

UNIVERSIDADE DO MINHO

MESTRADO EM ENGENHARIA INFORMÁTICA  
SENSORIZAÇÃO E AMBIENTE



Henrique Parola  
pg50415



Ana Henriques  
pg50196



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Domínio do projeto . . . . .	3
1.2	Motivação e objetivos . . . . .	3
1.3	Estrutura do relatório . . . . .	4
<b>2</b>	<b>Sistema Desenvolvido</b>	<b>5</b>
2.1	Arquitetura . . . . .	5
2.2	Funcionamento: A Rede Social . . . . .	5
2.3	Conexão ao Firebase . . . . .	10
<b>3</b>	<b>Sensores Utilizados</b>	<b>11</b>
3.1	Acelerômetro . . . . .	11
3.2	Giroscópio . . . . .	12
3.3	<i>Touch</i> . . . . .	13
3.4	Microfone . . . . .	13
<b>4</b>	<b>Modelos de Aprendizagem</b>	<b>14</b>
<b>5</b>	<b>Análise Crítica e Trabalho Futuro</b>	<b>16</b>



## Listas de Figuras

1	Arquitetura da aplicação Musarise . . . . .	5
2	Página para criar conta . . . . .	6
3	<i>Email</i> já é usado . . . . .	6
4	Página para fazer <i>login</i> . . . . .	6
5	Senha incorreta . . . . .	6
6	Página do meu perfil . . . . .	6
7	<i>Modal</i> com os seguidores . . . . .	6
8	Perfil de outro usuário . . . . .	6
9	Pesquisa de um perfil . . . . .	6
10	Seleção da imagem ao criar <i>post</i> . . . . .	7
11	Seleção de um áudio ao criar <i>post</i> . . . . .	7
12	Visão geral da criação de um <i>post</i> . . . . .	7
13	<i>Post</i> com áudio precisa de imagem . . . . .	7
14	Excerto do <i>feed</i> de publicações . . . . .	7
15	Secção de comentários . . . . .	7
16	Página da <i>waveform</i> do áudio . . . . .	7
17	Página com todos os instrumentos . . . . .	8
18	Página destinada à bateria . . . . .	8
19	Página destinada ao piano . . . . .	8
20	Página destinada à voz/microfone . . . . .	8
21	Página destinada à guitarra . . . . .	8
22	Captura de dados para treino . . . . .	8
23	Gravação do som produzido . . . . .	9
24	Criação e salvamento do som . . . . .	9
25	Biblioteca com os sons criados . . . . .	9
26	Exemplo de um desafio musical disponível . . . . .	9
27	Coleções na Firestore . . . . .	10
28	Movimento da bateria . . . . .	11
29	Movimento da guitarra . . . . .	12
30	Modelo do Piano . . . . .	13
31	Modelo do Microfone . . . . .	13
32	Dados coletados durante um movimento na guitarra . . . . .	14
33	<i>Accuracy</i> e <i>Loss</i> do modelo desenvolvido . . . . .	15



# 1 Introdução

## 1.1 Domínio do projeto

O domínio deste projeto insere-se em cinco áreas:

- **Sensorização em Dispositivos Móveis:** Com a revolução tecnológica dos *smartphones* na última década, os sensores de movimento embutidos nestes dispositivos têm proporcionado uma nova plataforma para inferência de atividades [1]. Essas atividades abrangem desde exercícios físicos para oferecer melhor qualidade de vida, até formas de entretenimento. Este trabalho tem como objetivo explorar a utilização de sensores voltados para o entretenimento e a vida social.
- **Vida Social:** As redes sociais desempenham um papel crucial nas interações humanas no mundo digital. Com a implementação de uma rede social, este trabalho contribui para conectar as pessoas ao criar uma plataforma *online* que permite interações e compartilhamentos de informações, indo além dos tradicionais *posts*.
- **Música:** Considerada uma das artes mais antigas da humanidade, o universo da música é um dos temas mais abordados no mundo digital. Assim, a criação e divulgação musical são os focos centrais da rede social desenvolvida neste trabalho.
- **Machine Learning:** No contexto da sensorização, os modelos de aprendizagem de máquina são comumente treinados para reconhecer com precisão os dados provenientes dos sensores. Lokesh Dhammi et al. [2] discutem exatamente a utilização destes modelos aliados aos sensores. Neste segmento, o presente trabalho também envolve a utilização de modelos de aprendizagem.
- **Entretenimento:** Bruno Gadelha et al. [3] mostram como pode ser interessante a inclusão de sensorização móvel para enriquecer o entretenimento de jogos digitais. Assim, aliar o conceito de uma rede social com o tema “música” garante como consequência o entretenimento dos utilizadores de uma aplicação.

## 1.2 Motivação e objetivos

Nos dias de hoje, a **música** desempenha um papel transcendente, unindo indivíduos de diferentes partes do mundo e despertando uma grande variedade de emoções. Assim como a matemática, também a música pode ser considerada uma linguagem universal. Contudo, é importante reconhecer que nem todas as pessoas têm a mesma igualdade de oportunidades para usufruir deste domínio artístico. O artigo [4] menciona precisamente o problema de cortes significativos nos orçamentos de educação do Reino Unido. Perante isto, foi introduzido o Exame de Bacharelado em Inglês (EBacc), com o objetivo de garantir que os alunos se dediquem especificamente a áreas científicas e humanísticas. Esta iniciativa contribuiu, porém, para a redução no número de estudantes que optam por disciplinas artísticas, limitando o desenvolvimento criativo dos jovens.

Para além da falta de orçamentos na educação, os escassos recursos financeiros, as divergências culturais e a aversão a novas tecnologias podem ser barreiras incontestáveis, que impossibilitam a sua participação ativa na criação sonora. Os sistemas de **sensorização**, por sua vez, estão cada vez mais presentes no quotidiano das pessoas. Frequentemente



associados ao **entretenimento**, estes sistemas trazem consigo uma revolução na indústria de aplicativos móveis. A título de exemplo, no estudo de Valpreda e Zonda [5], é criado um jogo educativo para ensinar as crianças sobre os problemas relacionados com o desperdício de alimentos. Para isso, a realidade virtual, juntamente com elementos da *Internet das Coisas* (IoT), são combinados num jogo para dispositivos móveis, de maneira a que os mais pequenos desenvolvam o seu sentido de responsabilidade, respeito e conscientização ambiental. Os sensores são, então, utilizados para caracterizar o ambiente em que a planta está a crescer, sendo esta representada por um avatar no jogo móvel.

É, neste contexto, que surge o projeto *Musarise*, uma **rede social** que procura superar as adversidades apontadas anteriormente, democratizando o acesso à música por meio de instrumentos virtuais. Combinando os termos "music", "musa" e "rise", enfatiza-se a ambição de impulsionar alguma inspiração artística e espírito autónomo nas pessoas.

Uma das características mais fascinantes desta aplicação é a sua capacidade de auxiliar artistas em períodos de estagnação criativa. De facto, todos os criadores, em algum momento, enfrentam a escassez de ideias e bloqueios na sua expressão artística. Ora, nessas circunstâncias mais desafiadoras, a *Musarise* emerge como uma ferramenta de apoio estimulante, fornecendo uma ampla gama de recursos musicais para fomentar a inspiração dos seus usuários. A partir de diferentes instrumentos virtuais e da interação com outros membros da rede social, conseguimos explorar novos talentos, expandir os nossos horizontes artísticos e enriquecer os outros com as nossas aprendizagens.

A *Musarise* objetiva, então, quebrar a mentalidade apática que caracteriza a sociedade contemporânea, ao estimular o espírito autónomo e criativo de cada indivíduo e ao transcender qualquer obstáculo que possa distanciar o ser humano do bem que é a música.

### 1.3 Estrutura do relatório

O presente relatório começa por explorar a arquitetura da aplicação, desde as suas funcionalidades, aos instrumentos virtuais disponíveis até a integração com o *Firebase*, possibilitando interações em tempo real. Para uma compreensão mais clara, serão incluídas capturas de tela da aplicação. Em seguida, é minuciosamente explicado como os sensores são utilizados para concretizar os objetivos da plataforma.

Paralelamente à aplicação *Musarise*, também foi implementado um pequeno modelo de aprendizagem profunda. Este modelo tem como propósito fazer previsões com base nos dados gerados pelos sensores, sendo isso explicado na secção 4. Por fim, uma conclusão abrangente e crítica do trabalho realizado é apresentada, juntamente com as perspectivas da equipa de trabalho para melhoramento da aplicação *Musarise*.

## 2 Sistema Desenvolvido

### 2.1 Arquitetura

O sistema desenvolvido possui uma arquitetura distribuída em duas componentes. Uma de suas componentes é a aplicação *mobile*, destinada ao utilizador final através da plataforma IOS. Será na aplicação *mobile* que os sensores serão diretamente utilizados. A restante componente baseia-se nos recursos de computação em nuvem do *Firebase* para a gestão do *back-end* da aplicação. Estas duas componentes interagem através do **protocolo HTTP** para realizar operações de autenticação e transferência de arquivos.

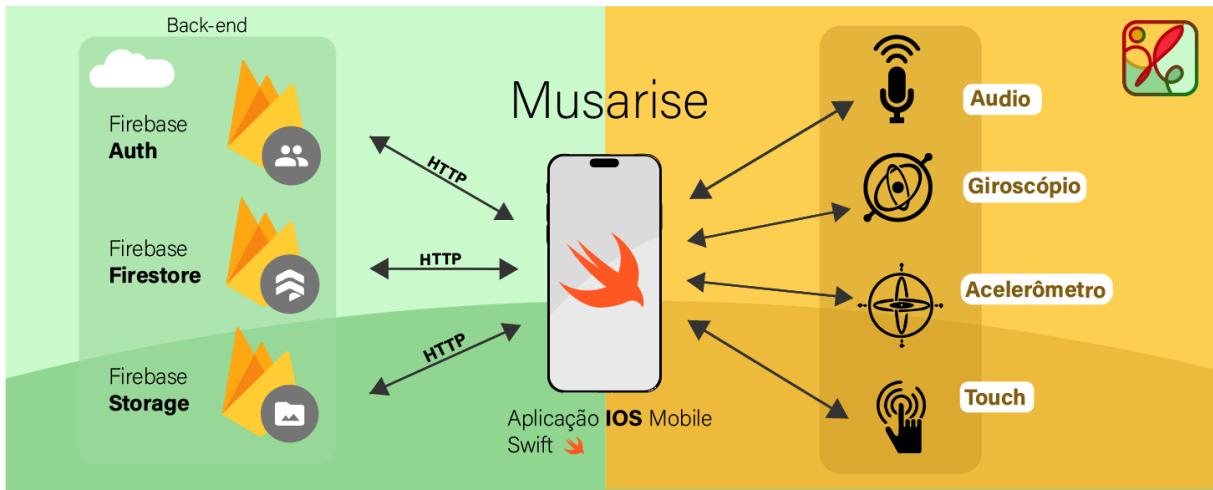


Figura 1: Arquitetura da aplicação Musarise

### 2.2 Funcionamento: A Rede Social

Nesta secção, são descritas as funcionalidades da aplicação *Musarise*, desde princípios básicos inerentes a uma rede social, até princípios mais avançados e focados na origem do projeto. Este tem como propósito a criação de música por meio de sensores, bem como a partilha dos resultados finais com outros membros da aplicação. Como tal, enumeramos, com o devido detalhe, as diversas ferramentas da *Musarise*, que, quando conjugadas, possibilitam a exploração de uma ampla gama de recursos e opções personalizáveis. Deste modo, os utilizadores da *Musarise* podem expressar a sua criatividade musical de maneiras emocionantes.

#### Registo da conta e *login*

Ao registrar uma conta na aplicação, o utilizador deve fornecer os seguintes dados pessoais: (1) o *email*, (2) o *username* e (3) a senha. Opcionalmente, é possível adicionar uma foto de perfil. No momento do *login*, o utilizador pode escolher inserir o seu *email* ou o *username*, juntamente com a senha. Nas Figuras 3 e 5, temos duas situações de erro que ocorrem quando já existe uma conta com o *email*, ou o nome de usuário, introduzido, e quando a palavra-passe fornecida está incorreta, respectivamente.

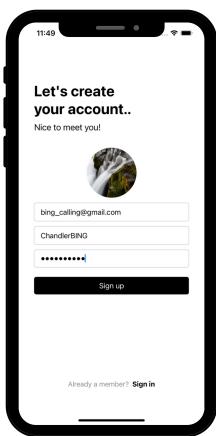


Figura 2: Página para criar conta

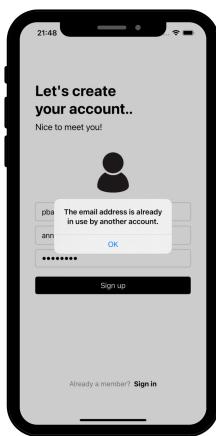


Figura 3: Email já é usado

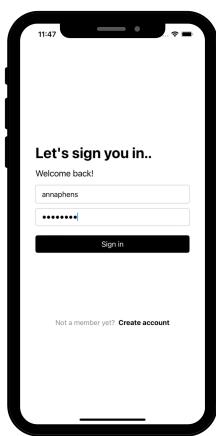


Figura 4: Página para fazer login

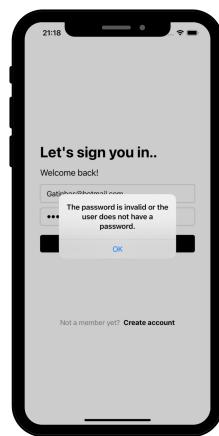


Figura 5: Senha incorreta

### Perfil de um usuário

Para além das informações pessoais fornecidas durante o registo da conta, a página de perfil de um utilizador exibe o número total de publicações feitas por ele, bem como *links* para o número de seguidores e de contas seguidas, que mostram exatamente esses indivíduos. A página apresenta, ainda, todos *posts* feitos pelo usuário em questão. Observe nas Figuras 6 e 8 que, quando se trata do perfil de outro usuário, a lista de publicações é apresentada como "Their Posts". No entanto, quando se trata do perfil do próprio usuário, a lista é denominada por "My Posts". É possível chegar ao perfil de outro usuário de diversas maneiras, nomeadamente através da opção de pesquisa – Figura 9.

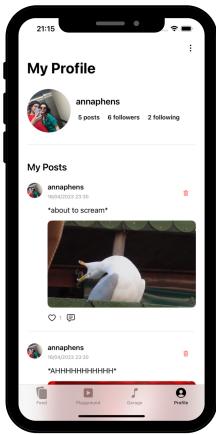


Figura 6: Página do meu perfil

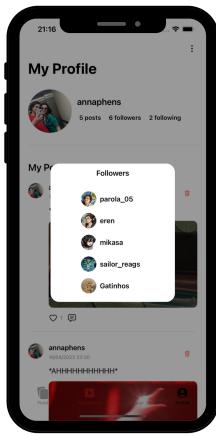


Figura 7: Modal com os seguidores

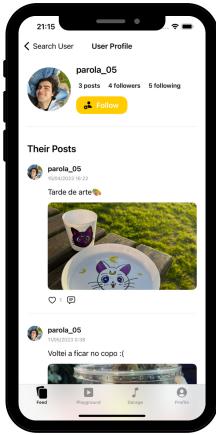


Figura 8: Perfil de outro usuário

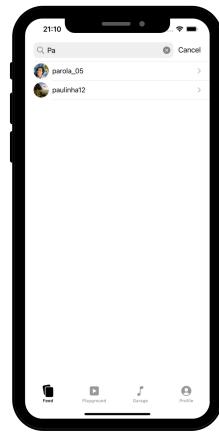


Figura 9: Pesquisa de um perfil

### Criação de uma publicação

A criação de um *post* pode incluir texto, imagem ou áudio. Contudo, ao adicionar um áudio previamente armazenado na garagem de sons que o usuário criou, é obrigatório também adicionar uma imagem. Isto porque, essa imagem servirá como um género de *cover* para o áudio. Na Figura 13, é ilustrado precisamente esse requisito.



Figura 10: Seleção da imagem ao criar *post*

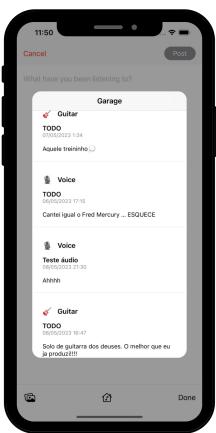


Figura 11: Seleção de um áudio ao criar *post*



Figura 12: Visão geral da criação de um *post*

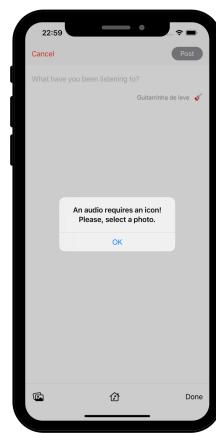


Figura 13: *Post* com áudio precisa de imagem

## Características de uma publicação

Após ser compartilhada com os demais membros da aplicação, a publicação assume a sua forma completa, composta pelo seu conteúdo, pelo *username* e pelo *icon* do usuário que a criou, juntamente com a data e hora em que foi publicada. Ao clicar na identificação do criador, a página é redirecionada para o perfil dele. Além dessas características, há também uma secção de *likes* e comentários, onde aqueles que têm acesso ao *post* – os seguidores do criador, podem interagir e fornecer seu *feedback*. Ao carregar na capa do áudio, a aplicação é redirecionada para a página com a *waveform* do áudio. Como se verifica na Figura 16, esta página tem como imagem de fundo a capa do som publicado, mostrando também a descrição da publicação.



Figura 14: Excerto do *feed* de publicações

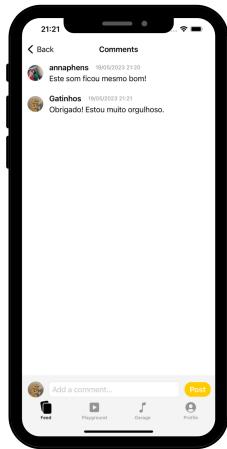


Figura 15: Secção de comentários



Figura 16: Página da *waveform* do áudio

## Playground com os vários instrumentos

Na Figura 17, temos a página "Playground", onde são apresentados todos os instrumentos disponíveis na aplicação: **guitarra** (21), **bateria** (18), **piano** (19) e **voz** (20). No caso da guitarra e da voz, as instruções de uso são claramente explicadas nas suas



respectivas páginas. O cenário é, no entanto, diferente para a bateria. Primeiramente, o usuário deve escolher um som de bateria para reproduzir. Após selecionar o som, ele pode imediatamente prosseguir para a criação de música, simulando o movimento de tocar o instrumento. Alternativamente, existe a opção de modo livre, onde são disponibilizadas três baterias, cada uma produzindo sons distintos. O piano, por sua vez, apresenta seis teclas que, ao serem tocadas, oferecem a animação de mudar de cor.

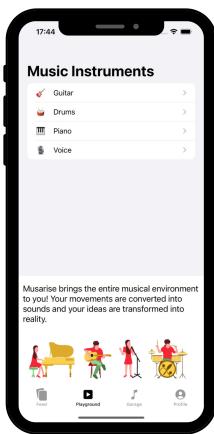


Figura 17: Página com todos os instrumentos

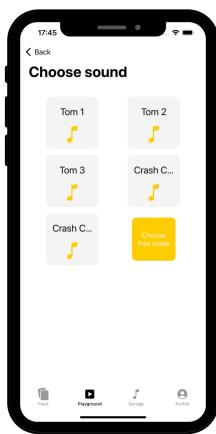


Figura 18: Página destinada à bateria

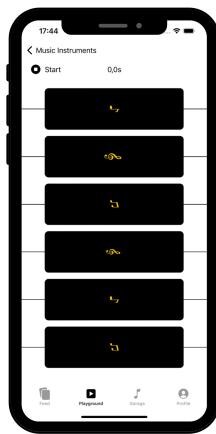


Figura 19: Página destinada ao piano

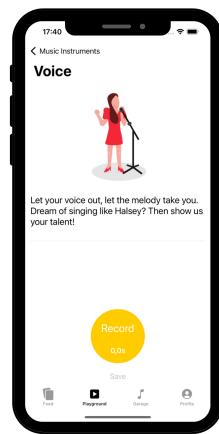


Figura 20: Página destinada à voz/microfone

### Registo dos valores do giroscópio para treinar modelos

A rede social oferece, adicionalmente, um modo de captura de valores gerados pelo giroscópio, visando a sua utilização para o aprimoramento da sensorização. Como cobaia, temos a guitarra virtual da *Musarise* que, ao carregar no botão localizado no canto inferior esquerdo, o utilizador pode decidir o momento exato em que os dados devem começar a ser registados. A consequência dessa ação é a mudança de cor da tela para sinalizar o acontecimento e a contribuição para um melhor funcionamento da aplicação.

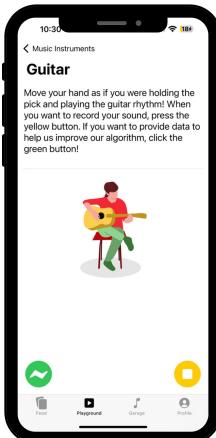


Figura 21: Página destinada à guitarra

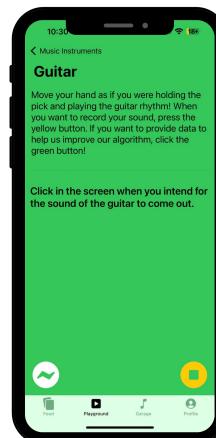


Figura 22: Captura de dados para treino

### Gravação e armazenamento do som

Depois de produzir um som, o utilizador não é obrigado a publicar o resultado final. Em vez disso, ele tem a opção de gravar e guardar a sua criação na sua biblioteca de áudios – Figura 25, para posterior audição e, quem sabe, compartilhamento com os seus seguidores. O formulário para salvar o resultado final está disponível na Figura 24, permitindo que sejam atribuídos um título e uma descrição.



Figura 23: Gravação do som produzido

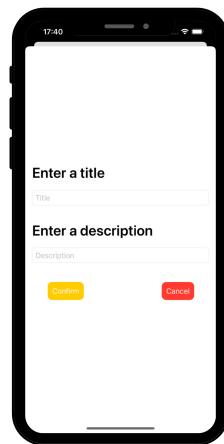


Figura 24: Criação e salvamento do som

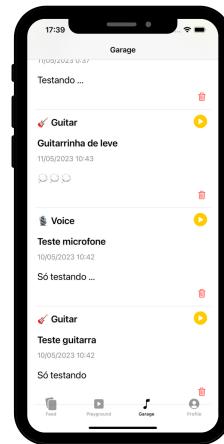


Figura 25: Biblioteca com os sons criados

### Participação em desafios musicais

A *Musarise* ambiciona entreter aqueles que usufruem dela, mas, acima de tudo, motivar e inspirar os amantes da música. Surgiu, assim, a ideia de desafiar os usuários a recriar sons bastante conhecidos. Basta consultar a página onde estão divulgados esses desafios, representada na Figura 26, e mergulhar na diversão que a música proporciona.

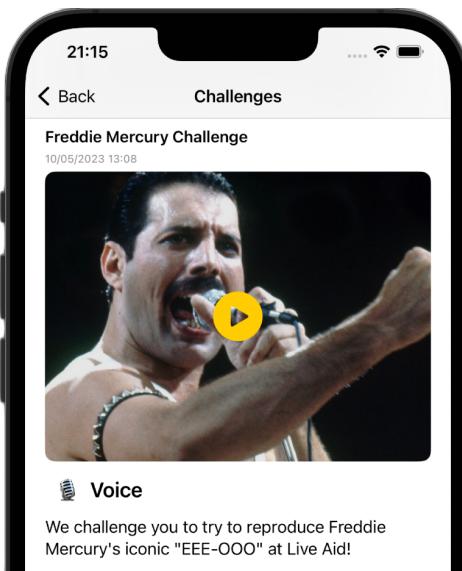


Figura 26: Exemplo de um desafio musical disponível



## 2.3 Conexão ao Firebase

A aplicação Musarise utiliza o *Firebase* para serviços de computação em nuvem de *backend*. Os produtos usados são:

- ***Firestore Database***: Escolhido por se tratar de uma base de dados NoSQL (utiliza o formato *JSON* para armazenamento), solução essa que é bastante recomendada para o contexto de "rede social" da aplicação.
- ***Storage***: Escolhido para o armazenamento de arquivos de áudios e imagens gerados pela aplicação.
- ***Authentication***: Escolhido para a integração do usuário na aplicação, assim como o processo de autenticação e *login*.

As coleções criadas no *Firestore* para dar suporte à aplicação não se limitam apenas aos *posts*, desafios e utilizadores criados. Todos os sons da aplicação estão também devidamente catalogados. Isto traz escalabilidade para a aplicação, podendo facilmente se expandir com a adição de novas notas musicais sem a exigência da alteração do código fonte. Outra utilidade desta tarefa é não tornar a aplicação cliente "pesada" com os arquivos de áudios. Assim, existem as coleções *Guitar*, *Drum* e *Piano*, em que cada uma possui as suas notas musicais identificadas por um nome, referenciando a *url* do arquivo de áudio correspondente no serviço de *Storage*.

Document ID	Document Type	Document Data
csound-967d4	Collection	+ Iniciar coleção Challenges Drums Guitar Piano Playground Posts Training Users
Guitar	Collection	+ Adicionar documento 1nQFDP5hN8bDXaGvF70H 4LqNg378bHqS2CIBXc6k eRvRuRyR2vGQIpV7vpY1 fjGGSPHSLjcXDnhAVeGj 12Z4Vn8JwEJXv4m53N6 uudza8CqjKLoKM7g9hLJ
eRvRuRyR2vGQIpV7vpY1	Document	+ Iniciar coleção + Adicionar campo name : "La Free" soundURL : "https://storage.googleapis.com/v0/b/csound-967d4.appspot.com/o/Instruments%2FGuitar%2FGuitarLaFree.mp3?alt=media&token=fadc28ac-4f73-4ce6-88eb-6060dbaa40ec"

Figura 27: Coleções na Firestore

Para além das coleções mencionadas, existem também as coleções *Playground* e *Training*. A *Playground* armazena os sons produzidos, mas não publicados pelos utilizadores. Deste modo, este recurso funciona como uma galeria virtual onde o utilizador pode guardar o que produziu e, posteriormente, ter a possibilidade de concretizar uma publicação com isso. Por fim, a *Training* armazena os dados cedidos pelos utilizadores durante os seus movimentos na utilização do modo guitarra. Estes dados poderão servir para o treino de modelos de aprendizagem para o algoritmo de emissão de som da guitarra. No capítulo 4, será explicado com mais pormenor a utilidade destes dados.



## 3 Sensores Utilizados

As informações dos sensores são obtidas através das *frameworks CoreMotion e ReplayKit* do *Swift*. A *framework CoreMotion* fornece acesso aos dados de sensores de movimento do dispositivo, como o acelerómetro e o giroscópio, essenciais para a simulação dos sons de alguns instrumentos, como a bateria e a guitarra. Por outro lado, o *ReplayKit* permite gravar o áudio do aplicativo. Com estes recursos é possível gravar não só a voz do utilizador, como também o áudio produzido durante a simulação dos instrumentos. De seguida, é listado mais minuciosamente cada um destes sensores no contexto da aplicação.

### 3.1 Acelerómetro

O **sensor acelerómetro** é utilizado para criar o som da **bateria**. Esse sensor mede as acelerações do dispositivo em diferentes direções, permitindo detetar o momento em que a batida é tocada e a intensidade dos golpes dados na bateria virtual. A aplicação disponibiliza 5 tipos de sons diferentes – 1 *floor tom*, 1 *rack tom* e 2 *crash cymbal*, todos extraídos a partir da plataforma *online Virtual Drumming*.

A estratégia adotada para a reprodução do som da bateria baseia-se na análise da variação do sinal da aceleração, tendo em conta a sua polaridade. No processo de execução, quando a baqueta começa o seu movimento descendente em direção ao instrumento, os valores da aceleração são registados como positivos. No entanto, à medida que se aproxima do ponto de impacto, ocorre uma transição para valores negativos, sendo este intervalo crucial para determinar o volume sonoro. Quanto mais rápido for esse intervalo de tempo em que a aceleração é negativa, maior será a intensidade resultante. Este intervalo termina assim que a aceleração voltar a ser positiva, representando este o momento em que a baqueta se eleva e se afasta do instrumento. A Figura 28 ilustra a posição que o dispositivo deve assumir para a concretização do processo descrito.



Figura 28: Movimento da bateria

No contexto da bateria virtual, existem dois modos possíveis de reprodução: o modo simples e o modo livre. No modo simples, é apenas tocado um dos 5 sons disponíveis ao longo da simulação. Em contrapartida, o no modo livre, já não inclui uma seleção pré-definida de um som. Em vez disso, existem 3 partes da bateria em posições diferentes. A

ideia é que, ao mover-se mais para a direita, esquerda ou centro, o resultado gerado varia de acordo com a tal parte do instrumento que se encontra nessa posição. Esta abordagem proporciona uma experiência mais realista e expressiva à simulação.

### 3.2 Giroscópio

O sensor **giroscópio** é utilizado para a criação do som da **guitarra**. Tal sensor mede a taxa de variação na qual um dispositivo gira em torno de um eixo espacial. Ou seja, é medida a velocidade angular do dispositivo. Os valores de rotação são medidos em radianos por segundo em torno do eixo específico. Os valores de rotação podem ser positivos ou negativos, dependendo da direção da rotação. O movimento considerado para a simulação da guitarra foi aquele que é gerado pela mão que segura a palheta. Isto é, a aplicação reproduz a mão que realiza o “ritmo” do som. A Figura 29 mostra o movimento de rotação simulado pela aplicação.



Figura 29: Movimento da guitarra

Para produzir o som da guitarra, foram consideradas as seis notas correspondentes às cordas soltas (Mi, Lá, Ré, Sol, Si e Mi), tocadas individualmente, sem a formação de acordes. É como se apenas a mão do “ritmo” fosse considerada. Os sons de cada nota foram obtidos a partir da plataforma [Recursive Arts](#).

O mecanismo por trás da reprodução dos sons segue o seguinte princípio: *a intensidade da variação da rotação reflete-se na quantidade de notas tocadas*. Quanto maior a variação obtida, mais notas são emitidas. Deste modo, foi preciso apenas determinar qual seria a variação detetada que iria corresponder a reprodução das seis notas de uma vez. A partir deste valor e de uma regra de três simples, pode-se obter o número de notas que devem ser reproduzidas através da variação de rotação atual.

As seis notas são armazenadas numa lista, seguindo a ordem “de cima para baixo” das cordas da guitarra. Para além disto, a última nota tocada é conhecida e mantida em registo. Desta forma, a partir do mecanismo de obtenção da quantidade de notas a serem tocadas juntamente com a direção do movimento, é possível identificar que notas devem ser reproduzidas, acedendo **n** notas à frente ou atrás da última nota tocada, conforme necessário. Após essa seleção, a última nota reproduzida deve ser atualizada.



### 3.3 Touch

O **sensor de touch** é usado para recolher o toque do usuário nas 6 teclas virtuais do piano. Os sons de cada tecla foram conseguidos através da plataforma [Recursive Arts](#). Quanto mais suave for o toque, melhor será a resposta às variações de pressão exercidas e, consequentemente, mais precisa será a sequência de sons produzida. Esta funcionalidade é preciosa para aspirantes a pianista que, com a capacidade de se expressarem musicalmente através de um toque, podem criar uma experiência interessante de tocar piano.

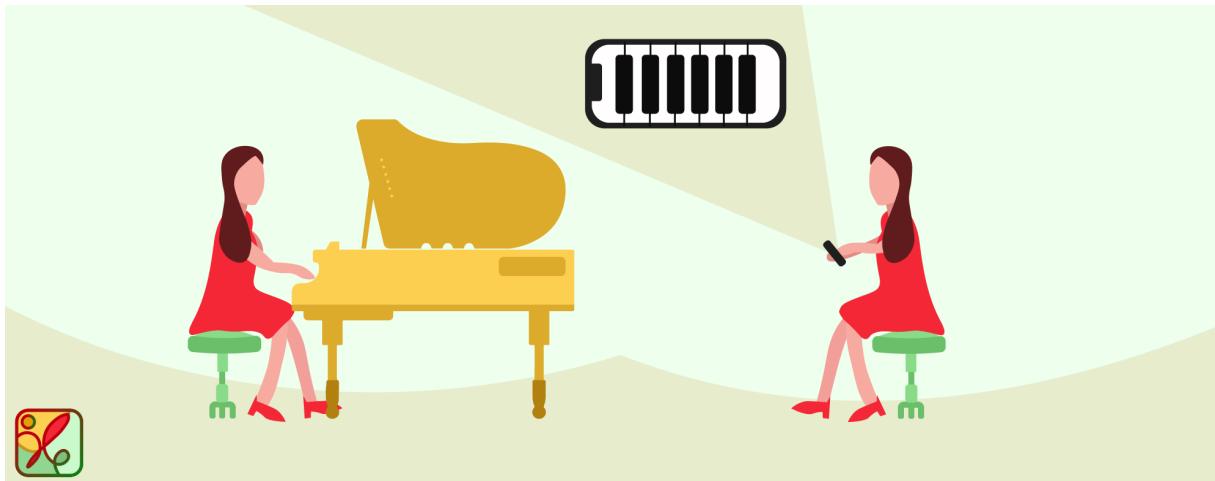


Figura 30: Modelo do Piano

### 3.4 Microfone

O **sensor do microfone** é utilizado para a captação do áudio do utilizador. Quanto mais perto da boca do utilizador, melhor as ondas sonoras são captadas pelo sensor. O objetivo desta *feature* é permitir o utilizador sentir-se como o “cantor de uma banda”.



Figura 31: Modelo do Microfone



## 4 Modelos de Aprendizagem

O sensor do giroscópio possui algumas limitações físicas face à reprodução do som da guitarra. Esta limitação ocorre em três níveis:

- Durante a aplicação de um ritmo, o tempo e a distância que separam duas notas é extremamente pequeno. Por vezes, este tempo pode até ser menor que o tempo de atualização dos dados do giroscópio;
- Não seria exequível tentar manter uma ideia de “distância percorrida” ou “localização no espaço”. Isto porque, o erro do sensor poderia se acumular e tornar muito difícil manter uma ideia da distância ou da localização do dispositivo ao longo do tempo. Daí termos optado por considerar a magnitude da rotação para determinar a quantidade de notas tocadas.
- Teríamos de ser capazes de “prever” atempadamente a nota que será tocada de modo a amenizar o *delay* do som.

Por estas razões, a guitarra possui um recurso de **coleção de dados do movimento de rotação** do dispositivo. Toda-via, invés do som da guitarra ser produzido conforme o método previamente explicado, o utilizador informa a aplicação quando é que deseja a emissão do som. Desta forma, podemos obter uma sequência de dados da velocidade angular do dispositivo ao longo do tempo, etiquetados com a informação acerca do som ter ou não sido emitido.

Esses dados sequenciais assemelham-se às séries temporais com quatro dimensões – três para os eixos de rotação e uma para a emissão do som. Estes dados podem facilmente servir de treino para algum modelo de *Deep Learning*, que possui memória de curto/longo prazo. O problema em questão seria: *“Dado uma sequência de  $n$  velocidades angulares ao longo do tempo, prever se a velocidade angular  $n + 1$  corresponde a uma emissão de som”*. Um modelo capaz de realizar estas previsões poderia melhorar consideravelmente o algoritmo para emissão do som das notas.

Inicialmente, utilizámos o modo de **coleção de dados** da "guitarra" da *Musarise* para a construção de um pequeno *dataset*. De seguida, procurámos concretizar um modelo de aprendizagem para testar se seria possível realizar a tarefa de previsão desejada.

rotationList	
▼	0
tapped:	false
x:	0.5182559490203857
y:	-0.3755669593811035
z:	0.08607283234596252
▼	1
tapped:	false
x:	0.48128753900527954
y:	-0.16809393465518951
z:	0.1556713581085205
▼	2
tapped:	false
x:	0.3291744291782379
y:	0.11953731626272202
z:	0.14705069363117218
▼	3
tapped:	false
x:	0.15825675427913666

Figura 32: Dados coletados durante um movimento na guitarra

A primeira etapa consistiu na conversão do *JSON* exportado para o formato de *array numpy*. Uma das decisões tomadas foi subdividir as sequências de velocidades angulares em intervalos menores. Imaginando que um utilizador usa o modo de coleção de dados por 30 segundos, não faria sentido treinar o modelo recebendo aproximadamente 29,5 segundos de dados de rotações, a fim de prever se a velocidade angular no instante 30' corresponde a uma emissão de som. Este último movimento muito provavelmente nem sequer corresponderia a uma produção sonora. Além disso, ao longo desses 30 segundos, pode ter ocorrido uma emissão de som, tornando a divisão em intervalos menores e pré-definidos um complemento bom para o modelo. Esta técnica é conhecida como "janela deslizante", comumente utilizada para dividir sequências temporais longas em sequências menores, para aumentar a quantidade de dados de treino. Feito este processo de divisão, conseguimos transformar 5 sequências iniciais de dados em 228!

A exploração dos *datasets* revelou uma enorme discrepância nos dados: cerca de 133 rotações foram registadas como "emissoras de som", enquanto que 5.620 foram classificadas como o oposto. Estes valores são congruentes, atendendo o curto intervalo de tempo em que a velocidade angular do dispositivo é registrada. Para dar a volta esta questão, recorremos à injeção sintética de dados através de técnicas de *Random Over Sample* e *Random Under Sample* para balancear os dados. O modelo concebido foi uma rede neural que utilizava somente uma simples camada LSTM. A Figura 33 mostra os valores de *accuracy* e de *loss* do modelo ao longo do treino.

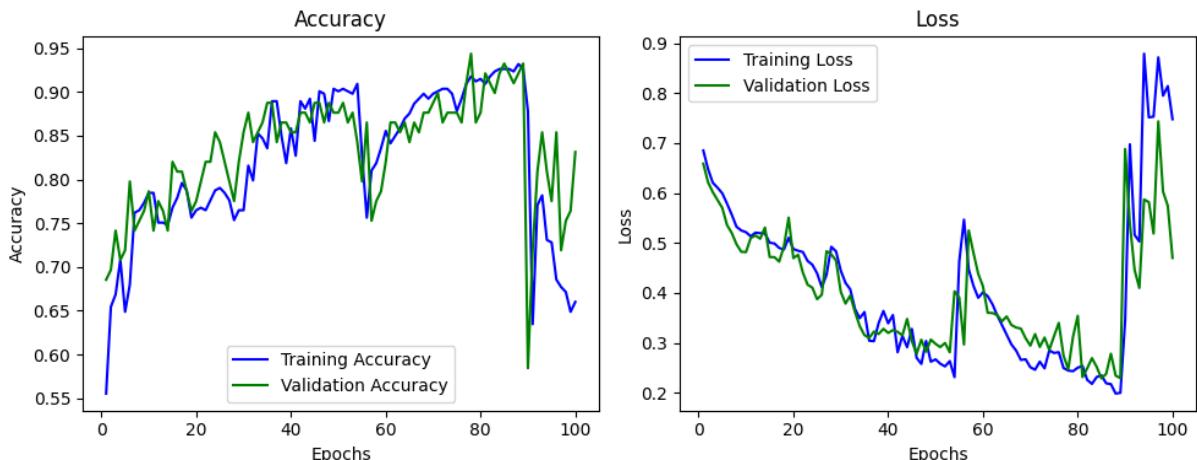


Figura 33: *Accuracy* e *Loss* do modelo desenvolvido

Um ponto importante a ser ressaltado é a implicação da injeção sintética de dados para equilibrar os dados de treino. Ao realizá-la, perde-se a noção temporal dos dados e a tarefa passa a ser a classificação de se uma determinada velocidade do dispositivo corresponde a uma emissão de som da guitarra. O artigo [6] apresenta técnicas para injeção sintética balanceada com ordenação temporal dos dados, mas não foi encontrado nenhuma forma consolidada e exequível para a sua execução em *Python*. A diferença significativa nos instantes de tempo em que ocorre, ou não, a emissão de som é um dos fatores que limitam a implementação de um modelo de *deep learning* para este problema.



## 5 Análise Crítica e Trabalho Futuro

A forma atual de armazenamento e reprodução dos áudios na aplicação sobressai pontos positivos e negativos. Como vantagens, destaca-se a flexibilidade ao adicionar e remover novos sons e instrumentos musicais diretamente através do *Firebase*, sem exigir modificar o código fonte. Em contrapartida, a reprodução dos áudios apresenta um *delay* devido à localização remota dos arquivos de áudio. Esse atraso compromete a experiência do usuário ao utilizar os instrumentos musicais da aplicação, embora não seja um problema grave durante a visualização de *posts*. Como trabalho futuro, propõe-se a migração dos arquivos de áudio dos instrumentos para o próprio dispositivo do usuário final, visando reduzir esse atraso. No *trade-off* entre **flexibilidade** e **experiência**, acredita-se que a experiência do usuário deve ser priorizada.

O modelo de *deep learning* desenvolvido foi apresentado com o principal propósito de demonstrar a utilidade para o aprimoramento da sensorização. Como trabalho futuro, sugere-se uma forma de mitigar a ausência dos dados discutidos na secção 4. Isso pode envolver a inserção sintética de dados que preservam a ordem temporal, como também a criação de formas alternativas para colectar a velocidade angular do dispositivo. Atualmente, o utilizador informa o momento de reprodução do som através de um clique. O objetivo é, então, possibilitar que o usuário avise que fará um movimento do tipo "ritmo X", e o sistema aprenda a detectar o padrão desse movimento. Esta é somente uma sugestão, entre outras possíveis. Além disso, seria interessante a interligação da aplicação com um possível modelo em funcionamento, no sentido de calcular com mais precisão os momentos de reprodução de um som de guitarra.

O grupo de trabalho deseja, ainda, implementar duas funcionalidades adicionais na *Musarise*: notificações e um *chat* para comunicação entre os membros. Ao seguir um usuário, ele receberá uma notificação informando que foi seguido por X. O mesmo ocorrerá quando alguém deixar de o seguir. Outros eventos comuns numa rede social, como gostar ou comentar uma publicação, também serão motivo de notificação. Além disso, sempre que novas ferramentas forem adicionadas ao aplicativo, todos os membros serão devidamente informados. Estas funcionalidades visam, por isso, promover uma melhor interação entre os usuários e uma maior aproximação à aplicação, garantindo que todos estão atualizados sobre eventos e modificações na *Musarise*.

## Referências

- [1] Alvina Anjum and Muhammad U. Ilyas. Activity recognition using smartphone sensors. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 914–919, 2013.
- [2] Lokesh Dhammi and Piyush Tewari. Human activity recognition based on smartphone sensors- a comparative study. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–6, 2021.
- [3] Gustavo de Freitas Martins, Rosiane de Freitas, and Bruno Gadelha. A mobile game based on participatory sensing with real-time client-server architecture for large entertainment events. In *2020 XLVI Latin American Computing Conference (CLEI)*, pages 332–339, 2020.
- [4] Karen Burland. Music for all: Identifying, challenging and overcoming barriers. *SAGE Journals*, 2020.
- [5] Fabrizio Valpreda and Ilaria Zonda. Grüt: A gardening sensor kit for children. *Special Issue Sensors for Entertainment*, 2016.
- [6] Pu Zhao, Chuan Luo, Bo Qiao, Lu Wang, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. T-smote: Temporal-oriented synthetic minority oversampling technique for imbalanced time series classification. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2406–2412. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.