTO DA APLICAÇÃO PROPOSTA

Para o início do desenvolvimento da aplicação foram levantados os requisitos e modelada a aplicação, levando-se em conta o fluxo desejado e alguns extras inseridos como função de lembrete para pacientes, criação de grupos e equipes entre médicos e pacientes. Neste sentido, a estrutura inicial da lógica da aplicação móvel são os arquivos *Javascript* [47] que estão estruturados em três partes: o controlador de páginas (*controllers.js*); serviços (*services.js*); e aplicativo (*app.js*).

O arquivo que controla o aplicativo (*app.js*) é responsável por chamar o controlador referente a cada página HTML exibida, presentes no controlador de páginas. Cada controlador de página é responsável por receber dados e *inputs* do usuário, processando-os e realizando as funções do aplicativo. O arquivo de serviços contém funções e variáveis de estado utilizadas por várias páginas do aplicativo, separando suas estruturas dos controladores específicos de cada arquivo HTML. São usadas, também, bibliotecas de desenvolvimento *Javascript* auxiliares com funcionalidades já implementadas. As bibliotecas auxiliares integradas são a *jQuery*, que possui funcionalidades extras de interação com *Javascript* e a biblioteca *Chart.js*, que é responsável pela exibição de gráficos, além de utilizar máscaras para *inputs* de dados de cadastro do usuário.

O fluxo de dados do aplicativo é realizado com o auxílio de uma API própria, hospedada em um servidor no serviço *Digital Ocean* [48]. A API do aplicativo foi desenvolvida em *Node.js*, com o módulo auxiliar *Express*. A documentação da API pode ser vista no Apêndice desta dissertação.

O *software* executado no servidor gerencia um banco de dados relacional de Linguagem de Consulta Estruturada, do inglês, *Structured Query Language (SQL)*, encarregado de armazenar os dados das pulseiras inteligentes e internos do aplicativo, tais como dados cadastrais e dados de usuários. Para ilustrar da melhor forma como o banco de dados da aplicação móvel está estruturado a Figura 6, ilustra o diagrama da estrutura da base de dados. Neste diagrama é possível ver todas as tabelas que são trabalhadas no interior da aplicação.

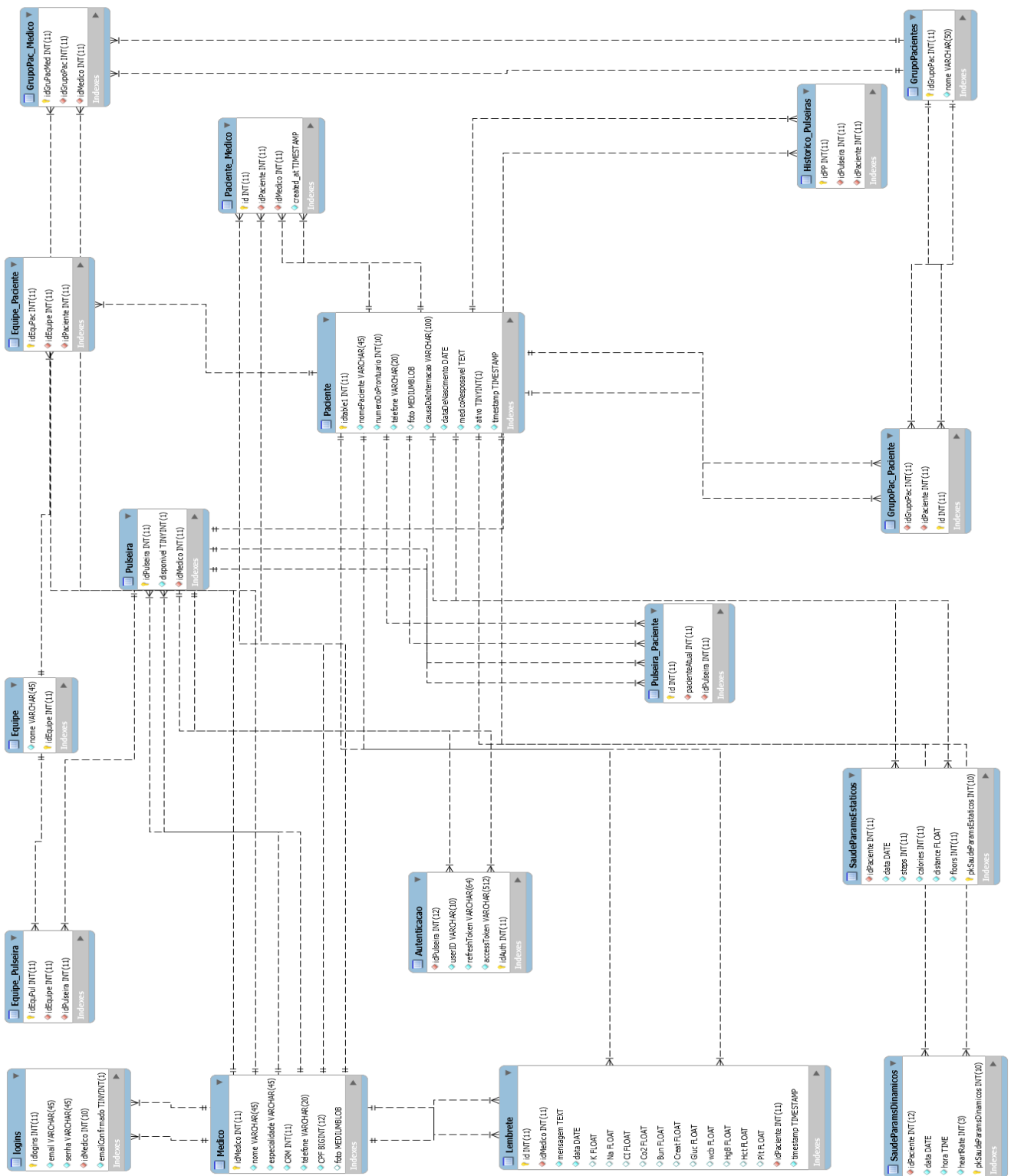


Figura 6 – Diagrama da estrutura da base de dados

Por meio de requisições HTTP, o aplicativo se comunica com sua API, recebendo e transmitindo informações necessárias à infraestrutura do programa. As requisições e transmissões de dados são realizadas a partir de funções *Javascript* próprias, sendo que os controladores também são encarregados de processar e exibir os dados como necessário.

Os dados de saúde provenientes da pulseira inteligente são tratados de maneira distinta do resto dos dados do aplicativo, dado que é obrigatório o uso da API proprietária da pulseira inteligente para adquirir informações de cada pulseira. Dessa forma, o servidor é encarregado de fazer requisições periódicas ao servidor próprio da pulseira, armazenando e transmitindo os dados das pulseiras de acordo com o comando do aplicativo.

Para dados dinâmicos, como por exemplo, frequência cardíaca, são feitas requisições pelo servidor a cada 1 minuto. Já para os dados estáticos, como consumo calórico, passos e elevação, são feitas requisições pelo servidor de 15 em 15 minutos. Internamente ao aplicativo, os dados de saúde são relacionados aos respectivos pacientes, por meio do banco de dados, no qual os dados de cada pulseira inteligente são associados ao paciente ao qual a pulseira inteligente foi especificamente atribuída. Desta forma, o aplicativo móvel faz requisições a partir da identidade de cada paciente, abstraindo para o servidor os dados proprietários da pulseira inteligente. Os dados adquiridos são vistos no perfil de cada paciente na aplicação móvel. A seção 4.2 detalha a aplicação móvel desenvolvida nesta dissertação.

4.2. APLICAÇÃO MÓVEL

Nesta seção será apresentada a modelagem da aplicação móvel desenvolvida nesta dissertação e a solução desenvolvida detalhadamente. A aplicação móvel tem como foco principal o monitoramento e o acompanhamento de pacientes, por meio da utilização de pulseiras inteligentes. Entretanto, a aplicação móvel, contém diversas ferramentas que constituem a aplicação como um todo.

Para o desenvolvimento da aplicação móvel foi composto um modelo inicial para sua estrutura. Tal

modelo foi criado através de necessidades já explicitadas nesta dissertação. A Figura 7 apresenta um diagrama simplificado das telas que compõe a aplicação móvel e como elas se relacionam uma com a outra. Na Figura 7 não são previstas as telas de login.

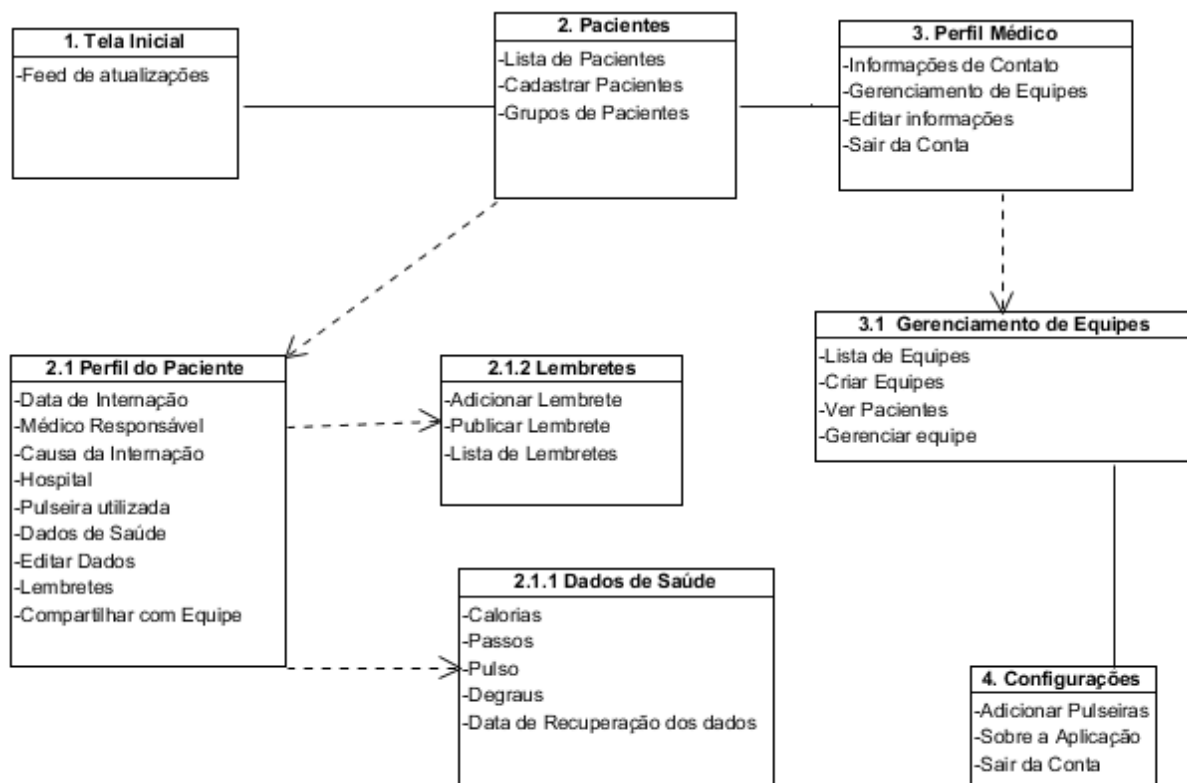


Figura 7 – Diagrama simplificado das telas da aplicação móvel

A Figura 7 ilustra as diversas telas que a aplicação possui. As linhas tracejadas significam que uma herança entre as conexões, como é o caso da tela (2) e (3) e suas heranças.

Na Tela Inicial (1) é possível visualizar o fluxo de atualizações dos pacientes do usuário, além de informações como lembretes e equipes.

Na tela Pacientes (2) é possível ter acesso a lista de pacientes inseridos pelo médico na aplicação móvel, além de conter recursos como cadastrar pacientes e fazer grupos de pacientes. Neste caso,

existe a opção de o médico fazer grupos onde é permitido inserir pacientes selecionados pelo mesmo.

Na tela Perfil do Paciente (2.1), é possível verificar a data da internação do paciente, médico responsável, causa da internação, hospital que o paciente se encontra, anexar e desanexar pulseiras inteligentes de monitoramento ao paciente, dados de saúde onde é possível verificar gráficos refletindo os sinais captados pela pulseira inteligente, edição de dados, lembretes onde é possível inserir anotações e informações médicas sobre o paciente e por fim o compartilhamento com equipes onde o médico pode criar uma equipe e compartilhar seus pacientes com outros médicos que cuidam do mesmo caso.

Na tela Dados de Saúde (2.1.1) é possível ver o monitoramento dos dados provindos da pulseira inteligente para cada paciente, dados como passos, calorias, pulso e degraus estão contidos nesta tela.

Na tela Perfil Médico (3) se encontram as informações sobre o médico, além disso, é possível gerenciar equipes criadas pelo médico, editar informações do cadastro pessoal do médico e por fim sair da conta na aplicação móvel.

O Gerenciamento de Equipes (3.1) permite o médico usuário da aplicação móvel criar, editar, excluir suas equipes tendo um gerenciamento completo de suas equipes.

Por fim na tela Configurações (4) é possível registrar novas pulseiras inteligentes na aplicação, além de conter um sobre o aplicativo e a opção de sair da conta.

Para ajudar no processo de desenvolvimento da aplicação foi desenvolvido um diagrama modelo de caso de uso da aplicação, este ilustrado na Figura 8. No diagrama de caso de uso é possível ver as ações que o ator médico pode realizar na aplicação. Além disso, é possível ver as ações que o ator paciente está associado.

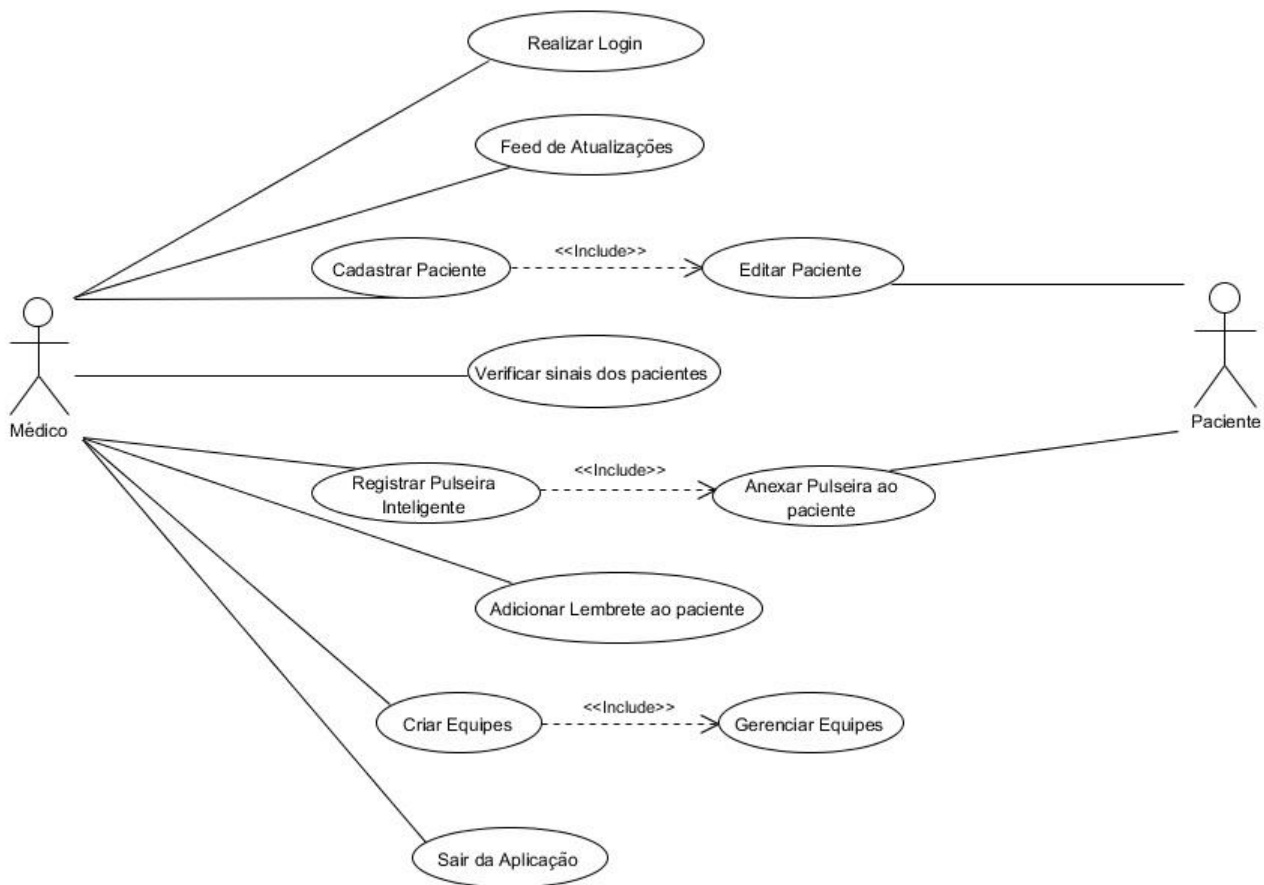


Figura 8 – Diagrama de Caso de Uso modelo da aplicação

Conforme a Figura 8 exemplifica o usuário médico pode realizar diversas ações como atualização e verificação do *feed* de atualizações, cadastrar paciente, editar paciente por meio das informações cedidas pelo paciente, também é possível registrar e anexar pulseiras inteligentes ao perfil de paciente para monitoramento, além de ser possível a verificação dos sinais vitais dos pacientes, adicionar lembretes, criar equipes de pacientes e gerenciar as mesmas.

A partir de diagramas e modelagem da aplicação foi desenvolvido a aplicação móvel, visando capturar os sinais das pulseiras inteligentes e distribuí-las por meio da aplicação móvel utilizada por médicos autorizados.

Por questão de segurança, na tela inicial da aplicação é necessário informar o *login* e a senha de

acesso, como exemplificado na Figura 9. Caso o usuário tenha errado alguma das informações inseridas a aplicação exibe uma notificação de erro ao usuário.

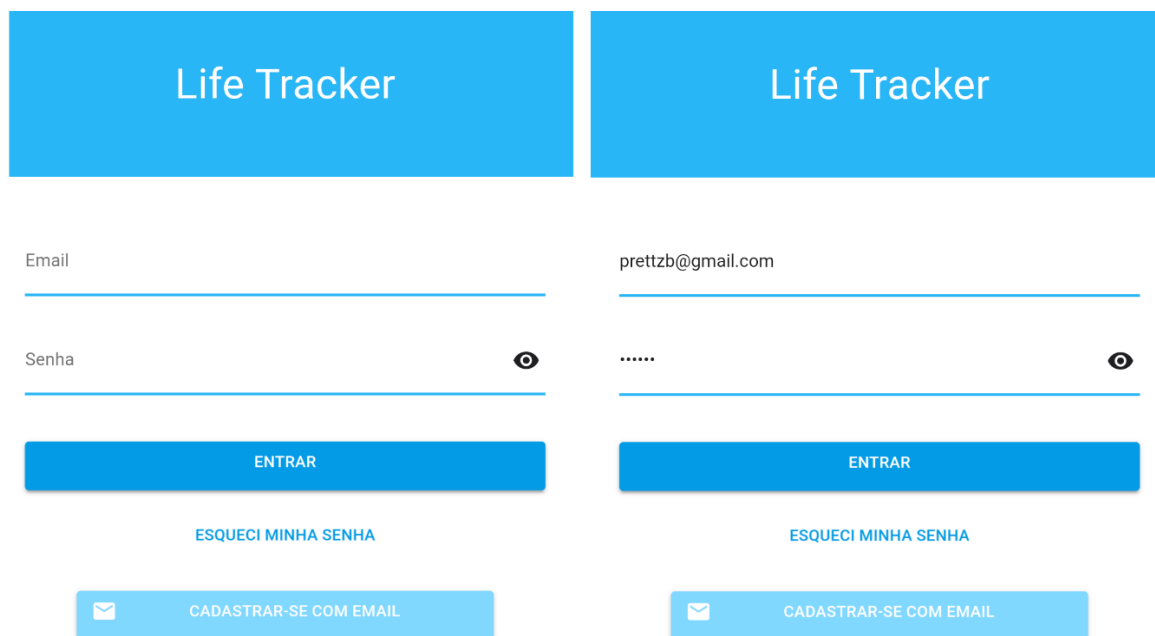


Figura 9 – Tela de *login* da aplicação

Em primeiro momento, a aplicação é iniciada com a tela de *login*, ilustrada pela Figura 9. Nesta tela o médico usuário da aplicação vai poder inserir seus dados e prosseguir para a aplicação. Existe o caso do usuário novo, que não tenha cadastro. Ele pode realizar o cadastro por meio do botão “Cadastrar-se com E-mail”, o mesmo vai poder inserir seu *e-mail* e dados e ter seu primeiro acesso à aplicação.

Caso o usuário esqueça a senha, é possível recuperar a mesma através da tela inicial (Figura 9) existe um botão chamado “ESQUECI MINHA SENHA”. Clicando nesse botão a aplicação exibe uma tela para inserção do *e-mail* do usuário que, caso inserido corretamente, permite, por meio de um *e-mail*

recebido, refazer a senha e validar seu acesso à aplicação.

Ao acessar a aplicação, a primeira tela que aparece é a o *feed* de informações. Nessa tela a aplicação traz apenas informativos e lembretes sobre pacientes. Caso o usuário esteja entrando pela primeira vez na aplicação, o *feed* será encontrado vazio, apenas com o informativo escrito “Sem atualizações de pacientes no momento. Pode-se adicionar um paciente na aba “Pacientes”, como exemplifica a Figura 10.

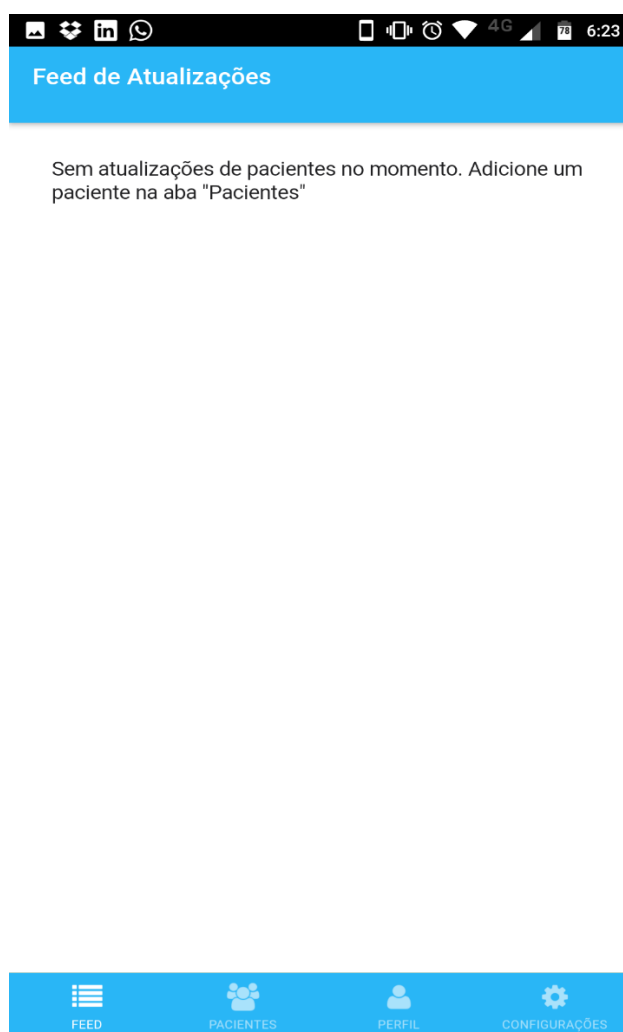


Figura 10 – *Feed* inicial da aplicação

Em seguida o usuário tem acesso à opção de ver seus pacientes, clicando no rodapé da aplicação, conforme ilustra a Figura 11. O mesmo tem a opção de ver uma lista de pacientes monitorados pela aplicação. Caso não tenha nenhum paciente listado é possível adicionar através da ferramenta.

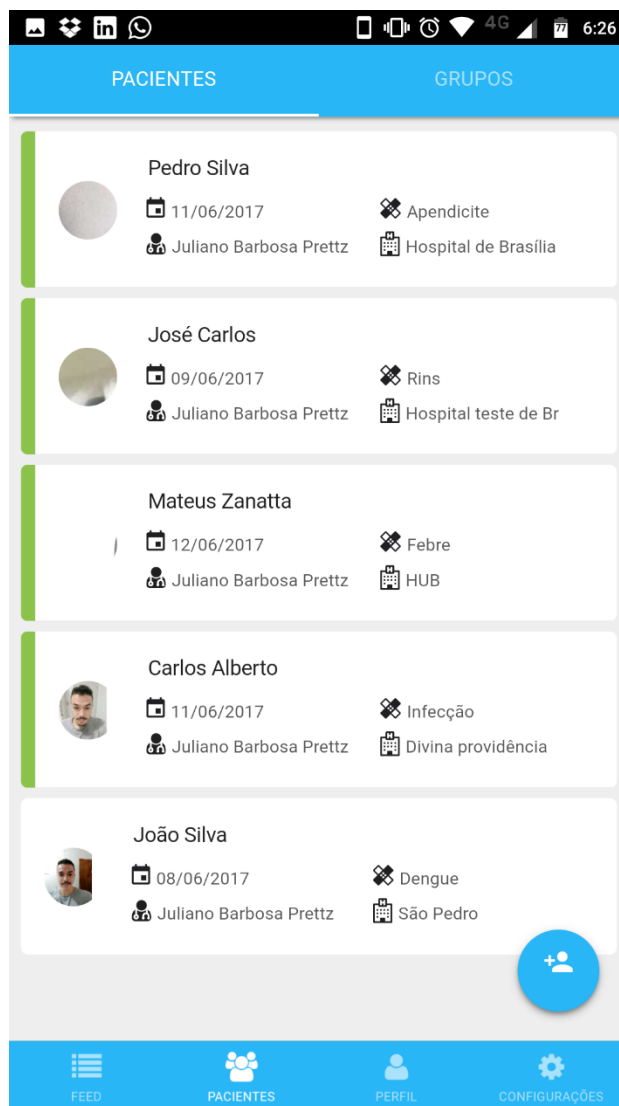


Figura 11 – Lista de pacientes na aplicação

O usuário também pode adicionar pacientes por meio do botão para adicionar pacientes na aplicação, conforme mostra a Figura 12.

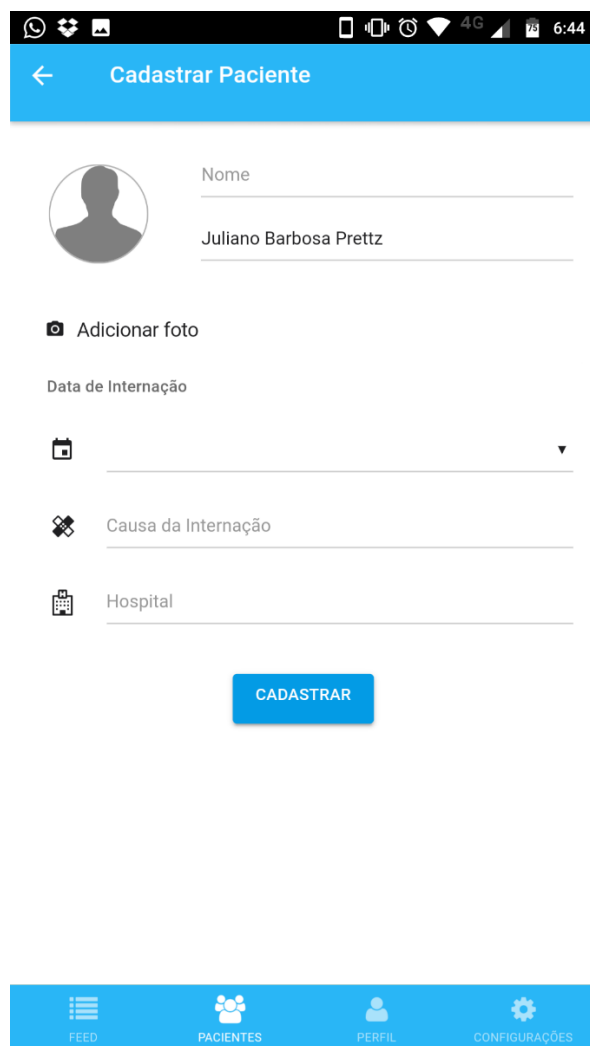


Figura 12 – Cadastro de Pacientes

Depois de ter já cadastrado um paciente, é possível ver uma série de informações no seu perfil, como data de internação do paciente, médico responsável pelo paciente, causa da internação, hospital que o paciente se encontra e pulseira utilizada pelo paciente.

Além disso, o perfil do paciente possui diversas informações, tais como dados de saúde onde os sinais captados pela pulseira são utilizados, lembretes onde é possível inserir informações para determinados pacientes, além de uma opção para compartilhar as informações do paciente com outros médicos, por meio de equipes médicas. A Figura 13 ilustra o perfil do paciente:

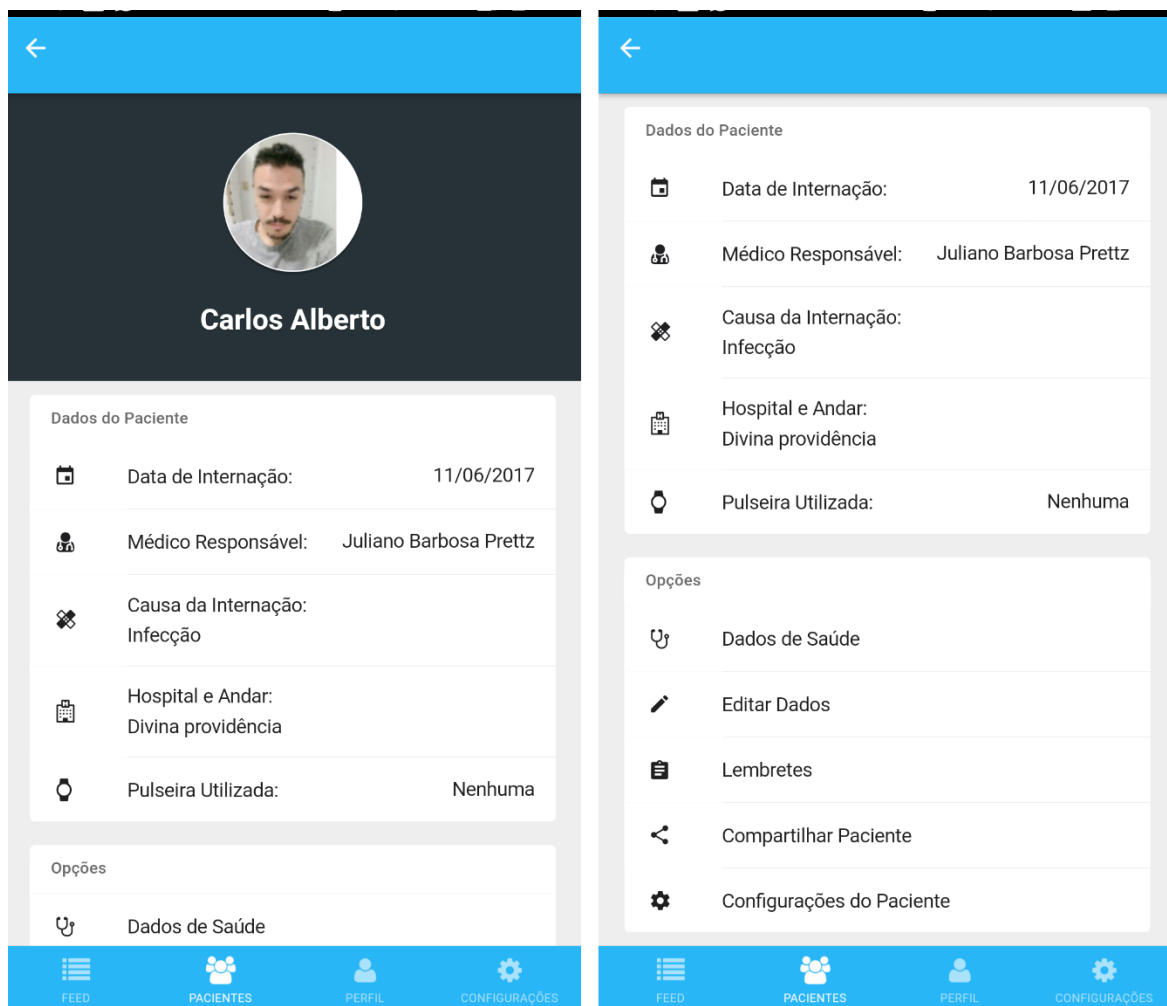


Figura 13 – Perfil do Paciente

Nas opções contidas no perfil de cada paciente, os “Dados de Saúde” mostram os gráficos dos sinais capturados pelas pulseiras inteligentes, onde é possível monitorar o paciente e verificar diversos sinais do mesmo, como mostra a Figura 14. Além disso, nota-se que, no primeiro gráfico tem-se a contagem de degraus que o usuário caminhou no dia, ou seja, sua elevação por meio da caminhada. No segundo gráfico é ilustrada a contagem de passos do paciente. Por fim, a Figura 14 ilustra as calorias gastas pelo usuário. Todos estes sinais são captados pela pulseira inteligente.



Figura 14 – Gráficos gerados pela aplicação através dos dados da pulseira inteligente

As informações ficam guardadas no banco de dados, o que permite um histórico para auxiliar os médicos nos seus atendimentos.

Um ponto importante é a opção “Lembrete”. Nesta opção o médico pode inserir diversos lembretes e informações técnicas sobre o paciente que podem ou não ser compartilhadas com outros médicos autorizados. A Figura 15 ilustra a tela de lembretes.

Ainda dentro do cadastro do paciente, o médico pode anexar ao mesmo a pulseira que estiver disponível como ilustra a Figura 16. Assim, a aplicação tem a lista das pulseiras disponíveis e o médico deve anexar ao paciente para iniciar o monitoramento.

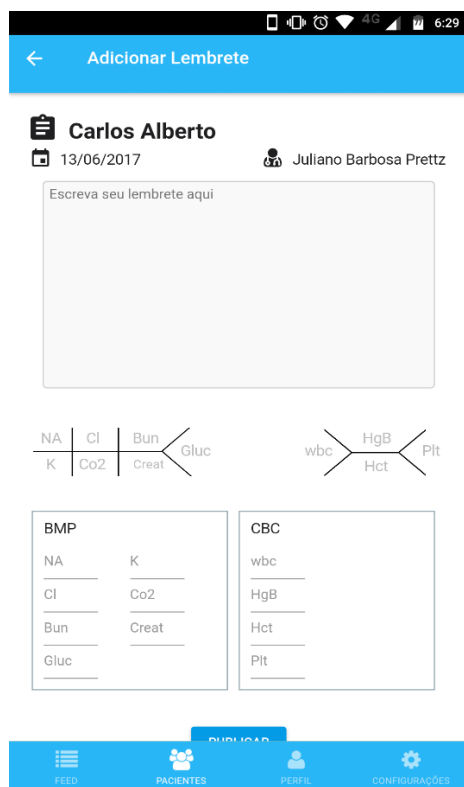


Figura 15 – Lembrete do paciente na aplicação



Figura 16 – Anexo de pulseira inteligente a paciente

Os lembretes podem ser acessados por outros médicos, mas para isso os profissionais da saúde devem estar na mesma equipe. A aplicação fornece um meio de criação de equipe, onde o usuário pode criar diversas equipes e nelas adicionar seus pacientes, semelhante ao grupo de pacientes. A diferença que nessas equipes o médico pode ser ou não o criador e ter acesso a diversos pacientes. A tela de equipes é ilustrada na Figura 17.

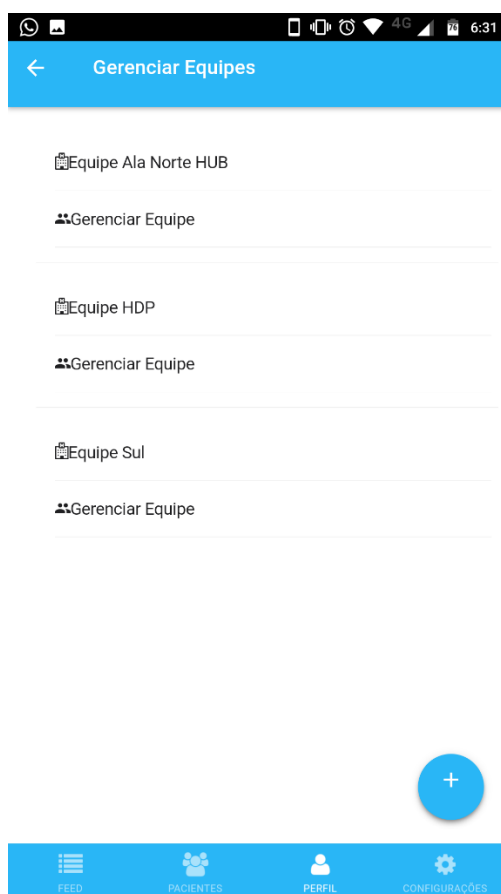


Figura 17 – Equipes da aplicação

A aplicação também contém um pequeno perfil do médico. O perfil ilustrado na Figura 18 mostra as informações do usuário e também uma opção de criação de equipes, além de prover a opção de editar o cadastro.

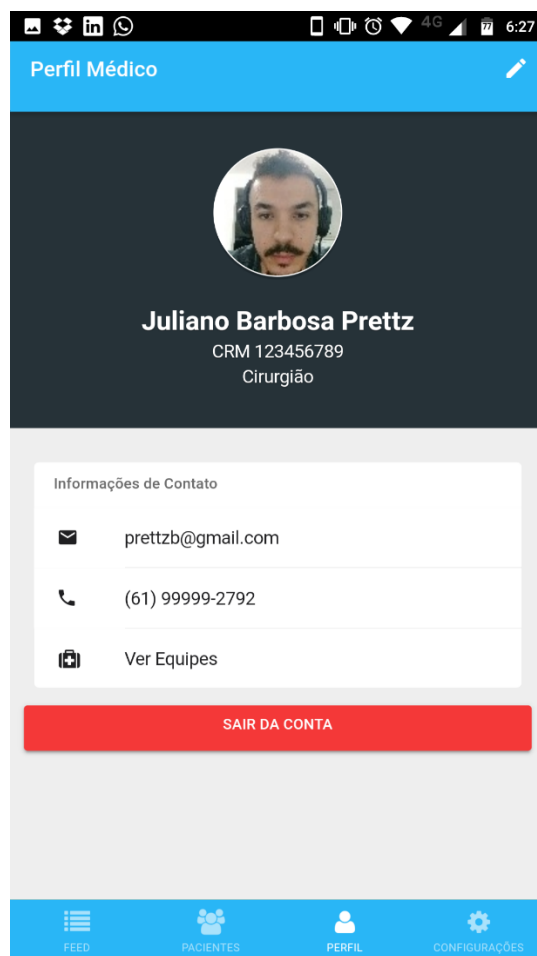


Figura 18 – Perfil Médico

Por fim a aplicação apresenta uma tela de configurações com as opções de “Sair da Conta” e adicionar pulseiras na aplicação. A Figura 19 ilustra a tela de configurações.

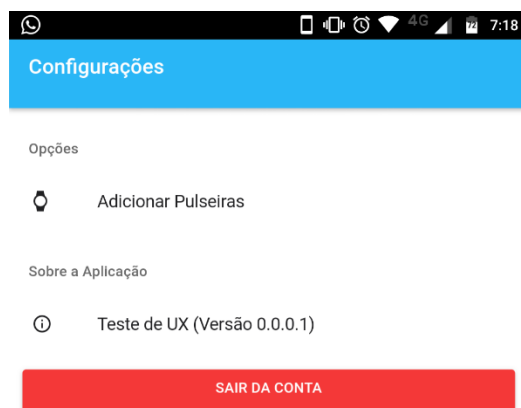


Figura 19 – Configurações da Aplicação

É possível notar que, na Figura 19, existe um botão para adicionar pulseiras. Já que a aplicação utiliza pulseiras inteligentes comerciais, os dados captados delas devem obrigatoriamente passar pela API proprietária antes de ser introduzidos na aplicação desenvolvida e, conseqüentemente, trabalhados pela API desenvolvida. Logo, para adicionar uma pulseira inteligente deve-se ter uma conta no *site* da empresa que comercializa a pulseira. Esta conta é gratuita e a pulseira deve ser adicionada apenas uma vez.

5. CONCLUSÕES

Nesta dissertação realizou-se o desenvolvimento de uma aplicação móvel para monitoramento de pacientes em clínicas e hospitais, apoiada na utilização de pulseiras inteligentes, otimizando uma arquitetura SISO e evoluindo a mesma para MIMO, por meio de um concentrador de sinais.

Com a revisão bibliográfica foi possível conhecer o estado atual do mercado de aplicação do mesmo gênero e os fundamentos da união entre tecnologia da informação e a área da saúde. Também foi possível notar que todas as aplicações e trabalhos científicos não contém nenhuma inovação no quesito arquitetura, utilizando apenas arquitetura SISO em todas as ocasiões.

O estudo também revelou que o mercado começa a mudar o foco atlético-*fitness* de suas aplicações e está olhando para a saúde como um todo.

É importante ressaltar que o trabalho desenvolvido nesta dissertação tem como conselheiro um médico especialista com mestrado, o qual é sempre solicitado para validar ideias para aplicação móvel.

Por fim, a arquitetura MIMO para pulseiras inteligentes com aplicação móvel para monitoramento de baixo custo para pacientes no pós-cirúrgico descreve a solução como uma aplicação para monitoramento de pacientes em hospitais e clínicas do Brasil. Os diversos sinais, captados através da pulseira inteligente que o paciente utiliza durante sua recuperação, fazem com que cada médico possa monitorar diversos pacientes simultaneamente. Além disso, cada paciente pode ter mais de um médico realizando o monitoramento e trocando informações sobre seu caso com outros médicos autorizados. O compartilhamento de informações se deu por meio da melhoria da arquitetura nativa, transformando uma conexão SISO para uma MIMO.

5.1 TRABALHOS FUTUROS

Como sugestões para trabalhos futuros, os resultados obtidos permitem apontar algumas direções importantes para o decorrer do trabalho.

Neste sentido como trabalho futuro visasse melhor o fluxo da aplicação como um todo, contando com o apoio de um especialista na área de Interface Humano Computador (IHC). Além disso, esperasse a substituição do concentrador de sinais por um *Raspberry Pi Zero*, baixando drasticamente o custo de um concentrador de sinais e Construção da própria pulseira inteligente, utilizando sensores otimizados e arquitetura aberta isso permitirá a captação de outros sinais que serão inseridos por meio de sensores na pulseira inteligente. Por fim, esperasse aplicar técnicas de mineração de dados no volume de informações gerados pela aplicação visando a extração de conhecimento.

Deste modo, é esperado que este trabalho possa contribuir significativamente para o crescimento da qualidade do acompanhamento e recuperação de pacientes, além de melhorar consideravelmente o trabalho dos médicos e enfermeiros.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] World Health Organization, WHO. "Global Health Observatory (GHO). <http://www.who.int/gho/health_workforce/physicians_density/en/> Acessado em: 04/04/2017.
- [2] Chen, J., Hillman, K., Bellomo, R., Flabouris, A., Finfer, S., Cretikos, M. The Impact of Introducing Medical Emergency Team System on the Documentations of Vital Signs. Official Journal of The European Resuscitation Council. 2009.
- [3] Chen, J., Hillman, K., Bellomo, R., Flabouris, A., Finfer, S., Cretikos, M. Introduction of The Medical Emergency Team (MET) System: a cluster-randomised controlled trial. The Lancet. 2005.
- [4] Laboissière, P. CFM: 93% dos brasileiros estão insatisfeitos com saúde pública e privada. <<http://agenciabrasil.ebc.com.br/geral/noticia/2014-08/pesquisa-diz-que-93-dos-brasileiros-estao-insatisfeitos-com-atendimento-na>> Acesso em: 03/06/2017.
- [5] Fitbit. <<https://www.fitbit.com/home>> Acesso em: 02/06/2017.
- [6] Tech Terms, Software Terms – API Definition. <<https://techterms.com/definition/api>> Acesso em: 10/05/2017.
- [7] Garmin International, Inc. Forerunner 305. <<https://buy.garmin.com/enUS/US/c10002-p1.html>> Acesso em: 17/05/2017.
- [8] Polar Electro Oy. Polar heart rate monitoring equipment. <<http://www.polar.fi>> Acesso em: 17/06/2017.
- [9] Apple, Inc. Nike+. <<https://www.apple.com/apple-watch-nike/>> Acesso em: 17/05/2017.
- [10] Runtastic. <<https://www.runtastic.com/>> Acesso em: 17/05/2017.
- [11] Prettz, J. B., Zanatta, M., Costa, J. P. C. L., Alvim, J., Miranda, R. K. Efficient and low cost MIMO communication architecture for smartbands applied to postoperative patient care. 2nd Russian-Pacific Conference on Computer Technology and Applications (RPC). 2017.

- [12] Kashem, A., Droogan, M., Santamore, W., Bove, A. Managing Heart Failure Care Using an Internet-Based Telemedicine System. Journal of Cardiac Failure. Elsevier. 2008.
- [13] Don, S., Chung, D., Choi, E., Min, D. An Awareness Approach to Analyse ECG Streaming Data. Journal of Medical Systems. Springer Science. 2013.
- [14] Suh, M., Chen, C., Woodbridge, J., Tu, M., Kim, J., Nahapetian, A., Evangelista, L., Sarrafzadeh, M. A Remote Patient Monitoring System for Congestive Heart Failure. Journal of Medical Systems. Springer Science. 2011.
- [15] Santos, G. Elaboração e Desenvolvimento de Aplicativo para Dispositivos Móveis para Prevenção do Pé Diabético. Dissertação de Mestrado do Curso de Enfermagem. Universidade Federal de Minas Gerais. 2013.
- [16] Edwards, B. “Wearable sensor by Affectiva can measure anxiety and is helping autism research”. <<http://www.imedicalapps.com/2011/10/wearable-sensor-byaffectiva-can-measure-anxiety-and-is-helping-autism-research/>> Acesso em: 02/06/2017.
- [17] Umbehr, J. “CPR Glove A Potential Lifesaver”. <http://www.medgadget.com/2007/02/cpr_glove_a_pot.html> Acesso em: 04/06/2017.
- [18] Bradford, T. “Motorola MotoACTV Is What the iPod Nano Should Be”. <<http://www.gottabemobile.com/2011/10/20/video-motorola-motoactv-iswhat-the-ipod-nano-should-be/>> Acesso em: 04/06/2017.
- [19] Fitbit API. <<https://dev.fitbit.com/>> Acesso em 03/04/2017.
- [20] Sony Mobile. <<https://www.sonymobile.com/global-en/products/smart-products/smartband-2/>> Acesso em 03/06/2017.
- [21] Android KitKat. <<https://www.android.com/versions/kit-kat-4-4/>> Acesso em 04/06/2017
- [22] Apple BR. <<https://www.apple.com/br/>> Acesso em: 04/06/2017.

- [23] MyFitnessPal. <<https://www.myfitnesspal.com/pt>> Acesso em: 04/06/2017.
- [24] Runkeeper. <www.runkeeper.com/> Acesso em: 04/06/2017.
- [25] Samsung Gear S. <<http://www.samsung.com/pt/promotions/galaxynote4/spec/gears/>> Acesso em 04/06/2017.
- [26] Garmin. <<https://www.garmin.com/pt-BR>> Acesso em 04/06/2017.
- [27] Do Bem. <<http://dobem.com/>> Acesso em 04/06/2017.
- [28] Apple iOS. <www.apple.com/br/ios/ios-10/> Acesso em: 04/06/2017.
- [29] Winkler, S., Schiebber, M., Lucke, S., Heinze, P., Schweizer, T., Wegertseder, D., Scherf, M., Nettelau, H., Henke, S., Braecklein, M., Anker, S., Koehler, F. "A new telemonitoring system intended for chronic heart failure patients using mobile telephone technology - Feasibility study. International Journal of Cardiology. Elsevier. V. 153. 2011.
- [30] Rudy, P. Smart notification bracelet for iPhone and Android. <[http://www.kickstarter.com/projects/embraceplus/embrace-1?ref=liveEMBRACE+, smart notification bracelet for iPhone and Android](http://www.kickstarter.com/projects/embraceplus/embrace-1?ref=liveEMBRACE+,smartnotificationbraceletforiPhoneandAndroid)> Acessado em: 02/06/2017.
- [31] EasyGlic - Dispositivo de Alerta Hipoglicêmico. <<http://www.easyglic.com/>> Acesso em: 04/06/2017.
- [32] Bluetooth Low-Energy. <<https://www.bluetooth.com/>> Acesso em: 10/06/2017.
- [33] Berthelot, M., Yang, G., Lo, B. Preliminary Study for Hemodynamics Monitoring using a Wearable Device Network. IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN). 2017.
- [34] Haartsen, J. C., Mattisson, S. Bluetooth-A New Low-Power Radio Interface Providing Short-Range Connectivity. Proceedings of the IEEE. V. 88. 2000.

- [35] Ferro, E., Potort, F. Bluetooth and Wi-Fi Wireless Protocols: A Survey and a Comparison. IEEE Wireless Communications. V. 12. 2005.
- [36] Node.js. <<https://nodejs.org/en/>> Acesso em: 05/06/2017.
- [37] WPI Analog Lab Resources, Bluetooth Specifications, Radio Specifications. <http://ece.wpi.edu/analog/resources/bluetooth_a.pdf> Acesso em: 06/06/2017.
- [38] Bluetooth. <<https://www.bluetooth.com/specifications/bluetooth-core-specification>> Acesso em: 10/06/2017.
- [39] Project Galileo. <<https://bitbucket.org/benallard/galileo/overview>> Acesso em: 04/06/2017.
- [40] Android. <https://www.android.com/intl/pt-BR_br> Acesso em: 03/04/2017.
- [41] Pressman, R. S. Engenharia de software. McGraw Hill. 2011.
- [42] Android SDK. <<https://developer.android.com/studio/index.html>> Acesso em: 05/04/2017.
- [43] Monaca App Development Fast, Easy and Flexible. <<https://monaca.io/>> Acesso em: 26/04/2017.
- [44] HTML5 - World Wide Web Consortium. <<https://www.w3.org/TR/html5/>> Acesso em: 03/06/2017.
- [45] PhoneGap. <phonegap.com> Acesso em: 02/06/2017.
- [46] Cordova. <<https://cordova.apache.org/>> Acesso em: 02/06/2017
- [47] Javascript. <<https://www.javascript.com/>> Acesso em: 02/06/2017.
- [48] DigitalOcean: Cloud computing designed for developers. <www.digitalocean.com> Acesso em: 03/06/2017.

[49] Meteor <<https://www.meteor.com/>> Acessado em: 06/06/2017.

[50] Ionic <<https://ionicframework.com/>> Acessado em: 06/06/2017.

[51] Bootstrap <<http://getbootstrap.com/>> Acessado em: 06/06/2017.

[52] AngularJS - Superheroic JavaScript MVW Framework. <<https://angularjs.org/>> Acessado em: 06/06/2017.

[53] jQuery <<https://jquery.com/>> Acessado em: 06/06/2017.

[54] Raspberry Pi - Teach, Learn, and Make with Raspberry Pi. < <https://www.raspberrypi.org/>> Acesso em: 06/06/07

APÊNDICES

1. Apendice I

Explicativo da aplicação móvel

A. Acesso ao aplicativo

Por questões de segurança a aplicação possui uma tela de login. Nesta tela é possível entrar na aplicação através da inserção de e-mail e senha já cadastrados. No caso de não existir cadastro esta mesma tela permite realizar um primeiro cadastro utilizando e-mail e dados referentes ao médico usuário como nome, CPF, especialidade do médico, hospital, CRM. Nesta tela possui ainda um botão onde é possível assinalar se deseja permanecer conectado ou fazer login a cada interação com a aplicação móvel. Também possui uma opção caso o usuário tenha esquecido a senha, podendo solicitar a troca de senha.

A.1 Login

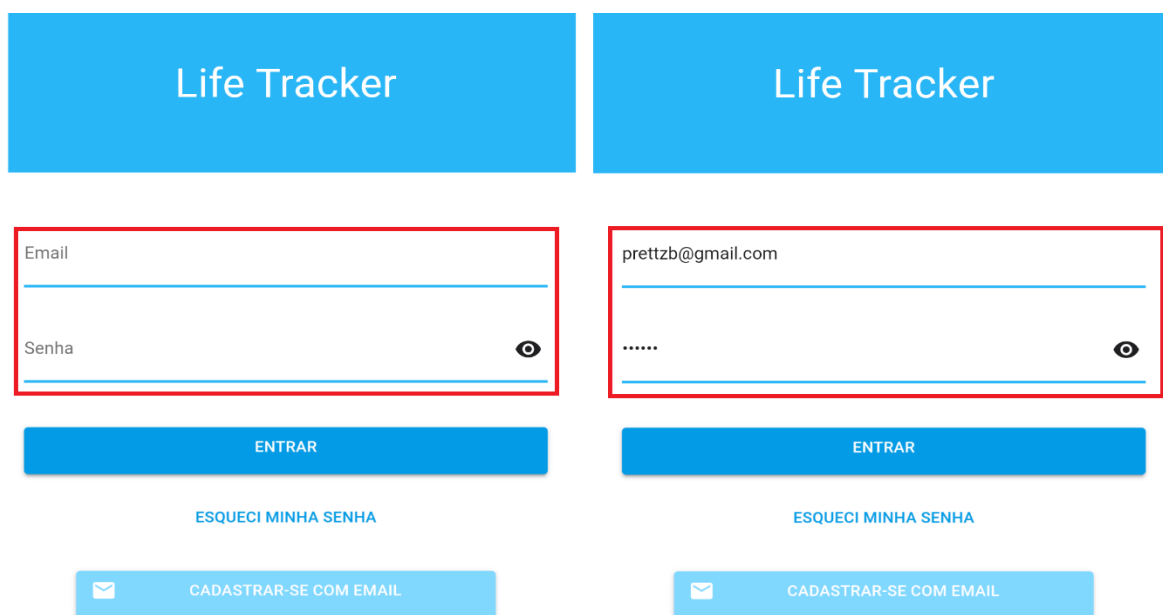


Figura 20 – Login da Aplicação

Para realizar o login, os passos são:

1. Passo: Inserir e-mail já cadastrado no campo e-mail;
2. Passo: Inserir senha já cadastrada no campo senha;
3. Passo: pressionar o botão “ENTRAR”

Assim o sistema irá realizar o processamento das informações e irá liberar o acesso para o usuário.

A.2 Primeiro Cadastro

A imagem mostra a interface de usuário da aplicação Life Tracker, dividida em duas seções principais. À esquerda, há um formulário de login com campos para 'Email' e 'Senha', um botão 'ENTRAR' e um link 'ESQUECI MINHA SENHA'. Abaixo, um botão 'CADASTRAR-SE COM EMAIL' está destacado com um retângulo vermelho. À direita, há uma tela de 'Cadastro' com um cabeçalho azul contendo um ícone de seta para trás e o texto 'Cadastro'. Abaixo, há um ícone de perfil humano e campos para 'Nome' e 'CPF'. Seguem-se ícones e campos para 'Adicionar foto', 'CRM', 'Especialidade', 'Email', 'Telefone', 'Senha' e 'Repetir Senha'. Um botão 'ENVIAR CONSULTA' está localizado na base da tela de cadastro.

Figura 21 – Primeiro cadastro na aplicação

Para realizar o primeiro cadastro na aplicação basta pressionar o botão “CADASTRAR SE COM EMAIL” e preencher algumas informações, em sua totalidade são:

1. Passo: pressionar o botão “CADASTRAR SE COM EMAIL”;
2. Passo: preencher os campos as com informações necessárias (Nome, CPF, Foto, CRM,

Especialidade, E-mail, Telefone, Senha e por fim a senha novamente);

3. Passo: pressionar o botão “ENVIAR CONSULTA”.

A aplicação móvel validará as informações e realizara o cadastro do usuário.

A.3 Esqueceu a senha

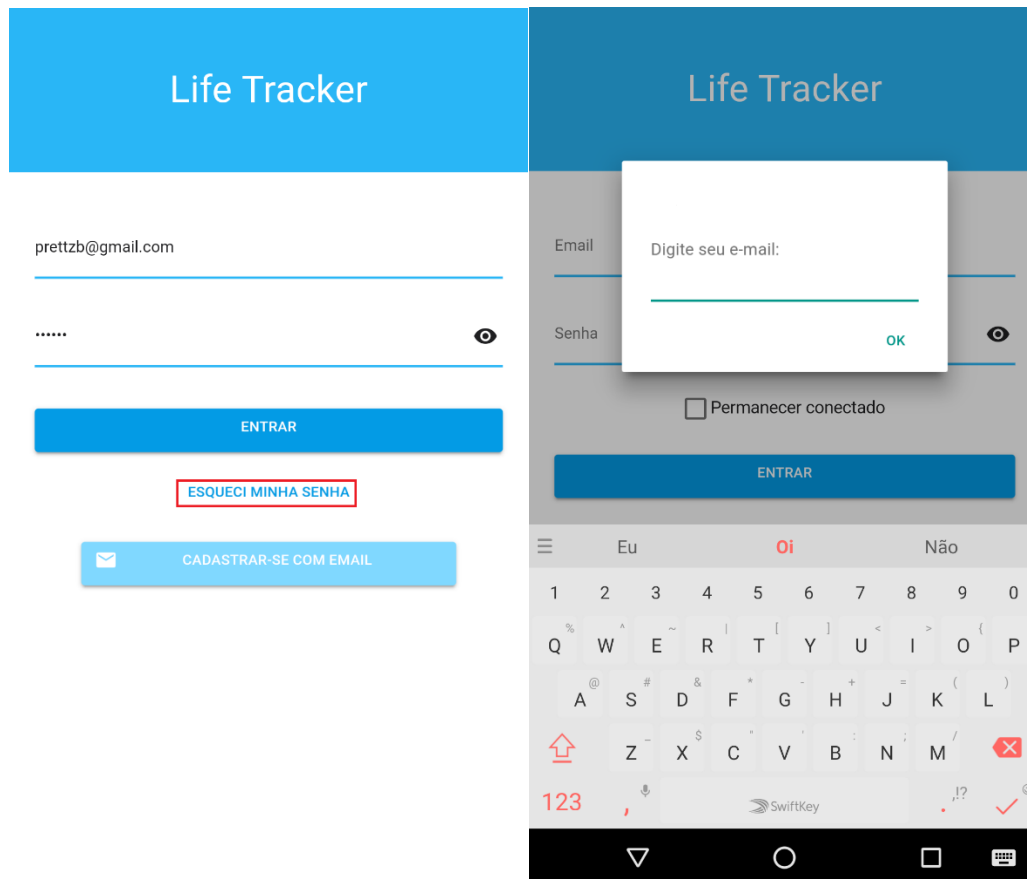


Figura 22 – Recuperação de senha

Caso o usuário esqueça a senha de acesso a aplicação existe uma maneira de recuperar, neste caso precisa se seguir alguns passos, como:

- Passo: pressionar o botão “ESQUECI MINHA SENHA”;
- Passo: insira o e-mail cadastrado no campo sugerido pela aplicação;
- Passo: pressionar o botão “OK”;

A aplicação irá enviar um e-mail com um link, basta acessar esse link e redefinir a senha, só irá funcionar se o e-mail inserido no campo for o mesmo utilizado no cadastro do perfil na aplicação.

B. *Feed de Atualizações*

O *feed* de informações concentra pequenas notificações como se fosse uma linha no tempo, informações como paciente adicionado, lembrete adicionada, paciente compartilhado e equipe criada aparece como informação no *Feed* de informações. O *feed* de informações é atualizado conforme as informações são inseridas na aplicação é também a primeira tela encontrada após a realização do login e/ou cadastro inicial.

B.1 *Feed de Atualizações*

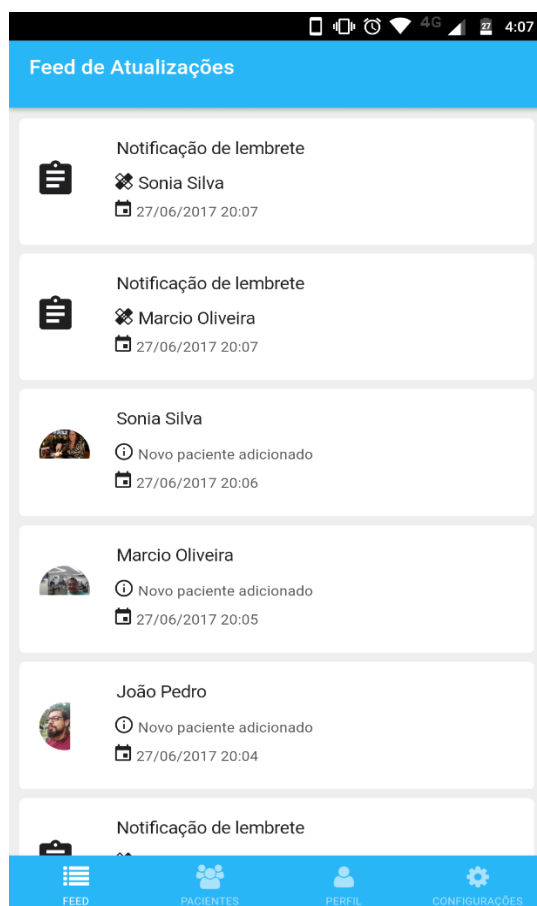


Figura 23 – *Feed* de Atualizações

Como ilustrado na Figura 23, o *feed* de atualizações concentra diversas atualizações no sistema, para acessar ele basta clicar no botão “FEED” que está presente no rodapé fixo da aplicação móvel. Além disso, toda a informação exibida no *feed* de atualizações são clicáveis, ou seja, não são informações fixas e estáticas na tela da aplicação.

C. Paciente e Grupos

Esta tela é capaz ser acessada através do ícone encontrado no rodapé da aplicação. Nela é possível encontrar uma lista com todos os pacientes adicionados pelo usuário e/ou compartilhados em equipes com o usuário. Além disso, é possível adicionar pacientes por meio de um botão que fica no canto inferior direito.

C.1 Paciente

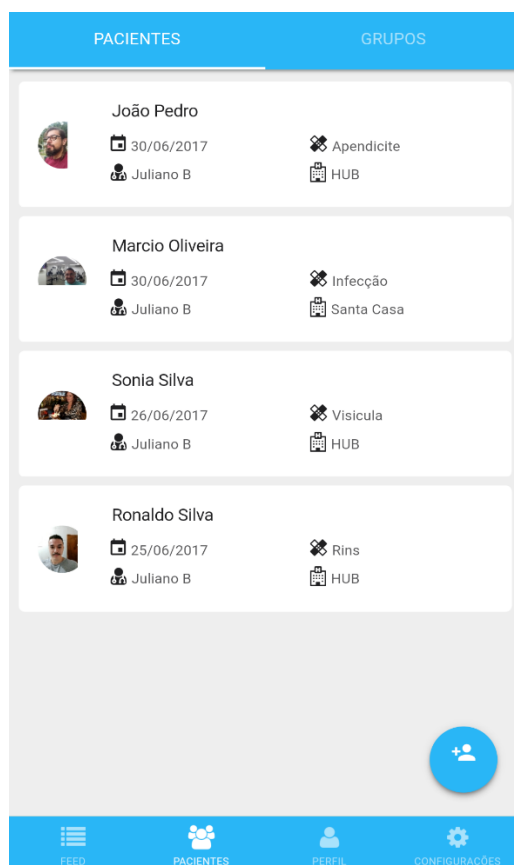


Figura 24 – Pacientes

Na tela de pacientes é possível ter acesso a lista de pacientes cadastrados na aplicação pelo usuário, conforme ilustra a Figura 24, para acesso basta pressionar o botão “PACIENTES” que está posicionado no rodapé fixo da aplicação.

C.2 Adicionar paciente

Na adição de pacientes devem ser informados uma série de informações como nome do paciente, data da internação, causa da internação, hospital que se encontra o paciente e foto do paciente, todos esses dados ficam armazenados no perfil do paciente que pode ser acessado pela lista inicial.

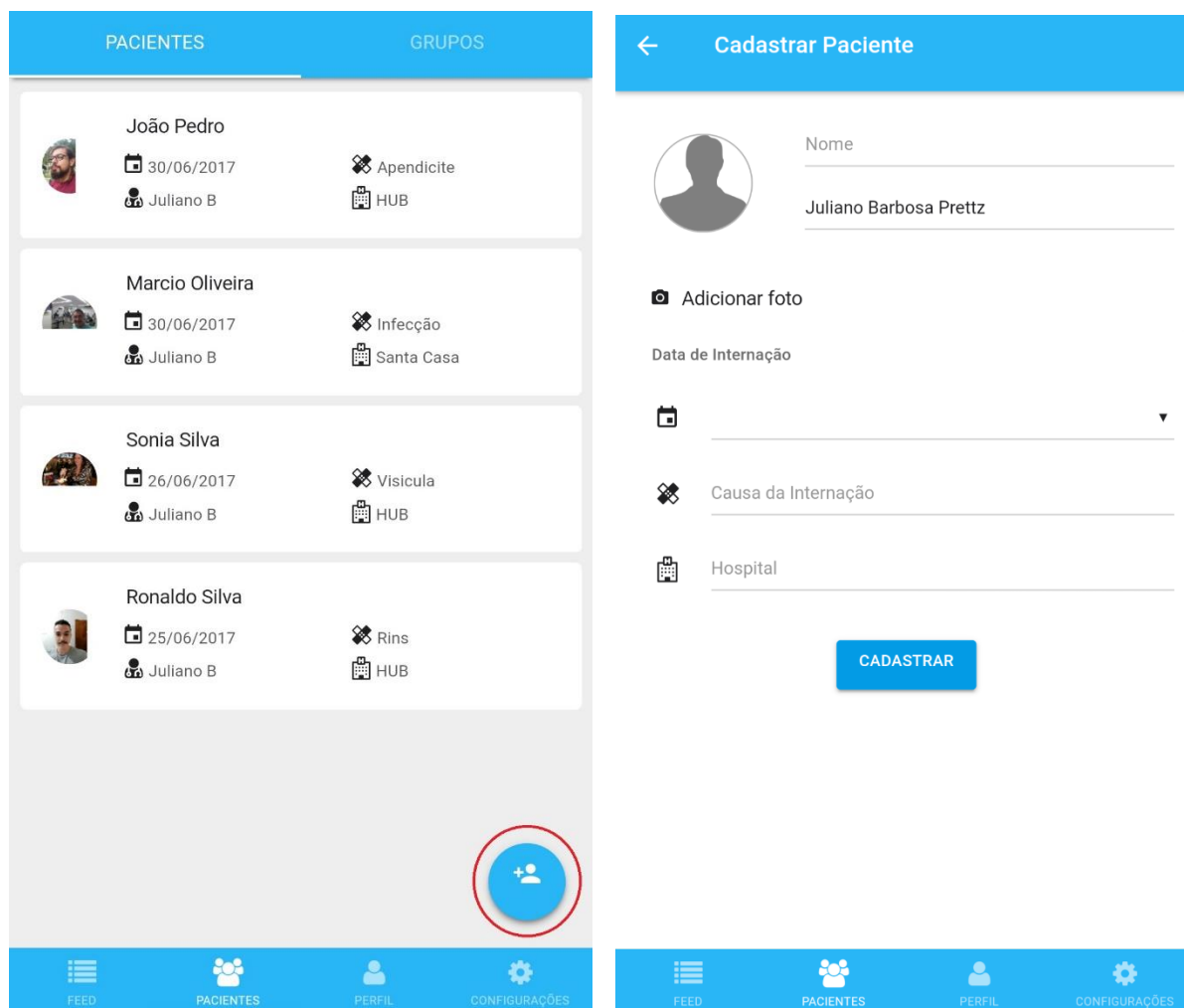


Figura 25 – Cadastrar paciente

Conforme ilustra na Figura 25, para cadastrar um paciente são necessários alguns passos, como:

1. Passo: acessar a área de pacientes na aplicação, basta pressionar o botão “PACIENTES” presente no rodapé fixo da aplicação móvel;
2. Passo: pressionar o botão presente no canto inferior direito na aplicação móvel, este botão está destacado na Figura 25;
3. Passo: o usuário deve preencher as informações solicitadas pela aplicação (nome, foto, data de internação, causa da internação e hospital de internação);
4. Passo: por fim basta pressionar o botão “CADASTRAR”.

C.3 Grupos de pacientes

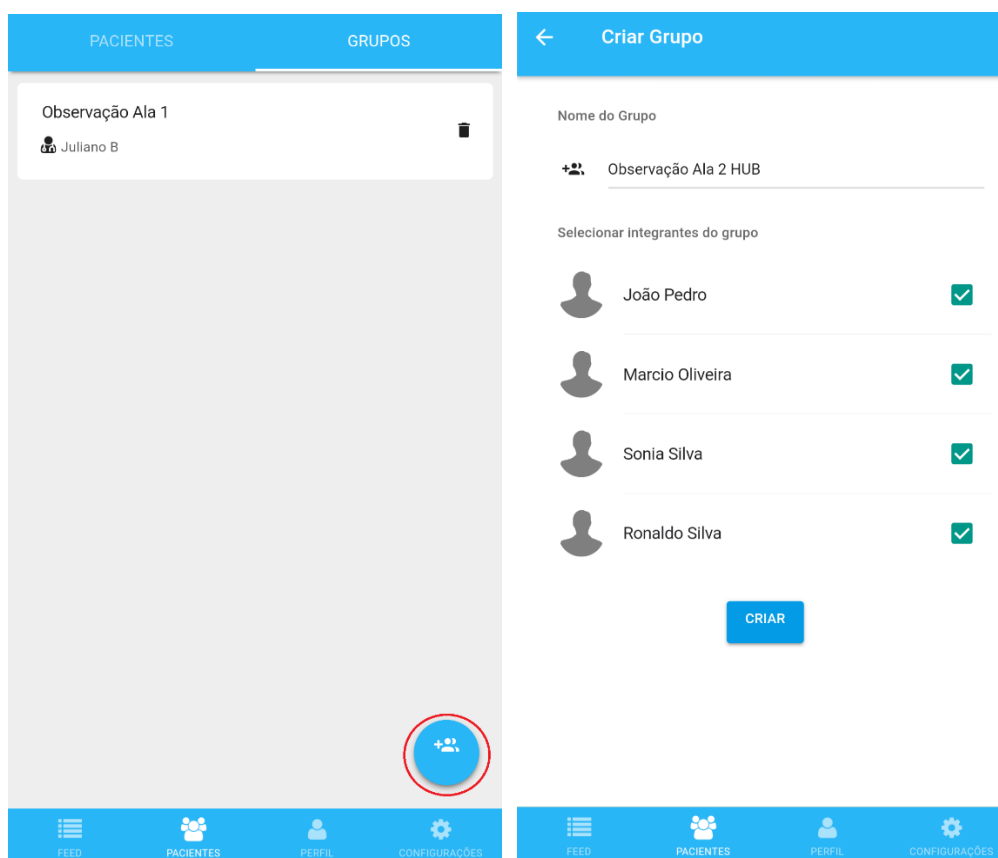


Figura 26 – Grupo de Pacientes

Conforme ilustrado na Figura 26, na mesma seção pacientes existe uma aba a direita do usuário chamada grupos. Nesta aba é possível criar grupos de pacientes, neste grupo é possível adicionar quantos pacientes quiser. O grupo de paciente pode servir para diversas situações como o médico organizar seus pacientes por hospital ou por estado no tratamento. Para criar um grupo é necessário seguir alguns passos, como:

1. Passo: acessar a segunda aba da tela de “PACIENTES” pressionando a segunda aba chamada “GRUPOS”;
2. Passo: pressionar o botão no canto inferior direito da aplicação móvel;
3. Passo: basta inserir um nome para o grupo e selecionar os pacientes desejados e clicar no botão criar. A aplicação deixa livre para o usuário utilizar a nomenclatura que quiser.

D. Perfil do paciente

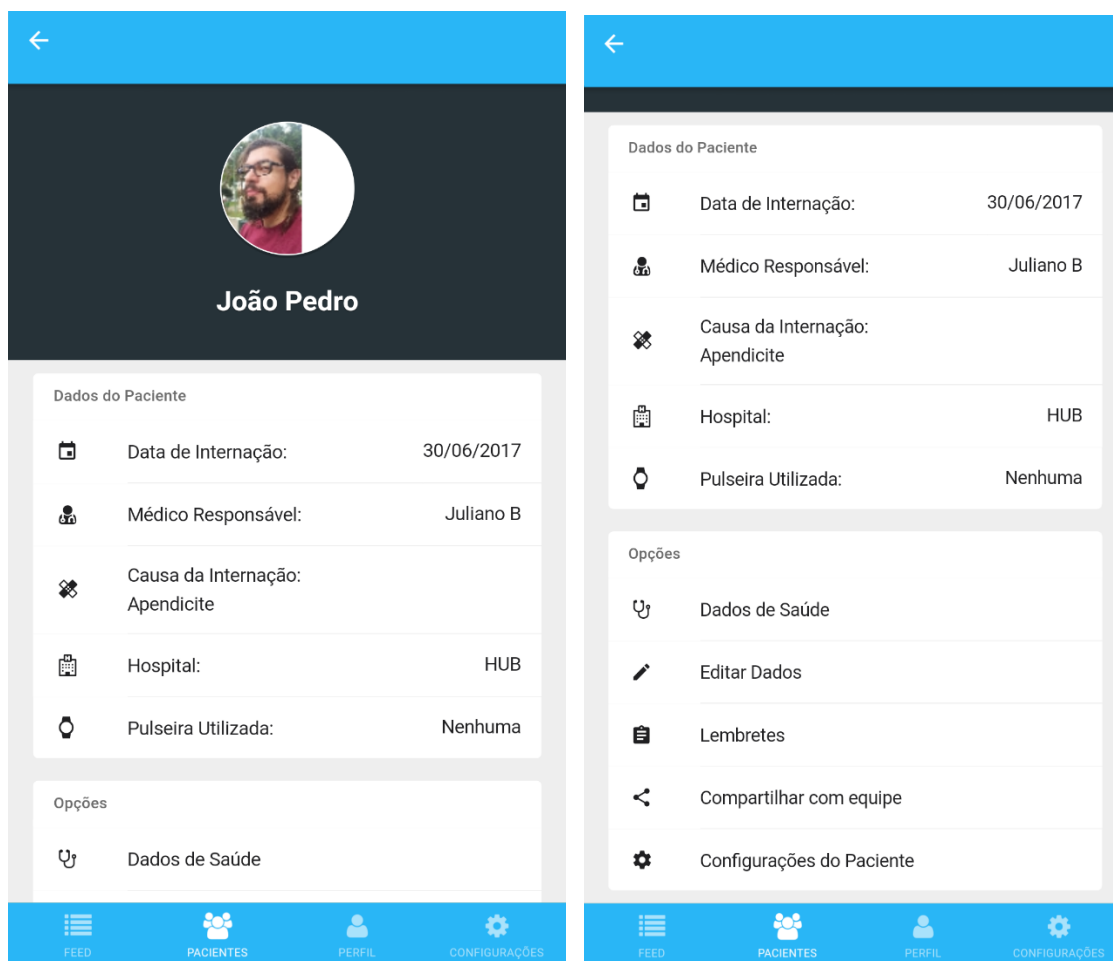


Figura 27 – Perfil do Paciente

Conforme ilustrado na Figura 27, o perfil do paciente pode ser acessado pelo rodapé da aplicação móvel, pressionando o botão “PACIENTES” e escolhendo o paciente desejado. Dentro do perfil do paciente contém diversas informações como data de internação, médico responsável, causa da internação, hospital e pulseira utilizada pelo paciente.

Existe algumas opções dentro do perfil como dados de saúde, edição de dados, lembretes, compartilhar com equipe e configurações do paciente.

D.1 Dados de saúde

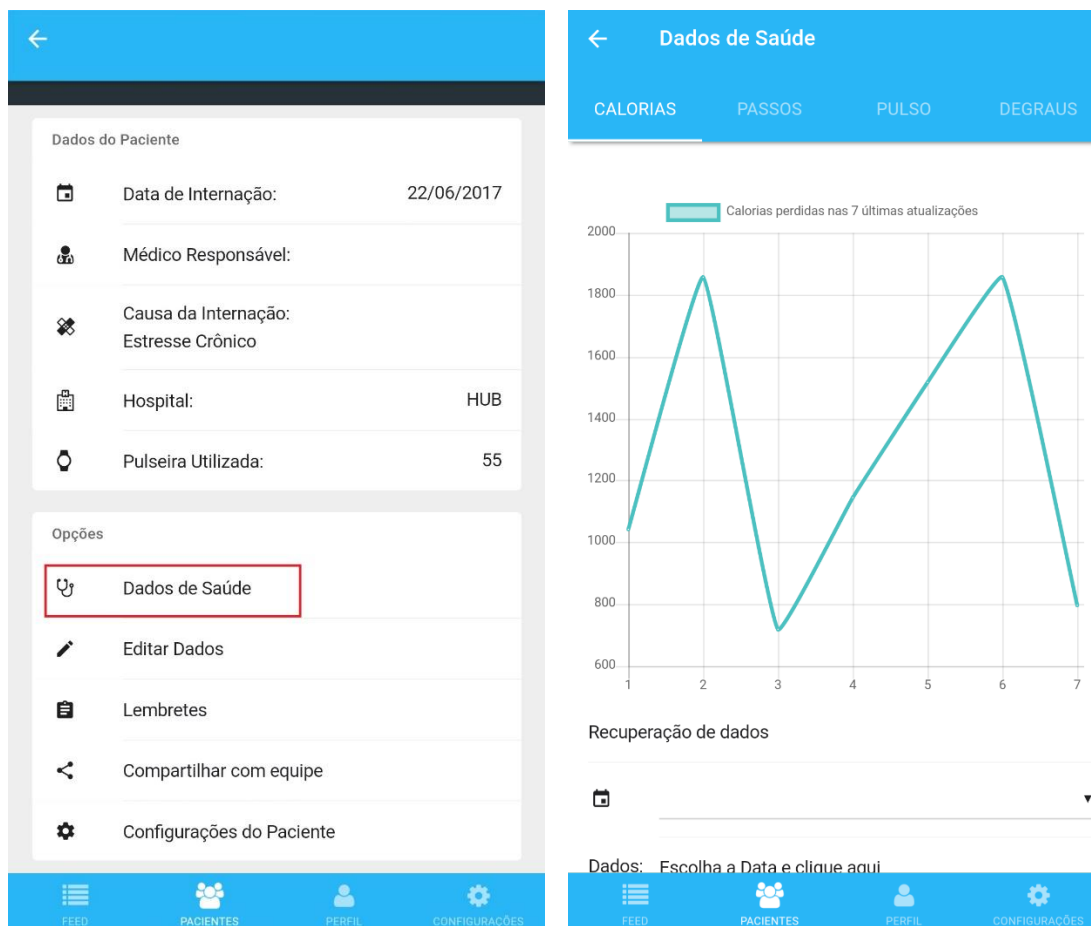


Figura 28 – Dados de Saúde

No perfil do paciente é possível encontrar a opção “Dados de Saúde”. Os dados de saúde podem ser acessados em qualquer perfil de paciente, porém o mesmo só irá fornecer gráficos quando conectados com a pulseira inteligente. É possível ver dados de calorias, passos, pulso, degraus.

D.2 Edição dos dados

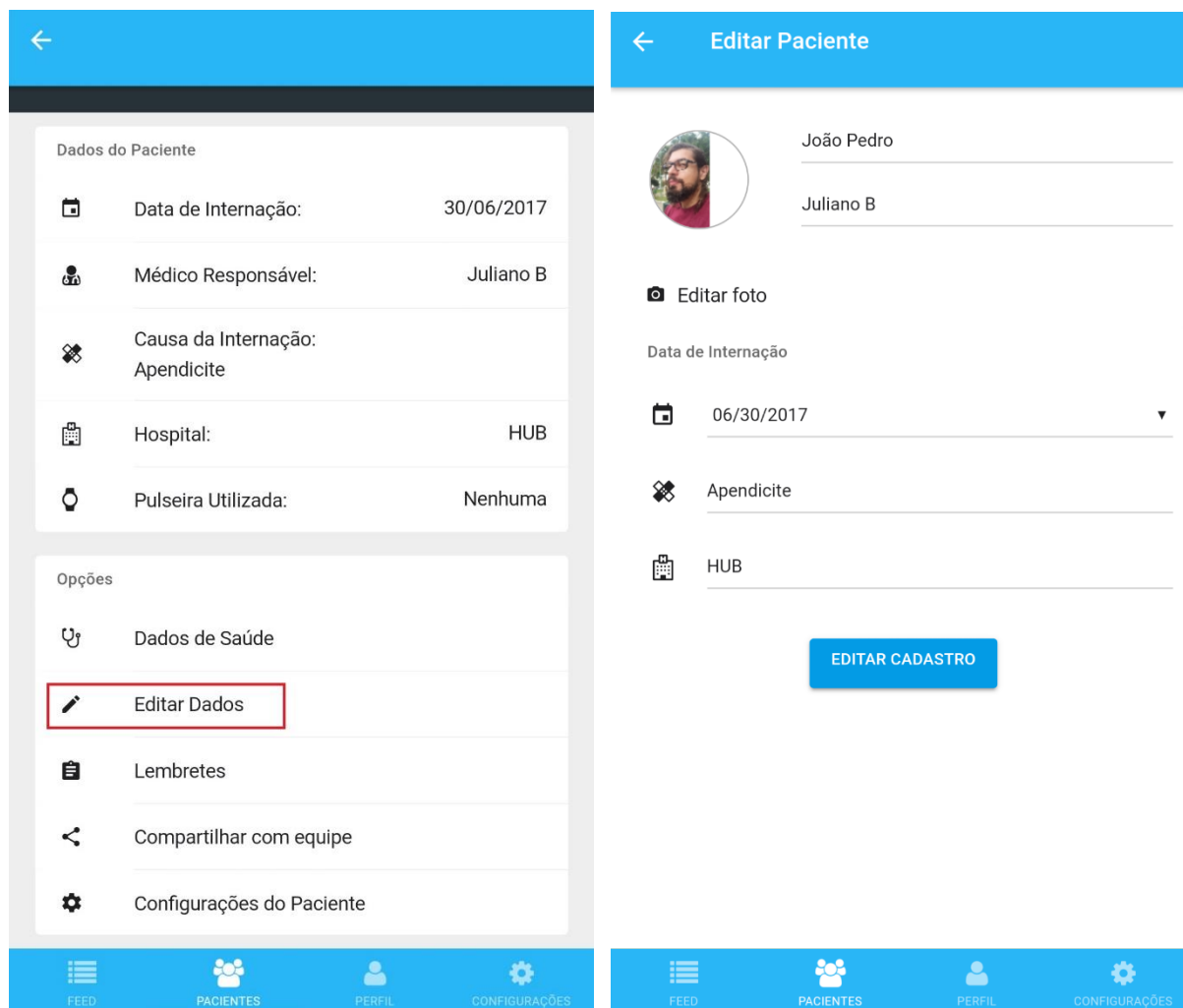


Figura 29 – Edição de dados de paciente

A edição de dados de paciente ilustrada na Figura 29, é proveniente de alguns passos como:

1. Passo: acesse o perfil do paciente desejado;
2. Passo: localize a opção “Editar Dados”;
3. Passo: edite os campos desejados;
4. Passo: pressione o botão “EDITAR CADASTRO”.

D.3 Lembretes

Ainda dentro do perfil do paciente se encontra uma opção chamada de lembretes, nesta opção permite o médico usuário a inserir informações referentes ao paciente desejado. As informações existentes iniciam se com anotações e podem ir até dados técnicos. Caso o médico usuário compartilhe o paciente com outro médico, o mesmo vai poder inserir informações nos lembretes também realizando assim uma troca de informações mais dinâmicas e guardando as mesmas para uma possível consulta posterior. A Figura 30 ilustra a tela de lembretes.

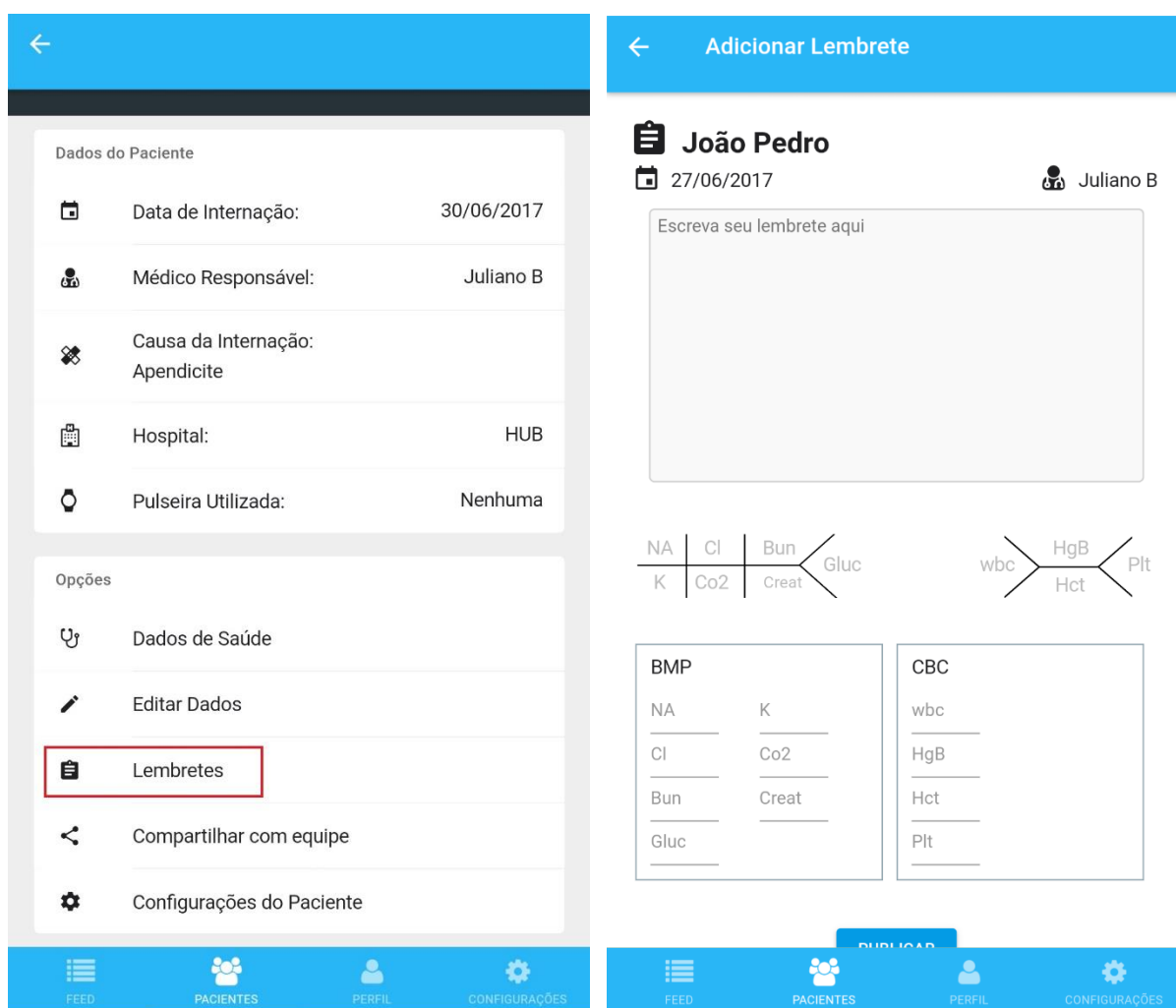


Figura 30 – Lembretes

Para acessar a tela de lembretes e escrever é necessário seguir alguns passos.

1. Passo: acessar perfil do paciente desejado;
2. Passo: localizar a opção “Lembretes”, destacado na Figura 30;
3. Passo: o usuário pode inserir informações de texto e técnicos específicos da área médica.
4. Passo: basta pressionar o botão “PUBLICAR”.

D.4 Compartilhar com equipe

Nesta opção localizada no perfil do paciente é possível adicionar pacientes as equipes criadas pelo usuário ou em equipes da qual o usuário faz parte. Neste sentido o usuário compartilha com outros médicos seus pacientes e estes podem interagir integralmente com o paciente. A Figura 31 ilustra a opção compartilhar com equipe.

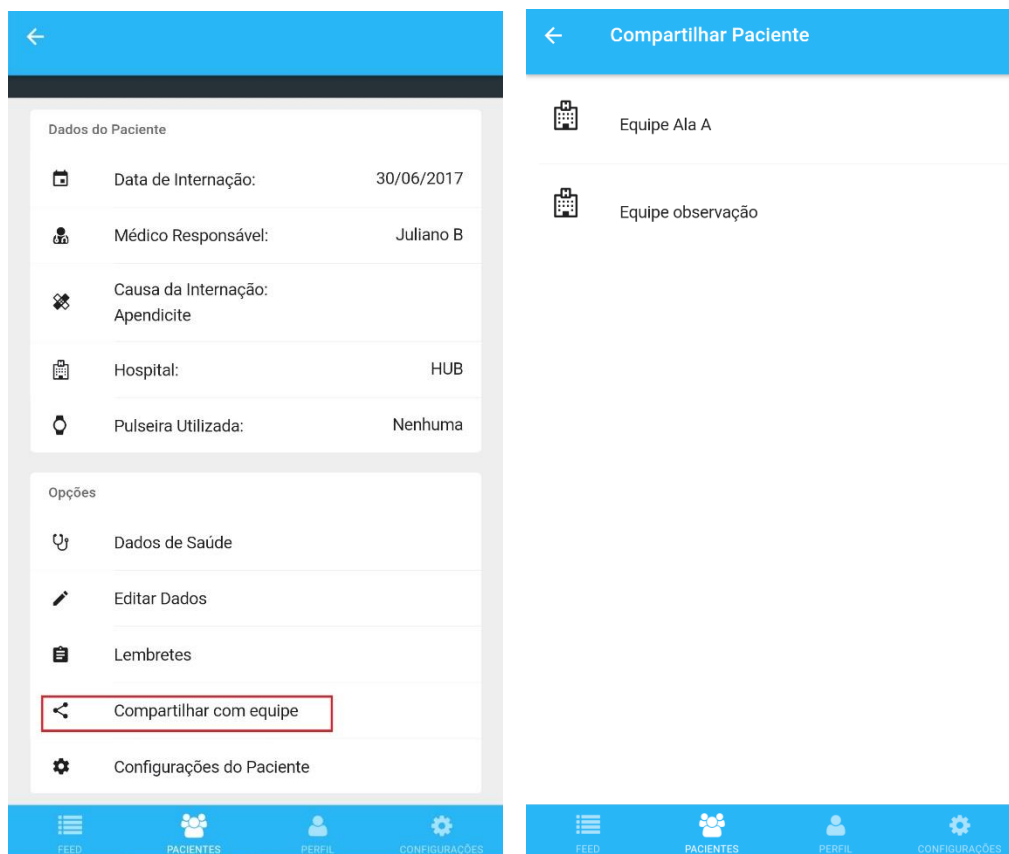


Figura 31 – Compartilhar com equipe

Conforme ilustrado na Figura 31, para compartilhar o paciente desejado em alguma equipe criada na aplicação móvel basta acessar o paciente e localizar a opção “Compartilhar com equipe” e escolher a equipe que deseja inserir o paciente.

D.5 Configurações do paciente

Nesta opção o usuário pode alterar a pulseira inteligente utilizada pelo paciente, além de ter a opção de excluir o paciente da aplicação. A Figura 32 ilustra as configurações do paciente.

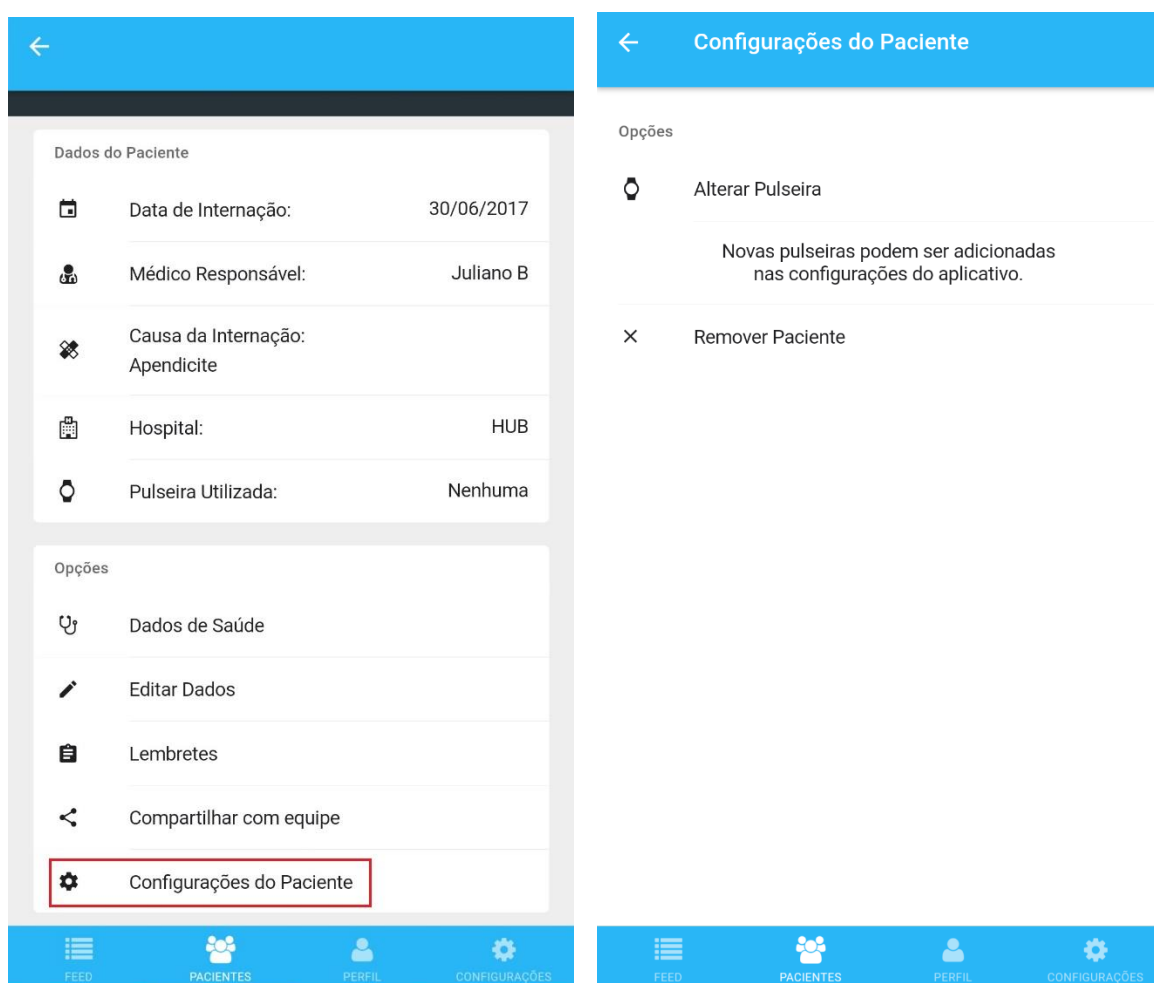


Figura 32 – Configurações do Paciente

As configurações do paciente são simples e com poucas opções, neste caso existe a opção de alterar a pulseira utilizada pelo paciente na opção “Alterar pulseira” e a opção “Excluir paciente” que remove o paciente do plano ativo da aplicação.

E. Perfil médico

O perfil do médico é acessado através do botão de mesmo nome existente no rodapé da aplicação móvel, o perfil do médico é simples e objetivo. Possui informações como edição dos dados do perfil pelo usuário, informações gerais de contato do médico usuário como e-mail, telefone, CRM e especialidade, nesta mesma tela é possível acessar as equipes do médico e sair da conta logada na aplicação. A Figura 33 ilustra o perfil médico.

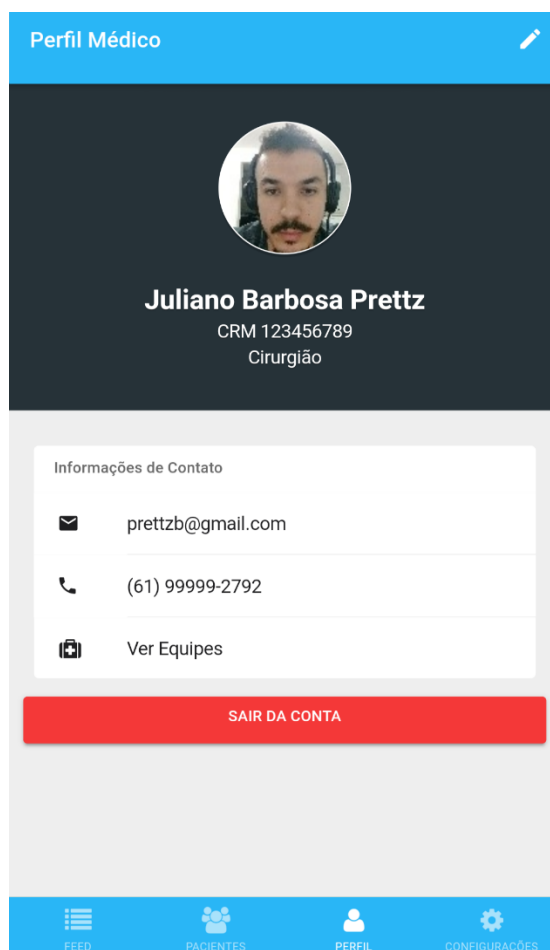


Figura 33 – Perfil médico

E.1 Edição de dados do perfil médico

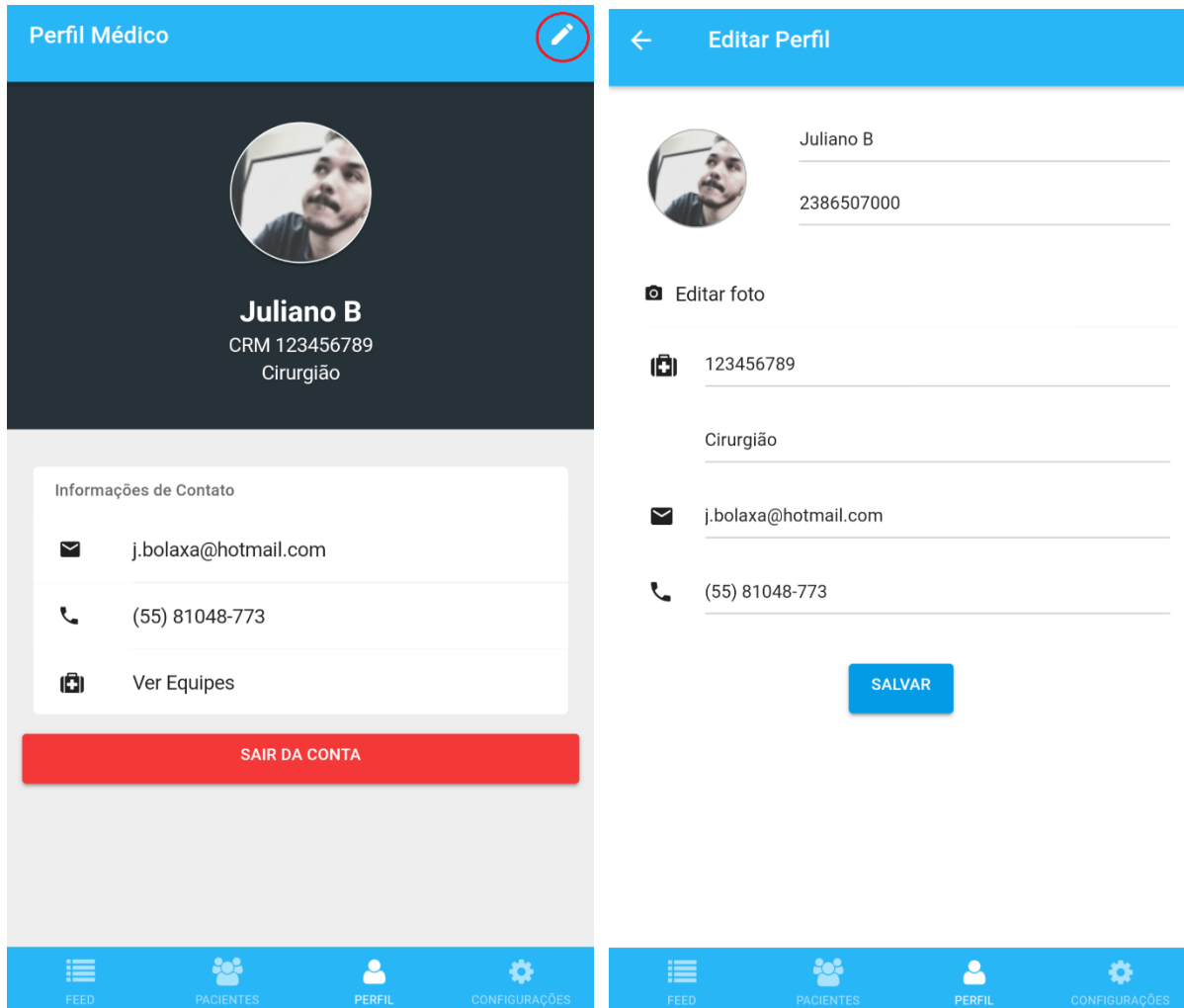


Figura 34 – Edição de dados do perfil médico

Conforme ilustrado na Figura 34, a edição de dados se faz em três passos:

1. Passo: acessar perfil médico através do botão “PERFIL” no rodapé e pressionar o botão de edição destacada no canto superior direito da aplicação;
2. Passo: basta alterar os dados necessários;
3. Passo: pressionar o botão “SALVAR”.

E.2 Equipes

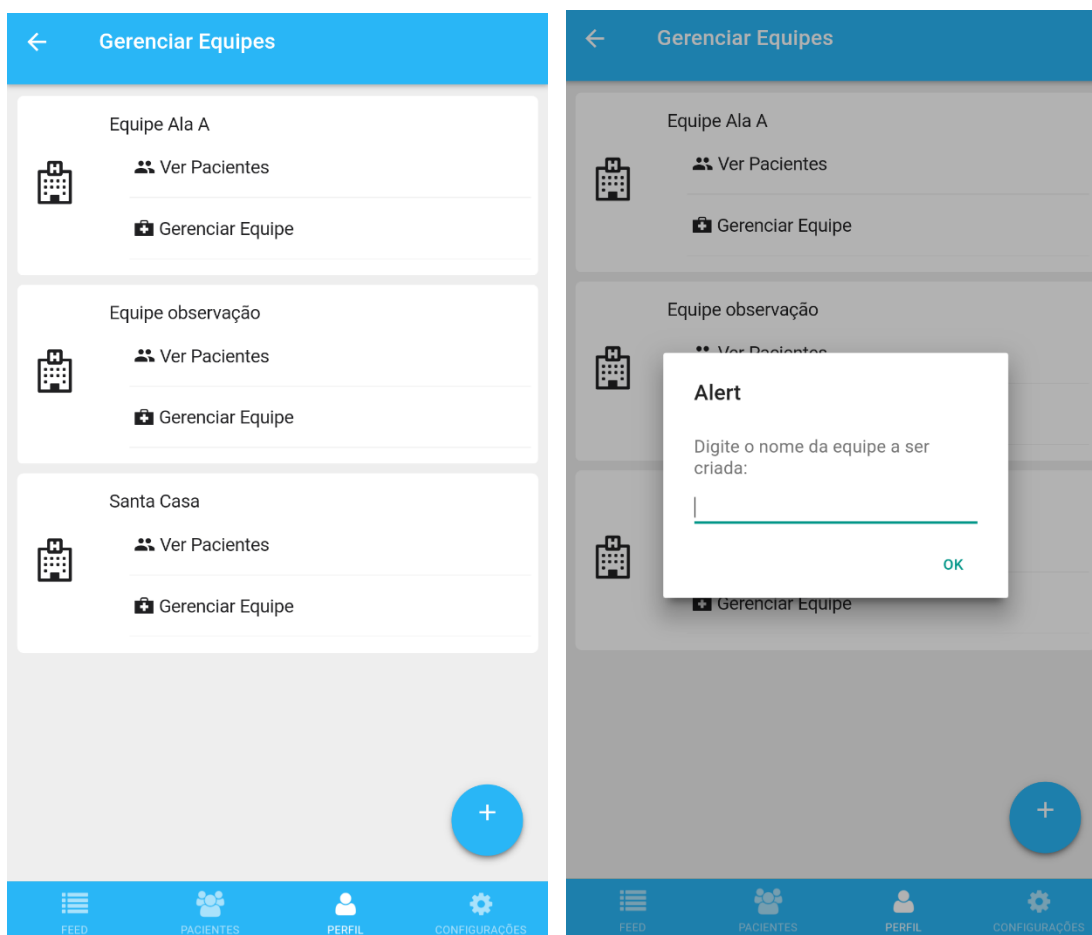


Figura 35 - Equipes

Opção contida dentro do perfil do médico é possível criar e gerenciar equipes. Para criação da equipe basta utilizar o botão contido no canto inferior direito da aplicação e inserir o nome da equipe a ser criada, como ilustra a Figura 35, fazendo isso você já tem uma nova equipe. Logo é viável adicionar pacientes as equipes e convidar médicos para participar da equipe criada, inserindo o e-mail do médico.

F. Configurações



Figura 36 - Configurações

Conforme ilustrado pela Figura 36, esta opção é acessada por meio do rodapé fixo da aplicação móvel, nas configurações é possível adicionar pulseiras inteligentes e fazer *logout* da aplicação móvel.

F.1 Adicionar pulseira

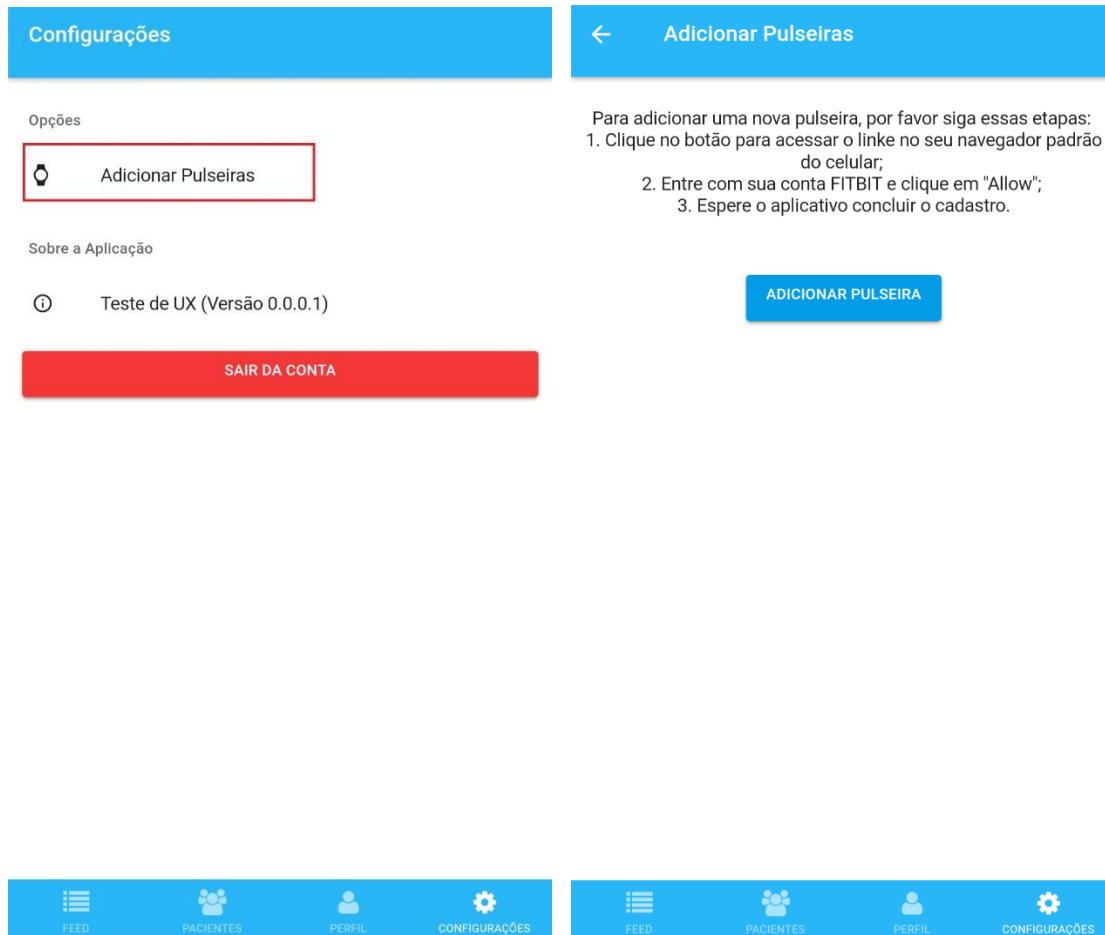


Figura 37 – Adicionar pulseiras

Para registrar uma pulseira na aplicação basta clicar na opção “Adicionar Pulseiras” destacada na Figura 37. Nesta tela é encontrado um pequeno passo a passo para adição de pulseiras inteligentes e um botão chamado “Adicionar Pulseira”. Para adicionar uma nova pulseira inteligente na aplicação móvel deve-se passar por etapas rápidas que são clicar no botão “Adicionar Pulseira” esse botão irá abrir um link no navegador padrão utilizado pelo smartphone. O link acessado por esse botão é www.fitbit.com/login basta inserir o login e senha da conta nativa da Fitbit e clicar no botão “Allow” que o aplicativo conclui o cadastro automaticamente e exibe a mensagem “Pulseira registrada com sucesso”.

2. Apêndice II

API própria desenvolvida em Node.js para trabalhar com a pulseira inteligente e a aplicação desenvolvida.

API's do projeto Pulseiras Inteligentes

Documentação das chamadas escritas para o desenvolvimento da aplicação móvel

Pulseiras Inteligentes

1. **POST em "url/api/pulseira" -> Método descontinuado - substituído por chamada em 2.** Adiciona uma pulseira à base de dados
 - a. **Use o link abaixo como redirect!!!! A própria fitbit vai chamar GET de 2. tirando necessidade de chamar pelo front.**
 - b. No corpo da requisição devem estar presentes `codigoOAuth` e `redirectUri`. Informações adicionais sobre estas variáveis podem ser encontradas em <https://dev.fitbit.com/apps/oauthinteractivetutorial>.
 - c. O `redirectUri` é "<http://julianop.com.br:3000/>" a menos que comunicados do contrário pela equipe do back end. Já o `codigoOAuth` é o código obtido ao acessar o único link presente na página citada em a. **(E TAMBÉM PRESENTE ABAIXO):**

Link:

https://www.fitbit.com/oauth2/authorize?response_type=code&client_id=227WRB&redirect_uri=http%3A%2F%2Fjulianop.com.br%3A3000%2Fapi%2Fpulseira%2Fcodigo&scope=activity%20heartrate%20location%20nutrition%20profile%20settings%20sleep%20social%20weight&expires_in=604800

2. **GET em "url/api/pulseira/codigo" -> chamada exclusiva da fitbit, realiza o processo de cadastro de pulseira**

3. **PUT em "url/api/pulseira" -> Edição de pulseira (alocar ou desalocar pulseira)**
 - a. O corpo da requisição deve conter os parâmetros **idPulseira** e **disponivel** obrigatoriamente, sendo que respectivamente um é o id da pulseira a ser editada e o outro é uma booleana que escolhe a ação a ser realizada.
 - b. Caso **disponivel** tenha valor 1 (true), a chamada vai liberar a pulseira do paciente, impossibilitando armazenagem de dados de saúde a partir daí. Se por acaso **disponivel = 0** (false), deve-se acrescentar aos parâmetros do corpo da requisição também **idPaciente** para indicar o paciente que irá usar a pulseira.

4. GET em "url/api/pulseira/disponivel" -> Retorna um array com os id's das pulseiras disponíveis

. **DELETE em "url/api/pulseira"** -> remove pulseira. A chamada leva id da pulseira a ser deletada

5. GET em "url/api/pulseira/idPaciente/:pacienteAtual" -> retorna um array de um único objeto que contém o id da pulseira que o paciente usa atualmente ou um array vazio caso o paciente não esteja usando pulseira no momento.

Paciente

1. POST em "url/api/paciente/geral" -> Adição de paciente na base de dados

- a. parâmetros no corpo da requisição: nomePaciente, causaDaInternacao, numeroDoProntuario, telefone, foto, dataDeNascimento (formato yyyy-mm-dd), idMedico.
- b. A maioria dos parâmetros é informação pessoal do paciente, porém menções especiais ao idMedico, que deve ser um id válido de médico.
 - idPulseira removido dessa chamada, usar PUT de Pulseira para alocá-la após criação de paciente
- c. Essa chamada pode ser acompanhada de PUT's da API pulseira para alocar a pulseira de id especificado ao paciente que acabou de ser criado.

2. PUT em "url/api/paciente/geral" -> Editar perfil de paciente na base de dados

- a. Edição de pacientes edita os parâmetros no corpo da requisição e não muda as informações que não vierem.
- b. idpaciente (reparem bem no p minúsculo) do perfil a ser editado deve ser colocado no header da requisição
- c. Parâmetros que podem ser editados sendo colocados no corpo da requisição: nomePaciente, numeroDoProntuario, telefone, foto, causaDaInternacao, dataDeNascimento, ativo.
- d. O campo ativo é uma booleana que determina se o paciente está ou não em tratamento (independente de uso de pulseira). Pacientes inativos permanecem nos registros com finalidade de histórico, mas já foram curados (receberam alta).
- e. Edição de pulseira desabilitada por aqui, favor usar PUT de pulseira para tal.

3. DELETE em "url/api/paciente/geral" -> Deletar paciente da base de dados

- . Recebe idPaciente no corpo da requisição para remover paciente da base de dados.
- a. MÉTODO DESCONTINUADO (não funciona mais por conta de acessos ao perfil para múltiplos medicos) -> Recomenda-se usar PUT na API de pulseiras com variável 'disponivel' = 1, para dar alta ao paciente.

4. GET em "url/api/paciente/geral/idMedico/:idMedico" -> Puxar um array de pacientes ATIVOS atualmente associados a um médico.

O propósito primário desse método é fornecer informações para página 'Lista de Pacientes' do app.

5. **GET em "url/api/paciente/geral/inativo/idMedico:idMedico"** -> Puxar um array de pacientes INATIVOS atualmente associados a um médico.
O propósito primário desse método é fornecer informações para página 'Lista de Pacientes' do app.
6. **GET em "url/api/paciente/health/static/:idPaciente"** -> puxa todos os dados estáticos de saúde do paciente de id especificado com ':' no url acima.
7. **GET em "url/api/paciente/health/static/:idPaciente/:data"** -> Bem similar ao método acima, porém este retorna somente os dados relacionados ao dia especificado em :data.
8. **GET em "url/api/paciente/health/dynamic/:idPaciente/:data"** -> puxa dados dinâmicos do paciente de id especificado na data especificada

Médico

1. **POST em "url/api/medico"** -> Cria novo perfil médico na base de dados
 - a. Parâmetros obrigatórios no corpo da requisição: nomeMedico, especialidade, CRM, telefone, email, senha, CPF
2. **PUT em "url/api/medico"** -> Edita perfil de médico com idMedico presente no corpo da requisição mudando informações que também estiverem no corpo e mantendo as que não foram citadas no corpo.
. Parâmetros que podem ser usados na edição: nomeMedico, especialidade, telefone, CPF, CRM, email*.
 - a. Mudança de email ainda vai ser discutida tendo em vista a necessidade de confirmação do novo email.
3. **DELETE em "url/api/medico"** -> remove perfil de médico com o idMedico citado no corpo da requisição da base de dados junto com todas as menções a este perfil
 - a. **IMPORTANTE:** Ao se remover o perfil médico da base de dados se remove **TAMBÉM TODOS** os lembretes feitos por ele, relações de médico-paciente que contém ele e seus dados de login.
 - b. Não vi menção a este tipo de feature no aplicativo, bom sinal porque o código ainda carece funcionalidade para remoção de perfis de pacientes que não tem mais relação com nenhum médico registrado
4. **GET em "url/api/medico/busca/ID/:idMedico"** -> Essa requisição retorna as informações de perfil de um médico dado o valor de seu **:idMedico** no url acima.
5. **GET em "url/api/medico/busca/email:emailMedico"** -> Essa requisição retorna as informações de id de um médico dado o valor de seu **:emailMedico** no url acima.

Login

- 1. POST em "url/api/login" ->** Requisição de validação da conta do médico
 - a. Parâmetros obrigatórios no corpo da requisição: email e senha
 - b. Método retorna o id do médico ligado às informações acima se este existir.
- 2. PUT em "url/api/login/" ->** Leva email no corpo do request, verifica se o email está devidamente cadastrado e confirmado no app e, se tudo estiver de acordo, dispara um email para o endereço do corpo da requisição com o link para a página de mudança de senha.
- 3. GET em "url/api/login/senha/change/:idMedico" ->** NENHUMA AÇÃO NECESSÁRIA AQUI, este é meramente o link que é mandado por email ao se realizar uma chamada bem sucedida ao PUT acima.
- 4. POST em "url/api/login/mudarSenha" ->** NENHUMA AÇÃO NECESSÁRIA AQUI, esta chamada recebe os parâmetros do formulário enviado pela página web do GET acima e executa a mudança de senha caso o processo tenha corrido corretamente.

Lembretes

- 1. POST em "url/api/lembrete" ->** Cria novo lembrete na base de dados
 - a. Parâmetros obrigatórios no corpo da requisição: data, mensagem, idMedico, idPaciente.
- 2. PUT em "url/api/lembrete" ->** Edita lembrete na base de dados com idLembrete especificado no corpo da requisição.
 - . Por hora a edição de lembrete só permite a mudança na mensagem do lembrete, sendo este o único parâmetro editável na época da escrita desta documentação.
- 3. DELETE em "url/api/lembrete" ->** Remove lembrete da base de dados
 - . Bem simples, dados o idLembrete no corpo da requisição este método remove da base de dados o lembrete com id correspondente.
- 4. GET em "url/api/lembrete/:idPaciente" ->** Método retorna todos os lembretes associados ao paciente cujo id tem valor :idPaciente independente de quais médicos os criaram.

Grupos de paciente

- 1. GET em "url/api/grupoPacientes/buscarGrupo/idMedico/:id"Medico ->** Método retorna todos os grupos associados ao médico cujo id tem valor :idMedico.
 - a. Use este método para ter acesso a todos os grupos pertencentes a um médico.

2. **GET em "url/api/grupoPacientes/buscarGrupo/paciente/:idGrupoPac/:idMedico" -**
> Método retorna todos os pacientes associados ao grupo cujo id tem valor :idGrupoPac.
Use este método para ter acesso a todos os pacientes presentes naquele grupo.
3. **GET em "url/api/grupoPacientes/buscarGrupo/grupo/:nome" ->** Método retorna o idGrupoPac correspondente ao nome do grupo criado.
Caso múltiplos grupos tenham um mesmo nome, todos serão retornados ao mesmo tempo
4. **POST em "url/api/grupoPacientes" ->** Gera um grupo vazio de pacientes na base.
Parâmetros obrigatórios no corpo da requisição: Nome do grupo a ser criado
5. **PUT em "url/api/grupoPacientes" ->** Edita o nome do grupo na base de dados.
Como os grupos são apenas uma “interface” para um array de pacientes, a única coisa a ser editada aqui é o nome do grupo, obrigatório para a edição. Tanto o idGrupoPac quanto o novo nome do grupo deve ser fornecidos.
6. **DELETE em "url/api/grupoPacientes" ->** Remove grupo completamente da base de dados, inclusive para outros médicos.
Fornecendo o idGrupoPac do grupo a ser deletado, o grupo é deletado permanentemente da base de dados.
7. **POST em "url/api/grupoPacientes/pacientes" ->** Adiciona um paciente a um grupo de pacientes.
Parâmetros obrigatórios no corpo da requisição: Ids do paciente e grupo para realizar a operação (idPaciente e idGrupoPac);
 - a. Um paciente pode pertencer a mais que um grupo ao mesmo tempo.
8. **DELETE em "url/api/grupoPacientes/pacientes" ->** Remove um paciente do grupo de pacientes.
Fornecendo idGrupoPac e idPaciente, esta rota deleta aquele paciente específico do grupo desejado.
9. **POST em "url/api/grupoPacientes/pacientes/multiplos" ->** Adiciona vários paciente a um grupo de pacientes.
Parâmetros obrigatórios no corpo da requisição: Ids do paciente e grupo para realizar a operação (idPaciente e idGrupoPac);
 - a. A variável idPaciente deve ser um vetor de tamanho mínimo 1, contendo todos os pacientes a serem inseridos.
10. **POST em "url/api/grupoPacientes/medicos" ->** Adiciona um médico como responsável por um grupo de pacientes.
Parâmetros obrigatórios no corpo da requisição: Ids do paciente e grupo para realizar a operação (idMedico e idGrupoPac);
 - a. Um grupo pode ser monitorado por mais que um médico ao mesmo tempo.

11. DELETE em "url/api/grupoPacientes/medicos" -> Remove um médico como responsável do grupo de pacientes.

. Fornecendo idGrupoPac e idMedico, esta rota deleta aquele médico específico do grupo desejado.

Compartilhamento de Paciente

1. POST em "url/api/compartilhamento/paciente" -> Método adiciona diretamente uma relação entre o médico-alvo e o paciente desejados.

a. Parâmetros obrigatórios no corpo da requisição: idPaciente (Paciente a ser compartilhado), idHospitalOrigem (Hospital do médico que deseja compartilhar um paciente), idMedicoDestino(Medico alvo que receberá o novo paciente)

b. O compartilhamento apenas ocorre quando dois médicos trabalham no mesmo hospital. Tentativas de adicionar médicos que pertencem a hospitais diferentes não serão aceitas.

2. POST em "url/api/compartilhamento/grupo" -> Método adiciona diretamente uma relação entre o médico-alvo e o grupo de pacientes desejados.

. Parâmetros obrigatórios no corpo da requisição: idGrupoPac (Grupo de pacientes a ser compartilhado), idHospitalOrigem (Hospital do médico que deseja compartilhar um grupo), idMedicoDestino(Medico alvo que receberá o novo grupo)

a. O compartilhamento apenas ocorre quando dois médicos trabalham no mesmo hospital. Tentativas de adicionar médicos que pertencem a hospitais diferentes não serão aceitas.

Hospitais

1. POST em "url/api/hospitais" -> Gera um hospital vazio na base.

a. Parâmetros obrigatórios no corpo da requisição: Nome do hospital a ser criado

2. PUT em "url/api/hospitais" -> Edita o nome do hospital na base de dados.

. Como os hospitais são apenas uma "interface" para um array de médicos, a única coisa a ser editada aqui é o nome do hospital, obrigatório para a edição. Tanto o idHospital quanto o novo nome do hospital devem ser fornecidos.

3. DELETE em "url/api/hospitais" -> Remove hospital completamente da base de dados, inclusive para outros médicos.

. Fornecendo o idHospital do hospital a ser deletado, o hospital é deletado permanentemente da base de dados.

4. POST em "url/api/hospitais/relacoes" -> Adiciona um médico a um hospital.

. Parâmetros obrigatórios no corpo da requisição: Ids do Médico e do hospital a serem relacionados (idMedico e idHospital).

a. Um médico pode pertencer a mais que um hospital ao mesmo tempo.

5. **DELETE em "url/api/hospitais/relacoes"** -> Remove um médico do hospital.

- . Fornecendo idHospital e idMedico, esta rota deleta aquele médico específico do hospital desejado.

6. **GET em "url/api/hospitais/medico/:idMedico"**

- . Fornecendo o id do Médico, é retornada a lista de todos os hospitais onde aquele médico trabalha.

Feed de Atulizações

1. **GET em "url/api/feed/:idMedico"** -> Fornece todos os Itens do Feed (alterações no banco) de um médico ordenados pelo mais recente. O Retorno é um *Array* com os itens. Hoje, retorna atualizações feitas somente para PACIENTE e LEMBRETE.

- o Um item possui o campo *type*, que indica o tipo do item, podendo ter o valor "Paciente" ou "Lembrete".
- o Possui também o campo *timestamp*, que é o horário que aconteceu a última atualização. Acredito que esse horário não esteja em GMT -3.
- o Caso o campo *type* assuma o valor "Paciente", um campo "patient" estará presente. Esse campo é um objeto que possui todos os dados do PACIENTE, iguais ao que estão no banco de dados.

i.Exemplo de Item do Feed para PACIENTE:

```
1. {
  "type": "Paciente",
  "timestamp": "2017-05-08T23:42:44.000Z",
  "patient": {
    "idtable1": 90,
    "nomePaciente": "Vitor",
    "numeroDoProntuario": 1111,
    "telefone": "HUB",
    "foto": "MTgzMDAzNDE0NzkyODEyMjQ=",
    "causaDaInternacao": "Provas",
    "dataDeNascimento": "2017-04-05T03:00:00.000Z",
    "ativo": 0,
    "timestamp": "2017-05-08T23:42:44.000Z"
  }
}
```

- o Caso o campo *type* assuma o valor "Lembrete", um campo "reminder" estará presente. Esse campo é um objeto que possui todos os dados do LEMBRETE, iguais ao que estão no banco de dados.

.Exemplo de Item do Feed para LEMBRETE:

```
1. {
  "type": "Lembrete",
  "timestamp": "2017-05-09T00:30:47.000Z",
  "reminder": {
    "id": 34,
    "idMedico": 20,
  }
}
```

```
"mensagem": "Internação no quarto atualizado",
"data": "2017-03-07T03:00:00.000Z",
"K": 0,
"Na": 0,
"Cl": 0,
"Co2": 0,
"Bun": 0,
"Creat": 0,
"Gluc": 0,
"wcb": 0,
"HgB": 0,
"Hct": 0,
"Plt": 0,
"idPaciente": 39,
"timestamp": "2017-05-09T00:30:47.000Z"
}
```

- Parâmetros opcionais na requisição:

.limit: Limita a quantidade de itens a serem retornados.

i.offset: Indica a partir de qual item o primeiro resultado deve ser buscado.

ii.Exemplos:

1. **"url/api/feed/:idMedico?limit=10"** - Retorna somente os 10 primeiros itens.
2. **"url/api/feed/:idMedico?offset=10"** - Retorna todos os itens a partir do 11o resultado.
3. Caso não haja resultados o retorno é um *Array* vazio - `[]`.
4. **Exemplo para ir pedindo de 10 em 10 itens:**
 - a. **"url/api/feed/:idMedico?offset=0&limit=10"** - Retorna os primeiros 10 resultados.
 - b. **"url/api/feed/:idMedico?offset=10&limit=10"** - Retorna 10 resultados, começando a partir do 11o.
 - c. **"url/api/feed/:idMedico?offset=20&limit=10"** - Retorna 10 resultados, começando a partir do 21o.