



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

RAFAEL DANTAS SANTOS DE AZEVÊDO

**RECONHECIMENTO DE EVENTOS USANDO SENSORES
MÓVEIS**

CAMPINA GRANDE - PB

2022

RAFAEL DANTAS SANTOS DE AZEVÊDO

**RECONHECIMENTO DE EVENTOS USANDO SENSORES
MÓVEIS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientador : Herman Martins Gomes

CAMPINA GRANDE - PB

2022

RAFAEL DANTAS SANTOS DE AZEVÊDO

**RECONHECIMENTO DE EVENTOS USANDO SENSORES
MÓVEIS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

Herman Martins Gomes

Orientador – UASC/CEEI/UFCG

Hyggo Oliveira de Almeida

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 02 de Setembro de 2022.

CAMPINA GRANDE - PB

RESUMO

A partir da disseminação do uso de *smartphones* nos últimos anos, tem crescido o interesse em utilizar os dados produzidos por seus sensores para monitorar temporalmente ações e eventos associados ao usuário ou ao ambiente onde se encontra o dispositivo. Tal monitoramento pode ajudar na compreensão do comportamento de portadores de *smartphones* de determinada localização geográfica e na formação de possíveis agrupamentos a partir das sequências de eventos/ações realizados ao longo do dia (e.g. caminhar, pular, correr, sentar etc.). O resultado deste processo pode ser útil ao levantamento de perfis de grupos populacionais. Neste contexto, o presente trabalho de conclusão de curso propõe um método para o treinamento e a predição de eventos/ações por meio de dados adquiridos de sensores móveis. Para tal, foi criado um aplicativo para realizar a aquisição dos dados, sendo obtidos dados do acelerômetro, giroscópio, microfone e GPS para formação de uma base de dados integrada. Ao executar o aplicativo os dados são gravados e o usuário pode rotular determinados eventos que estiverem acontecendo num determinado instante de tempo. Posteriormente, tais rótulos irão servir para que algoritmos de aprendizagem de máquina possam construir um modelo preditivo para os eventos.

Reconhecimento de eventos usando sensores móveis

Rafael Azevêdo

Universidade Federal de Campina Grande

Campina Grande, Paraíba, Brasil

rafael.azevedo@ccc.ufcg.edu.br

Herman Martins

Universidade Federal de Campina Grande

Campina Grande, Paraíba, Brasil

hmg@computacao.ufcg.edu.br

RESUMO

A partir da disseminação do uso de *smartphones* nos últimos anos, tem crescido o interesse em utilizar os dados produzidos por seus sensores para monitorar temporalmente ações e eventos associados ao usuário ou ao ambiente onde se encontra o dispositivo. Tal monitoramento pode ajudar na compreensão do comportamento de portadores de *smartphones* de determinada localização geográfica e na formação de possíveis agrupamentos a partir das sequências de eventos/ações realizados ao longo do dia (e.g. caminhar, pular, correr, sentar etc.). O resultado deste processo pode ser útil ao levantamento de perfis de grupos populacionais. Neste contexto, o presente trabalho de conclusão de curso propõe um método para o treinamento e a predição de eventos/ações por meio de dados adquiridos de sensores móveis. Para tal, foi criado um aplicativo para realizar a aquisição dos dados, sendo obtidos dados do acelerômetro, giroscópio, microfone e GPS para formação de uma base de dados integrada. Ao executar o aplicativo os dados são gravados e o usuário pode rotular determinados eventos que estiverem acontecendo num determinado instante de tempo. Posteriormente, tais rótulos irão servir para que algoritmos de aprendizagem de máquina possam construir um modelo preditivo para os eventos.

PALAVRAS-CHAVE

Sensoriamento móvel, Inteligência Artificial, Modelo preditivo.

LINKS ÚTEIS

Código fonte do aplicativo e notebook:

<https://github.com/Rafaeldsa/sensores-expo>

1. INTRODUÇÃO

Neste trabalho, sensores móveis se referem a qualquer um de vários tipos diferentes de dispositivos de aquisição de dados instalados em um *smartphone*, geralmente operados em conjunto com um aplicativo móvel [1]. A partir destes sensores é possível detectar ações, como aceleração, inclinação, vibração, direção do movimento, dentre outros.

Sabendo que o mundo está em constante progresso de transformação e de avanço digital, os *smartphones* tornam-se viáveis alternativas para o monitoramento, a coleta e o armazenamento de dados de eventos/ações. A ferramenta contém

tecnologias capazes de oferecer um conjunto de recursos e de informações, como também a possibilidade de acesso e de compartilhamento de conteúdos.

Posto isso, esta pesquisa visa ajudar na compreensão do comportamento de portadores de *smartphones* de determinada localização geográfica, formando possíveis agrupamentos a partir das sequências de eventos/ações realizados ao longo do dia (e.g. caminhar, pular, correr, sentar etc.).

É visto que existem câmeras de monitoramento, de segurança e de trânsito, porém somente com elas não há como inferir um perfil de uma população a partir de seus eventos/ações, pois estas câmeras apenas monitoram trajetórias e a identidade do objeto (e.g. pessoas, veículos, animais etc.). Ademais, existe também o *smartwatch* que inclui o monitoramento por meio de sensores. Porém, há lacunas pois o *smartwatch* é focado no monitoramento de funções vitais durante a realização de atividades físicas e compreende um conjunto limitado de sensores.

Neste trabalho, tem-se como embasamento teórico pesquisas relacionadas ao processamento de sinais com filtragem de ruídos com o objetivo de filtrar os dados coletados a partir do *smartphone*. Técnicas de aprendizagem de máquina (predição, agrupamento, classificação) também compõem o embasamento da pesquisa, dado que, ao final da coleta dos dados, modelos de aprendizagem de máquina são utilizados para realizar predições.

Neste contexto, foi desenvolvido um aplicativo móvel, chamado de Sensores, para fazer a coleta dos dados dos seguintes sensores: acelerômetro, giroscópio e GPS. Busca-se neste trabalho compreender a coleta desses dados e utilizá-los para prever as atividades de uma determinada localização e formar possíveis agrupamentos, ou seja fazer um levantamento demográfico em tempo real.

2. SOLUÇÃO

2.1 Visão Geral

O Sensores é um aplicativo Android voltado para os usuários de *smartphones*. O aplicativo possui um menu de seleção para especificação das ações durante a atividade que o usuário vai realizar como: correr, pular, etc. Além disso, há um menu para iniciar, parar e enviar a coleta dos dados dos sensores móveis, inclusive um cronômetro para visualização de tempo. (Figura 1)

Por fim, foi codificado um *notebook* utilizando a linguagem de programação Python para visualização dos dados adquiridos via

aplicativo, juntamente com os algoritmos de aprendizagem de máquina para treinamento e predição.

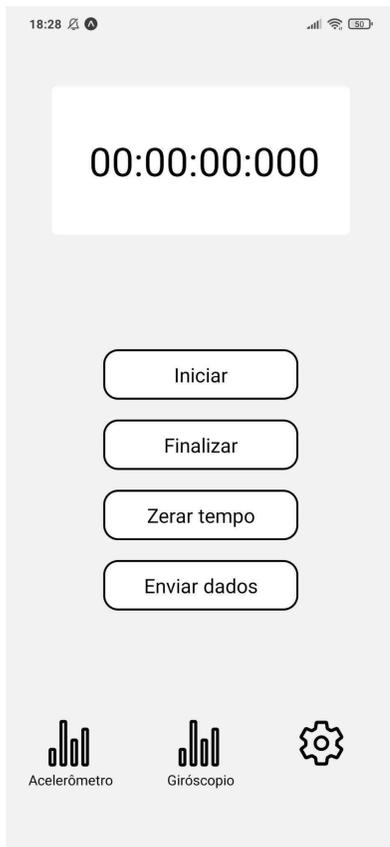


FIGURA 1. Tela inicial com cronômetro e menu.

2.2 Descrição

Neste trabalho foi desenvolvido um aplicativo para coleta de dados dos sensores dos *smartphones*, para levantar agrupamentos sociais de acordo com perfis de suas atividades realizadas. Além disso, também foi desenvolvido um código utilizando a linguagem Python para: visualização dos dados adquiridos e treinamento e avaliação de modelos de aprendizagem de máquina com os dados coletados pelos usuários para fins de predição.

Ao clicar no botão de configurações, visto como uma engrenagem na Figura 1, é apresentado ao usuário um menu com o objetivo de configurar/escolher quais sensores irá utilizar naquela coleta. É possível também, que, se o usuário escolher acelerômetro ou giroscópio, ele também pode escolher a taxa de coleta dos dados desses sensores. Essas taxas, podem variar no intervalo de 500 a 900 milissegundos.

Caso um desses 2 sensores sejam escolhidos, são apresentados na tela inicial 2 botões com ícones de gráficos. Dessa forma, ao clicá-los poderá observar o comportamento dos eixos de cada sensor (Figura 2). Por fim, o notebook programado na linguagem Python possui funções para visualização dos eixos do

acelerômetro e giroscópio, para isso foi utilizada a biblioteca matplotlib para plotagem dos gráficos. Além disso, foi utilizada a biblioteca tensorflow para criação e treinamento de redes neurais, com a finalidade de ter vários modelos sendo treinados e, ao final da predição dos dados, determinar qual obteve o melhor resultado. Além disso, foi gerada representação gráfica do modelo, como visto na Figura 3.

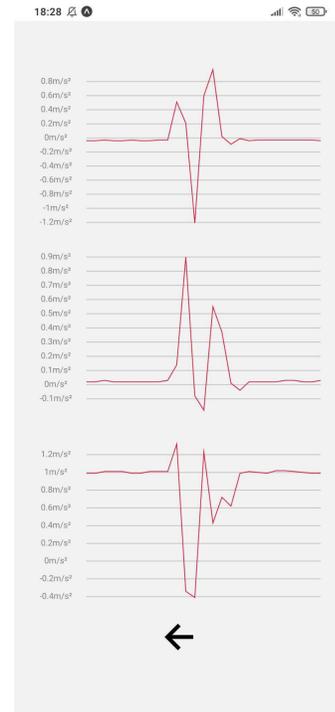


FIGURA 2. Tela com os eixos do acelerômetro.

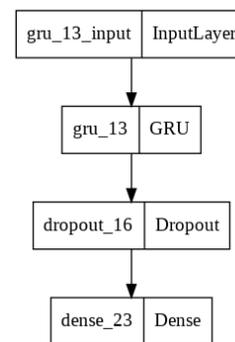


FIGURA 3. Representação do modelo GRU.

2.3 Funcionalidades

Segue abaixo uma lista de funcionalidades que foram planejadas e implementadas neste projeto:

- **Escolha dos sensores:** definição dos sensores que serão utilizados na coleta dos dados. Podendo escolher um ou mais.
- **Rotulagem dos dados:** menu com rótulos de ações e com botões de iniciar/parar, para que o usuário rotule os dados dado a ação/evento realizada naquele instante. (Figura 4)
- **Visualização dos dados:** demonstrar por meio de gráficos os dados coletados dos sensores do acelerômetro e giroscópio, juntamente com os rótulos. O intuito é visualizar como o dado do sensor se comporta em determinada ação/evento, como, por exemplo, os 3 eixos do sensor acelerômetro na ação de pulo. (Figura 5)
- **Preparação dos dados:** conjunto de funções para preparar os dados coletados para o treinamento das redes neurais. Implementando filtragem de ruídos, junção de todos os dados em um único *dataset* juntamente com os rótulos, com o objetivo de ter um janelamento (por se tratar de séries temporais) dos dados para posterior treinamento dos algoritmos de inteligência artificial.
- **Treinamento e predição dos dados:** definição dos modelos e realização do treinamento a partir dos rótulos obtidos com a coleta dos dados, com a finalidade dos algoritmos produzirem um modelo preditivo dos dados.



FIGURA 4. Tela de rótulos de ações/eventos.

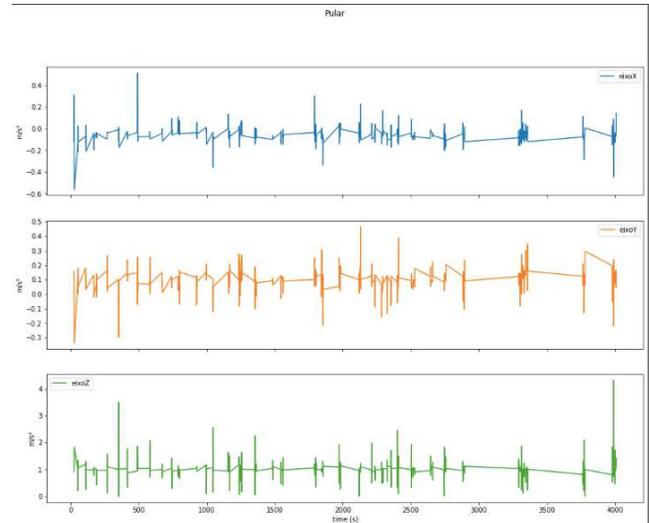


FIGURA 5. Eixos do acelerômetro na ação de pular.

2.4 Arquitetura

Para o desenvolvimento, o projeto foi dividido em *front-end*, que é onde os usuários interagem diretamente com o aplicativo, e o *notebook* em Python que é responsável pela visualização dos dados e treinamento dos algoritmos de aprendizagem de máquina.

2.4.1 Tecnologia do Aplicativo

No aplicativo, utilizou-se o React Native¹. O React Native é um framework baseado no já aclamado React, desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações *mobile*, tanto para Android, como para IOS, utilizando apenas a linguagem Javascript. O framework React Native possui bibliotecas de boa performance, além de funcionalidades para aquisição dos dados.

As vantagens do React Native são sua simplicidade de uso que possibilita um desenvolvimento ágil, e permite a criação de aplicativos nativos. Além disso, outra ferramenta utilizada foi o Expo².

O Expo é uma ferramenta utilizada no desenvolvimento móvel junto do React Native que permite o fácil acesso às APIs nativas do dispositivo sem precisar instalar qualquer dependência ou alterar código nativo.

Por esse fácil acesso às APIs nativas, foi escolhida esta ferramenta para ter acesso aos dados dos sensores do Android. Os dados então são armazenados em arquivos *csv* para serem enviados para a nuvem, Google Drive, utilizando a biblioteca expo-mail-composer. Logo após, os dados são tratados no Jupyter Notebook através do Google Colab.

Na Figura 6, ilustra-se a arquitetura do aplicativo.¹

¹ <https://reactnative.dev/>

² <https://expo.dev/>

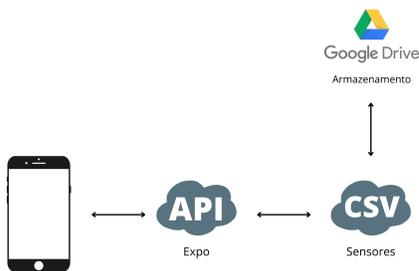


FIGURA 6. Arquitetura da aplicação mobile.

2.4.2 Notebook Python

Para este projeto foi decidido utilizar o Google Colab, mantido pelo Google, fornecendo suporte ao desenvolvimento de notebooks na linguagem Python. A linguagem Python é amplamente utilizada na área de ciência de dados por possuir uma vasta gama de bibliotecas que auxiliam nesse âmbito.

Já que os dados coletados vêm em forma de planilhas em formato csv, utilizou-se também a biblioteca Pandas por ter facilidade de tratar o formato desses arquivos, com objetivo de formar o dataset para as outras atividades seguintes.

Para visualização dos dados utilizou-se da biblioteca Matplotlib, já que possui um vasto ferramental para plotagem de gráficos. Além disso, foi utilizada a biblioteca Seaborn para calcular medidas estatísticas como, por exemplo, a média, para realizar uma pré-análise dos dados antes do seu treinamento/predição. Calculou-se a média de cada eixo tanto do acelerômetro quanto do giroscópio e utilizou-se a função pairplot da biblioteca Seaborn para mostrar as distribuições bivariadas de pares das médias referente a cada rótulo do evento/ação. (Figura 7)

Utilizou-se de bibliotecas bastante conhecidas como Numpy, que suporta o processamento de grandes, multi-dimensionais arranjos e matrizes, para manipular os dados coletados. Além disso, a biblioteca Keras para implementação de algoritmos de aprendizagem de máquina e realizar o treinamento, assim como a predição dos dados.

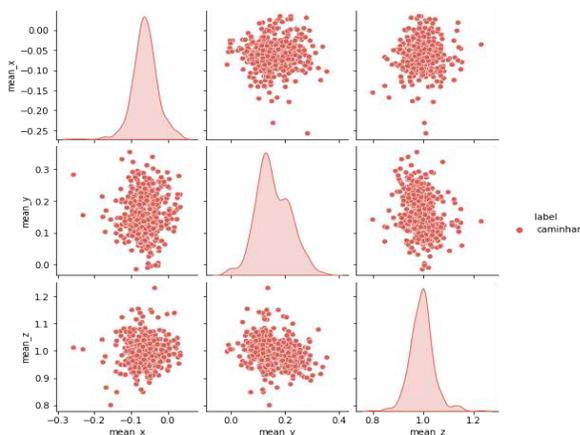


FIGURA 7. Distribuição das médias dos eixos na ação caminhar.

3. METODOLOGIA

Inicialmente, investigou-se como seria feita a coleta dos dados dos sensores dos smartphones. Levantaram-se 2 tecnologias como sendo para esta etapa: Android Studio, que é um ambiente de desenvolvimento integrado para desenvolver para a plataforma Android, e o React Native.

Então, após a criação do aplicativo, criou-se um planejamento para a coleta dos dados pelos usuários, de modo a padronizar a aquisição e evitar disparidades.

Após a coleta dos dados, a partir de seus eventos/ações, são montadas planilhas automaticamente pelo aplicativo, as quais são enviadas para o email criado exclusivamente para receber esses dados, para que depois sejam salvas no google drive para fins de uso no Notebook Python. (Figura 8)

Para fins iniciais, este trabalho propõe apenas utilizar dois sensores dos smartphones, o acelerômetro e o giroscópio. Consideraram-se os sensores mais importantes para a realização do reconhecimento das atividades humanas, pois com eles consegue-se medir a aceleração do movimento e posição do smartphone, possuindo assim dados que correlacionam com os movimentos.

Por serem séries temporais, foi feito um janelamento dos dados para prepará-los para o treinamento dos algoritmos de aprendizagem de máquina.

	tempo	eixoX	eixoY	eixoZ	label
0	00:00:00:106	-0.129802	0.198872	0.917782	normal
1	00:00:00:634	-0.118571	0.227192	0.961971	normal
2	00:00:01:134	-0.233805	0.356831	0.952693	normal
3	00:00:01:635	-0.204997	0.336079	0.862850	normal
4	00:00:01:685	-0.204997	0.336079	0.862850	caminhar
5	00:00:02:111	0.017416	0.333637	0.996638	caminhar
6	00:00:02:611	-0.225260	0.225728	0.713924	caminhar
7	00:00:03:111	-0.032389	0.031147	1.013973	caminhar
8	00:00:03:628	-0.170572	0.225728	1.055965	caminhar
9	00:00:04:129	-0.076334	0.162007	1.175106	caminhar

FIGURA 8. Exemplo do data frame gerado com dados do acelerômetro.

4. RESULTADOS

Posteriormente à coleta dos dados, foi estruturando um conjunto de treinamento para os algoritmos de aprendizagem de máquina. Na Tabela 1, tem-se a quantidade de cada rótulo coletado, significando a quantidade de atividades humanas coletadas que o aplicativo engloba, que são elas: caminhar, correr, pular, sentar e escadas (significando subir e descer os degraus).

Ação/Evento	Total
Caminhar	9124
Correr	286
Pular	1421
Sentar	965
Escada	547

Tabela 1. Total de amostras de dados por Ação/Evento.

Após ser feita uma pré-análise dos dados, definiu-se que não seria necessário realizar o processamento desses dados coletados. Como podemos visualizar nos gráficos da Figura 9, há uma taxa baixíssima de *outliers* (dados que se diferenciam drasticamente de todos os outros), e que os dados dos diferentes rótulos se comportam de forma distintas (Figura 10).

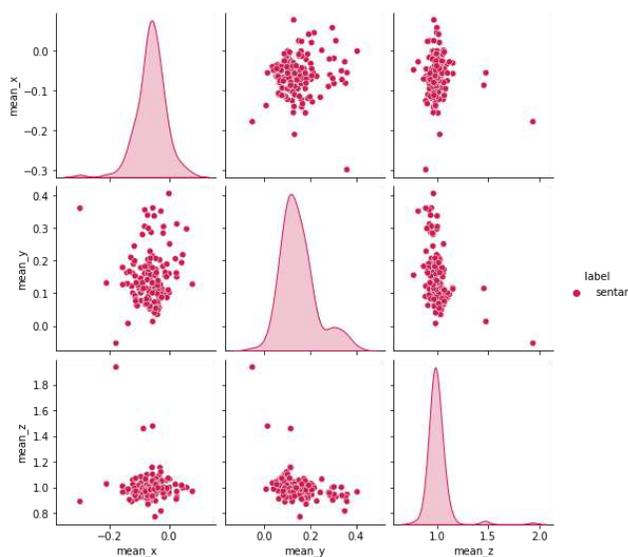


Figura 9. Representação dos eixos do acelerômetro para o rótulo sentar.

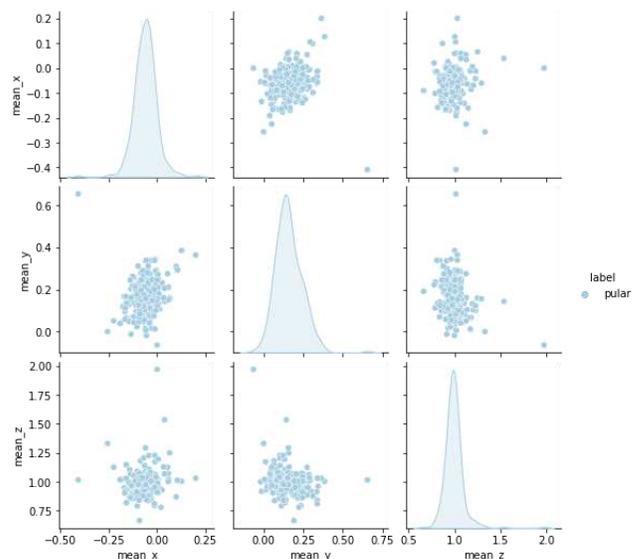


Figura 10. Representação dos eixos do acelerômetro para o rótulo pular.

Foram experimentados 3 algoritmos de aprendizagem de máquina, são eles Simple RNN, rede neural recorrente simples; LSTM, rede neural de memória de curto prazo e GRU, nova geração de redes neurais recorrentes e é bastante semelhante a uma LSTM.

LSTM e GRU são modelos recorrentes com estado interno, adequados ao processamento de informações temporais.

Após separar 60% dos dados para treinamento e 40% para a predição, aleatoriamente, e realizado o treinamento, os seguintes resultados foram obtidos para os sensores acelerômetro e giroscópio, respectivamente:

Acelerômetro: (Figura 11)

- **Simple RNN:** Obteve 80,33% de acurácia no conjunto de teste após o treinamento após o treinamento.
- **LSTM:** Obteve 86,10% de acurácia no conjunto de teste após o treinamento após o treinamento.
- **GRU:** Obteve 86,79% de acurácia no conjunto de teste após o treinamento após o treinamento.

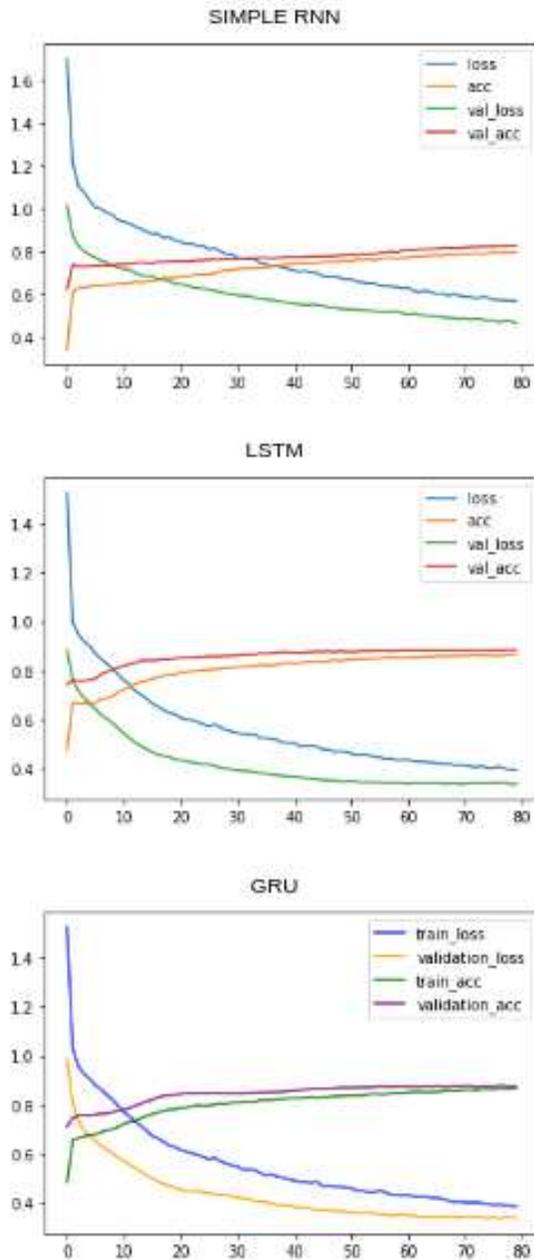


Figura 11. Gráficos pós treinamento dos dados do acelerômetro.

Giroscópio: (Figura 12)

- **Simple RNN:** Obteve 78,30% de acurácia no conjunto de teste após o treinamento após o treinamento.
- **LSTM:** Obteve 85,33% de acurácia no conjunto de teste após o treinamento após o treinamento.
- **GRU:** Obteve 81,82% de acurácia no conjunto de teste após o treinamento após o treinamento.

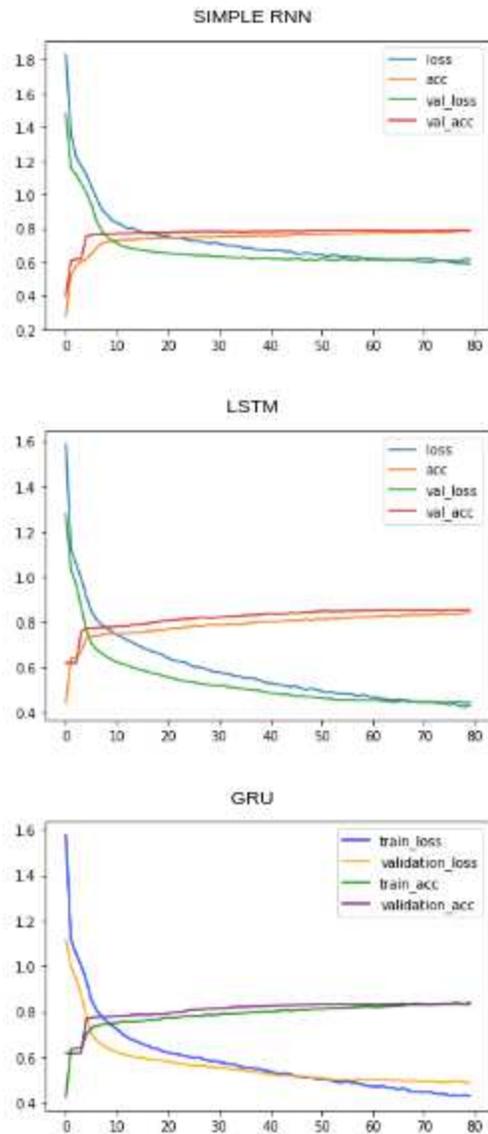


Figura 12. Gráficos pós treinamento dos dados do giroscópio.

5. EXPERIÊNCIA E LIÇÕES APRENDIDAS

Nesta seção será descrita a experiência propiciada pelo processo de desenvolvimento do aplicativo Sensores e o Notebook em Python. Assim como os desafios e trabalhos futuros.

5.1 Processo de desenvolvimento

Inicialmente, foram escolhidas as tecnologias que seriam utilizadas no desenvolvimento do projeto. Levou-se em consideração a experiência de outras aplicações desenvolvidas

quanto às tecnologias mais utilizadas atualmente. Após isso, elaborou-se a arquitetura para definir a estrutura tanto do aplicativo quanto do notebook.

Com a estruturação completa e todas as funcionalidades definidas, iniciou o processo de desenvolvimento tendo como base a metodologia Scrum, sendo ajustada de acordo com os horários disponíveis do aluno e do orientador. O motivo pelo qual essa metodologia foi escolhida é que estimula a entrega rápida de pequenas funcionalidades do sistema, e a evolução incessante no desenvolvimento através das *sprints* (ciclos de entrega).

5.2 Principais desafios

Um desafio foi a definição de um *layout* do aplicativo simples e bem estruturado que fosse fácil para o usuário utilizar para coletar os dados. Um segundo desafio foi estabelecer a forma de estruturar os dados no *notebook* já que provinham de arquivos separados. Com isso foi necessário associar os rótulos das ações com seus devidos dados coletados. Tendo isso em vista, escolheu-se utilizar de data frames em python para estruturá-los.

5.3 Limitações

As limitações encontradas, foi o impedimento de não ter como disponibilizar o aplicativo Sensores para o sistema IOS, já que sem isso, atualmente só é permitido a utilização em dispositivos com sistema operacional Android. Ocorreu devido a falta de conhecimento e falta de tempo devido a demanda principal da parte de desenvolvimento.

5.4 Trabalhos futuros

O trabalho teve como objetivo inicial a utilização dos sensores acelerômetro e giroscópio para predição de atividades humanas. Sabendo-se disso, trabalhos futuros podem ser implementados no Sensores e no *Notebook* Python, como as seguintes funcionalidades:

- Implementar o uso de outros sensores, como GPS, microfone e câmera do smartphone.
- Envio automático dos dados para o google drive.
- Implementar no notebook funções para tratar a representação da fala e poder usá-la nos treinamentos.
- Implementar algoritmos adicionais de aprendizagem de máquina para investigar se produzem melhor desempenho na predição dos dados.

Nesta perspectiva, as propostas de melhorias serão analisadas e projetadas para serem desenvolvidas futuramente, com intuito de proporcionar um aplicativo e um notebook mais robusto.

6. AGRADECIMENTOS

Agradeço a todos que estiveram, de forma ativa, presente no desenvolvimento do Sensores, ajudando com comentários para um bom aplicativo, em especial ao meu orientador Herman Martins Gomes, que deu todo o apoio e direções necessárias para o desenvolvimento deste TCC. Agradeço aos meus pais por todo apoio desde criança, que em todos os seus anos de vida me apoiaram e me motivaram a ir em frente em busca dos meus sonhos. À minha namorada, Rosmânia, por estar presente, apoiando minhas decisões e me ajudando no que precisava. Agradeço aos meus amigos, em especial aos amigos queridos do pizza day e do DQIU, João, Wesley, Eduardo, Roberto, Matheus, Gustavo e Gabriel, e a todos da graduação que a todo o momento estiveram dispostos em me ajudar durante o curso. Obrigado por todos os momentos que criamos e vivemos juntos durante esses anos.

7. REFERÊNCIAS

- [1] Wigmore, Ivy. Techtarget - Whats Is ?, 2014. <https://bit.ly/3b41mll>
- [2] Zhu, S.; Chollet, F. Working with RNNs, 2019. <https://bit.ly/3Bd1ccv>
- [3] Meta Platforms, Inc. React Native, 2022. <https://bit.ly/3cFt2UO>
- [4] Fernandes, Diego. Expo: o que é, para que serve e quando utilizar? <https://bit.ly/3vhmUbM>
- [5] Matsumoto, Fernando. Data Science | Visualização de Dados, 2019. <https://bit.ly/3Sh5e9Z>
- [6] Pedamallu, Hermanth. RNN vs GRU vs LSTM, 2020. <https://bit.ly/3SjdSF1>
- [7] Tensorflow Core. Recurrent Neural Networks (RNN) with Keras, 2022. <https://bit.ly/3CsCmGt>
- [8] Tensorflow Core. Layers LSTM, 2022. <https://bit.ly/3Ab9Fe9>
- [9] Tang, Yujian. Build a GRU RNN in Keras, 2022. <https://bit.ly/3Ab9Fe9>
- [10] Drumond, Claire. O que é scrum?, 2022. <https://bit.ly/3pyHNf9>

Sobre o autor:

Rafael Dantas Santos de Azevêdo é graduando de Ciência da Computação na Universidade Federal de Campina Grande, atualmente cursando o 9º período. Atualmente, trabalha como Aluno de PDI no Virtus, além de atuar em projetos de autoria própria, como desenvolvedor *Full Stack*. (<https://github.com/Rafaeldsa>)