

DESENVOLVIMENTO DE SISTEMA WEB PARA REGISTRAR E MONITORAR SINAIS VITAIS DURANTE A ATIVIDADE FÍSICA DE PESSOAS COM DEFICIÊNCIA INTELECTUAL E MÚLTIPLA UTILIZANDO UMA PULSEIRA INTELIGENTE

Fernanda Di Queiroz Freitas*

Orientador: Adilson Jair Cardoso**

Resumo

O presente trabalho apresenta o desenvolvimento de um protótipo de uma aplicação *web* para armazenar dados da frequência cardíaca de uma pulseira inteligente, sendo seu desenvolvimento, na maior parte, realizado na linguagem em JavaScript, suas bases de dados em Knex, e PostgreSQL. O projeto inicia introduzindo o conceito de aplicação *web*, cuja conceituação é explanada para uma melhor compreensão da temática, e dos assuntos relacionados, auxiliando no desenvolvimento do protótipo. A tecnologia e as ferramentas implementadas no projeto têm como objetivo principal dar o suporte ao desenvolvimento do protótipo. Por fim, o desenvolvimento da aplicação resultou em um protótipo, a qual permite que usuários, alunos e atividades físicas sejam cadastrados e visualizados. Além disso, resultou-se também na identificação de UUIDs que correspondem aos serviços e características da frequência cardíaca, tornando possível a leitura desse parâmetro e a comunicação entre a aplicação e uma pulseira inteligente através do BLE.

Palavras-Chave: Aplicação Web. Bluetooth Low Energy. Banco de dados. VueJS.

1 INTRODUÇÃO

Segundo Casino et al., (2015) o envelhecimento da sociedade e a necessidade de promover hábitos saudáveis entre a população são grandes desafios para os sistemas de saúde pública. Esse setor tem sido sempre muito ativo na busca de novas tecnologias que possam ser integrados, de modo a melhorá-lo.

* Acadêmica do curso de Engenharia Mecatrônica do Instituto Federal de Santa Catarina. fernanda1queiroz44@gmail.com

** Professor Doutor do Instituto Federal de Santa Catarina. adilson.jair@ifsc.edu.br

Em todo o mundo, os modelos de cuidados de saúde estão constantemente mudando para abordagens centradas nos pacientes em que esses não são mais passivos e sim contribuintes ativos na sua própria saúde. Por trás desta mudança de paradigma, encontra-se um papel fundamental nas tecnologias de informação e comunicação no setor da saúde, em conjunto com a utilização de dispositivos móveis.

No contexto, os sistemas de informação tornaram-se uma das principais tecnologias computacionais capazes de propagar de maneira rápida, fácil e segura as informações necessárias para mediar a aprendizagem e auxiliar os profissionais no processo de trabalho. Estes, enquanto instrumentos de trabalho na saúde, são importantes recursos de apoio às ações administrativo burocráticas e àquelas ancoradas em conhecimentos técnico-científicos, sobretudo as que dependem de informações atualizadas. A tecnologia da informação representa, portanto, uma ferramenta que pode disponibilizar maior facilidade na realização dos trabalhos e incrementar a assistência prestada. O fator chave para a deficiente utilização dos dados na geração de informação é a falta de sistemas informatizados para processar dados e disponibilizar informações em um formato que seja facilmente compreendido pelas pessoas. O tempo despendido na recuperação das diversas informações do cuidado prestado ao paciente em sistema de registro manual compromete, sobremaneira, o processo de tomada de decisão (PARRO & MARTINEZ ÉVORA, 2013).

As tecnologias vestíveis - dispositivos inteligentes que o usuário literalmente veste e usa como se fosse um acessório - de vigilância da saúde podem auxiliar na busca por métodos mais rápidos e seguros para o armazenamento de informações da saúde. Estes são capazes de medir os parâmetros fisiológicos significativos como frequência cardíaca, pressão arterial, temperatura do corpo e da pele, saturação de oxigênio, frequência respiratória, eletrocardiograma, etc. As medições obtidas são comunicadas através de uma rede sem fio ou um link para um nó central, por exemplo, um assistente pessoal digital ou um microcontrolador, que pode por sua vez pode apresentar informações numa interface de usuário ou transmitir sinais vitais agregada a um centro médico (PANTELOPOULOS & BOURBAKIS, 2010).

Pessoas com deficiência, em geral, apresentam maior número de fatores de risco, pois, pela necessidade de modificarem seus hábitos de vida, possuem a tendência de aumentar usualmente de peso e tornam-se sedentários, o que favorece

ainda mais o desenvolvimento de doenças do aparelho cardiovascular (HADDAD ET AL., 2005). As atividades de alta intensidade implicam em maiores velocidades de movimento e maiores frequências cardíacas, aumentando os fatores de risco para intercorrências patológicas tanto musculoesqueléticas como cardiovasculares (SANTARÉM, 2001). No Brasil, as doenças cardiovasculares são uma das principais causas de morbimortalidade e ainda um grande desafio à saúde pública, especialmente por sua prevalência ao longo dos anos (MOREIRA, GOMES, & SANTOS, 2010).

Alinhado ao assunto, o objeto deste trabalho é um sistema de coleta dos dados de sinais vitais de uma pessoa. As informações coletadas serão utilizadas para o processo de tomada de decisão na área da saúde. A pergunta a qual o presente trabalho busca responder é: Como a utilização de uma pulseira inteligente pode auxiliar no monitoramento de sinais vitais na atividade física em pessoas com deficiência intelectual e múltipla?

Dessa forma, a Figura 1 abaixo evidencia o objetivo final do projeto, no qual tem como focos principais a comunicação da pulseira com o sistema *web* e sua interação com os usuários.

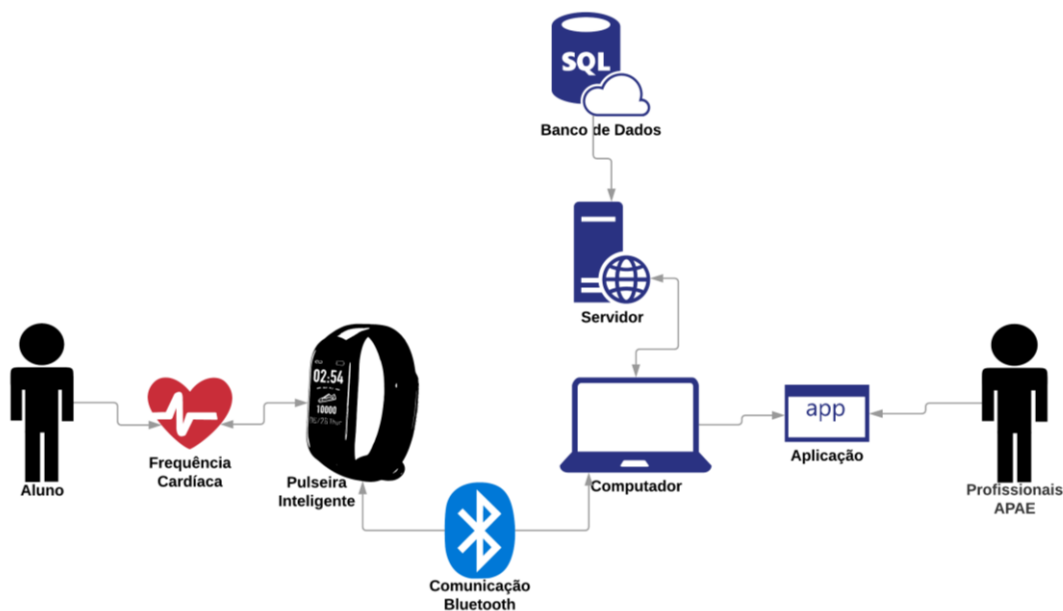


Figura 1 - Diagrama do sistema proposto.

Fonte: Elaborado pela autora.

1.1 Definição do Problema

Em 2018 foi realizado uma visita na APAE – Criciúma com o intuito de encontrar demandas tecnológicas na associação e assim criar soluções que possam

ser desenvolvidas no Instituto Federal de Santa Catarina – campus Criciúma. Uma das demandas encontradas foi a necessidade de monitorar a frequência cardíaca dos alunos durante as atividades físicas para avaliar o impacto de cada atividade em cada aluno.

1.2 Justificativa

Este trabalho final de curso justifica-se pela importância da coleta de dados de frequência cardíaca para uma futura análise da saúde do paciente. Tratando-se de um projeto voltado para suprir necessidades provenientes da comunidade local, onde os dados de entrada são informações subsequentes do desempenho de cada paciente durante uma atividade física específica.

Além disso, o trabalho busca identificar ou suprir informações antes não identificados no paciente em relação à saúde cardiovascular. Segundo Radovanovic et al., (2014) as Doenças Cardiovasculares (DCV) são, atualmente, a maior causa de mortes no mundo. Dentre as DCVs, a hipertensão arterial constitui importante fator de risco para complicações cardíacas e cerebrovasculares, sendo considerada um problema de saúde pública em âmbito mundial.

De acordo com o gráfico da Figura 2, a proporção de indivíduos de 18 anos ou mais que referem diagnóstico de hipertensão arterial no Brasil foi de 21,4% em 2013, o que corresponde a 31,3 milhões de pessoas. Dessas, 22,9% encontram-se na região Sul, local onde o projeto visa ser executado.

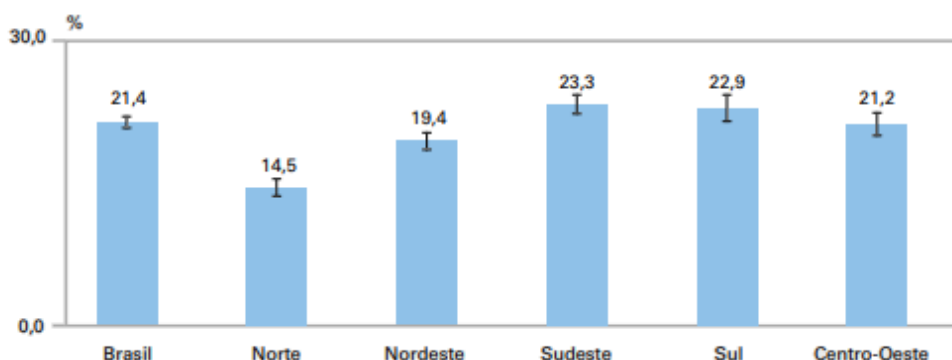


Figura 2 - Proporção de indivíduos de 18 anos ou mais que referem diagnóstico médico de hipertensão arterial, com indicação do intervalo de confiança de 95%, segundo as Grandes Regiões – 2013.

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional de Saúde Nacional (2013).

1.3 Hipótese

Por meio do desenvolvimento de um sistema *web* que utiliza uma pulseira

inteligente como dispositivo de aquisição de dados dos sinais vitais, pode-se registrar o resultado da atividade física na saúde de pessoas com deficiência intelectual e múltipla, promovendo auxílio aos profissionais da saúde na APAE - Criciúma.

1.3.1 Objetivo geral

Desenvolver um sistema *web* para registrar e monitorar os sinais vitais durante a atividade física de pessoas com deficiência intelectual e múltipla utilizando uma pulseira inteligente.

1.3.2 Objetivos Específicos

- a) Conceituar aplicação *web*;
- b) Conceituar pulseira inteligente;
- c) Identificar os principais sinais de uma pulseira inteligente que devem ser registrados durante uma atividade física;
- d) Desenvolver aplicação *web front-end*;
- e) Desenvolver aplicação *web back-end*;
- f) Persistir informações no banco de dados;
- g) Implementar comunicação entre pulseira inteligente e aplicação *web*.

1.4 Fundamentação teórica

Esta seção tem por objetivo descrever os aspectos teóricos que formam a base do processo de coleta de dados de uma pulseira inteligente até a construção do sistema *web*. Para tanto, será conceituado o que é aplicação *web*, pulseira inteligente, além do embasamento teórico sobre o sistema de aquisição de dados, instrumentação e plataforma de programação utilizada no desenvolvimento desse projeto.

1.4.1 Aplicação Web

Pode-se definir uma aplicação *web* como uma forma de uso de *software* que acessa dados persistentes através do serviço *web*, e permite a construção dinâmica de páginas para manipular estes dados. Em outras palavras, aplicação *web* é um sistema que permite aos usuários executarem lógica de negócio com um navegador *web* ou *browser* (WINCKLER & PIMENTA, 2002).

A aplicação *web* utiliza-se de uma página em HTML (Linguagem de Marcação de Hipertexto), interpretada pelo navegador *web* ou *browser*, para interagir com o usuário, formando a camada de apresentação. Outras tecnologias podem ser

misturadas ao HTML para a construção de uma interface com um visual adequado, além de proporcionar recursos que o HTML isoladamente não é capaz (COSTA, 2001).

A seguir, um breve resumo sobre essas tecnologias, as quais são responsáveis pela construção da interface com o usuário e tratamento dos dados:

1.4.1.1 HTML

O *HyperText Markup Language* ou Linguagem de Marcação de Hipertexto (HTML) é interpretado por qualquer *browser*, em qualquer plataforma (COSTA, 2001). Por meio dessa linguagem, o desenvolvedor pode especificar atributos para um texto (como fonte, tamanho e cor) ou criar marcações que definam um vínculo entre diversos documentos criados a partir de marcações específicas. O HTML não é uma linguagem de programação propriamente dita, mas sim uma linguagem de formatação de textos ou de definição da estrutura de um documento (PEREIRA ALVES, 2015), ou seja, HTML utiliza marcas ou *tags* para estruturar ou formatar os documentos para a *web* em formato de texto (WINCKLER & PIMENTA, 2002).

1.4.1.2 CSS

As Folhas de Estilos em Cascata ou *Cascading Style Sheets* (CSS) é uma linguagem de folhas de estilo que formata a informação HTML, possibilitando realizar a devida separação entre o modo de apresentação e o conteúdo do documento. Entende-se como informação qualquer elemento criado, como imagem, texto, áudio ou vídeo. O CSS prepara todos esses tipos de informação para que sejam exibidos da melhor forma para o usuário (SCHMITT, 2009).

1.4.1.3 JavaScript

JavaScript é a linguagem de programação da *web*. A maioria dos sites modernos usa JavaScript, e todos os navegadores modernos - em *desktops*, consoles de jogos, *tablets* e *smartphones* - incluem intérpretes de JavaScript. O HTML é utilizado para especificar o conteúdo de páginas da *web*, CSS para especificar a apresentação de páginas da *web* e JavaScript para especificar o comportamento de páginas da *web* (FLANAGAN, 2013).

No processo de programação de sistemas *web*, há linguagens que rodam no lado do cliente e que rodam no lado do servidor. JavaScript é utilizada principalmente para rodar *scripts* no lado do cliente, embora também seja crescente o número de aplicações desenvolvidas para rodar no lado do servidor (IEPSEN,

2018).

1.5 Framework

Segundo Minetto (2007) um *framework* de desenvolvimento é um conjunto de bibliotecas ou componentes que são usados para criar uma base onde uma aplicação será construída. É uma coleção de códigos-fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos *softwares*.

1.6 Arquitetura Cliente-Servidor

O modelo cliente-servidor, evidenciado na Figura 3, é definido como uma arquitetura onde o processamento da informação é dividido em processos ou módulos distintos. Desta forma, um processo é responsável pela obtenção ou acesso aos dados pelo usuário (cliente) e o outro pela manutenção da informação (servidor). O meio de transporte desses dados é a rede (SILVA & FERRARI, 2015).

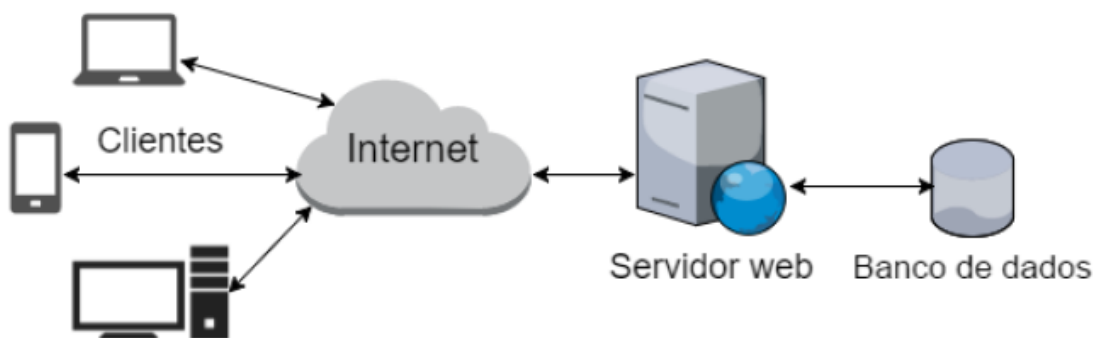


Figura 3 - Fluxo cliente-servidor.

Fonte: Silva & Ferrari (2015).

O processo cliente realiza o envio das mensagens fazendo algum pedido ao servidor. Essa é a parte que interage com o usuário e possui a interface que o usuário utiliza para requisitar as tarefas ao servidor, sendo chamado de *front-end* da aplicação. Os processos clientes gerenciam as atividades dos usuários e realizam as validações dos dados informados por estes (FILETO, 2006).

O processo servidor é chamado de *back-end* e é a parte que responde a uma mensagem solicitando a realização de alguma tarefa por parte do cliente. Os serviços podem ser realizados diretamente pelo processo servidor ou através de processos escravos criados por este para atender cada pedido do cliente, o que libera o processo mestre do servidor para receber outras solicitações (FILETO, 2006).

Segundo Silva & Ferrari (2015) é nos servidores onde se encontram as regras

de negócio da aplicação, sendo ele responsável pela manipulação das informações e pela persistência no banco de dados.

1.7 Formato de Dados

De acordo com Silva Júnior & Caetano Silva (2018) a necessidade da definição de um padrão para troca de mensagens, trouxe à *web* a Linguagem de Marcação Extensível (*Extensible Markup Language – XML*). A XML permitiu não só a troca de mensagens, mas a possibilidade de definir estruturas específicas para as mais diversas necessidades. Porém para o uso desta tecnologia, é necessário a criação de estruturas de texto que torne viável a comunicação rápida dos dados em redes com pouca capacidade de banda, sem exigir da rede uma maior quantidade de tráfego de dados.

Com o passar do tempo, o uso de comunicações assíncronas, processo em que não é necessário um tempo de resposta para que o outro evento seja acionado, tornou-se quase que pré-requisito no desenvolvimento de sistemas. Desta forma, o Javascript se fortaleceu com o uso do Javascript Assíncrono e XML (*Asynchronous Javascript and XML – AJAX*). Mas, a manipulação de arquivos XML tornou-se complexa no processo de desenvolvimento de software. Assim, com o surgimento da necessidade de uma linguagem comum que seja compacta, de fácil leitura e que exige uma menor quantidade de processamento, surgiu o JSON – *JavaScript Object Notation* ou Notação de Objetos JavaScript (SILVA JÚNIOR & CAETANO SILVA, 2018).

JSON é uma técnica de serialização de objetos ou formato de dados que se baseia na sintaxe de inicialização de objetos JavaScript, especificamente arranjos (*arrays*) e objetos literais. Definições de JSON pode ser incorporada dentro de arquivos JavaScript e acessado sem as análises que vem junto com linguagens baseadas em XML, pois utiliza a sintaxe do JavaScript. Esse modelo de transmissão de informações é baseado no formato texto e muito usado em serviços *web* que utilizam aplicações AJAX e arquiteturas de transferência de estado representacional (*Representational State Transfer - REST*) (HAQ, KHAN, & HUSSAIN, 2014). A principal vantagem de JSON sobre XML é o tamanho do arquivo, sendo muito mais compacto, e ainda consegue ser auto descritivo, utilizando uma notação de pares chave/valor para descrever os campos (SILVA & GONÇALVES, 2012).

1.8 Protocolo HTTP

O protocolo HTTP (*Hyper Text Transfer Protocol*), traduzido como Protocolo para Transferência de Hipertexto, é um protocolo baseado na arquitetura requisição-resposta. Para cada requisição que o browser envia para um servidor, há uma resposta. HTTP não é orientado a conexão e não mantém estado entre as chamadas, isso significa que toda informação necessária para que uma requisição seja processada deve ser passada na própria requisição. Essa abordagem promove uma maior escalabilidade, pois os recursos não ficam alocados no servidor por muito tempo (SILVA & GONÇALVES, 2012).

Para a requisição são utilizados métodos ou comandos específicos do HTTP, evidenciado na Tabela 1, e cada um desses realiza uma requisição específica.

Tabela 1 - Métodos HTTP.

MÉTODO	DESCRIÇÃO
GET	Solicita a leitura de uma página da <i>web</i>
HEAD	Solicita a leitura de um cabeçalho de página da <i>web</i>
PUT	Solicita o armazenamento de uma página da <i>web</i>
POST	Acrescenta a um recurso (por exemplo, uma página da <i>web</i>)
DELETE	Remove a página da <i>web</i>
TRACE	Ecoa a solicitação recebida
OPTIONS	Consulta opções

Fonte: Tanenbaum (2002).

1.9 Pulseira Inteligente

O termo pulseira inteligente faz parte do meio da tecnologia dos computadores *wearables* ou vestíveis, que possuem como definição o computador totalmente funcional, autoalimentado e independente, que é usado no corpo e permite o acesso a informações e a interação com informações, em qualquer lugar e a qualquer hora. Esse termo se diferencia de dispositivos móveis, como celular, por meio da possibilidade de captar informações, tanto do usuário como do ambiente, tornando seu funcionamento mais interativo (FANTONI, 2016).

Segundo Viseu (2003) sistemas *wearables* são dispositivos com constante comunicação em rede que conectam os indivíduos ao ambiente, composto por seres humanos ou computadores. Nessa relação, corpo e tecnologia interagem moldando um ao outro, em um processo de criação de uma nova entidade híbrida com suas

próprias dinâmicas e particularidades.

Embora a definição de *wearables* incorpore desde jaquetas inteligentes até colares com alguma espécie de conectividade, seus modelos mais conhecidas são as pulseiras inteligentes (*smartbands*) e os relógios inteligentes (*smartwatches*). Essas pulseiras são capazes de medir batimentos cardíacos, algumas são resistentes à poeira e água, e contam com sistema de posicionamento global (TECMUNDO, 2019).

1.10 Comunicação

A conexão sem fio da pulseiras inteligentes com outros dispositivos é geralmente realizada através da tecnologia Bluetooth (AMAZON, 2019).

Essa tecnologia *wireless* surgiu como um novo padrão para transmissão de dados digitais a curto alcance. Ela incorpora diversas vantagens como baixo consumo, baixo custo e operação industrial, científica e médica (ISM) disponível mundialmente (MARTINCOSKI, 2003).

Bluetooth Low Energy (BLE) - conhecido também como Bluetooth 4.0 ou *Bluetooth Smart* - é uma tecnologia sem fio desenvolvida para a comunicação de curto alcance. Em contraste com as tecnologias de Bluetooth anteriores, o BLE foi concebido como uma solução de baixa potência para aplicações de controle e monitorização (GOMEZ, OLLER, & PARADELLS, 2012).

A pilha de protocolo BLE é composto de duas partes principais: o controlador e o *host* (Hospedeiro), conforme Figura 4. O controlador compreende a camada física e a camada de enlace. O *host* é executado em um processador de aplicativos e inclui a funcionalidade da camada superior, ou seja, o protocolo L2CAP (*Logical Link Control and Adaptation Protocol*), o Protocolo de Atributos, do inglês ATT - *Attribute Protocol*, o perfil de Atributo Genérico, do inglês GATT - *Generic Attribute Profile*, o Protocolo de Gerenciamento de Segurança, do inglês SMP- *Security Manager Protocol* e o perfil de acesso genérico, do inglês GAP - *Generic Access Profile*. A comunicação entre o *host* e o controlador é padronizada com uma interface controladora de hospedeiro, do inglês HCI - *Host Controller Interface* (GOMEZ, OLLER, & PARADELLS, 2012). Por fim, a aplicação que é o bloco mais alto da pilha e representa a interface direta com o usuário, é quem define perfis que permitem a interoperabilidade entre diferentes aplicações, os quais são definidos pelo *Bluetooth Special interest group* (SIG), organização que supervisiona e gerencia os padrões Bluetooth.

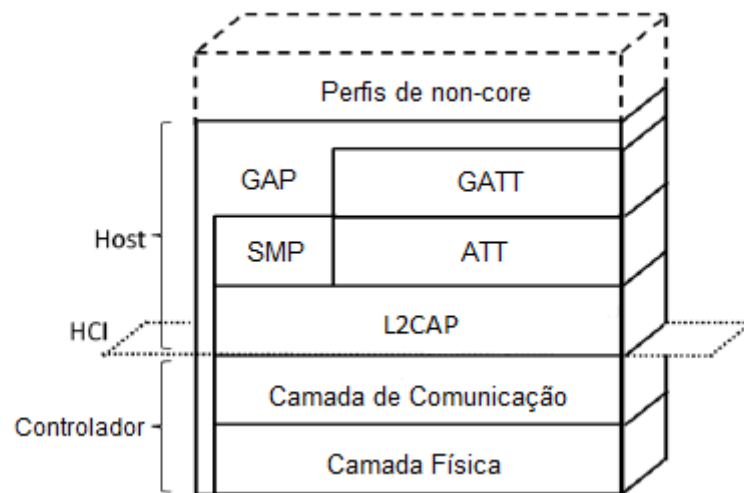


Figura 4 - Pilha de protocolo BLE

Fonte: Adaptado de Gomez, Oller, & Paradells (2012)

O protocolo L2CAP é quem entrelaça as camadas superiores as camadas de hardware, fragmenta e remonta pacotes e detecta erros. A camada de protocolos inclui protocolos de terceiros, padrões industriais e alguns desenvolvidos pelo próprio SIG (PATRICIA MCDERMOTT-WELLS, 2005).

Segundo Silveira (2018), *Generic Access Profile* (GAP) é responsável pela construção da interface diretamente com a camada de aplicação e, portanto, com o usuário, que pode definir todos os parâmetros que a rede necessita. É o protocolo responsável por implementar e controlar todas as camadas inferiores da pilha. Além disso, especifica papéis, modos e procedimentos do dispositivo, bem como o de gerenciar o estabelecimento da conexão e a segurança.

Security Manager Protocol (SMP) gerencia o emparelhamento de dispositivos, da distribuição das chaves de segurança e é projetado para minimizar os requisitos de recursos para os dispositivos escravos, deslocando a carga de trabalho para dispositivos mestres que possuem maior capacidade de processamento (SILVEIRA, 2018).

1.10.1 Perfil de atributo genérico – GATT

O Perfil de atributo genérico (GATT) estabelece em detalhes como trocar todos os dados de perfil e usuário por uma conexão BLE. Em contraste com o GAP, que define as interações de baixo nível com os dispositivos, o GATT lida apenas com procedimentos e formatos reais de transferência de dados (TOWNSEND ET AL., 2014).

Segundo Townsend et al., (2014) GATT também fornece a estrutura de

referência para todos os perfis baseados em GATT, que cobrem casos de uso precisos e garantem a interoperabilidade entre dispositivos de diferentes fornecedores. Todos os perfis BLE padrão são, portanto, baseados no GATT e devem obedecê-lo para funcionar corretamente.

O GATT usa o Protocolo de Atributo (ATT) como seu protocolo de transporte para trocar dados entre dispositivos. Esses dados são organizados hierarquicamente em seções chamadas serviços, que agrupam partes de dados do usuário conceitualmente relacionadas, denominadas características (TOWNSEND ET AL., 2014).

Como em qualquer outro protocolo ou perfil na especificação Bluetooth, o GATT começa definindo as funções que os dispositivos em interação podem adotar:

O cliente GATT envia solicitações para um servidor e recebe respostas (e atualizações iniciadas pelo servidor). Esse não possui informações sobre os atributos do servidor; portanto, ele deve primeiro consultar a presença e a natureza desses atributos executando a descoberta de serviço. Depois de concluir a descoberta do serviço, ele pode começar a ler e gravar atributos encontrados no servidor, além de receber atualizações iniciadas pelo servidor (TOWNSEND ET AL., 2014).

O servidor GATT recebe solicitações de um cliente e envia respostas de volta. Ele também envia atualizações iniciadas pelo servidor quando configurado para fazê-lo, e é a função responsável por armazenar e disponibilizar os dados do usuário para o cliente, organizados em atributos. Todo dispositivo BLE vendido deve incluir pelo menos um servidor GATT básico que possa responder às solicitações do cliente, mesmo que apenas para retornar uma resposta de erro (TOWNSEND ET AL., 2014).

1.10.1.1 UUID

Um identificador universalmente exclusivo (UUID) é um número de 128 bits (16 bytes) que tem uma alta probabilidade de ser globalmente exclusivo. Os UUIDs são usados em muitos protocolos e aplicativos além do Bluetooth (TOWNSEND ET AL., 2014).

1.10.1.2 Atributos

Os atributos são a menor entidade de dados definida pelo GATT (e ATT). São informações endereçáveis que podem conter dados relevantes do usuário

sobre a estrutura e o agrupamento dos diferentes atributos contidos no servidor. O GATT e o ATT podem trabalhar apenas com atributos; portanto, para clientes e servidores interagirem, todas as informações devem ser organizadas neste formulário (TOWNSEND ET AL., 2014).

1.10.1.3 Serviços

Os serviços do GATT agrupam atributos conceitualmente relacionados em uma seção comum das informações de atributo definidas no servidor GATT. Um serviço primário é o tipo padrão de serviço GATT que inclui funcionalidade padrão relevante exposta pelo servidor GATT.

Embora a declaração de serviço deva sempre ser o primeiro atributo do serviço, muitos outros podem segui-la antes da próxima declaração de serviço, geralmente na forma de características e descritores.

1.10.1.4 Características

Características são contêineres para dados do usuário. Eles sempre incluem pelo menos dois atributos: a declaração da característica (que fornece informações que crescem aos dados reais do usuário) e o valor da característica (que é um atributo completo que contém os dados do usuário em seu campo de valor). Todas as características do GATT sempre fazem parte de um serviço e, portanto, sempre podem ser encontradas em um.

2 METODOLOGIA

Com base no exposto, este trabalho caracteriza-se, conforme mencionado por Silva e Menezes (2005), como uma pesquisa aplicada, pois tem como objetivo gerar conhecimento para aplicação prática e dirige-se à solução de problemas específicos, ou seja, envolve verdades e interesses locais. Além disso sua abordagem é qualitativa, pois a mesma não precisa de métodos ou técnicas que envolvam números ou estatísticas, possuindo como foco principal a descrição do processo.

Para a realização do trabalho houve uma divisão, por etapas, de todos os passos necessários para a conclusão da pesquisa. Apesar de existirem diversos métodos e técnicas para a consecução da mesma, optou-se, na primeira fase, pela pesquisa bibliográfica, onde foram analisadas obras, artigos e documentos de autores que trabalharam com o tema, para dar suporte ao projeto. A pesquisa bibliográfica foi desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos.

A segunda etapa da pesquisa referiu-se à parte de comunicação do computador, que possui sistema operacional Linux Ubuntu, com um modelo de pulseira inteligente. Nesse processo, definiu-se o modelo de pulseira a ser utilizado, a camada do protocolo Bluetooth a ser estudada e suas características dependentes a serem identificadas.

A terceira etapa da pesquisa referiu-se à parte de análise de requisitos do protótipo. Nesta etapa foi onde especificou os elementos como funcionalidade da aplicação.

A última etapa foi voltada ao desenvolvimento das atividades e integrações exigidas pelo software, ou seja, a explicação e demonstração da parte do código, com imagens, das funcionalidades do mesmo, além de ser também etapa que mostra a maneira que o layout foi construído e as interações entre as telas, para um melhor entendimento do sistema.

A necessidade de coleta e armazenamento de dados da frequência cardíaca resultou no desenvolvimento de uma aplicação *web*. A Figura 5, evidencia os passos executados para o desenvolvimento da mesma, na qual utilizou-se as variáveis expostas no referencial teórico.

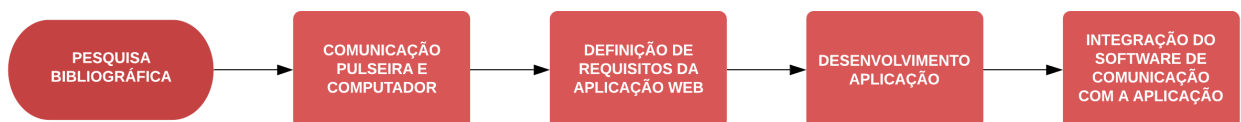


Figura 5 - Resumo da Execução do Projeto.

Fonte: Elaborado pela autora.

Para a coleta dos dados de frequência cardíaca, foi utilizado uma pulseira inteligente comercial, disponível na instituição onde desenvolveu-se o projeto, essa possui como tecnologia de comunicação, o BLE.

Nesse contexto, o BLE é baseado na especificação GATT, que estrutura a comunicação ou transferência de dados entre cliente e servidor. Portanto tornou-se necessário identificar os atributos presentes nesse protocolo para a obtenção dos dados de frequência cardíaca da pulseira inteligente. O conceito é evidenciado na Figura 6.

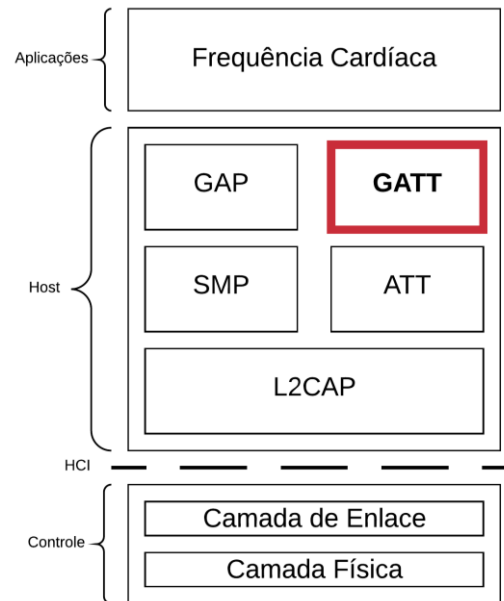


Figura 6 - Pilha de protocolos BLE com ênfase no protocolo GATT.

Fonte: Elaborado pela autora.

2.1 Ferramentas utilizadas

Esta seção tem por objetivo descrever as ferramentas utilizadas que formam a base do processo de coleta de dados de uma pulseira inteligente até a construção do sistema *web*.

2.1.1 Linux

O projeto utilizou-se do sistema operacional de código aberto Linux com a distribuição Ubuntu 18.04. Ubuntu é uma das mais populares distribuições do Linux, seu principal objetivo é que qualquer pessoa (usuário) possa utilizá-lo sem dificuldades, independentemente de nacionalidade, nível de conhecimento ou limitações físicas (“Get Ubuntu”, [s.d.]).

2.1.2 HCITool

Host Controller Interface Tool é uma ferramenta de linha de comando do sistema operacional Linux usada para detectar e se conectar com dispositivos periféricos, essa também é capaz de exibir informações adicionais permitidas pelo dispositivo.

2.1.3 GATTTool

Generic Attribute Tool é uma ferramenta de linha de comando do sistema operacional Linux capaz de ler e escrever características para um dispositivo acessível que suporta o protocolo GATT.

2.1.4 nRF Connect

O nRF Connect é uma ferramenta multiplataforma que permite testar e desenvolver BLE. Ele permite a configuração fácil de conexões com outros dispositivos e usa essas conexões para ler e gravar os nós externos (“nRF Connect”, [s.d.]).

2.1.5 Wireshark

Wireshark é um programa que analisa o tráfego de rede, e o organiza por protocolos. Através dessa aplicação é possível controlar o tráfego de uma rede e monitorar a entrada e saída de dados do computador, em diferentes protocolos, ou da rede à qual o computador está ligado. Também é possível analisar protocolos de um determinado dispositivo através de arquivos no formato de logs (“Wireshark”, [s.d.]).

2.1.6 Visual Studio Code

O Visual Studio Code é um editor de código-fonte que suporta um número de linguagens de programação e um conjunto de recursos que podem ou não estarem disponíveis para a dada linguagem (“Visual Studio Code - Code Editing. Redefined”, [s.d.]).

2.1.7 NodeJS

Node.js é uma plataforma de aplicação para JavaScript, sua execução é assíncrona e orientada a eventos de código, sua função é migrar a programação do JavaScript do cliente (*front-end*) para os servidores, manipulando milhares de conexões/eventos simultâneas em tempo real numa única máquina física (“About | Node.js”, [s.d.]).

2.1.8 NPM

Node Package Manager (Gerenciador de Pacotes do Node) é um repositório online para publicação de projetos de código aberto para o Node.js; além disso, ele é um utilitário de linha de comando que interage com este repositório online, que ajuda na instalação de pacotes, gerenciamento de versão e gerenciamento de dependências (“npm | build amazing things”, [s.d.]).

2.1.9 PostgreSQL

O PostgreSQL é um sistema de banco de dados objeto-relacional de código aberto com mais de 30 anos de desenvolvimento ativo que lhe garantiu uma forte reputação de confiabilidade, robustez de recursos e desempenho, além disso é um

dos sistemas que possuem as maiores comunidades de usuários de bancos de dados de código livre (“PostgreSQL”, [s.d.]).

2.1.10 Knex

O Knex.js é um construtor de consultas SQL para bancos de dados relacionais. Ele apresenta retornos de chamada, interface para controle de fluxo assíncrono, interface de fluxo, construtores de consulta e esquema com recursos completos e respostas padronizadas entre diferentes clientes de consulta e dialetos. Sua utilização principal é programar bancos de dados na linguagem JavaScript (“Knex.js”, [s.d.]).

2.1.11 VueJS

VueJS ou simplesmente *Vue* é um *framework* JavaScript para o desenvolvimento de componentes reativos para interfaces *web* modernas. Componentes são um conjunto de código que contém marcação, estilo e comportamento (*HTML*, *CSS* e *JavaScript*) e que juntos podem compor interfaces reaproveitáveis (“Introduction — Vue.js”, [s.d.]).

2.2 Extração dos dados de frequência cardíaca da pulseira inteligente

Definido o tipo de comunicação no qual o presente projeto está inserido, executou-se os seguintes passos para o desenvolvimento da comunicação:

- Pesquisou-se modelos de pulseiras inteligentes comerciais disponíveis e suas tecnologias de comunicação;
- A partir dos modelos encontrados e adquiridos, identificou-se os atributos genéricos de cada dispositivo;
- Desenvolveu-se aplicação de comunicação utilizando os atributos identificados.

2.2.1 Identificação de atributos BLE

No contexto deste trabalho, os módulos BLE (servidores) são utilizados para transferir os dados coletados a partir do sensor de frequência cardíaca para o *gateway* BLE (cliente) que, por sua vez, encaminha os dados para serem registrados no banco de dados. BLE é baseada na especificação GATT, que define como comunicar ou transferir dados entre o cliente e o servidor. As informações que estão sendo enviadas e recebidas são chamadas de atributos. Cada dispositivo BLE possui alguns serviços e cada serviços têm características dos descritores de características (se a característica tem mais de um parâmetro ou é do tipo ler ou

notificação). Algumas das características têm apenas acesso de leitura/gravação (por exemplo, hora atual, o status da bateria ou notificações).

O processo iniciou-se pela leitura de dispositivos Bluetooth disponíveis na proximidade utilizando Hcitol que é uma ferramenta capaz de detectar e conectar a dispositivos periféricos. Através do terminal em um sistema operacional Linux inseriu-se o comando “sudo hcitool lescan”. Através desse comando, é possível identificar o endereço Bluetooth da pulseira inteligente e conectar ao computador, conforme Figura 7.

```
Fernanda@FernandaPC:~$ sudo hcitool lescan
LE Scan ...
EB:8C:CC:96:40:EA (unknown)
CC:14:74:7C:6D:16 (unknown)
FC:8F:90:36:A3:3F (unknown)
F0:39:28:C2:68:FE L11_68FE
F0:39:28:C2:68:FE (unknown)
CC:14:74:7C:6D:16 Mi Band 3
EB:8C:CC:96:40:EA Mi Smart Band 4
```

Figura 7 - Comando para identificar endereço Bluetooth.

Fonte: Elaborado pela autora.

A segunda etapa refere-se à identificação dos serviços e características da pulseira inteligente, portanto utilizou-se a ferramenta gatttool inserindo os comandos “primary”, conforme Figura 8. Isso gerou uma lista de todos os serviços disponíveis em execução no dispositivo e “characteristics” para gerar todas as características disponíveis. Estes são valores mascarados de bits que descrevem modos de leitura/gravação e requisitos de permissão. Este comando evidencia o Bluetooth com UUIDs, que descrevem os tipos de dados.

```
[CC:14:74:7C:6D:16][LE]> primary
attr handle: 0x0001, end grp handle: 0x0007 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x0008, end grp handle: 0x000b uuid: 00001801-0000-1000-8000-00805f9b34fb
attr handle: 0x000c, end grp handle: 0x0016 uuid: 0000180a-0000-1000-8000-00805f9b34fb
attr handle: 0x0017, end grp handle: 0x001c uuid: 00001530-0000-3512-2118-0009af100700
attr handle: 0x001d, end grp handle: 0x0023 uuid: 00001811-0000-1000-8000-00805f9b34fb
attr handle: 0x0024, end grp handle: 0x0026 uuid: 00001802-0000-1000-8000-00805f9b34fb
attr handle: 0x0027, end grp handle: 0x002c uuid: 0000180d-0000-1000-8000-00805f9b34fb
attr handle: 0x002d, end grp handle: 0x0057 uuid: 0000fee0-0000-1000-8000-00805f9b34fb
attr handle: 0x0058, end grp handle: 0x006c uuid: 0000fee1-0000-1000-8000-00805f9b34fb
```

Figura 8 - Comando para identificar serviços.

Fonte: Elaborado pela autora.

A terceira etapa refere-se à associação dos serviços e características aos comandos relacionados com sensor de frequência cardíaca. Dessa maneira conectou-se a pulseira inteligente ao computador e utilizou-se a ferramenta nFR Connect para reconhecer os serviços disponíveis e identificar a denominação dos serviços e características

A quarta etapa refere-se à identificação das permissões de acesso dos

atributos identificados, conectou-se a pulseira inteligente ao computador e registrou os logs de comunicação entre os dispositivos. Em sequência, utilizou-se o programa Wireshark para analisar os protocolos dos logs registrados, ou seja, comparou-se os UUIDs enviados durante a conexão e identificou-se as características e serviços de emparelhamento.

2.2.2 Desenvolvimento aplicação de comunicação

A primeira etapa para o desenvolvimento da aplicação de comunicação é a importação de um pacote node – “web-bluetooth” - responsável por realizar a interação entre navegadores da *web* e dispositivos Bluetooth próximos. Esse pacote provê funcionalidades como a busca de dispositivos, busca de serviços, leitura e escrita nas características.

A segunda etapa refere-se a encontrar os dispositivos Bluetooth próximos, ou seja, a pulseira inteligente. Para isso, utiliza-se o método “Bluetooth.requestDevice” que mostra uma lista interativa com todos os dispositivos descobertos que encontrou, conforme Figura 9. Esse método usa um objeto obrigatório que define filtros. Esses podem ser de uma variedade de serviços UUIDs e são usados para retornar apenas dispositivos que correspondam a serviços anunciados do Bluetooth GATT ou o nome do dispositivo. Na figura, o serviço “0xfee0” corresponde ao serviço de hardware da pulseira inteligente.

```
const device = await bluetooth.requestDevice({
  filters: [{ services: [0xfee0] }],
  optionalServices: optionalServices
});
```

Figura 9 - Método para encontrar a pulseira inteligente.

Fonte: Elaborado pela autora.

2.3 Definição do funcionamento da aplicação

A Tabela 2 apresenta as funcionalidades principais do sistema.

Tabela 2 - Funcionalidade da aplicação.

Requisito Funcional	Descrição
Cadastrar/Consultar Aluno	Função do sistema para inclusão e visualização de novos alunos.
Cadastrar/Consultar atividade do aluno	Função do sistema para inclusão e visualização de novas atividades.
Registrar frequência cardíaca	Função do sistema para inclusão da frequência cardíaca no registro de atividades.

 Acessar Sistema

Possibilita o acesso ao sistema de páginas.

 Fonte: Elaborado pela autora.

2.4 Desenvolvimento aplicação *Back-end*

Para o desenvolvimento da aplicação *back-end*, utilizou-se o sistema de banco de dados relacional PostgreSQL e as tabelas foram construídas com a linguagem JavaScript utilizando o construtor de consultas SQL knex.js.

Com o intuito de realizar o requisito funcional da aplicação, no banco de dados, foram construídas duas tabelas: A tabela de alunos e a tabela de histórico de atividades. A última recebe da primeira o identificador único do aluno, para que cada atividade esteja relacionada a um único aluno, possibilitando a consulta de históricos por aluno, conforme Figura 10.

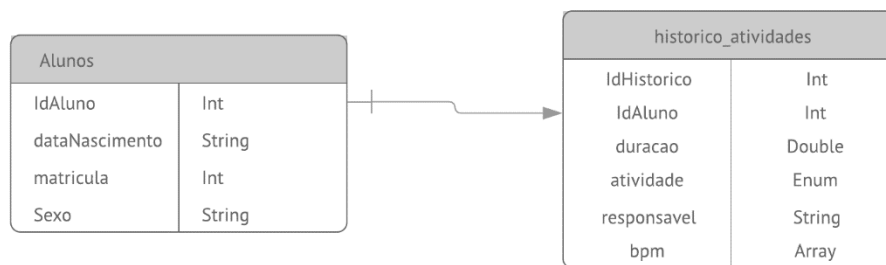


Figura 10 - Diagrama de relacionamento entre as tabelas.

Fonte: Elaborado pela autora.

Além disso, para o acesso ao sistema foi construído a tabela de usuário, conforme a Figura 11, essa não possui relacionamento com as outras tabelas, mas possui um parâmetro de administrador, responsável pela permissão de acesso a determinadas telas no *front-end*.

Usuários	
IdUsuario	Int
nome	String
email	String
senha	String
administrador	Boolean

Figura 11 - Tabela Usuários.

Fonte: Elaborado pela autora.

Em todas as tabelas, os métodos HTTP utilizados foram GET para buscar dados da tabela, POST para inserir dados nas tabelas, PUT para atualizar dados já cadastrados e DELETE para remover dados das tabelas.

2.5 Desenvolvimento aplicação *Front-end*

A aplicação no lado do cliente, foi desenvolvida utilizando o framework VueJS para a estruturação e a construção visual. Para a comunicação com *back-end* foi utilizado a biblioteca axios, ou seja, um cliente HTTP que interage com a interface HTTP do node quando realizado as requisições do browser para o servidor. As requisições e respostas da aplicação são realizadas no formato de dados JSON.

3 RESULTADOS

Este capítulo demonstra os resultados obtidos da extração de dados da pulseira inteligente e do desenvolvimento do protótipo da aplicação *web*.

3.1 Comunicação com pulseira inteligente

Os serviços e características essenciais para a comunicação e leitura de dados da pulseira inteligente podem ser evidenciados nas Tabelas 3 e 4. Esses são responsáveis pelo emparelhamento e leitura da frequência cardíaca.

Tabela 3 – Serviços identificados para o emparelhamento.

Nome	UUID
Serviço principal	0000fee1-0000-1000-8000-00805f9b34fb
Característica de autenticação	00000009-0000-3512-2118-0009af100700

Fonte: Elaborado pela autora.

Tabela 4 – Serviços identificados para o sensor de frequência cardíaca.

Nome	UUID
Serviço de Hardware	0000fee0-0000-1000-8000-00805f9b34fb
Serviço de monitoramento cardíaco	0000180d-0000-1000-8000-00805f9b34fb
Característica de medição da frequência cardíaca	00002a37-0000-1000-8000-00805f9b34fb
Característica de controle do monitoramento cardíaco	00002a39-0000-1000-8000-00805f9b34fb
Característica do sensor	00000001-0000-3512-2118-0009af100700

Fonte: Elaborado pela autora.

3.2 Aplicação web











A versão atual do protótipo da aplicação *web* totaliza seis telas principais de interação com o usuário, conforme Figuras 12, 13, 14, 15, 16, 17, 18 e 19. Estas interfaces permitem que os seguintes fluxos de atividade sejam realizados pelos usuários:

- Cadastro, edição e remoção de Alunos;
- Cadastro, edição e remoção de usuário;
- Cadastro de Atividade Física;
- Visualização de dados dos alunos;
- Visualização de dados das atividades físicas;
- Visualização da frequência cardíaca em cada atividade física.



Figura 12 - Tela de *login* e cadastro de usuário.

Fonte: Elaborado pela autora.

Código	Nome	E-mail	Administrador	Ações
2	Alan	alan@gmail.com.br	Não	 
3	Admin	admin@gmail.com.br	Não	 
4	UserAdmin	user1@gmail.com.br	Sim	 
5	Freitas	freitas@gmail.com	Sim	 
1	Fernanda Di Queiroz	fernanda1queiroz@gmail.com.br	Sim	 

Desenvolvido com ♥ por Fernanda

Figura 13 - Tela de visualização visão de administrador.

Fonte: Elaborado pela autora.

Cadastrar Aluno Gerenciamento de Alunos UserAdmin

Administração do Sistema

Cadastro e edições

Usuários

Nome: E-mail:

Administrador?

Senha: Confirmação de Senha:

Figura 14 - Tela de cadastro visão de administrador.

Fonte: Elaborado pela autora.

Cadastrar Aluno Gerenciamento de Alunos UserAdmin

Cadastro de Alunos

Cadastro

Nome: Matricula:

Sexo
 Feminino Masculino

Data de Nascimento

Desenvolvido com ❤️ por Fernanda

Figura 15 - Tela de Cadastro de alunos.

Fonte: Elaborado pela autora.

Cadastrar Aluno Gerenciamento de Alunos UserAdmin

Alunos

Administração Sair

Histórico de Atividades	Matricula	Nome	Data de Nascimento	Sexo	Ações
	124549045	Joao luiz	18.07.2003	Masculino	
	124466567	Maria	02.09.1999	Feminino	
	7654321	Ana	09.11.2000	Feminino	
	98765432	Pedro	02.11.1998	Masculino	

Desenvolvido com ❤️ por Fernanda

Figura 16 - Tela de visualização de alunos.

Fonte: Elaborado pela autora.



Figura 17 - Tela de visualização de atividades.

Fonte: Elaborado pela autora.

Cadastro

Profissional Responsável:

Duração da Atividade:

Tipo de Atividade:

Data da atividade:

Figura 18 - Tela de cadastro de atividades.

Fonte: Elaborado pela autora.

Conectar Pulseira

bpm

Salvar Cancelar

Desenvolvido com ❤️ por Fernanda

Figura 19 – Tela de cadastro de atividades – conexão com a pulseira.

Fonte: Elaborado pela autora.

4 DISCUSSÕES

Conforme o tema do trabalho, o desenvolvimento da aplicação *web* está

relacionado a uma necessidade da comunidade local, a qual deve registrar os dados de frequência cardíaca durante uma atividade para análises posteriores ou em tempo real daquele que está praticando tal atividade.

As ferramentas utilizadas na construção da aplicação *web* permitiram que as operações necessárias fossem implementadas com êxito. Não houve limitações no desenvolvimento decorrentes do uso das ferramentas (frameworks, linguagens de programação, aplicativos, etc). Os dados de frequência capturados foram precisos em relação a pulseira utilizada.

Dessa maneira, os serviços identificados em formatos UUID proporcionaram a leitura e posterior armazenamento dos dados de frequência cardíaca, concluindo, portanto, o objetivo específico de identificar os principais sinais de uma pulseira inteligente que devem ser registrados durante a atividade física.

Como forma de validar os dados identificados, construiu-se a aplicação *web* possuindo três tabelas no banco de dados e seis telas no lado cliente da aplicação.

Das tabelas construídas, a tabela de usuários possui como chave única os parâmetros “*IdUsuario*” e “*email*”, o parâmetro “*administrador*” da mesma possui como foco dar permissão à tela de admiração do *front-end*, nessa é possível manipular e visualizar os dados de usuários cadastrados na aplicação. A tabela de alunos possui como chave única os parâmetro “*IdAluno*” e a “*matricula*”. A tabela de histórico de atividade possui como chave estrangeira o “*IdAluno*” da tabela de alunos e possui como chave única o parâmetro “*IdHistorico*”. Todos os dados das tabelas são obrigatórios, ou seja, não podem ser nulos.

Das telas desenvolvida, a tela de cadastro de *login/cadastro* destina-se a incluir os usuários na aplicação e atribuir permissão para entrar no sistema. Nesta tela, também, são feitas validações das informações, como e-mail e senha, que são obrigatórios.

A tela de visualização e cadastro de usuários destina-se a manipular os dados de usuários com ações para adicionar, editar e remover. Esta é apenas acessada se o usuário possuir o parâmetro “*administrador*” como verdadeiro.

A tela de cadastro de alunos destina-se a adicionar ou editar alunos, sendo todos os campos obrigatórios. O redirecionamento para esta tela é realizado por meio do botão “*cadastrar aluno*” no cabeçalho da aplicação ou por meio do botão “*editar aluno*” na tela de visualização de alunos.

A tela de visualização de alunos destina-se a gerenciar os alunos cadastrados.

Esta é a tela principal da aplicação, o seu redirecionamento ocorre logo após a aprovação do *login*. Além da funcionalidade de visualização, essa possui botões que redirecionam para outras telas, como o botão de cadastrar atividade, botão de visualização de histórico, botão de editar e botão de remover.

A tela de cadastro de atividade destina-se a adicionar atividades de cada aluno, sendo todos os campos obrigatórios. A mesma possui o botão “conectar pulseira” que gera a busca do dispositivo e solicita a conexão. Após conectado quando o usuário selecionar o modo de atividade física, os dados de frequência cardíaca serão evidenciados no terminal. No término da atividade, quando o usuário pressionar o botão “salvar”, todos os dados serão adicionados no banco de dados e estarão disponíveis para visualização na tela de visualização de histórico.

A tela de visualização de histórico de atividades, destina-se a evidenciar as atividades de um único aluno. Nessa, os dados de frequência cardíaca são demonstrados no formato de gráficos de linhas.

Além do desenvolvimento das telas, avaliou-se que a distância de comunicação entre o computador e a pulseira inteligente foi de até quatro metros sem barreiras, não se realizou uma análise em relação ao resultado desse limite de comunicação.

Através das informações descritas, percebe-se que os objetivos específicos de desenvolver a aplicação *back-end* e *front-end*, desenvolver o banco de dados e integrar a aplicação com a pulseira inteligente foram concluídos.

Como este projeto tratou-se de um protótipo de aplicação, o conjunto completo de funcionalidades pode ser estendido para futuras melhorias, como: identificar dados e diferentes parâmetros de outros modelos de pulseira inteligente, avaliar as distâncias físicas de comunicação dessas; desenvolver campo de busca nas telas de visualização; ajustar dados para corresponder com os requisitos dos profissionais da saúde voltados para pessoas com deficiências.

5 CONCLUSÃO

No projeto desenvolvido foi construído um protótipo de aplicação *web* que pode extrair os dados de frequência cardíaca de uma pulseira inteligente (*smartband*) e armazená-los no banco de dados, fazendo com que profissionais da saúde possam utilizar a aplicação em seus alunos ou pacientes durante uma determinada atividade física. Além disso, o protótipo permite a visualização das

informações de forma prática e rápida, facilitando a consulta e uso dessas informações para tomada de decisões futuras.

Ademais, como já foi explicitado anteriormente, com a melhoria proposta para o protótipo, pode-se expandir os modelos de pulseiras utilizadas e os tipos de dados adquiridos.

O tópico relacionado à pesquisa bibliográfica permitiu que houvesse um incremento do entendimento da parte conceitual da aplicação. Além disso, evidenciou a importância do protocolo GATT para identificação dos dados de frequência cardíaca.

A construção desse protótipo possibilitou aplicação prática de diversos conhecimentos, como análise de sistemas, programação *web*, banco de dados, BLE, entre outros.

Diferentes tecnologias de programação foram utilizadas ao longo do projeto, optando-se por opções modernas e de código aberto. Todas essas características permitiram a construção com sucesso de um aplicativo de relativa complexidade em curto período de tempo.

Por fim, este trabalho mostrou-se de extrema importância para a formação profissional e pessoal do autor. Diversos momentos de aprendizado foram proporcionados ao longo do processo de desenvolvimento.

DEVELOPMENT OF A WEB APPLICATION TO REGISTER AND MONITOR VITAL SIGNS DURING PHYSICAL ACTIVITY FOR PEOPLE WITH INTELLECTUAL AND MULTIPLE DISABILITIES USING A SMARTBAND

Abstract: The present study presents the development of a prototype of a web application to store heart rate data of a smartband, and its development, for the most part, focused on the JavaScript language and its data bases on Knex and PostgreSQL. The project starts by introducing the concept of a web application, whose conceptualization is explained for a better understanding of the theme and related matters, assisting in the development of the prototype. The technology and tools implemented in the project has its main objective to support the development of the prototype. The development of the application resulted in a prototype, which allows users, students and physical activities to be registered and displayed, in addition, it also resulted in the identification of UUIDs that correspond to services and

characteristics of heart rate, making it possible to read this parameter and the communication between the application and a smartband through the BLE.

Keywords: Web Application. Bluetooth Low Energy. Data Base. VueJS.

REFERÊNCIAS

- About | Node.js.** Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 10 nov. 2019.
- CASINO, F. et al. Context-aware recommender for smart health. **2015 IEEE 1st International Smart Cities Conference, ISC2 2015**, n. September 2017, p. 10–12, 2015.
- COSTA, C. G. A. DA. Desenvolvimento e Avaliação Tecnológica de um Sistema de Prontuário Eletrônico do Paciente, Baseado nos Paradigmas da World Wide Web e da Engenharia de Software. 2001.
- EDNA LÚCIA DA SILVA, D.; ESTERA MUSZKAT MENEZES, M. Metodologia da Pesquisa e Elaboração de Dissertação. **Portal**, v. 29, n. 1, p. 121, 2005.
- FANTONI, A. Dispositivos Wearable para o Campo da Saúde: reflexões acerca do monitoramento de dados do corpo humano. **Temática**, v. 12, n. 01, p. 185–198, 2016.
- FILETO, R. Sistemas Cliente-Servidor Cliente - Servidor. 2006.
- FLANAGAN, D. **Javascript: O guia Definitivo**. [s.l: s.n.].
- Get Ubuntu | Download | Ubuntu.** Disponível em: <<https://ubuntu.com/download>>. Acesso em: 23 out. 2019.
- GOMEZ, C.; OLLER, J.; PARADELLS, J. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. **Sensors (Switzerland)**, v. 12, n. 9, p. 11734–11753, 2012.
- HADDAD, S. et al. Efeito do treinamento físico de membros superiores aeróbio de curta duração no deficiente físico com hipertensão leve. **Arquivos Brasileiros de Cardiologia**, v. 69, n. 3, p. 169–173, 2005.
- HAQ, Z. U.; KHAN, G. F.; HUSSAIN, T. A Comprehensive analysis of XML and JSON web technologies. **New Developments in Circuits, Systems, Signal Processing, Communications and Computers**, p. 102–109, 2014.
- IBGE. Análise dos resultados Percepção do estado de saúde. 2013.
- IEPSEN, E. F. **Lógica de Programação e Algoritmos com JavaScript: Uma introdução à programação de computadores com exemplos e exercícios para iniciantes**. 1. ed. São Paulo: [s.n.].

Introduction — Vue.js. Disponível em: <<https://vuejs.org/v2/guide/>>. Acesso em: 11 dez. 2019.

Knex.js - Um construtor de consultas SQL para Javascript. Disponível em: <<http://knexjs.org/>>. Acesso em: 23 out. 2019.

MARTINCOSKI, D. H. Sistema para telemetria de eletrocardiograma utilizando tecnologia bluetooth. 2003.

MINETTO, E. L. Frameworks para Desenvolvimento em PHP. p. 192, 2007.

npm | build amazing things. Disponível em: <<https://www.npmjs.com/>>. Acesso em: 23 out. 2019.

nRF Connect for Desktop - Downloads - nordicsemi.com. Disponível em: <<https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF-Connect-for-desktop/Download>>. Acesso em: 23 out. 2019.

O que são wearables e por que você vai querer usar um em breve. Disponível em: <<https://www.tecmundo.com.br/wearables/117937-samsung-wearables-dispositivos-vestiveis-realidade-virtual-camera-360.htm>>. Acesso em: 26 maio. 2019.

PANTELOPOULOS, A.; BOURBAKIS, N. G. A survey on wearable sensor-based systems for health monitoring and prognosis. **IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, v. 40, n. 1, p. 1–12, 2010.

PARRO, M. C.; MARTINEZ ÉVORA, Y. D. Desenvolvimento de software para a organização da informação de um serviço de saúde ocupacional hospitalar. **Revista Eletrônica de Enfermagem**, v. 13, n. 3, p. 386–394, 2013.

PATRICIA MCDERMOTT-WELLS. What is Bluetooth? **IEEE**, n. December 2004, p. 1–71, 2005.

PEREIRA ALVES, W. **Java para Web – Desenvolvimento de Aplicações - William Pereira Alves - Google Livros.** São Paulo: Editora Érica, 2015.

PostgreSQL: O banco de dados de código aberto mais avançado do mundo. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 26 maio. 2019.

RADOVANOVIC, C. A. T. et al. Arterial Hypertension and other risk factors associated with cardiovascular diseases among adults. **Revista Latino-Americana de Enfermagem**, v. 22, n. 4, p. 547–553, 2014.

Relógio Sport Xiaomi Mi Band 2: Amazon.com.br: Celulares e Telefonia.

Disponível em: <https://www.amazon.com.br/Xiaomi-Mi-Band-Original-lacrada/dp/B01MSAZGDW?tag=goog0ef-20&smid=A3DVR3YMB215Q4&ascsubtag=go_1676964938_67796570520_3252207>

16474_pla-637226056969_c_>. Acesso em: 26 maio. 2019.

SANTARÉM, J. M. Texto de apoio ao curso de Especialização Atividade Física Adaptada e Saúde Promoção da saúde do idosos: a importância da atividade física. p. 1–6, 2001.

SCHMITT, C. CSS Cookbook, 3rd Edition. **Chemistry & ...**, p. 732, 2009.

SILVA, M. S. DA; FERRARI, F. C. Integração de frameworks front-end para desenvolvimento de interfaces ricas com JavaServer Faces. p. 68–77, 2015.

SILVA, R. F. DA; GONÇALVES, P. R. Web Services – Uma Análise Comparativa Web Services – Uma Análise Comparativa. n. Dezembro, 2012.

SILVA JUNIOR, J. B.; CAETANO SILVA, P. Análise da representação semântica de modelos de dados do formato JSON. **Revista de Sistemas e Computação - RSC**, v. 8, n. December, p. 196–209, 2018.

SILVEIRA, V. M. DA. Sistema de manutenção preditiva utilizando redes ble e wi-fi com aprendizado de máquina. 2018.

TANENBAUM, A. S. Computer networks (4. ed.). p. I–XX, 1–891, 2002.

THEREZA MARIA MAGALHÃES MOREIRA; GOMES, E. B.; JÊNIFA CAVALCANTE DOS SANTOS. Fatores de risco cardiovasculares em adultos jovens com hipertensão arterial e/ou diabetes mellitus. **Revista Gaucha de Enfermagem**, v. 4, n. 4, p. 662–9, 2010.

TOWNSEND, K. et al. **Introdução ao Bluetooth Low Energy**. [s.l.] O'Reilly Media, Inc., 2014.

WISEU, A. N. A. Shaping Technology / Building Body (Nets). p. 128–133, 2003.

Visual Studio Code - Code Editing. Redefined. Disponível em:

<<https://code.visualstudio.com/>>. Acesso em: 23 out. 2019.

WINCKLER, M.; PIMENTA, M. Avaliação de Usabilidade de sites Web. **X Escola de Informática da SBC-Sul (ERI2002)**, p. 85–137, 2002.

Wireshark - Go Deep. Disponível em: <<https://www.wireshark.org/>>. Acesso em: 23 out. 2019.