

**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**PRÓ-REITORIA DE PESQUISA**  
**COORDENAÇÃO GERAL DE PROGRAMAS ACADÊMICOS E DE**  
**INICIAÇÃO CIENTÍFICA**

**PROGRAMA DE INICIAÇÃO CIENTÍFICA PIBIC/PIBITI/PIBIC-AF/PIVIC/PIVITI**

**Título do Projeto:**

Análise Computacional de Exercícios Físicos Adaptados para Pacientes com Lesões  
Cerebrais e Medulares

**Título do Plano:**

Uma Proposta de Plataforma Computacional de Baixo Custo para Aferição do  
Condicionamento Aeróbico de Pacientes com Lesão Medular

CLAUIRTON DE ALBUQUERQUE SIEBRA

Departamento de Sistemas de Computação - Centro de Informática

LAURA DE FARIA MARANHÃO AYRES

Engenharia de Computação/Centro de Informática

**João Pessoa, 29 de julho de 2020**

## **RESUMO**

A Lesão Medular (LM) é uma condição de insuficiência parcial ou total do funcionamento da medula espinhal, na qual a parcela da população nessa circunstância é caracterizada como sedentária, promovendo um aumento de incidências relacionadas a complicações secundárias. Portanto, os objetivos deste trabalho incluem a construção de um software de controle, além de uma plataforma computacional de baixo custo para aferição do condicionamento aeróbio de pacientes com lesão medular. Dessa forma, com o auxílio do sensor tcr5000, um sensor de frequência cardíaca e do módulo NodeMCU WiFi ESP8266, são obtidos os dados do exercício e compartilhados - através do banco de dados em tempo real chamado Firebase - com o aplicativo inicialmente criado na plataforma App Inventor, sendo posteriormente reformulado com uso do React Native. Com isso, os valores da velocidade média, aceleração média e a distância percorrida são armazenados no banco de dados principal MongoDB, que também possui as características do indivíduo com a lesão. Finalmente, com base nessas estatísticas e através da técnica de clusterização, o aprendizado de máquina é usado para auxiliar nos próximos casos, comparando as similaridades características entre os indivíduos e seu histórico de progresso, determinando a melhor intensidade de treinamento a ser realizado.

**PALAVRAS CHAVE:** Lesão medular. IOT. Aprendizado de máquina.

## **SUMÁRIO**

<b>1. Introdução</b>	<b>3</b>
<b>2. Procedimentos metodológicos</b>	<b>4</b>
<b>3. Resultados e Discussão</b>	<b>8</b>
<b>4. Conclusão</b>	<b>11</b>

## 1. INTRODUÇÃO

A Lesão Medular (LM) é uma condição de insuficiência parcial ou total do funcionamento da medula espinhal, causando alterações a nível motor, sensitivo e autonômico na vida do indivíduo (CEREZETTI et al., 2012). Nesse contexto, constata-se que essa parcela da população é caracterizada como sedentária, promovendo um aumento de incidências relacionadas a complicações secundárias - incluindo diabetes mellitus, hipertensão e perfis lipídicos aterogênicos (NOREAU; SHEPHARD, 1995). Desse modo, uma vez que o Brasil possui uma incidência maior que 10 mil casos novos de lesões medulares por ano (BRASIL, 2015, p. 7), torna-se significativa a presença de análises focadas nos resultados obtidos nas práticas de exercício desses indivíduos.

Com efeito, a elaboração de projetos e pesquisas direcionadas para indivíduos com lesão cerebral e medular são importantes para auxiliar no cotidiano deles. Estudos têm utilizado métodos de treinamento aeróbico para os usuários, como Hoffman (1986) que em uma revisão de 13 estudos de treinamento cardiorrespiratório, revelou melhorias na capacidade de trabalho físico e no consumo máximo de oxigênio ( $VO_{2max}$ ) após 4-20 semanas de treinamento, demonstrando eficácia para indivíduos com lesão na medula espinhal. Um desses métodos de treinamento aeróbico é conhecido como Treinamento Intervalado de Alta Intensidade (HIIT), o qual já demonstra benefícios de sua prática pela população geral. Apesar desse tipo de treino não ser muito examinado, uma pesquisa de Elizondo (2015) utilizando esse protocolo adaptado para indivíduos usuáries de cadeiras de rodas identificou um efeito aeróbico positivo, porém sem melhoria no consumo máximo do oxigênio.

Com isso, realizou-se a construção de um software de controle e de uma plataforma computacional de baixo custo, utilizando como base o treinamento HIIT para aferição do condicionamento aeróbico de pacientes com lesão medular. Assim, Foi-se desenvolvido na plataforma de eletrônica aberta “Arduino” um circuito com uma cinta cardíaca e um sensor óptico reflexivo tcr5000, o qual contabilizava as rotações do cilindro do ergômetro em um período de tempo, calculando os dados da velocidade e distância, que em seguida eram

repassados da placa Arduino UNO através da comunicação entre o módulo bluetooth AT-09 e o aplicativo Serial Bluetooth Terminal.

Assim, para facilitar o envio e visualização dos dados, a comunicação passou de bluetooth para WiFi, utilizando o microcontrolador ESP8266, o banco de dados Firebase e uma plataforma de desenvolvimento de aplicativos chamada App Inventor. Além disso, por problemas de conexão e recebimento dados, a cinta cardíaca foi substituída por um sensor de frequência cardíaca, que envia o sinal analógico para o microcontrolador da placa Arduino UNO, transmitindo os dados para o Firebase por meio da comunicação serial com a NodeMCU WiFi ESP8266.

Cogitou-se como substituição ao Firebase a utilização do banco de dados físico MySQL, que propunha estabilidade e fácil manuseio. Outrora, à medida que o projeto avançou para geração de relatórios em tempo real com React e React Native, a implementação do MySQL foi-se tornando cada vez mais difícil, haja vista que Firebase é um banco na nuvem atualizado em tempo real, tal mudança comprometeria o objetivo anteriormente mencionado. Então Firebase foi mantido como banco de dados intermediário, e primário foi o MongoDB, para armazenar as informações processados de exercícios e os dados cadastrais. Com base nesses registros, foi-se aplicada a técnica de clusterização através do método K Nearest Neighbor (kNN), o qual auxiliará nos próximos casos, determinando a melhor intensidade de treinamento a ser realizado.

## **2. PROCEDIMENTOS METODOLÓGICOS**

Cada sessão foi constituída por treino aeróbio intervalado seguido do resistido, trabalhando com o método de treino HIIT. Para determinar a sua intensidade, foi-se executado um teste incremental no próprio ergômetro de cadeira de rodas utilizado nos treinos. Desse modo, coloca-se uma marcação no cilindro traseiro do mecanismo que possui circunferência calculada, utilizando-o para contabilizar a quantidade de rotações realizadas em um período de dois segundos, através da utilização de um sensor tcr5000, como mostrado na Figura 1.

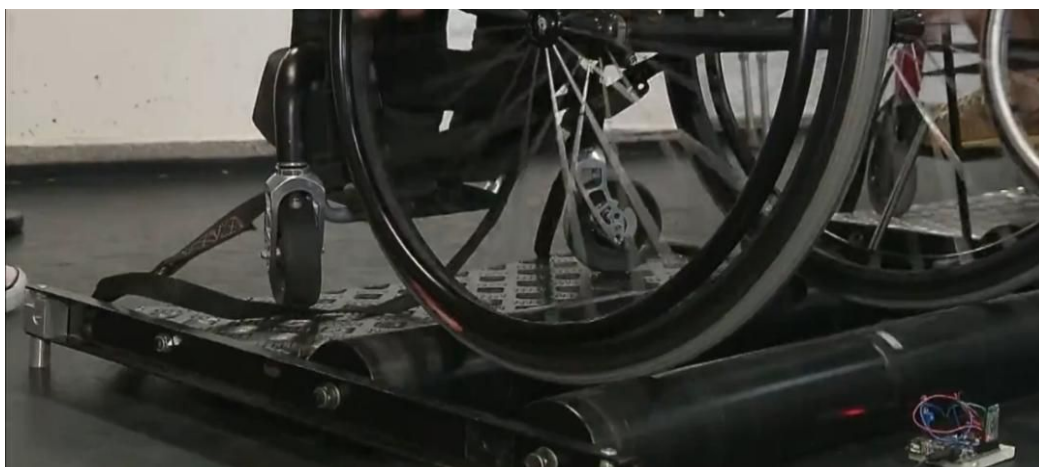


Figura 1 - Circuito posicionado para obter os dados da velocidade e distância percorrida.

A partir desse circuito elaborado, foi-se possível calcular os dados da velocidade e distância percorrida pelo indivíduo por meio de um código escrito na linguagem de programação C/C++ na plataforma Arduino. Inicialmente fez-se uso da placa Arduino UNO e um módulo bluetooth AT-09 para visualizar os dados no aplicativo Serial Bluetooth Terminal, no qual a finalização do exercício era controlada manualmente. Além disso, os dados de batimento cardíacos por minuto (BPM) eram adquiridos através de uma cinta cardíaca Bluetooth, a qual exibia os valores em outro dispositivo através de aplicativos como: Endomondo, RunKeeper, Runtastic.

Após a detecção de falhas, foi-se realizado um aprimoramento na implementação do aplicativo e do recebimento das informações referente a BPM do indivíduo. Passou-se a utilizar a plataforma de desenvolvimento App Inventor na confecção do aplicativo, a qual recebe os dados dos sensores pelo banco de dados Firebase. Assim, o microcontrolador ESP8266 tornou-se responsável pelo envio dos valores adquiridos para o Firebase Realtime Database (observado na Figura 2), já que o sensor óptico reflexivo passou a ser conectado na entrada digital da placa NodeMCU WiFi ESP8266, mas usufruindo da placa anterior para alimentação.

```

void calcula_velocidade(){

    int line_detector;                                // Inicializa a variável

    line_detector = digitalRead(sensor_vel);          // Detecta se a marcação no cilindro passou pelo sensor tct5000

    if (line_detector){                               // Ativou o sensor
        flag = 1;
    }

    if(!line_detector && flag){                        // Contabiliza no intervalo entre as marcações
        flag = 0;
        count += 1;
    }

    tempo = millis() - correcaoTempo;                // Calcula o intervalo de tempo

    if (tempo%2000 == 0) {                            // Verdadero caso tenha passado 2 segundos

        distancia = count * comprimento;              // Calcula a distância nos ultimos 2 segundos
        distAcumulada += distancia;
        Firebase.pushFloat("Distancia", distAcumulada); // Push na pilha e envia para o Firebase

        velocidade = (3.6 * distancia) / 3;          // Calcula a velocidade em km/h
        Firebase.pushFloat("Velocidade", velocidade);

        distancia = 0.0;                             // Reset na distância e contador
        count = 0;
    }
}

```

Figura 2 - Função do cálculo da velocidade e distância percorrida na plataforma Arduino, tendo comprimento =  $[2 * 3.14 * \text{raio\_do\_cilindro}]/100$ .

Outrossim, substitui-se a cinta por um sensor de frequência cardíaca colocado no lóbulo do indivíduo, o qual envia o sinal à entrada analógica da placa trabalhada, com auxílio do código “Getting\_BPM\_To\_Monitor” encontrado na biblioteca <PulseSensorPlayground.h>. Com isso, uma vez que o sensor estava conectado com a placa Arduino UNO, a comunicação serial foi-se necessária para repassar as informações, sendo conectada a porta Transmitter (Tx) da NodeMCU na Receiver (Rx) do Arduino. Já no caminho oposto, a Tx do Arduino passa por um regulador de tensão 3.3 volts(V) (LM1117T) para fazer a conexão com o Rx da ESP8266. Com isso, o código do sensor imprimia os valores no monitor serial, que posteriormente era lido e enviado para o Firebase.

Como tentativa de substituição do Firebase, utilizou-se o servidor de banco de dados MySQL por intermédio do XAMPP. Destarte, para realizar a comunicação, acrescentou-se as bibliotecas “MySQL\_Connection.h” e “MySQL\_Cursor.h” no código principal. Após algumas tentativas, no modelo final do projeto (Figura 4) optou-se por utilizar: o Firebase

como um banco de dados intermediário, o microcontrolador ESP8266 e o Arduino como fonte de alimentação dos sensores. Ademais, para facilitar a integração com a API do projeto, o banco de dados principal que passou a ser utilizado foi o MongoDB, por meio da biblioteca Javascript ‘mongoose’.

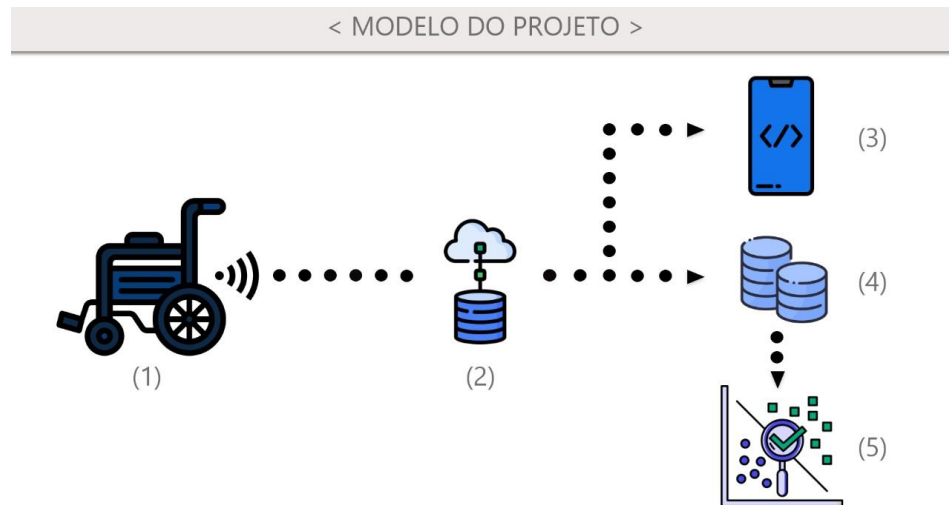


Figura 4 - (1) Circuito com o sensor tcr5000 e a NodeMCU Wifi ESP8266; (2) Firebase; (3) Aplicativo; (4) MongoDB; (5) Análise por clusterização.

Por fim, para fazer a análise do próximo caso, exporta-se em formato CSV os dados contidos no banco principal (Idade; Sexo; Causa da LM; Nível da LM; Categoria da LM; Gravidade da LM; Classe), sendo utilizados futuramente como a base de dados na técnica de clusterização. Com esses dados, aplica-se o método K Nearest Neighbor (kNN) por meio de um código produzido na linguagem de programação Python (destacado na Figura 5), determinando em qual das três classes distribuídas o novo indivíduo se classifica e registrando-a no banco de dados.

```
from sklearn.neighbors import KNeighborsClassifier # Importa biblioteca
Knn = KNeighborsClassifier(n_neighbors=3)
Knn.fit(X,y) # Passa as características para treinamento
Knn.predict(new_object) # Classifica o novo indivíduo
```

Figura 5 - Código em Python para classificar o indivíduo. Sendo y = [classes] e X = [Idade; Sexo; Causa da LM; Nível da LM; Categoria da LM; Gravidade da LM].



### 3. RESULTADOS E DISCUSSÃO

O projeto sofreu três fases de alterações, na qual as duas primeiras foram focadas na leitura e armazenamento com maior clareza dos dados da velocidade, batimentos cardíacos por minuto e distância percorrida. No contexto retratado, o primeiro aplicativo utilizado para exibir os dados calculados era o “unWired Lite”, o qual recebia os dados do circuito via bluetooth, mas após apresentar problemas foi substituído pela versão 1.31 do “Serial Bluetooth Terminal”. Contudo, o aplicativo não armazenava os detalhes do exercício, com isso apenas a média da velocidade final era transcrita, impossibilitando o reconhecimento do progresso do indivíduo durante a atividade. Além disso, o aplicativo era finalizado de forma manual, causando uma perda na precisão da velocidade média final, uma vez que o indivíduo precisa parar seu exercício para finalizar o programa.

Inicialmente a cinta cardíaca deveria ter seus dados também compartilhados no aplicativo, porém não foi possível conectá-la ao módulo bluetooth, uma vez que só funcionava o pareamento entre eles. Desse modo, tornou-se necessária a presença de dois aparelhos durante o exercício, sendo um para analisar os dados do circuito e outro mostrando os batimentos cardíacos por minuto do indivíduo, como apresentado na Figura 6.



Figura 6 - Visualização dos dados de BPM (esquerda) e dos gerados através do código na plataforma Arduino, sendo observados pelo aplicativo “Serial Bluetooth Terminal” (direita).

Em seguida, foi-se realizada a primeira aprimoração do sistema, o qual teve como principal mudança a construção de um aplicativo. Com base nisso, a plataforma App Inventor facilitou a visibilidade dos dados, porém não automatizou a finalização do programa, como visualizado na Figura 7. Ademais, tornou-se necessária a inclusão da NodeMCU WiFi ESP8266, trocando informações com a placa Arduino UNO através da comunicação serial. Contudo, surgiu um problema com relação a diferença de tensão, já que o Arduino trabalha com 5 volts(V) e a ESP8266 com 3.3 volts(V). A solução surgiu na utilização de um regulador de tensão, rebaixando a tensão vinda do Arduino.



Figura 7 - Código na plataforma App Inventor (esquerda) e a Tela do aplicativo durante o exercício (direita).

Após a conclusão do segundo protótipo do projeto, foi realizada uma visita ao Centro de Ciências Médicas - UFPB para realizar um teste do aplicativo e dos sensores. Atestou-se que a comunicação do hardware e software com o Firebase possuía um curto delay no recebimento de dados, porém não interferia de forma drástica na visualização das informações, podendo ser mantido no decorrer do projeto como um banco de dados intermediário. Entretanto, houveram falhas durante a visita no que diz respeito a precisão de

valores do sensor de frequência cardíaca e a sua localização, uma vez que o aparelho utilizado não se mantém estável caso haja muita movimentação.

Assim, foram realizadas algumas alterações nas técnicas e tecnologias utilizadas para abordagem do problema, e no que diz respeito aos equipamentos utilizados, a principal mudança foi na interrupção da conexão entre Arduino e ESP8266, com o objetivo de manter uma comunicação mais direta com o banco e evitar problemas de transmissão de dados. Essa ação tornou-se necessária, haja vista que a comunicação entre as duas placas não era perfeita e produziu lixo de memória, dados incorretos que precisaram ser descartados.

Com a interrupção nessa comunicação, optou-se primeiramente por manter o Arduino, substituindo o banco de dados na nuvem Firebase pelo banco MySQL. Contudo, apesar da plataforma App Inventor fornecer uma fácil integração com o Firebase, surgiu a necessidade da substituição do aplicativo, uma vez que o React - uma biblioteca em JavaScript que proporciona a criação de interfaces - possibilita uma visibilidade melhor dos dados. Por isso, a implementação direcionada para o uso do MySQL não prosseguiu, já que o banco de dados MongoDB possui uma integração mais prática com a API trabalhada nesta parte do projeto. Além disso, esse banco possui uma excelente performance em relação a outros bancos, um bom custo (gratuito e de código aberto), a possibilidade de estruturar bem os dados processados e simplicidade no que diz respeito a curva de aprendizado.

Portanto, o protótipo prosseguiu com o Firebase sendo o banco de dados intermediário, o Arduino como fonte de alimentação dos sensores, o MongoDB como banco de dados principal e o microcontrolador ESP8266. Por fim, devido a impossibilidade de realizar testes, não houve informações suficientes para gerar uma base de dados, que seriam utilizados posteriormente na prática de reconhecimento de padrões com a técnica de clusterização.

#### 4. CONCLUSÃO

Nos resultados deste estudo, foram apresentaram alguns erros, como o sensor de batimento cardíaco, que não captava corretamente os valores de batimento por minuto, podendo ser amenizado colocando-o no pododáctilo I.

O atraso de informações ou a presença de dados corrompidos entre a NodeMCU WiFi ESP8266 e o Arduino UNO foi solucionada, já que a captação dos dados feita pelo sensor passa diretamente pelo microcontrolador ESP8266, interrompendo a comunicação serial que existia entre ambas as placas.

Com a nova implementação do aplicativo desenvolvida na outra parte do projeto, os dados tornaram-se mais visíveis e o aplicativo foi automatizado, uma vez que programa-se o tempo de duração e inicia-se uma contagem regressiva, auxiliando na precisão da velocidade média final contabilizada.

Aconselha-se a realização de estudos com grupos de diferentes níveis de lesão medular e níveis de atividade física distintos, para a implementação de uma larga base de dados, usando-a como parâmetro na aprendizagem de máquina. Contudo, reforça-se o efeito positivo da intervenção realizada, já que construi-se uma plataforma utilizando softwares gratuitos e sensores de baixo custo, cumprindo quase todos os objetivos incluídos no plano de trabalho.

## REFERÊNCIAS

BRASIL, Diretrizes de Atenção à Pessoa com Lesão Medular. **Brasília: Ministério da Saúde**, 2ª edição, 2015.

CEREZETTI, C.et al. Lesão Medular Traumática e estratégias de enfrentamento: revisão crítica. **O mundo da Saúde**, São Paulo, v. 36, n 2, p. 318-326, 2012.

ELIZONDO,T. Effects of upper body HIIT training on recreationally trained wheelchair athletes. **Eastern Washington University**. Spring, 2015.

HOFFMAN, Martin D. Cardiorespiratory fitness and training in quadriplegics and paraplegics. **Sports Medicine**, v. 3, n. 5, p. 312-330, 1986.

NOREAU, Luc; SHEPHARD, Roy J. Spinal cord injury, exercise and quality of life. **Sports Medicine**, v. 20, n. 4, p. 226-250, 1995.