

**Uso de SmartWatch no Auxílio a Monitoração de Arritmias Cardíacas****Usage of SmartWatch to Assist Monitoring of Cardiac Arrhythmias**

DOI:10.34117/bjdv6n10-108

Recebimento dos originais:01/10/2020

Aceitação para publicação:06/10/2020

**Leoni Kurt G. N. M. da Silva**

Bacharel em Ciência da Computação pela UNICAP

Instituição: Stefanini Document Solutions

Endereço: Avenida, Cais do Apolo, 222 – 7º andar, Recife – PE, Brasil

E-mail: leonikurt@gmail.com

**Sérgio Murilo Maciel Fernandes**

Doutor em Ciência da Computação pela UFPE

Instituição: Universidade Católica de Pernambuco – UNICAP

Endereço: Rua do Príncipe, 526 –Boa Vista, Recife –PE, Brasil

E-mail: sergio.murilo@unicap.br

**Robson Cavalcanti Lins**

Doutor em Ciência da Computação pela UFPE

Instituição: Universidade Católica de Pernambuco – UNICAP

Endereço: Rua do Príncipe, 526 –Boa Vista, Recife –PE, Brasil

E-mail: robson.lins@unicap.br

**RESUMO**

Doenças cardiovasculares são as que mais matam no mundo, e dentre estas as mais comuns são as arritmias cardíacas. O socorro imediato desses pacientes se faz necessário. Com a popularização dos Smartwatches, esses dispositivos têm se tornado uma ferramenta alternativa para a monitoração desses indivíduos. Assim, o objetivo desse trabalho é auxiliar no monitoramento constante desses pacientes, utilizando um Smartwatch para medir sua frequência cardíaca (FC), e gerando um alerta a um contato de emergência ao detectar alguma anormalidade. Para isso foi desenvolvido o Sistema de Monitoramento de Arritmias Cardíacas que consiste em um Back-end em Node.JS para o desenvolvimento da API e de MongoDB para o banco de dados, e como Front-end um aplicativo Android. Dessa maneira, o sistema possibilita um rápido socorro à indivíduos com arritmia. E ainda gera um histórico de FC enquanto o paciente executa diferentes atividades diárias gerando um diagnóstico mais assertivo.

**Palavras-chave:** Relógio Inteligente, Arritmia Cardíaca, Aplicativo.

**ABSTRACT**

Cardiovascular diseases are the most deadly in the world, and among these the most common are cardiac arrhythmias. The immediate assistance of these patients is necessary. With the popularization of Smartwatches, these devices have become an alternative tool for monitoring these individuals. So, the objective of this work is to assist in the constant monitoring of these patients, using a Smartwatch to measure their heart rate (HR), and generating an alert to an emergency contact when detecting any abnormality. For this, the Cardiac Arrhythmia

Monitoring System was developed, which consists in a Back-end in Node.JS for the development of the API and MongoDB for the database, and an Android application made in Java as the Front-end. In this way, the system allows quick assistance to individuals with arrhythmia. It also generates a history of HR while the patient performs different daily activities for a more assertive diagnostic.

**Keywords:** Smartwatch, Cardiac Arrhythmia, App.

## 1 INTRODUÇÃO

As doenças cardiovasculares são um grupo de doenças do coração e dos vasos sanguíneos, que segundo a Organização Mundial de Saúde (OMS), são as enfermidades que mais matam no mundo atualmente (WHO, 2020).

Dentre essas doenças, as mais comuns são as arritmias cardíacas, que acometem mais de 20 milhões de brasileiros e geram mais de 300 mil mortes súbitas ao ano no país, segundo a Sociedade Brasileira de Arritmias Cardíacas (2020). Arritmias cardíacas são doenças caracterizadas pela má formação e condução irregular do impulso elétrico do coração, fazendo com que ele pulse mais acelerado (taquicardia) ou mais devagar (bradicardia) (SOBRAC, 2020). A frequência cardíaca em um indivíduo saudável e em repouso é em média 75 batimentos por minuto (bpm) (Guyton, 2006). Taquicardia caracteriza-se por uma frequência cardíaca acima de 100 bpm, podendo ser indício de uma insuficiência cardíaca (Solimene, 1985). Já a bradicardia caracteriza-se por frequência cardíaca inferior a 50 bpm (Solimene, 1985).

Segundo a Associação Americana do Coração (AHA) para determinar a normalidade – ou anormalidade - da frequência cardíaca durante um esforço ou realização de atividade física, deve ser levada em consideração a idade (Tabela 1). De acordo com a AHA, durante o exercício, a frequência cardíaca deve ficar entre 50% e 85% do seu valor máximo. Comumente o valor máximo de cada indivíduo é calculado subtraindo a sua idade do valor máximo geral (220 bpm). Valores de frequência cardíaca abaixo ou acima dessa faixa são consideradas arritmias (Epstein *et al.*, 2008).

**Tabela 1.** Exemplo de faixa de frequência cardíaca normal, durante esforço físico

Idade	Frequência Cardíaca (bpm)
20	100 - 170
40	90 - 153
60	80 - 136

Com o ritmo descompassado, o coração não consegue bombear eficientemente o sangue para o restante do corpo, o que pode ocasionar até um desmaio, necessitando de um socorro

imediatamente. Em indivíduos idosos e/ou com comorbidade é importante um sistema para a monitoração diária da sua frequência cardíaca (FC).

Os dispositivos vestíveis têm sido cada vez mais utilizados como alternativa para a monitoração contínua destes indivíduos. Dentre esses dispositivos, os *Smartwatches* (relógios inteligentes) têm se tornado cada vez mais populares devido às suas inúmeras funções como, por exemplo, monitoramento da qualidade do sono, aferição da FC, atendimento de telefonemas e recebimento de notificações. Os dados fisiológicos coletados do usuário por esse tipo de dispositivo vêm sendo estudados e utilizados em diferentes tipos de tecnologias assistivas, como em Pandelo (2016) que utilizou *Smartwatch* para detectar e alertar quedas humanas. Silva (2017) realizou uma revisão da literatura quanto a diferentes tecnologias vestíveis para o auxílio de pessoas com deficiência visual.

Tendo em vista a importância que seja prestado socorro ao indivíduo com sintomas de arritmia o mais rápido possível, e que esses sintomas possam deixar o indivíduo incapaz de solicitar ajuda, se faz necessário o desenvolvimento de sistemas que identifiquem a ocorrência do sintoma e gere um alerta para algum familiar, um cuidador ou um profissional de saúde.

Assim, este trabalho tem por objetivo auxiliar no diagnóstico durante as atividades diárias, e monitoração constante de pacientes com histórico de cardiopatia. Para isso será utilizado um *Smartwatch* para aferir a FC do usuário e notificar automaticamente, por meio de um aplicativo, um contato de emergência cadastrado.

## 2 FUNDAMENTAÇÃO TEÓRICA

Antes da explanação da metodologia utilizada neste trabalho serão introduzidos alguns conceitos básicos para o entendimento dos métodos.

### 2.1 NODE.JS

A arquitetura *Blocking-Thread* possui como característica a paralisação de um processamento enquanto utiliza um I/O no servidor. Diversos sistemas para web possuem este tipo de arquitetura (Pereira, 2014). Alguns dos grandes problemas dessa arquitetura são: dispêndio de tempo, pelo fato de bloquear o sistema enquanto um processo não é finalizado; e um consumo maior de memória e de CPU (Pereira, 2014).

Tendo em vista os problemas desse tipo de arquitetura, Ryan Dahl criou o Node.JS com a arquitetura *Non-Blocking Thread* (Pereira, 2014). Dahl utilizou a ideia de *Single-Thread*, onde a execução de um aplicativo Node.Js ocorre em um único processo, sem a necessidade de um novo encadeamento para solicitações futuras (Node.JS, 2020). Além disso, também foi utilizado I/O

assíncrono nas suas bibliotecas, ou seja, ao invés de aguardar o retorno de uma solicitação (seja um envio de e-mail, um acesso ao banco, entre outros) ele retoma a operação apenas quando obtiver esse retorno (Node.JS, 2020). E geralmente, as bibliotecas são gravadas com paradigmas não bloqueantes, fazendo que um bloqueio seja tratado como uma exceção (Node.JS, 2020).

O Node.JS utiliza a linguagem de programação *JavaScript*, que é uma das mais utilizadas em todo o mundo (Cantelon *et al.*, 2014). Essa popularidade apresenta muitas vantagens, entre elas: aplicações web podem utilizar a mesma linguagem, sem ter a necessidade de troca de contexto entre cliente e servidor; é possível utilizar o JSON (*JavaScript Object Notation*) nativamente; e *JavaScript* está presente em vários bancos de dados NoSQL como, por exemplo, o *shell* do MongoDB, facilitando a comunicação do Node.JS com esses bancos (Cantelon *et al.*, 2014).

## 2.2 MONGODB

Em 1970 o matemático Frank Codd introduziu o modelo de dados relacional, que seria baseado em uma estrutura de dados como uma coleção de relações e com sua fundamentação teórica baseada na álgebra relacional (Spoto, 2000). A linguagem SQL tem sido amplamente utilizada como a linguagem padrão para manipular os dados nos bancos relacionais, por oferecer uma linguagem matemática que facilita e melhora o desempenho das consultas (Oracle, 2020).

Em 1998 foi a primeira vez que o termo NoSQL foi utilizado para citar um banco de dados que não utilizaria a linguagem SQL, mas ainda estaria utilizando a arquitetura relacional (Aniceto *et al.*, 2014). Com o passar do tempo, o termo veio se popularizando e em 2009 passou a ser utilizado para referenciar um banco de dados não relacional (Aniceto *et al.*, 2014). O uso do esquema de tabela de linhas e colunas que podem ser encontrados nos bancos de dados tradicionais não são utilizados em um banco não relacional (Microsoft, 2018). Nele é utilizado um modelo de armazenamento otimizado como pares chave/valor, e documentos JSON ou grafo (Microsoft, 2018).

O MongoDB é um exemplo de banco de dados NoSQL, de código aberto, que é orientado a documentos e possui alta escalabilidade e flexibilidade (Mongodb, 2020). A estrutura de armazenamento utilizada é a versão binária do JSON, o BSON (Binary JSON), no qual são permitidas várias estruturas de dados como *arrays* e objetos, como acontece no JSON (Abramova, 2013).

Em aplicações que a prioridade é velocidade, os bancos não relacionais superam os relacionais, como na análise apresentada por Silva e Ferreira (2017), que compararam o MongoDB e o SQL Server.

### 2.3 FIREBASE

O FireBase é uma plataforma de desenvolvimento de aplicativos para dispositivos móveis e web, que foi comprada pela Google desde 2014. Essa plataforma oferece diversos serviços, os mais utilizados são (FireBase, 2020): *Analytics*, *Realtime database*, *Authentication*, *Cloud Messaging* e *Storage*.

O serviço *Cloud Messaging* oferece o envio de mensagens e notificações gratuitamente para usuários de várias plataformas (Android, iOS ou Web). As mensagens podem ser enviadas para dispositivos únicos, grupos de dispositivos, segmentos de usuários ou tópicos específicos. Neste trabalho o FireBase foi utilizado apenas como notificador, enviando mensagens de alerta ao usuário cadastrado como cuidador, ou seja, serviço *Cloud Messaging*. Mesmo o FireBase possuindo seu serviço de banco de dados, o MongoDB foi selecionado por sua facilidade de se conectar ao Node.js. A dependência Mongoose, facilita a comunicação, trazendo uma melhor fluidez.

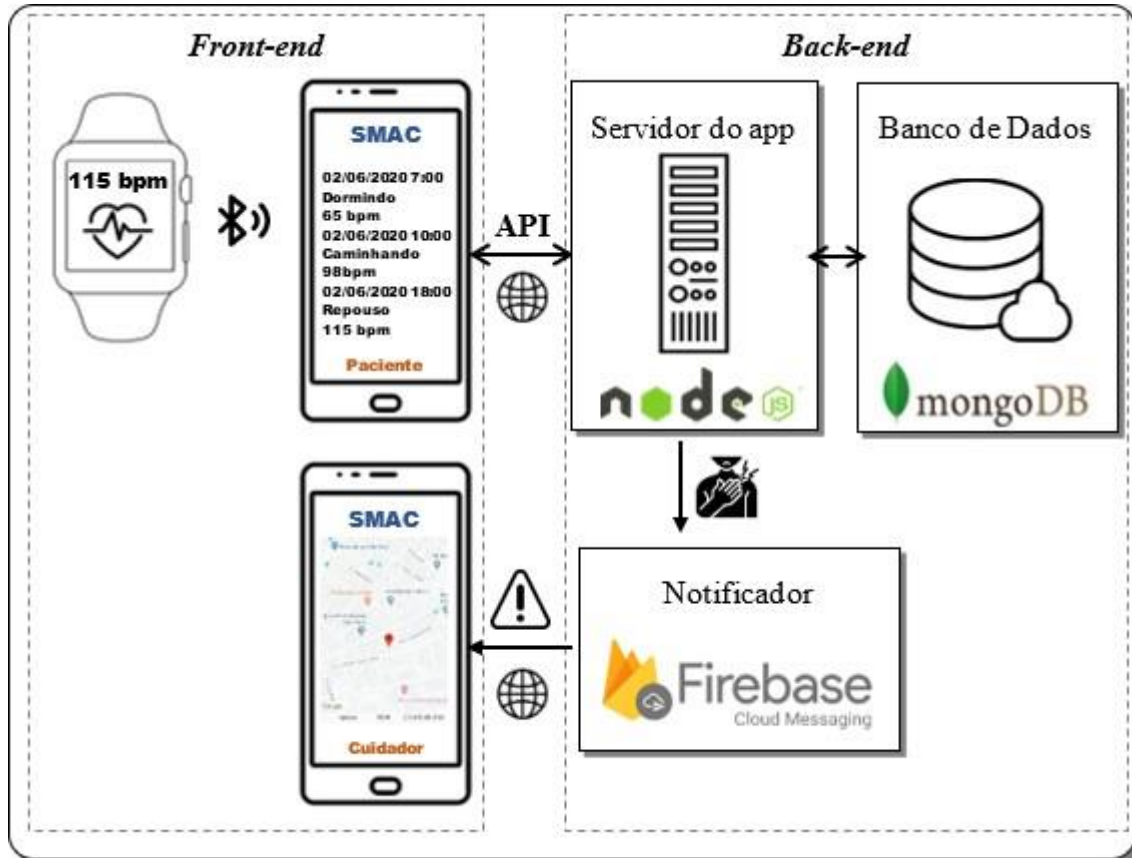
### 3 METODOLOGIA

O sistema de monitoramento de arritmias cardíacas (SMAC) desenvolvido neste trabalho consiste em um arranjo de *back-end* com diferentes tecnologias, que se comunica com um *front-end* em forma de aplicativo (app) para *smartphone*. Esse app recebe os dados de FC de um indivíduo, captados por um *Smartwatch*. O diagrama geral do SMAC pode ser visualizado na Figura 1.

O dispositivo vestível, *Smartwatch*, foi emulado para que os testes de validação do sistema fossem realizados. Para a emulação foi utilizado o emulador do AndroidStudio, a mesma ferramenta que foi utilizada para desenvolver o *Front-end* do aplicativo. Para o emulador se comunicar com o aplicativo, foi necessário conectar um *SmartPhone* ao computador via porta USB. No *SmartPhone* é necessário que o modo de depuração pela USB esteja habilitado, o que possibilita a configuração da porta do *Smartwatch* para apontar para a USB.

Com o *SmartPhone* e emulador conectados, consegue-se realizar a comunicação do aplicativo com o *Smartwatch* emulado, com isso, acionar o sensor de medição de FC. Pelo fato de ser um emulador, não é possível obter um valor de FC real, então, os valores passados para o aplicativo foram gerados manualmente para a realização dos testes.

Figura 1. Diagrama do Sistema de Monitoramento de Arritmias Cardíacas (SMAC)



### 3.1 BACK-END

A arquitetura do sistema desenvolvido foi dividida em *Back-end* e *Front-end*. Para o desenvolvimento do *Back-end* foram utilizadas as seguintes tecnologias: Node.JS como servidor da aplicação; MongoDB como banco de dados, que irá armazenar todas as informações coletadas pelo aplicativo; e FireBase para realizar o envio de notificações.

O Node.JS é o responsável pela parte do servidor da aplicação, onde foram desenvolvidas as APIs (*Application Programming Interface*) para que o *Front-end* possa consumir. Para operacionalizar esse processo, foram criadas rotas para o *Front-end* realizar requisições HTTP (*Hypertext Transfer Protocol*), com o objetivo de trocar (enviar ou receber) informações com o *Back-end*. Com essas informações, o *Back-end* fica encarregado de inserir, coletar ou remover informações do banco de dados. No caso de uma inserção, por exemplo, o *Front-end* envia no corpo da sua requisição as informações necessárias para inserir um registro no banco de dados. Por exemplo, no cadastro de um novo usuário o corpo da requisição deve conter as informações do usuário, como: nome, idade, CPF, telefone e e-mail. Para mostrar o histórico da FC do usuário, é necessário que o *Front-end* envie apenas o Id associado ao usuário no banco de dados, de modo que o *Back-end* possa realizar uma solicitação de busca no banco de dados. Após essa busca, o



*Back-end* envia, em formato JSON, uma lista contendo os registros salvos de quando o dispositivo aferiu a FC do indivíduo, contendo: a data e o horário da medição; a FC; e a atividade que o usuário estava realizando no momento da medição, a qual pode ser selecionada no aplicativo.

Existem duas formas de operacionalizar o banco de dados MongoDB: utilizando o navegador, ou baixando a aplicação MongoDB *Compass Community* para realizar o gerenciamento. Para esta aplicação, foi feito o *download* do Atlas, e em seguida a criação da conta para se conectar ao Node.JS. Dentro do *Compass*, são disponibilizadas a chave e a URL para conexão. Na parte do Node.JS, foi instalado a dependência Mongoose para a conexão com o MongoDB e para realizar as operações de inserir, buscar e remover informações. No banco, foram criadas duas *collections*, sendo elas, uma para os usuários e a outra para os históricos de FC.

Ainda, utilizou-se o FireBase, que ficou responsável por enviar uma notificação para o celular do contato de emergência do usuário quando detectada uma anormalidade na FC do usuário. Para isso, foi criada uma conta no FireBase e configurado o serviço de *Push Notification* no FireBase *Cloud Messaging*. Tendo o serviço configurado, é necessário salvar no banco de dados o código do celular do usuário. Isso é realizado quando o usuário cria uma conta no aplicativo e sempre que realiza o *login*. Tendo o código do celular, é realizada uma requisição à API do FireBase, onde são enviados: o token gerado na sua conta do FireBase para poder se conectar; o código do celular do usuário, que está salvo no banco de dados; e o corpo da mensagem, onde deve conter o título e o conteúdo da notificação.

### 3.2 FRONT-END

*Front-end* é a parte da aplicação que interage diretamente com o usuário. Como *Front-end* do SMAC foi desenvolvido um aplicativo móvel para dispositivos Android. O Android é o sistema operacional móvel da Google e o mais utilizado no mundo, estando disponível para diversas plataformas, como *smartphones* e tablets, TV (Google TV), relógios (*AndroidWear*), óculos (Google Glass) e carros (Android Auto) (Lecheta, 2016). Além disso, o Android é uma plataforma para aplicações móveis completamente livre e de código aberto (*open source*), este sistema operacional possui uma grande vantagem competitiva, uma vez que desenvolvedores do mundo inteiro podem contribuir para seu aprimoramento.

O aplicativo foi desenvolvido para *smartphones* com o sistema operacional Android, e que se comunica com *Smartwatches* que possuem o sistema operacional *AndroidWear*. Para o desenvolvimento desse aplicativo, foi utilizada a linguagem de programação JAVA, que no

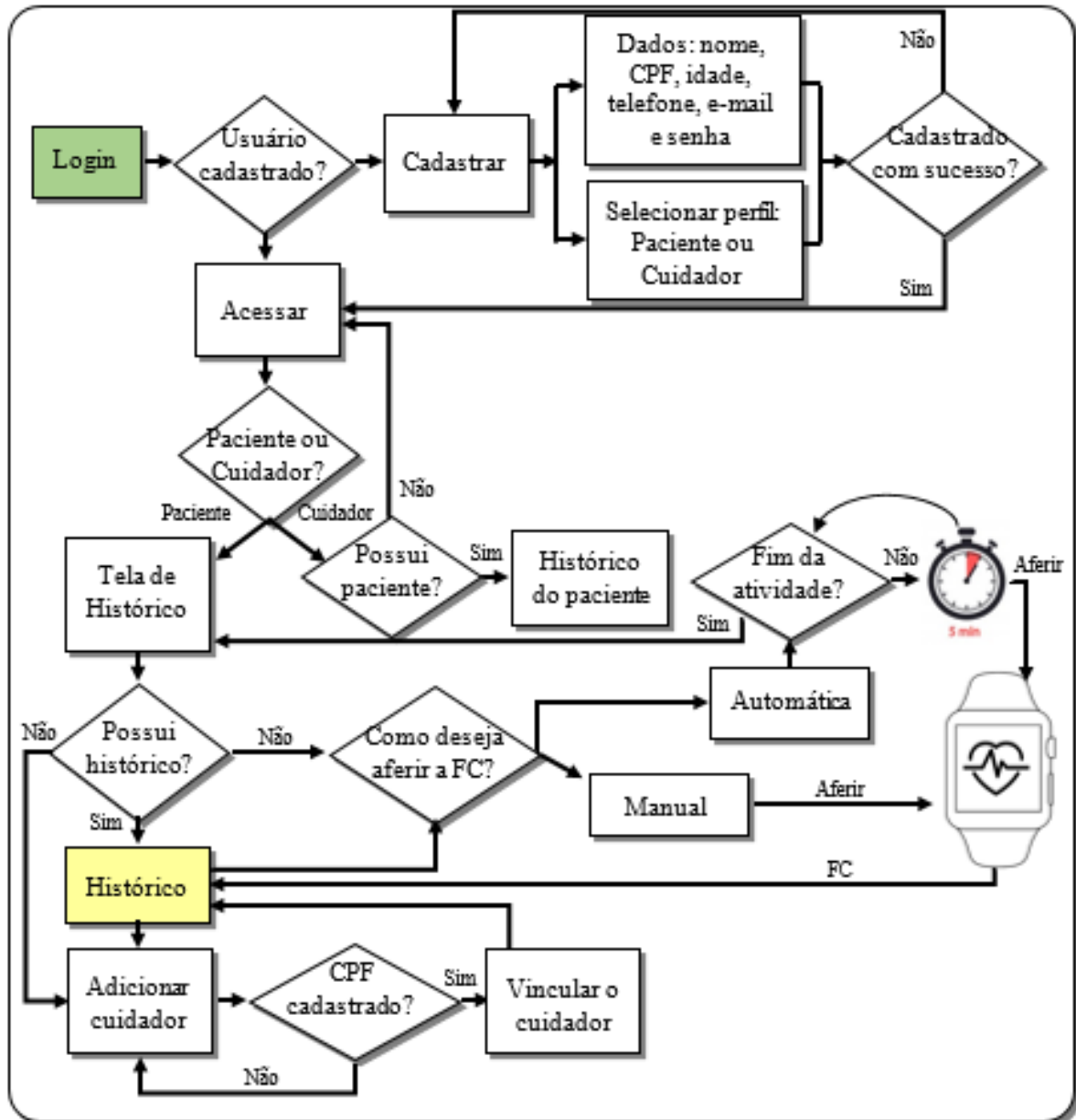
desenvolvimento para Android é a mais popular. Sendo assim, possui uma documentação completa e diversos fóruns para ajudar a desenvolver uma aplicação.

O aplicativo possui uma interface homem-máquina de fácil manipulação pelo usuário. O fluxograma que resume toda a programação do app pode ser visualizado na Figura 2. Ao iniciar o aplicativo uma tela de *login* é exibida, onde o usuário poderá selecionar a opção de acesso ou cadastro. Em um novo cadastro o usuário precisa preencher os campos obrigatórios para que o cadastro seja finalizado com sucesso. Para isso, é necessário preencher o nome, idade, CPF, telefone, e-mail e senha, e selecionar o seu tipo de perfil: paciente ou cuidador. Caso o usuário já seja cadastrado, basta acessar utilizando seu número de CPF e senha. O aplicativo possui funcionalidades diferentes para os dois tipos de usuários:

- Paciente: é possível obter acesso ao histórico de frequências cardíacas adquiridas; medir sua FC; adicionar um cuidador para monitorar seu histórico; e indicar a atividade diária (repouso, dormir, comer, dirigir, pedalar, correr, subir escada, exercício físico, ou caminhar) que está realizando no momento da aferição, além de detalhes sobre a atividade.
- Cuidador: permite acessar o histórico de frequências cardíacas do paciente ao qual está vinculado; receber notificação quando o paciente apresentar anormalidade na FC; e acessar a localização do paciente por mapa, em caso de emergência.



Figura 2. Fluxograma da programação do app do SMAC



O aplicativo é responsável pela comunicação com o *Smartwatch*, podendo solicitar que o dispositivo acione o sensor responsável por aferir a FC do usuário. Após aferir, o *Smartwatch* retorna o valor da FC para o app. A medição da FC pode ser configurada de dois modos:

- Manual: aciona uma única medição da FC pelo *Smartwatch*, por exemplo, ao usuário sentir algum desconforto ou dor no peito;
- Automática: aciona repetidas medições da FC pelo *Smartwatch*, com uma periodicidade selecionada pelo usuário. A sequência de medições só é finalizada quando o usuário clica no botão de parar. Assim, esse modo permite a monitoração da FC durante determinada atividade diária.

Quando o aplicativo verifica uma anormalidade na FC, automaticamente é enviada uma requisição ao *Back-end* para enviar um alerta ao contato de emergência (Cuidador) cadastrado para aquele usuário. O Cuidador recebe a notificação no seu celular, e no aplicativo recebe as informações quanto ao ocorrido e a localização do usuário, para que possa socorrê-lo de forma mais rápida.

Para obter a localização do usuário o app utiliza uma API da Google, a *Geolocation* API. Essa API é responsável por descobrir a localização através de requisições via internet com uma acurácia de 10 metros.

Além de auxiliar o usuário na medição da FC e disparar o alerta para um contato de emergência, no aplicativo é possível obter o histórico das últimas 30 medições que o usuário realizou, podendo ser filtrado selecionando um intervalo de dias. Isso pode auxiliar em uma consulta médica, onde o médico poderá acompanhar a FC do paciente durante suas atividades diárias.

#### **4 RESULTADOS**

O Sistema de monitoração de arritmias cardíacas foi desenvolvido, e para realização de sua validação foram instalados o servidor da aplicação e o banco de dados em um notebook. O app foi instalado em 2 smartphones: um Asus Zenfone 5 com Android 9.0, que foi utilizado como usuário do tipo paciente; e um Samsung Galaxy J6 com Android 10.0, que foi utilizado como usuário do tipo cuidador.

Após instalar o app em cada dispositivo, foram solicitados os cadastros de cada usuário na tela de Login (Figura 3a), e na tela de Cadastro do Usuário (Figura 3b) as informações dos usuários, que foram inseridas com sucesso. Utilizando o usuário Paciente vinculou-se o seu cuidador (Figura 3c). Em seguida, na tela de Aferir a frequência cardíaca (Figura 4a) foram realizados testes com cada tipo de atividade diária. Os testes foram realizados nos modos manual e automático. Como pode ser visualizado na Figura 4b os históricos dos testes foram gerados e mostrados na Tela de Histórico.

Todo o *back-end* funcionou conforme projetado, e podem ser visualizados exemplos de informações dos usuários (Figura 5a), e os registros de aferição de frequência cardíaca (Figura 5b) salvos no banco de dados.

Figura 3. Telas do SMAC: (a) login, (b) cadastro de usuário e (c) vincular cuidador

(a) Login

CPF

Senha

CADASTRAR ACESSAR

(b) Cadastro de Usuário

Nome

CPF

Idade

Telefone

Email

Senha

SALVAR

(c) Adicionar responsável

CPF

Seleção de opção

Paciente

Cuidador

Figura 4. Telas do SMAC: (a) seleção de tipo de atividade diária e (b) histórico

(a) Aferir frequência cardíaca

Correndo

Correndo no Parque da Jaqueira

Aferir automaticamente

Tempo para aferir automaticamente

1 minuto

2 minutos

5 minutos

10 minutos

DATA INICIO

DATA FIM

(b) Histórico Cardíaco

Atividade: Dormindo Frequência: 50.0 Data: 12/06/2020 13:33 Descrição: Dormindo

Atividade: Pedalando Frequência: 140.0 Data: 12/06/2020 13:32 Descrição: Pedalando no parque

Atividade: Repouso Frequência: 75.0 Data: 12/06/2020 13:31 Descrição: Descanso

Atividade: Caminhando Frequência: 120.0 Data: 09/06/2020 22:09 Descrição: Caminhando até o trabalho

Atividade: Correndo Frequência: 120.0 Data: 09/06/2020 22:06 Descrição: Correndo - Automático

Atividade: Exercício físico Frequência: 120.0 Data: 09/06/2020 22:05 Descrição: Musculação

Atividade: Subindo escada Frequência: 120.0 Data: 09/06/2020 22:05 Descrição: Terceiro degrau

Atividade: Dormindo Frequência: 120.0 Data: 09/06/2020 22:04 Descrição: Zzzzz

Atividade: Pedalando Frequência: 120.0 Data: 09/06/2020 22:03 Descrição: Pedalando orla de Olinda

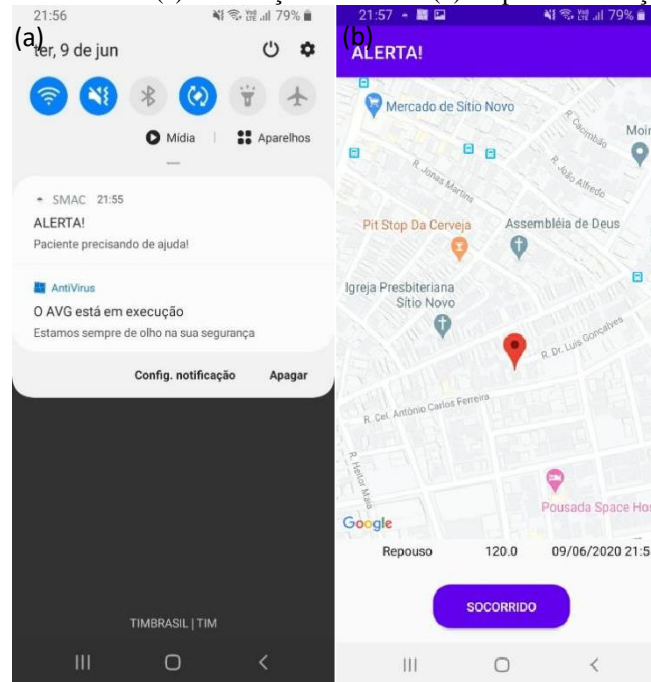
Atividade: Correndo Frequência: 120.0 Data: 09/06/2020 22:03 Descrição: Correndo no

Figura 5. Telas do banco de dados: (a) registros dos usuários e (b) registros de FC

(a)	<pre> _id: ObjectId("5ee02e8641e2cec71c7bf4b6") cell_phone: "81912345678" name: "Paciente" age: 22 cpf: "12345678901" email: "paciente@unicap.br" user_type: 1 password: "1" alert: false __v: 0 device_token: "cyVGADkg8cQ:APA91bEOB807NE" </pre>	<pre> _id: ObjectId("5ee02eb241e2cec71c7bf4b7") cell_phone: "32415547" name: "Cuidador" age: 38 cpf: "12345678910" email: "cuidador@unicap.br" user_type: 2 password: "1234" alert: false __v: 0 device_token: "d07Q4QkQ2iwrV2yMUSvfr:APA9" emergency_contact: "12345678901" </pre>
(b)	<pre> _id: ObjectId("5ee02f9b41e2cec71c7bf4b9") user_id: "5ee02e8641e2cec71c7bf4b6" heart_rate: 120 type: "Reposo" description: "Sentado na sala" date: 2020-06-10T00:55:55.362+00:00 __v: 0 </pre>	<pre> _id: ObjectId("5ee0315141e2cec71c7bf4ba") user_id: "5ee02e8641e2cec71c7bf4b6" heart_rate: 120 type: "Correndo" description: "Correndo no parque" date: 2020-06-10T01:03:13.550+00:00 __v: 0 </pre>

Após a instalação do aplicativo no smartphone do Cuidador, o dispositivo fica apto a receber mensagens de alerta do Firebase. Na Figura 6a pode ser visualizada a notificação recebida no dispositivo do cuidador. Ao clicar na notificação, o aplicativo é inicializado mostrando um mapa com a geolocalização do paciente, e as informações referentes ao ocorrido, como pode ser visualizado na Figura 6b. Clicando no botão “Socorrido”, o alerta é removido e o aplicativo volta ao seu fluxo normal.

Figura 6. Telas do SMAC: (a) notificação recebida e (b) mapa e informações do ocorrido.



A fim de observar o comportamento do SMAC, foram realizadas simulações de atividades diárias, podendo estar associados ou não à arritmia do usuário. Exemplos de simulações realizadas podem ser observadas na Tabela 2.

**Tabela 2.** Exemplos de simulações para a validação do SMAC

Simulação	Idade	Atividade	FC Normal por atividade	FC	Alerta
1	22	Repouso Dormir Comer Dirigir	50 - 100	120	Sim
2	46	Pedalar Correr Subir escada Exercício físico Caminhar	87 - 148	100	Não
3	63	Pedalar Correr Subir escada Exercício físico Caminhar	79 - 134	70	Sim

Por meio das simulações realizadas, pode-se observar o funcionamento do SMAC, que responde conforme projetado, distinguindo a frequência cardíaca normal de arritmias, durante as diferentes atividades diárias realizadas por indivíduos de diferentes faixas etárias.

## 5 CONCLUSÃO

Neste trabalho, foi proposto um sistema de monitoração de arritmias cardíacas, SMAC, que permite o acionamento de um *Smartwatch* para a medição da FC do paciente e envio via *Bluetooth* para um aplicativo em seu *smartphone*. Essas informações são armazenadas em um banco de dados em nuvem, e podem ser consultadas gerando um histórico da FC atrelada a atividade realizada pelo paciente. Esse histórico pode auxiliar o cardiologista do paciente para um tratamento mais assertivo, visto que a análise da FC é realizada enquanto o indivíduo realiza suas atividades diárias.

O aplicativo do SMAC também possui usuário com perfil de cuidador, que fica vinculado a um paciente. O app do paciente monitora sua FC e verifica se de acordo com sua idade e atividade que está realizando, encontra-se dentro da normalidade. Caso contrário, o app detectará taquicardia ou bradicardia e é gerado um alerta no app do cuidador, juntamente com a localização do paciente, para que o socorro seja prestado mais rapidamente.

Todas as simulações realizadas para a validação do sistema funcionaram conforme projetado. E como trabalhos futuros, pretende-se realizar testes com um *Smartwatch* físico, e também adicionar uma camada de segurança ao sistema, para proteger os dados contra possíveis invasões.

### REFERÊNCIAS

- Abramova, V., Bernardino, J. (2013) “NoSQL Databases: MongoDB vs Cassandra”, Instituto Politécnico de Coimbra, Coimbra, Portugal.
- Cantelon, M., Harter, M., Holowaychuk, T. J., Rajlich, N. (2014) Node.js in Action.
- Epstein, A. E. *et al.* (2008), ACC/AHA/HRS Guidelines for Device-Based Therapy of Cardiac Rhythm Abnormalities, Journal of the American College of Cardiology Volume 51, Issue 21, May 2008 DOI: 10.1016/j.jacc.2008.02.032.
- FireBase, Disponível em: <https://firebase.google.com/products?hl=pt-br>, acessado em: 02/06/2020.
- Guyton, H. (2006), Tratado de Fisiologia Médica, Tradução da 11ª Edição, Elsevier.
- Lecheta, R. R. (2016), Introdução ao Android. In: Lecheta, Ricardo R. Android Essencial. São Paulo: Novatec. Cap. 1, p. 17 – 18.
- Microsoft (2018), disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>>, acessado em: 17 de maio de 2020.
- MongoDB, disponível em: <<https://www.mongodb.com/what-is-mongodb>>, acessado em: 16 de maio de 2020.
- Node.JS, disponível em: <<https://nodejs.org/>>, acessado em: 16 de maio de 2020.
- Oracle, disponível em: <<https://www.oracle.com/br/database/what-is-a-relational-database>>, acessado em: 17 de maio de 2020.
- Pandelo, H. R. M. (2016) “IoT e Dispositivos Vestíveis Aplicados à Área da Saúde”, Trabalho de conclusão de curso, Bacharelado Em Sistemas De Informação, Centro Universitário Eurípides De Marília - Fundação De Ensino “Eurípides Soares Da Rocha”, Marília, São Paulo, Brasil.
- Pereira, C. R. (2014), Aplicações web real-time com Node.js. São Paulo: Casa do Código.
- Silva, G. A., Medola, F. O., Rodrigues, S. T. (2017) “Smartwatch: Um Possível Dispositivo Eletrônico Vestível de Tecnologia Assistiva”, 16º Ergodesign – Congresso Internacional de Ergonomia e Usabilidade de Interfaces Humano Tecnológica, Santa Catarina, Brasil.
- Silva, G. J., Ferreira, J. C. O. (2017), “Análise Comparativa de Desempenho de Consultas entre um Banco de Dados Relacional e um Banco de Dados Não Relacional”, trabalho de conclusão do curso de Engenharia da computação, Universidade de Uberaba, Paraná, Brasil.
- SOBRAC – Sociedade Brasileira de Arritmias Cardíacas, disponível em: <<https://sobrac.org/home/release-dia-mundial-do-coracao-e-a-necessidade-de-atencao-para-as-arritmias-cardiacas>>, acessado em: 13 de maio de 2020.

## **Brazilian Journal of Development**

Solimene, M. C. (1985) “Disritmias Cardíacas — Aspectos Clínicos E Terapêuticos”, Rev. Med., vol. 66, N-1:18 a 22.

Spoto, E. S. (2000) “Teste Estrutural De Programas De Aplicação De Banco De Dados Relacional”, Universidade Estadual De Campinas, Campinas, São Paulo, Brasil.

World Health Organization, disponível em: <[https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab\\_2](https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_2)>, acessado em: 13 de maio de 2020.