



# **Técnicas emergentes de Inteligência Artificial em computação móvel para promoção de atividade física no contexto da mHealth**

Mestrado em Engenharia Informática - Computação Móvel

Alexandre Louro de Sousa

Leiria, março de 2024



# **Técnicas Emergentes de Inteligência Artificial em Computação Móvel para Promoção de Atividade Física no Contexto da mHealth**

Mestrado em Engenharia Informática - Computação Móvel

Alexandre Louro de Sousa

Trabalho de Projeto realizado sob orientação de Professor Doutor José Carlos Bregieiro Ribeiro ([jose.ribeiro@ipleiria.pt](mailto:jose.ribeiro@ipleiria.pt)), Professor Doutor Fernando José Mateus da Silva ([fernando.silva@ipleiria.pt](mailto:fernando.silva@ipleiria.pt)) e Professora Doutora Joana Patrícia dos Santos Cruz ([joana.cruz@ipleiria.pt](mailto:joana.cruz@ipleiria.pt)).

Leiria, março de 2024



# Resumo

A finalidade do projeto OnTRACK é a promoção de atividade física em pacientes com doença pulmonar obstrutiva crónica, sendo esta acompanhada diretamente por um profissional de saúde para uma recomendação de atividade física mais personalizada às necessidades de cada um.

O projeto OnTRACK desenvolveu uma plataforma digital inovadora de treino personalizado. Esta é uma plataforma constituída por duas aplicações, *web* e *mobile*, em que a de *web* é exclusivamente para os profissionais de saúde e a *mobile* para os pacientes.

Este projeto focou-se no desafio de tornar a execução dos exercícios deste treino personalizado mais rigorosa, para garantir uma maior qualidade e segurança. Outra grande parte deste projeto é a “gamificação”, pois este ajuda no envolvimento do paciente e dá-lhe mais motivação para realizar os exercícios, por ser algo mais divertido.

O projeto está estruturado em 5 partes: Introdução, Estado da arte, Desenvolvimento, Verificação e Validação e Conclusão.

Na introdução é apresentada uma breve contextualização e explicação do projeto.

No estado da arte são revistos artigos e projetos relevantes ao tema, abordando tópicos como o de sensores, *body pose estimation* e realidade aumentada.

A secção de desenvolvimento detalha o processo de desenvolvimento do projeto, incluindo a implementação de cada componente, *body pose*, realidade aumentada, entre outros, com exemplos de código.

Na verificação e validação são feitos testes com utilizadores reais e, ao final, são avaliados por meio de um questionário, além da utilização de *System Usability Scale* visando obter conclusões mais precisas a respeito do *feedback* recebido.

As conclusões alcançadas incluem, a necessidade de sensores externos e/ou o uso de *Machine Learning*. O uso de realidade aumentada e a “gamificação” são como ferramentas que auxiliam os pacientes a manterem a motivação.

Das tecnologias, bibliotecas e *frameworks* utilizadas destacam-se o ARCore, o TensorFlow Lite e a API do Google Maps.

Palavras-chave: atividade física, AR, exercícios, ARCore, TensorFlow Lite, Google Maps.

# Abstract

The purpose of the OnTRACK project is to promote physical activity in patients with Chronic Obstructive Pulmonary Disease, with direct guidance from a healthcare professional for a more personalized physical activity recommendation tailored to each individual's needs.

The OnTRACK project has developed an innovative digital platform for personalized training. This platform consists of two applications, web and mobile, with the web app exclusively for healthcare professionals and the mobile app for patients.

This project focused on the challenge of making the execution of exercises in this personalized training more rigorous to ensure greater quality and safety. Another significant aspect of this project is gamification, as it helps engage patients and provides them with more motivation to perform exercises by making it more enjoyable.

The project is divided into four parts: State of the Art, Development, Verification and Validation, and Conclusion.

In the State-of-the-Art section, relevant articles and projects related to the topic are reviewed, including topics such as sensors, body pose estimation, and augmented reality.

The Development section details the development process of the project, including the implementation of each component, such as body pose, augmented reality, among others, with code examples.

Verification and validation involve testing with real people and evaluating with a survey, as well as using the System Usability Scale to draw better conclusions about the feedback.

The conclusions reached include the need for external sensors and/or the use of Machine Learning for exercise validation. The use of augmented reality and gamification helps patients maintain motivation. Among the technologies, libraries, and frameworks used, notable mentions include ARCore, TensorFlow Lite, and the Google Maps API.

**Keywords: physical activity, AR, exercises, ARCore, TensorFlow Lite, Google Maps.**

# Índice

<b>Resumo .....</b>	<b>iv</b>
<b>Abstract .....</b>	<b>vi</b>
<b>Lista de Figuras .....</b>	<b>ix</b>
<b>Lista de siglas e acrónimos.....</b>	<b>x</b>
<b>1. Introdução .....</b>	<b>1</b>
<b>2. Estado da arte .....</b>	<b>3</b>
<b>2.1. Soluções envolvendo sensores .....</b>	<b>3</b>
2.1.1. Human physical activity recognition using smartphone sensors.....	5
2.1.2. Gymskill: a personal trainer for physical exercises.....	9
2.1.3. Discussão .....	10
<b>2.2. Soluções envolvendo reconhecimento de poses .....</b>	<b>10</b>
2.2.1. Discussão .....	19
<b>2.3. Soluções envolvendo Augmented Reality .....</b>	<b>19</b>
2.3.1. Discussão .....	23
<b>3. Desenvolvimento .....</b>	<b>24</b>
<b>3.1. Conceitos e tecnologias relevantes.....</b>	<b>24</b>
3.1.1. Realidade aumentada .....	24
3.1.2. ARCore .....	24
3.1.2.1. Conceitos fundamentais do ARCore .....	24
3.1.2.2. ARCore Geospatial.....	25
3.1.3. TensorFlow Lite .....	25
3.1.3.1. Pose estimation .....	26
3.1.3.2. Video classification .....	26
3.1.4. Sensores do dispositivo móvel .....	26
3.1.5. Google Maps API .....	26
<b>3.2. Metodologia de desenvolvimento .....</b>	<b>27</b>
3.2.1. Prototipagem evolutiva.....	27
3.2.2. Levantamento de requisitos .....	28
<b>3.3. Arquitetura .....</b>	<b>29</b>
3.3.1. Arquitetura da plataforma.....	29
3.3.2. Arquitetura do módulo desenvolvido – Aplicação <i>Web</i> .....	30
<b>3.4. Descrição da solução desenvolvida.....</b>	<b>31</b>
3.4.1. Funcionalidades da app móvel .....	31

3.4.2.	Funcionalidade de realidade aumentada .....	31
3.4.3.	Funcionalidade do mapa.....	34
3.4.4.	Sensores de acelerómetro .....	36
3.4.5.	Funcionalidade do <i>body pose estimation</i> .....	36
3.4.6.	Funcionalidades da app web .....	39
3.4.7.	Funcionalidades do backend .....	43
<b>4.</b>	<b>Verificação e validação .....</b>	<b>44</b>
<b>4.1.</b>	<b>Aplicação Web .....</b>	<b>44</b>
<b>4.2.</b>	<b>Aplicação móvel.....</b>	<b>46</b>
<b>5.</b>	<b>Conclusão .....</b>	<b>51</b>
<b>6.</b>	<b>Bibliografia .....</b>	<b>54</b>

# Lista de Figuras

Figura 1: Padrão da atividade "andar", figura 2 do artigo [24] .....	6
Figura 2: Padrão da atividade "correr", figura 3 do artigo [24] .....	7
Figura 3: Padrão da atividade "sentar", figura 4 do artigo [24] .....	7
Figura 4: Padrão da atividade "de pé", figura 5 do artigo [24] .....	8
Figura 5: Padrão da atividade "subir escadas", figura 6 do artigo [24].....	8
Figura 6: Padrão da atividade "descer escadas", figura 7 do artigo [24] .....	9
Figura 7: Arquitetura da plataforma .....	30
Figura 8: Captura de ecrã do modelo animado AR 3D.....	32
Figura 9: Captura de ecrã do mapa na plataforma de mobile .....	35
Figura 10: Captura de ecrã do body pose na plataforma mobile .....	39
Figura 11: Captura de ecrã do <i>score</i> na plataforma <i>mobile</i> .....	39
Figura 12: Captura de ecrã do mapa na plataforma web .....	40
Figura 13: Captura de ecrã adicionar exercício na plataforma web.....	40
Figura 14: Questionário tarefa 1 .....	45
Figura 15: Questionário tarefa 2.....	45
Figura 16: Questionário tarefa 3.....	46
Figura 17: Questionário tarefa 1 aplicação móvel .....	47
Figura 18: Questionário tarefa 2 aplicação móvel .....	48
Figura 19: Questionário tarefa 3 aplicação móvel .....	48
Figura 20: Questionário tarefa 4 aplicação móvel .....	49
Figura 21: Questionário tarefa 5 aplicação móvel .....	49

# Lista de siglas e acrónimos

Elemento a figurar, **quando aplicável**.

3DPCK	3D percentage of correct keypoints
AL	Activity Learning
ANN	Artificial Neural Networks
AP	Average Precision
AR	Realidade Aumentada, Augmented Reality
ARC	Activity Recognition and Classification
ASM	Average Euclidean distance between ground-truth and predicted joint positions
ATL	Activity and Transition Learning
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
CRF	Conditional Random Field
DC-NN	Deep Convolutional Neural Networks
DNN	Deep Neural Network
DPOC	Doença Pulmonar Obstrutiva Crónica
EDM	Euclidean Distance Matrices
EEFV	Estações de Exercício Físico Virtuais
ESTG	Escola Superior de Tecnologia e Gestão
FLIC	Frames Labeled In Cinema
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model with Expectation-Maximization
IA	Inteligência Artificial
IEF	Iterative Error Feedback
LSP	Leeds Sports Dataset
LSTM	Long Short-Term Memory
MEI-CM	Mestrado em Engenharia Informática – Computação Móvel
MFAP	Máximo à Posteriori
ML	Machine Learning
MLP	Multilayer Perceptron

MPII	Max Planck Institute for Informatics Human Pose
MPJE	Mean Per Joint Position Error
MoViNets	Mobile Video Networks
NMS	Parametric Pose Non-Maximum-Suppression
ORPM	Occlusion-Robust Pose-Maps
OSTAR	Optical See-Through Augmented Reality
PAF	Part Affinity Fields
PCK	Percentage of Correct Keypoints
PDJ	Porcentagem de Articulações Detetadas
PGPG	Pose-Guided Proposals Generator
PIFPAF	Part Intensity Field Part Association Field
PT	Personal Trainer
PrSVM	Probability Support Vector Machine
SEQ	Single Ease Question
SMPL	Skinned Multi-Person linear model
SURREAL	Synthetic hUmans foR REAL tasks
SUS	System Usability Scale
SVM	Support Vector Machine
TFLite	TensorFlow Lite
VGG-LG	Very Deep Convolutional Networks for Large-Scale
VPS	Visual Positioning System
WAF	We Are Family
WGS	World Geodetic System



# 1. Introdução

A doença pulmonar obstrutiva crónica (DPOC) provocou, cerca de 2500 mortes em Portugal no ano 2019 [1]. A DPOC [2] é uma das doenças respiratórias crónicas mais comuns globalmente, sendo definida por sintomas respiratórios persistentes, como a dispneia, tosse e produção de expectoração, bem como pela diminuição do fluxo de ar devido a modificações nas vias aéreas e/ou no tecido pulmonar, resultantes da exposição a substâncias gasosas ou partículas nocivas.

Reconhecendo que a atividade física reduz a mortalidade desta doença [3], o projeto OnTRACK visou a promoção de atividade física em pacientes com DPOC. Por outras palavras, o projeto OnTRACK pretendeu incentivar as pessoas com DPOC a praticar regularmente atividade física o que, por consequência, pode ter impacto na melhoria da qualidade de vida e redução da progressão da doença.

O projeto OnTRACK deu resposta a este desafio através do desenvolvimento, avaliação e disseminação de uma plataforma digital inovadora de treino personalizado - a plataforma OnTRACK. Esta é uma plataforma dividida em duas aplicações, *web* e *mobile*, sendo que a parte *web* se destina a profissionais de saúde e a *mobile* a pacientes de DPOC.

Como referido anteriormente, a plataforma OnTRACK tem duas vertentes, *web* e *mobile*, sendo que a plataforma *web* é manuseável apenas pelos profissionais de saúde para fazerem recomendações de atividade física e exercícios específicos, com o propósito de gerirem dados e informações dos pacientes. A aplicação móvel é para a utilização dos pacientes, sendo que esta aplicação disponibiliza informação sobre os seus objetivos de atividade física, tendo estes sido definidos pelos profissionais de saúde. Esta aplicação móvel tem também a integração com *wearables* (*smartbands*) para recolha de informação sobre a atividade física, tal como passos, dados e calorias, para facilitar a disponibilização de dados e mantê-los atualizados automaticamente.

O presente documento foca-se no módulo de Estações de Exercício Físico Virtuais (EEFV), desenvolvido no contexto da Unidade Curricular de Projeto do Mestrado em Engenharia Informática - Computação Móvel (MEI-CM), e integrado na plataforma OnTRACK com o propósito de tornar mais apelativa a realização de exercícios por parte dos utilizadores/pacientes de DPOC, e de potenciar o alcance dos objetivos prescritos pelos profissionais de saúde. Pretende-se, também, através deste módulo, incentivar os utentes a utilizar a aplicação e a conseguirem realizar os exercícios com qualidade e segurança.

O módulo de EEFV visa, pois, dotar a plataforma OnTRACK de funcionalidades diferenciadoras no contexto da *e-Health* aplicada à problemática da DPOC, com enfoque na utilização de técnicas emergentes de Inteligência Artificial (IA) em Computação Móvel, e especificamente de computação cognitiva, sensibilidade ao contexto e realidade aumentada (AR), para potenciar a promoção da atividade física. Pretende-se, nomeadamente:

- aplicar técnicas de *Machine Learning* (ML) *on-device* (e.g., *Body Pose Estimation* [4]), para dotar a app móvel de mecanismos que permitam validar a correta execução dos exercícios prescritos de forma automatizada;
- conceber experiências de AR geolocalizadas [5] com o objetivo de criar “estações de exercício virtual” em locais reais identificados como sendo particularmente adaptados à prática de exercício físico (e.g., parques, jardins, passadiços), e onde a AR poderá ser empregue para demonstrar os exercícios prescritos;
- utilizar ferramentas de geolocalização e *geofencing* [6] para guiar os utilizadores ao encontro das EEFFV, tornando o desafio mais apelativo e promovendo, no processo, a realização de exercício (e.g., caminhadas, corridas) com base no contexto e na localização dos utilizadores.

Transversalmente, foi também objetivo deste trabalho dotar a plataforma de uma maior robustez, estabilidade e manutenibilidade, utilizando as melhores e mais recentes boas práticas de desenvolvimento móvel e *web*, a fim de criar um protótipo funcional que possa ser utilizado por pacientes e profissionais de saúde.

Para este efeito, usou-se a metodologia de, primeiramente, fazer uma investigação de outros projetos e artigos existentes, para se ter uma melhor ideia do que existe, e a partir daí tirar as conclusões se as tecnologias, realmente, são as indicadas para este módulo em específico, denominando-se estado da arte. Passou-se então ao desenvolvimento já com uma ideia do que existia e o que se poderia fazer.

Depois fez-se a verificação e validação deste desenvolvimento, e, por fim, tirou-se as conclusões do projeto.

Deste processo resultou um novo protótipo funcional da plataforma OnTRACK, que integra as funcionalidades de EEFFV desenvolvidas no contexto deste trabalho, e que foi alvo de validação em ambiente controlado (verificação e validação referida, anteriormente).

Este documento organiza-se da seguinte forma:

- Estado da arte: Neste capítulo representam-se outros documentos e plataformas em que são abordados o uso dos sensores de *smartphones*, *body pose estimation* e AR, na saúde.
- Desenvolvimento: Este capítulo apresenta a metodologia de desenvolvimento, descreve o processo de implementação do projeto no âmbito das novas *features*, a arquitetura utilizada e fornece uma explicação das tecnologias e conceitos relevantes.
- Verificação e Validação: Este capítulo apresenta os testes de usabilidade, com um questionário *System Usability Scale* (SUS) no final para dar a entender a opinião dos utilizadores em relação às aplicações.
- Conclusão: Na conclusão faz-se a análise do projeto no geral e das tecnologias na saúde. No capítulo faz-se também uma reflexão do que poderá ser melhorado e/ou acrescentado.

## 2. Estado da arte

Neste capítulo são abordados outros documentos e plataformas retratando o uso de sensores de telemóveis, *body pose estimation* e AR, em particular na área da saúde e para confirmação dos exercícios.

### 2.1. Soluções envolvendo sensores

Nesta secção são retratadas, inicialmente, as aplicações existentes, não puramente académicas; e seguidamente os estudos, artigos e revisões encontrados em que se usem sensores para reconhecer os exercícios e/ou fazer a confirmação. Irá fazer-se uma breve síntese de todos, desenvolvendo dois exemplos, selecionados com base no contexto, deste ser similar ao do projeto OnTRACK. Foi usado como referência o artigo de Chaurasia & Redd [7].

O objetivo desta secção é o de demonstrar a dificuldade de se obter uma boa precisão apenas com os sensores do dispositivo para exercícios algo complexos. Nos exemplos escolhidos, optou-se por usar atividades básicas do quotidiano.

Começando por Rabbi e colaboradores [8], que criaram uma aplicação denominada MyBehavior, em que o objetivo é o de guardar os dados de atividade física e rotina de refeições e fazer sugestões ao utilizador para melhorar esses hábitos. Esta aplicação é relevante porque, assim como no projeto OnTRACK, o objetivo geral é promover a atividade física.

Em CalFit App [9] foi utilizada uma aplicação que conta os passos e que envia notificações para os utilizadores, com o número dos passos, tendo estes um objetivo de passos diários. Com este objetivo diário, os utilizadores são incentivados a caminhar.

Na aplicação TODAY [10] o objetivo é de ter todos os dias atividades diferentes para promover a movimentação, algo tão simples como ir à secretária próxima do utilizador, ao invés de enviar *email*. Este fornecia os objetivos de atividades, juntamente com o seu contexto.

Estas duas aplicações são semelhantes ao projeto OnTRACK pois também são dados objetivos, diários ou não, aos utilizadores, promovendo assim a sua locomoção e/ou outra atividade física.

No artigo “*Wearable Computing: Accelerometers*”, [11], os autores forneceram três ambientes sensíveis ao contexto que denominaram de Amoeba. Esta é uma aplicação baseada em *Android* e os dados recolhidos são de um *smartphone*. Este recolhe os dados de GPS, acelerómetro e Wi-Fi. Para os sensores são utilizados diferentes totais de valores, ou seja, para o acelerómetro utiliza-se a média, desvio padrão e frequência máxima; para o GPS utiliza-se a velocidade e para o Wi-Fi as semelhanças entre duas impressões digitais

consecutivas. Este sistema só deteta um modo de cada vez, e por isso confunde a atividade de caminhar com estar parado e de correr com andar. A precisão deste sistema para casos de vida real está entre 80% e 98%.

No documento “*Activity recognition using cell phone accelerometers*” [12] desenvolveram uma aplicação que utiliza o GPS e acelerómetro do *smartphone*, em os dados são recolhidos, de 29 utilizadores, através de uma aplicação móvel. A melhor exatidão que teve foi de 91,7%, recorrendo a redes neuronais multicamada. Estas são validadas através de validação cruzada, feita 10 vezes (*ten-fold cross validation*). No entanto, esta aplicação é gerada *offline*, ou seja, não é em tempo real, logo o utilizador não tem *feedback*, além disso o resultado muitas vezes confunde as ações de subir e de descer escadas.

Os autores Reyes-ortiz *et al* [13] encontra-se representado um sistema com arquitetura *transition-aware* para reconhecimento de atividades, também utilizando o *smartphone*, que permite identificar a transição postural e as atividades básicas, com uma duração mais extensa, isto em tempo real. São utilizados três *datasets* diferentes, o SBHAR, *dataset* feito pelos autores; e o PAMAP2 e REALDISP, que são *datasets* públicos. Os autores decidiram usar 17 medidas: média aritmética, desvio padrão, desvio absoluto mediano, máximo, mínimo, assimetria, curtose, o componente com máxima frequência, energia, área de sinal de magnitude, entropia de sinal, alcance interquartil, auto regressão, coeficiente da correlação de Pearson, frequência mínima, banda de energia e ângulo.

Foram feitas duas implementações com *Activity Learning* (AL), uma destas implementações tem em conta só as atividades básicas e considera a transição como atividade desconhecida. Por outro lado, temos a *Activity and Transition Learning* (ATL) que consiste numa implementação na qual se considera a transição durante o treino do modelo. As probabilidades baseam-se em *Support Vector Machines* (SVM) e usou-se o filtro heurístico para a classificação que identifica as classes de *input* com as probabilidades de cada classe. Utilizou-se o erro de sistema para demonstrar a eficácia da implementação. Os resultados da *Probability Support Vector Machine* (PrSVM) no sistema SBHAR é, no geral, de 7,4% +/- 4,8 (desvio padrão) com implementação AL e 4,2% +/- 4,2 com ATL. No PAMAP2 só o AL foi testado, pois o *dataset* não inclui dados de transição e o resultado de PrSVM foi de 6,96% +/- 2,5. Por fim, o REALDISP teve um resultado no PrSVM de 0,63% +/- 0,9 na implementação AL, pois, como o anterior, este *dataset* não continha transições posturais, logo não seria possível a implementação ATL.

No artigo de Yonggang Lu e seus colaboradores [14] reconheceu-se o problema de recolher dados classificados e decidiu-se utilizar uma solução com métodos não supervisionados para o reconhecimento das atividades, solução esta que apenas utiliza o acelerómetro. Usou-se o método *Molecular Complex Detection* com 5 métodos de *clustering* diferentes: *Gaussian Mixture Model with Expectation-Maximization* (GMM) [15], método hierárquico de *clustering* [16], dois métodos baseados em *centroids* (K-means++ [17] e K-medoids [18]) e um método baseado em gráficos (*spectral clustering*) [19]. Este consegue atingir uma precisão de 74% nas atividades do quotidiano.

Os autores de “*A Tutorial on Spectral Clustering*” [19] utilizaram um sistema com *convolutional neural networks* (CNN) e usaram o acelerômetro e o giroscópio do *smartphone*. Neste sistema os dados são envolvidos em 3 canais parciais de CNN, sendo estes o *Fine* (mais complexa), o *Medium* e o *Coarse* (menos complexa). Receberam-se os dados de 30 utilizadores, com o *smartphone* na vertical, tendo sido realizadas 6 atividades: andar, subir e descer as escadas, sentar, estar de pé e estar deitado. Usou-se a função de ativação ReLU para esta estratégia, alcançando uma taxa de sucesso de 100% tanto no *dataset* público do grupo WISDM quanto no *dataset* UCI HAR, obtiveram-se os resultados de 100%.

No documento de Carlos Avilés-Cruz e de seus coautores [20] explorou-se a fusão de métodos com *features* manuais e *features* automáticas, utilizando *Long Short-Term Memory* (LSTM) para melhorar a performance de *Activity Recognition and Classification* (ARC). Como é frequente os humanos mudarem a sua atividade, utilizou-se o conceito de máximo à posteriori (MFAP), é um algoritmo que tem em conta informação antecedente e a informação obtida pela *framework* criada pelos autores. Os resultados deste foram de, no geral, 0,9644 para a *framework* proposta e 0,9885 para o MFAP, tendo em conta o *dataset* ICU, no *dataset* dos autores, os resultados foram de 0,9867 para a *framework* e 0,9958 para o MFAP.

No relatório de Chen e partícipes [21] utilizaram-se as atividades: estar parado, andar, correr, andar de bicicleta e conduzir. Usou-se o serviço Amoeba o qual consiste em três detetores de contexto: atividade, condição *indoor* ou *outdoor*, e *Geofence*. Recorreu-se aos sensores acelerômetro, WiFi e GPS. Usando o serviço referenciado anteriormente os seus resultados são iguais de 80% a 98%, para o detetor *indoor/outdoor* a confiabilidade é de 83,3% quando a latência é de 10 segundos, esta passa a 91,9% quando a latência é de 60 segundos. No *geofence* os resultados são similares aos de um GPS normal, no entanto, este consome menos energia.

O proposto por Hassan e seus colaboradores [22] é um sistema robusto baseado nos sensores do *smartphone*, que reconheça as atividades humanas. Para os testes e treino, usou-se uma base de dados pública, encontrada em [23], que consiste em 12 atividades: estar de pé, sentar, estar deitado, andar, andar para cima e andar para baixo, transição de estar sentado e se levantar e vice-versa, transição de estar sentado para se deitar e vice-versa, e transição de estar de pé para deitado e vice-versa. Os resultados usando *Artificial Neural Networks* (ANN), mostram que a média de precisão foi de 89,06%, usando SVM, a precisão foi de 94,12%. Recorrendo a *Deep Belief Network*, a precisão foi de 95,85%, sendo essa a configuração eleita pelos autores.

### 2.1.1. Human physical activity recognition using smartphone sensors

O objetivo deste artigo é o reconhecimento de atividade física a partir de sensores do dispositivo móvel. O âmbito é o uso de IA no ambiente e garantir o “*aging in place*” que se refere à capacidade de uma pessoa viver sozinha na sua casa e comunidade, de forma independente e confortável. Segundo um inquérito realizado em 2010, nos Estados Unidos

([24]), a grande maioria das pessoas com mais de 65 anos (88%) quer manter-se na sua residência o máximo de tempo possível.

Este artigo aborda o reconhecimento de atividades físicas através de sensores, nomeadamente, acelerómetro, giroscópio e sensor de gravidade, do *smartphone*, devido a este ser ubíquo e usar técnicas de ML.

Separaram-se os dados em dois grupos, teste e treino. Guardou-se estas atividades usando reconhecimento de atividade, podendo estas ser andar, que é o *default*, correr, sentar, estar de pé, subir ou descer escadas. Têm um intervalo de 50 milissegundos de *default* e grava por 10 segundos, também *default*.

Para testar utilizaram-se 5 indivíduos. O objetivo era recolher os dados enquanto se faziam as atividades referidas; durante estas atividades apenas o acelerómetro, giroscópio e sensor de gravidade foram utilizados e como ML não funciona bem com grandes quantidades de dados seguidos, foram criados segmentos de 10 segundos para esta leitura. Por segundo, em média, o acelerómetro leu 164, o giroscópio e o sensor de gravidade 124.

Os resultados da atividade andar geraram um certo padrão, representado na Figura 1. No correr consegue-se perceber que é semelhante ao andar, mas executado mais rapidamente, Figura 2.

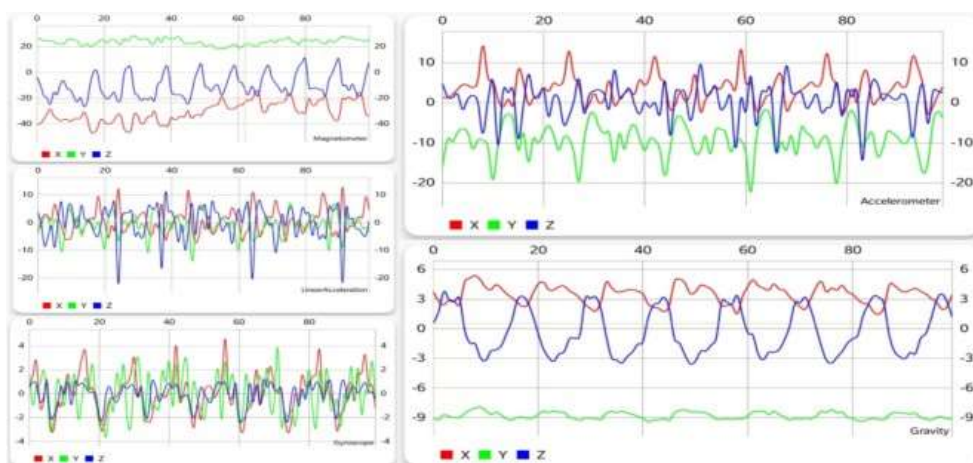


Figura 1: Padrão da atividade "andar", figura 2 do artigo [24]

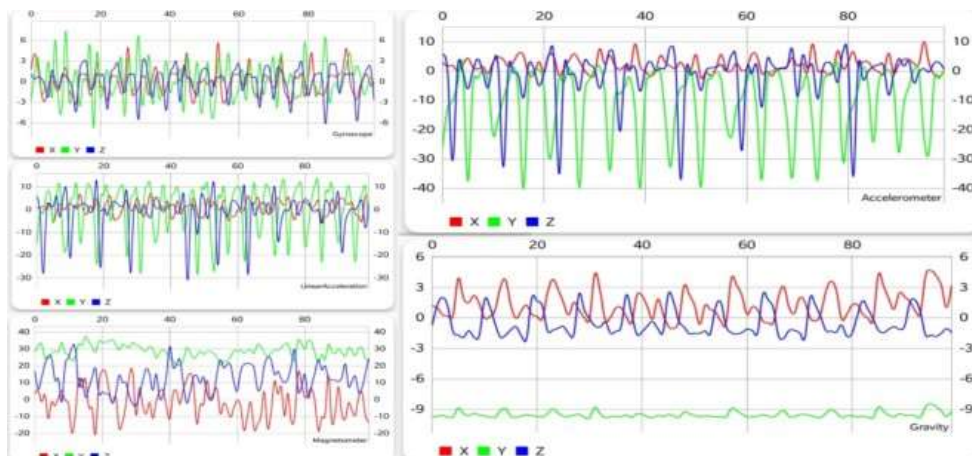


Figura 2: Padrão da atividade "correr", figura 3 do artigo [24]

Na atividade sentar os resultados obtidos são completamente diferentes dos anteriores; estes são quase estáticos, à exceção da aceleração linear e giroscópio, como representado na Figura 3.

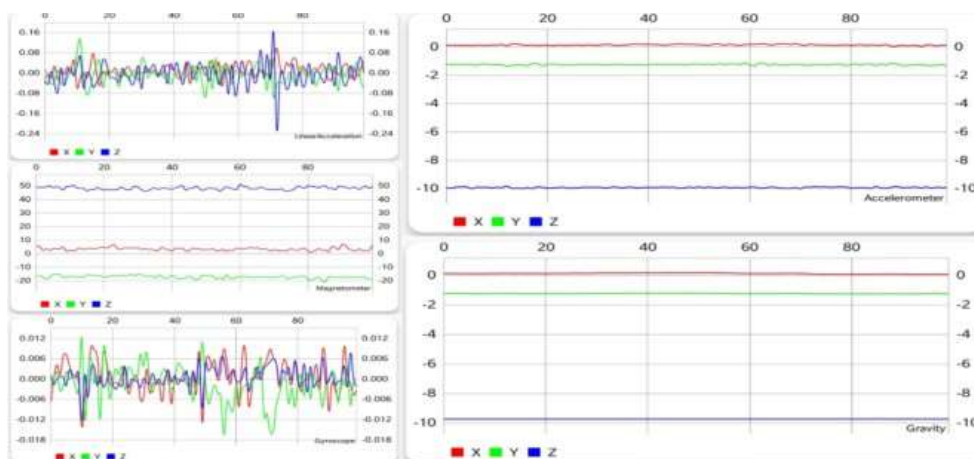


Figura 3: Padrão da atividade "sentar", figura 4 do artigo [24]

Sendo estar de pé uma atividade que também não envolve movimento, tem um comportamento parecido ao sentar; no entanto vemos que os eixos Y e Z são diferentes entre a atividade sentar (Figura 3) e a atividade de pé (Figura 4), uma vez que estas são bem reconhecíveis uma da outra.

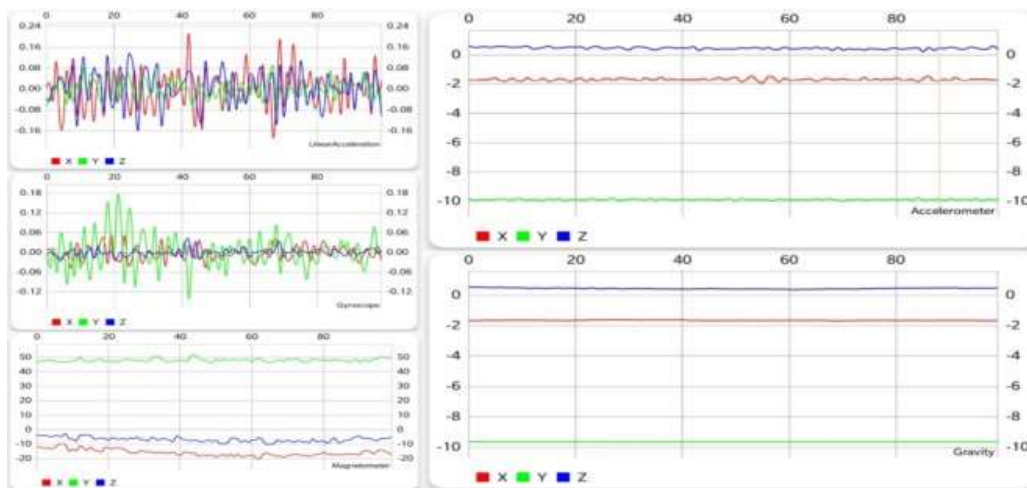


Figura 4: Padrão da atividade "de pé", figura 5 do artigo [24]

Na atividade de subir as escadas, Figura 5, consegue-se ver algumas semelhanças, em concreto no acelerómetro e giroscópio.

Na atividade de descer as escadas, e tal como no subir as escadas, este padrão é difícil de definir. Como se pode observar os padrões de subir escadas, Figura 5, e descer escadas, Figura 6, são parecidos e daí ser difícil para o algoritmo distinguir.

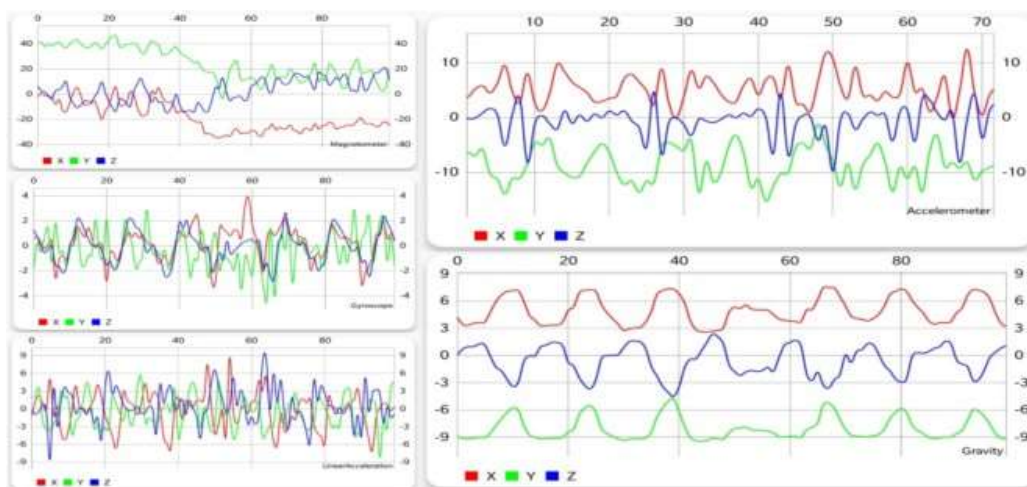


Figura 5: Padrão da atividade "subir escadas", figura 6 do artigo [24]

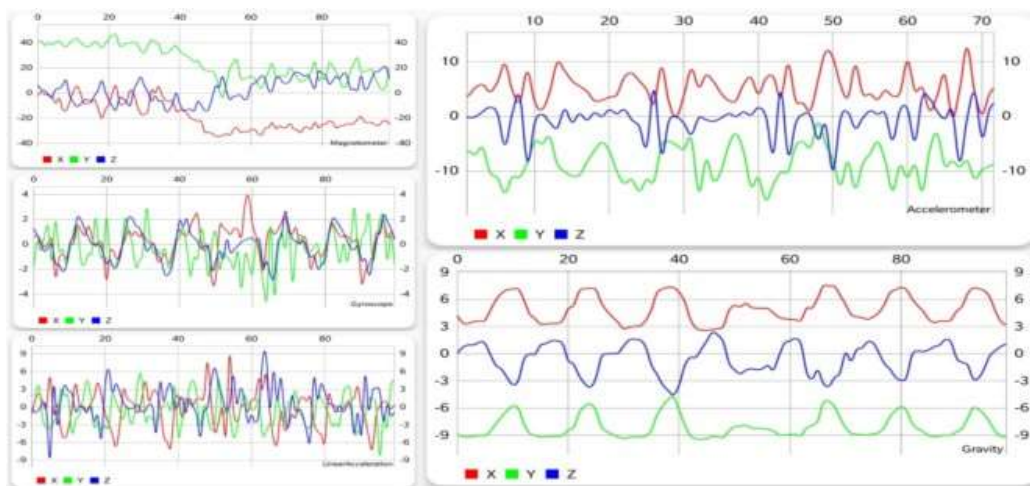


Figura 6: Padrão da atividade "descer escadas", figura 7 do artigo [24]

No final os resultados obtidos são de 92%, não envolvendo escadas, e de 73%, envolvendo. Conclui-se que seria possível usar os sensores para reconhecimento de atividades.

### 2.1.2. Gymskill: a personal trainer for physical exercises

*GymSkill* [25], que basicamente é um *personal trainer* (PT), baseado num *smartphone*, este monitoriza e avalia a atividade física usando equipamento comum de *fitness*.

Torna-se difícil para as pessoas manterem um regime de atividade física, porque é necessário bastante motivação e tempo. No entanto, um PT tem bastante impacto na aderência e na qualidade da realização dos exercícios, consequências deste treino ser personalizado.

Este PT no *smartphone* foi criado não só por meios financeiros, mas, também, por comodidade. Utiliza principalmente os sensores acelerómetro e giroscópio, dando *feedback* visual e auditivo.

Compondo-se, este sistema, por um *smartphone* e um servidor, servidor este que torna disponível a componente de análise. Estando o *smartphone* adjunto ao equipamento de *fitness*, este grava os dados de aceleração e orientação, recebendo *feedback*, baseado numa análise inicial do número de repetições que faltam e se o exercício está dentro dos parâmetros aceitáveis, recebendo *feedback* em gráficos e áudio.

Este artigo teve como caso de estudo 6 participantes, e realizou-se numa estação de equilíbrio, estando o *smartphone* pousado no centro desta mesma. Os participantes treinaram durante 5 dias e fizeram 20 exercícios, duas vezes por dia. Para conseguirem resultados qualitativos, eram feitos questionários aos participantes, no primeiro *set* de exercícios, e um questionário mais completo depois do último exercício.

Fizeram-se vídeos para um físico rever em termos de regularidade do movimento, ângulos, *tempo* e postura corporal, analisando e comparando com os resultados da aplicação, e assim verificar se os resultados eram plausíveis, pois os sensores do *smartphone* não conseguem garantir que os participantes não estão apoiados (e.g., mãos na parede).

Comparando os resultados dos primeiros dias com os últimos, notou-se um melhoramento moderado na qualidade do exercício. Os resultados qualitativos, dado pelos participantes, foram divididos pelas 5 *features* e o global, *feedback* individual com um *rating* de 5.0, sugestões de exercícios *rating* de 4.8, o diário de treinos 4.0, avaliação e repetição de exercícios, ambos com *rating* de 3.8, por último o global teve o *rating* de 4.2.

Conclui-se então que esta aplicação teve resultados encorajadores para uma continuação e até mesmo integração com outros dispositivos, como *wearables*, para melhorar o critério, podendo ser usado em fisioterapia, reabilitação, entre outros.

### 2.1.3. Discussão

Como podemos ver ao longo desta secção existem soluções que têm uma precisão algo alta, no entanto, grande parte destas soluções utilizam ML para aumentar a referida precisão. Ainda assim, e de notar no artigo referido em 2.1.1, algo simples como subir e descer escadas pode reduzir drasticamente a precisão (de 92% para 73%, no caso concreto).

Conclui-se que, além de apresentarem boa precisão, quando utilizados em conjunto com técnicas de ML, os sensores são inadequados, devido às suas limitações. Quanto mais complexa é a tarefa, menor é a precisão alcançada pelos sensores.

## 2.2. Soluções envolvendo reconhecimento de poses

Nesta secção retratar-se-ão artigos, estudos ou revisões incidindo sobre o reconhecimento de poses [26], com enfoque nas aplicações de promoção de atividade física.

Decidiu-se fazer um estado da arte de *pose estimation*, que é uma técnica de visão computacional que prevê a posição espacial e orientação de objetos dentro de imagens ou vídeos. Esta estima diretamente as coordenadas ou ângulos dos *keypoints* ou articulações, eliminando a necessidade de deteção explícita de partes individuais, uma vez que esta técnica melhora as limitações dos sensores quando usados isoladamente, fornecendo um modelo capaz de identificar com precisão poses, independentemente de seu nível de complexidade, sem depender de outras ferramentas. De referir que este modelo necessita de um sensor visual (câmara). *Pose estimation* pode-se utilizar com 2D e 3D, ambos com apenas uma pessoa, ou várias. Daí haver vários estudos, com âmbitos diferentes.

O autor Toshev e seus coautores [27] apresentaram então uma *pose regression*, aplicação de *pose estimation* como uma regressão, em que para uma imagem  $x$  regride para um vetor normalizado de pose, ou seja, aplica-se um *refining regressor* numa subimagem que refina a previsão com base no estado anterior, baseada em *Deep Neural Network* (DNN) para identificar as articulações corporais; usam-se os *datasets Frames Labeled In Cinema* (FLIC) [28], *Leeds Sports Dataset* (LSP) [29] e sua extensão [30], os últimos referidos são usados em conjunto.

Utilizou-se, para testes, com o modelo já treinado, o *dataset Image Parse* [31]. As métricas usadas por este artigo são a percentagem de partes corretas (PCP) e percentagem de

articulações detetadas (PDJ). Com isto conseguimos observar que a PDJ vai aumentando conforme o aumento da distância normalizada das articulações, em ambos os *datasets*, e que a PCP, usando o LSP juntamente com a sua extensão, tem uma média de 61%; usando o *dataset Image Parse* uma média de 69%.

No documento “*Human pose estimation with iterative error feedback*” [32] utilizaram-se redes convolucionais, criando o método *Iterative Error Feedback* (IEF). Os *datasets* aplicados foram o *Max Planck Institute for Informatics Human Pose* (MPII) [33] e o LSP.

Usou-se o método de validação o *percentage of correct keypoints* (PCK)<sub>h@0,5</sub> (significa PCK normalizada ao comprimento da cabeça, com um offset de 0,5) e o resultado foi de 81,3% para o corpo todo, com o MPII no *set* de teste, e de 72,5% para o *Person Centric PCP*, no *dataset LSP*, usando o IEF.

No artigo de Sun e seus colaboradores [34], propõe-se uma abordagem *structure-aware* usando *compositional pose regression*, em que se usa ossos em vez de articulações. Artigo analisa posições 2D e, também, 3D, servindo-se dos *datasets* MPII para 2D e Human3.6M [35] para 3D.

Para o reconhecimento 3D, a métrica é a média de *PA Joint Error* e de *Joint error* em que este teve os resultados de 48,3% e 59,1%, respetivamente, sendo o *Baseline*, ou dados de controlo, 51,4% e 64,2%. Para o 2D usou-se o PCK<sub>h@0,5</sub> e o resultado foi de 86,4%.

O proposto por Picard e partícipes [36] foi, também, uma abordagem baseada em *pose regression* e utiliza-se a operação *soft-argmax*. Utilizaram-se os *datasets* LSP e o MPII e as métricas são o PCP e o PCK<sub>h@0,5</sub>, com os resultados de 89,4% e 91,2%, respetivamente.

Os autores de “*Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation*” [37] apresentam uma combinação entre redes convolucionais *Part Detector* e um modelo espacial. Usaram-se os *datasets* FLIC e a versão estendida do LSP. A métrica de avaliação é a *detection rate / Normalized distance error* (pixels) em que foram feitas 2 para FLIC, com o cotovelo (90%) e pulso (87%), em que os resultados foram aproximadamente os indicados anteriormente. Para o LSP, tornozelo e joelho, com um resultado de aproximadamente 65% para o tornozelo e 71% para o joelho.

No artigo de Ouyang e seus coautores [38] propôs-se uma *framework* que combina *Deep Convolutional Neural Networks* (DC-NN) com mistura deformável expressiva de partes. Utilizaram-se os *datasets* LSP, FLIC e o *Image Parse*. As métricas são a PCP e a PDJ e as arquiteturas basearam-se em *Very Deep Convolutional Networks for Large-Scale* (VGG-LG) ([39]). Os resultados da média para o *dataset* LSP tendo em conta a PCP e a arquitetura VGG-LG é de 81,1%; com o FLIC e com a arquitetura VGG-T (VGG com modelo estruturado em árvore) é de 93,8%. Com *Image Parse* e com a arquitetura VGG-LG é de 81,0%. Os resultados PDJ com o *dataset* LSP para cotovelos foram de 80,3%, para pulsos 73,5%, joelhos 82,8% e tornozelos 68,8%; com o FLIC, apenas se fez para cotovelos (96,6%) e para pulsos (94,5%).

No trabalho “*Stacked Hourglass Networks for Human Pose Estimation*” [40] introduziu-se uma arquitetura de rede convolucional denominada “*stacked hourglass*”. Usaram-se os *datasets* FLIC e MPII e a métrica foi PCK@0,2 para FLIC e PCKh@0,5 para o MPII. Os resultados foram de 99,0% para o cotovelo e 97,0% para os pulsos, usando o FLIC, com o MPII o resultado total foi de 90,9%.

No documento de Ramakrishna e seus colaboradores [41] apresentou-se uma arquitetura sequencial composta por redes convolucionais que operam diretamente com mapas *belief* de fases anteriores. Usaram-se os *datasets* FLIC, LSP e MPII. A métrica é PCKh@0,5 para LSP e MPII, e PCK@0,2 para o FLIC. Os resultados, MPII é de 88,52%, LSP é de 90,5% e FLIC é de 97,59% nos cotovelos e 95,03% nos pulsos.

No artigo “*Multi-context attention for human pose estimation*” [42] usou-se um *Conditional Random Field* (CRF) com um mapa de atenção para fazer as correlações entre regiões vizinhas. Os *datasets* são o MPII e o LSP, as métricas são a PCKh@0,5 para o MPII e PCK@0,2 para os LSP. Os resultados da média destes são 91,5% para o MPII e 92,6% para o LSP.

O proposto por Yang e partícipes [43] é uma *Pyramid Residual Module* usando *deep CNN*. Usaram-se os *datasets* MPII e LSP, as métricas são PCKh@0,5 e PCK@0,2, respetivamente. O resultado para o MPII é de 92,0% e o LSP é de 93,9%.

Os autores de “*Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation*” [44] propuseram o uso de uma *Generative Adversial Network* (GAN) em que se usam dois discriminadores para distinguir se a pose é real e, se não for, a rede aprende com os anteriores. Os *datasets* são o MPII com a métrica de PCKh@0,5 e o LSP com a métrica de PCK@0,2. Os resultados são de 91,9% e 93,1%, respetivamente.

No trabalho de Metaxas e seus coautores [45] propôs-se uma rede de aumento usada contra uma rede alvo e assim o alvo aprende com as aumentações para alcançar melhor *performance*. Os *datasets* são o MPII e LSP. Métricas são a PCKh@0,5(MPII) e PCK@0,2(LSP). Os resultados deste são para o MPII 91,5% e para o LSP de 94,5%.

No *paper* de Lyu e contribuintes [46] propôs-se uma melhoria aos modelos *conv-deconv hourglass* através da *multi-scale supervision*, *multi-scale regression network*, *structure-aware loss* e *keypoint masking scheme*. Utilizaram-se os *datasets* MPII e o FLIC. Usaram-se as métricas PCK@0,2 para o FLIC e PCKh@0,5 para o MPII. O resultado foi de 99,2% para cotovelo e 97,3% para pulsos com o *dataset* FLIC e de 92,1% para o MPII.

No documento “*Deeply Learned Compositional Models for Human Pose Estimation*” [47] propôs-se uma *framework* que usa o modelo *Deeply Learned Compositional*, usa DNN. Utilizaram-se os *datasets* FLIC, LSP e MPII. Métricas para FLIC e LSP PCK@0,2 e para MPII PCKh@0,5. Os resultados foram de 99% para o FLIC, 95,1% para o LSP e 92,3% para o MPII.

O proposto por Liu e colaboradores [48] foi usarem-se várias fusões *multi-scale* com várias representações *high-to-low resolutions*, recebendo estas representações dados de outras

paralelas para alcançar representações ricas. O *dataset* utilizado foi o MPII com resultado de 92,3% PCKh@0,5.

Os autores de “*Does learning specific features for related parts help human pose estimation?*” [49] propuseram uma *data-driven approach* para agrupar partes relacionadas, conforme quanta informação partilham e usa um *part-based branching network* (PBN) para representar partes específicas de cada grupo. Explorando-se os *datasets* MPII e o LSP. Obtiveram-se os resultados de 92,7% PCKh@0,5 para o MPII e de 94,5% PCK@0,2 para o LSP.

Os artigos referidos anteriormente são de *pose estimation* 2D para apenas uma pessoa, exceto o “*Compositional Human Pose Regression*” [34]. Os próximos são também de 2D, mas para várias pessoas.

Começando com o artigo de Iqbal e Gall [50], em que se propôs usar *pose estimation* por pessoa e depois usar-se *integer linear programming* para retirar os *outliers* dos pontos encontrados. Aplicou-se uma parte do *dataset* MPII com imagens de múltiplas pessoas. A métrica, a *average precision* (AP) e o resultado foi de 43,1%.

No documento “*RMPE: Regional Multi-person Pose Estimation*” [51] propôs-se uma *framework* que consiste em 3 componentes, *Symmetric Spatial Transformer*, *Parametric Pose Non-Maximum-Suppression* (NMS), and *Pose-Guided Proposals Generator* (PGPG) que, em conjunto, conseguem retirar as caixas envoltas das pessoas e as detecções redundantes. Utilizaram-se os *datasets* MPII e o *Common Objects in Context* (COCO) [52], mais exatamente o *MSCOCO Keypoints Challenge* 2016. A métrica foi a AP e obtiveram-se os resultados de 76,7% para o MPII e para o MSCOCO usando o test-dev 63,3%.

O proposto por George Papandreou e seus coautores [53] foi uma *top-down approach*, em que usa, primeiramente, detetores RCNN para prever a localização e escala das caixas e na segunda parte estima os *keypoints* da pessoa em cada caixa. Para cada *keypoint* preveem *heatmaps* usando redes convolucionais. Usou-se o *dataset* COCO, a métrica executada foi a AP e o resultado foi de 64,9%.

Os autores de “*A Coarse-Fine Network for Keypoint Localization*” [54] propuseram uma *keypoint localization network* composta por vários ramos e construída sobre uma *feature layer* com CNN, e um ramo *fine detector* sobre múltiplas *feature layers*. Empregaram-se os *datasets* 2016 COCO Keypoint Challenge e teve o resultado de 72,2% AP, o LSP que teve como resultado 92,6% PCK@0,2.

No trabalho de Gkioxari e partícipes [55] propôs-se um método denominado *Mask R-CNN* que se baseia no *Faster R-CNN*, mas adiciona um ramo para prever as máscaras de segmentação.

Este modelo corre a 5 FPS. Usou-se o COCO *test-dev dataset* e o resultado foi de 63,1% AP.

No *paper* de Xiao e seus colaboradores [56], propuseram-se métodos de patamares simples e eficazes. Baseando a *pose estimation* em camadas deconvolucionais numa rede ResNet

[57]. Usou-se o *dataset* COCO *test-dev* 2017, obteve-se o resultado de 73,7%, utilizando-se a métrica AP.

No artigo “*Cascaded Pyramid Network for Multi-person Pose Estimation*” [58], propôs-se uma estrutura denominada *Cascaded Pyramid Network*, organizada em dois passos: o *GlobalNet*, que consiste numa funcionalidade para localizar *keypoints* simples, e o *RedefineNet*, responsável pelo tratamento dos *keypoints* mais desafiadores. Utilizou-se o conjunto de dados MS COCO para realizar as avaliações. Empregou-se a métrica AP, obtendo-se resultados de 73%.

No documento “*PoseFix: Model-agnostic General Human Pose Refinement Network*” [59] propôs-se uma rede de refinamento da pose humana, que estima a partir de *input images* e *input poses*. Usou-se o *dataset* MS COCO 2017. Os resultados apresentados pelos autores referem-se à eficácia da sua solução juntamente com outros métodos. O desempenho mais elevado, avaliado através da métrica de AP, foi alcançado atingindo-se um resultado de 73,6%, juntamente com *Simple* (ResNet-152).

O proposto por Pishchulin e coautores [60], foi uma formulação de particionamento e rotulagem de um conjunto de hipóteses de partes do corpo, geradas com detetores de partes do corpo. Utilizaram-se os *datasets* “*We are family*” (WAF) [61] e um *subset* de MPII para *multi-person*. Utilizou-se a métrica PCP para o WAF e teve resultados com média de 82,2%, a métrica do MPII foi a *mean* AP tendo um resultado de 51,4%.

O autor Insafutdinov e seus colaboradores [62], propuseram melhorar os detetores das partes do corpo, uma nova *image-conditioned pairwise terms* e uma estratégia de otimização na procura do espaço em redor. Foi usado o MPII *subset multi-person* e teve o resultado de 58,7% de AP.

No trabalho “*Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*” [63] propôs-se uma representação não paramétrica, denominada *Part Affinity Fields* (PAF) que associa as partes do corpo com individuais na imagem, estes individuais podem ser *blobs* de certas cores e orientações, que capturam a resposta de filtros locais. Usaram-se os *datasets* MPII o *subset* e o COCO 2016 *challenge*. Usou-se a métrica AP e o resultado para o MPII foi 79,7%, no *testing subset*, e para o COCO 2016 foi de 61,8%, no *subset test-dev*.

No *paper* de Newell e Deng [64] propôs-se uma rede que, simultaneamente, é ensinada *por output detections* e *por group assignments*. Os *datasets* foram os MS COCO e o MPII *subset multi-person*. Em que o resultado para a MPII foi de 77,5% de AP e para o MS COCO *test dev* foi de 65,5% de AP.

No artigo “*Pose Partition Networks for Multi-person Pose Estimation*” [65] propôs-se uma *Pose Partition Network* em que favorece a baixa complexidade e uma alta precisão na deteção de articulações. Esta faz regressões das articulações globais num espaço determinado. Utilizaram-se os *datasets* MPII com um total de 80,4% de AP, o PASCAL-*Person-Part* [66] com um total de 46,6% de AP e o WAF com um total de 84,8%.

No documento de Gidaris e partícipes [67], propôs-se um modelo que usa uma rede convolucional. Rede esta que aprende a detetar *keypoints* e prever as suas mudanças relativas. Usou-se o COCO *dataset* com a métrica de AP e resultado de 68,7%.

O proposto por “*MultiPoseNet: Fast Multi-Person Pose Estimation Using Pose Residual Network*” [68] foi uma *MultiPoseNet* que aborda os problemas da deteção de pessoas, segmentação e da *pose estimation*. Usa uma *Pose Residual Network* que produz poses com alta exatidão. COCO (*test dev*) foi o *dataset* que se utilizou e o resultado foi de 69,6% de AP.

Os autores de “*PifPaf: Composite fields for human pose estimation*” [69] propuseram um método novo denominado *PifPaf*, *Part Intensity Field* (PIF) e *Part Association Field* (PAF) que associa diferentes partes do corpo, criando poses completas. O conjunto de dados que se utilizou para treino e teste foi o COCO (*test-dev*). Obteve-se o resultado de 66,7% AP.

Tem também referência a *pose-estimation* em 3D, estes são todos testados no *dataset* Human3.6M. Começando com o artigo “*3D human pose estimation from monocular images with deep convolutional neural network*” [70] em que se usa uma rede de *multi task* para prever qual a parte do corpo com junção de *sliding windows* e *pose estimation* 3D, sendo o resultado 132,2 *Mean Per Joint Position Error* (MPJE) em milímetros (mm).

O documento de Zhan e colaboradores [71], retrata a computação do *score* de correspondência entre imagem e pose. Obteve-se os resultados de 145,74 mm de MPJE usando o *StructNet-Max*, que é uma tarefa de previsão de poses.

O proposto por Salzmann e seus coautores [72] foi um *auto-encoder* que aprende a fazer uma representação *high-dimensional* de uma pose 3D e usa redes CNN para essa aprendizagem. Obtiveram-se os resultados de 129,06 mm de *Average Euclidean distance between ground-truth and predicted joint positions* (ASM).

Os autores de “*Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation*” [73] propuseram usar *heatmaps* 2D para as articulações primeiro, seguidamente usar uma arquitetura de *trainable fusion* para combinar os *heatmaps* com as *features* e, por fim, usar um módulo 2D pré treinado com MPII. Obtiveram-se os resultados de 69,73 mm de MPJE.

No trabalho de Ching-Hang Chen e Deva Ramanan [74] propôs-se, primeiramente, estimar poses 2D a partir das imagens, depois estimar a profundidade comparando com bibliotecas de poses 3D e por fim usar um módulo 2D pré treinado com MPII. Obteve-se o resultado de 82,37 mm de MPJE.

No *paper* de Francesc Moreno-Noguer [75], usaram-se as *Euclidean Distance Matrices* (EDM) para fazer *encoding* das distâncias entre as articulações corporais 2D e 3D, de seguida treinar a rede para aprender 2D-para-3D usou-se regressão EDM e, por fim, treinar essa rede juntamente com outro *dataset* 3D. Os resultados são de 87,3 mm de MPJE, com o protocolo 1 (11 sujeitos; 1,5,6,7 e 8 usados para treinar e 9,11 para teste).

No artigo “*Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose*” [76] propôs-se uma representação volumétrica 3D para poses humanas, um esquema de previsão *coarse-to-fine* e um módulo 2D pré treinado com MPII. Obteve-se o resultado de 71,9 mm de MPJE.

No documento “*Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach*” [77] propôs-se uma perda induzida a partir de uma restrição geométrica para dados 2D, restrições por tamanho de osso e juntamente treinar com o MPII *dataset*. O resultado que se obteve foi de 64,9 mm de ASM.

A proposta “*A simple yet effective baseline for 3d human pose estimation*” [78] consistiu em mapear poses 2D diretamente para 3D utilizando duas camadas lineares. O módulo 2D foi pré-treinado com MPII e foi processado em tempo real. O resultado obtido foi de 62,9 mm no MPJE, conforme o protocolo 1.

Os autores de “*3D Human Pose Estimation in the Wild by Adversarial Learning*” [79] propuseram *adversarial learning* para adaptar os domínios dos *datasets* de 2D/3D, um discriminador *multi-source* com imagens, uma estrutura geométrica de pares com localização das articulações e um módulo pré treinado com MPII. Obtiveram-se os resultados de 58,6 mm de MPJE.

No trabalho de Daniilidis e partícipes [80], propôs-se uma representação volumétrica para poses humanas 3D, que usa anotações, adicionais, de profundidades ordinais para as articulações e treina juntamente 2D e 3D *datasets*. Obteve-se o resultado de 56,2 mm de MPJE.

No *paper* “*Integral human pose regression*” [81] os autores propuseram uma representação volumétrica para poses humanas 3D, que usa operações integrais para unificar a representação do *heatmap* e a regressão articular e treinar com um *dataset* 2D. Os resultados, com o protocolo 1, foram de 40,6 mm de MPJE.

No artigo de Chen Li e Gim Hee Lee [82], propuseram-se diversas hipóteses de poses 3D, derivadas a partir de poses 2D, com a escolha da melhor entre as reproduções 2D, sendo o módulo 2D previamente treinado com o MPII. Obteve-se o resultado de 52,7 mm no MPJE.

No documento de Kanazawa e seus colaboradores [83] propôs-se um modelo estatístico de formas corporais, ajustar este mesmo a um com articulações 2D para minimizar a distância destas com as articulações em 3D. O resultado obtido foi de 82,3 mm de MPJE.

O que se propôs por Zhou e coautores [84], foi um modelo cinético, que integra um objeto modelo numa rede para gerar objetos articulados de *pose estimation*, com restrições de orientação e de rotação. O seu resultado foi de 107,26 mm de MPJE.

No trabalho “*Monocular 3D Human Pose Estimation Using Transfer Learning and Improved CNN Supervision*” [85] apresentou-se um modelo cinético que realiza transferência da aprendizagem das *features* adquiridas para o espaço 2D. Ele realiza previsões de poses 2D como uma tarefa auxiliar, antecipa as articulações relativas às

localizações, seguindo o modelo da árvore cinética, e é treinado em conjunto com MPII e LSP. Alcançou-se o resultado de 74,11 mm no MPJE.

No *paper* de Nie e seus colaboradores [86] propôs-se, também, um modelo cinético, em que junta estimação de profundidade, pertencentes às poses 2D, com um esqueleto LSTM e partes do corpo com *patch*-LSTM. O módulo 2D é pré treinado com MPII e o resultado que se obteve foi de 79,5 mm de MPJE.

No artigo “*End-to-end Recovery of Human Shape and Pose*” [87] propôs-se um modelo, denominado *Skinned Multi-person linear model* (SMPL), modelo este com *adversarial learning*, para adaptação do domínio de imagens 2D e modelos 3D. Propôs-se, também, uma *framework* para aprender os parâmetros de SMPL, e treinar juntamente com *datasets* 2D, esta é processada em tempo real. O resultado foi de 87,97 mm de MPJE.

No documento “*Ordinal Depth Supervision for 3D Human Pose Estimation*” [88] apresentou-se um modelo SMPL em que, inicialmente, realiza a previsão dos *heatmaps* 2D das articulações e da silhueta humana. Seguidamente gera parâmetros para o modelo SMPL, sendo que o módulo 2D foi treinado com MPII e LSP. O resultado obtido foi de 75,9 mm de MPJE.

O proposto por Omran e partícipes [89] foi um modelo SMPL que, primeiramente, previa partes segmentadas 2D de imagens RGB e depois usava essa segmentação para prever os parâmetros do SMPL. O resultado obtido foi de 59,9 mm de MPJE.

Os autores de “*BodyNet: Volumetric Inference of 3D Human Body Shapes*” [90] propuseram um modelo SMPL, que previa poses 2D e partes do corpo, também 2D, segmentadas, depois previa as poses 3D e finalmente previa a forma volumétrica para encaixar no modelo SMPL. Os módulos 2D foram treinados com MPII e *Synthetic hUmans foR REAL tasks* (SURREAL) [91]. Obteve-se o resultado de 49,0 mm de MPJE.

No trabalho “*Exploiting temporal context for 3D human pose estimation in the wild*” [92] propôs-se um modelo SMPL em que usa *keypoints* 2D e estimação de parâmetros da câmara, faz ajustes *offline* com restrições temporais, o módulo 2D é treinado com COCO. Obteve-se o resultado de 77,8 mm de MPJE.

No *paper* de Tomè e coautores [93], propôs-se um modelo probabilístico pré-treinado de poses 3D para fazer a fusão probabilística entre as poses 3D e uma arquitetura CNN *multi stage*, usa também *landmarks* 3D com suas localizações, processado em tempo real e em que o módulo 2D é pré-treinado com MPII. Resultado que se obteve foi de 88,39 mm de MPJE.

No artigo “*Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation*” [94] propôs-se um modelo de corpo variável latente aprendido de imagens *multi-view*, também foi proposto pelos autores um *encoder-decoder* para prever as novas imagens, *view*, a partir de uma dada previamente, o *encoder* foi pré-treinado para adicionar camadas "vazias" para prever as poses 3D a partir de imagens. O resultado obtido deste foi de 131,7 mm de MPJE numa rede semi-supervisionada.

O autor Mehta e partícipes [95] propuseram *occlusion-robust pose-maps* (ORPM) para inferência de poses de corpo inteiro, combina poses 2D com campos de afinidade para inferir instâncias de pessoas. Usou-se a métrica 3D *percentage of correct keypoints* (3DPCK) no geral com o resultado de 65%, usando o *dataset* dos autores, que denominaram MuPoTS-3D. É explicado que na altura não existiam *datasets 3D multi person*.

O proposto por Weinzaepfel e colaboradores [96] foi localizar as caixas envolventes dos humanos com *Faster R-CNN*, classificar o mais próximo *anchor-pose* para cada proposta de pose e fazer a regressão até chegar à pose final. Usou-se o *dataset* MPII, os resultados são por classe e utilizou-se o protocolo 2, tendo no geral 74,2% em PCKh@0,5.

O autor Zanfir e seus coautores [97], propuseram um processo de *feed forward* na semântica das partes do corpo e na estimação de poses 3D, o processo *feed backwards* para fazer a redefinição dos parâmetros e formas do modelo corporal SMPL. Utilizou-se o *dataset* CMU *Panoptic* [98] e os resultados que se obtiveram foram de 153,4 mm de MPJE.

Em “*XNect: Real-time Multi-person 3D Human Pose Estimation with a Single RGB Camera*” [99] a proposta consistiu numa arquitetura em tempo real, fundamentada em CNN, que utilizou conexões seletivas de grande e baixo alcance. Esta arquitetura utilizou *features* de previsão para poses 2D e 3D, além de atribuições de identidade para todas as articulações de cada indivíduo. Efetuou a reconstrução completa da pose 3D mesmo em casos de oclusão de articulações, apresentando uma sintonia entre estabilidade temporal e o esqueleto cinético apropriado. Usou-se o *dataset* MuPoTS-3D e a métrica é a 3DPCK e o resultado foi de 72,1%.

Dushyant Mehta e partícipes [100] propuseram uma *timeline* em tempo real com um filtro para um modelo baseado em esquelética cinética 3D. Foi utilizado uma CNN juntamente com o esqueleto cinético. Foram usadas articulações 2D e 3D em conjunto e em tempo real, para controlo de personagens. Este sistema foi todo feito a partir de uma câmara mono-ocular RGB.

Neste sistema foi aplicada a *pose estimation* ao esqueleto cinético conseguindo-se os resultados a partir da *framework* que otimiza as articulações (2D e 3D) com filtros ao longo do tempo e *smoothing*, para obter um resultado mais confiável. Utilizaram-se os *datasets* MPII e *Human 3.6M* e obtiveram-se os resultados de 76,6 PCK e 80,5mm de MPJE, respetivamente. Algumas das limitações foram os erros de estimação quando a profundidade era diferente das imagens estimadas e as previsões erradas das articulações 2D para 3D.

O módulo foi pré treinado com MPII e LSP, sendo processado em tempo real, e consegue fornecer altura do corpo. O resultado obtido foi de 80,5 mm de MPJE.

No artigo “*Deep learning approaches for workout repetition counting and validation*” [101] propôs-se um sistema para contar e validar as repetições de exercícios baseado num *set* de *features* baseadas em esquelética obtida por uma rede de *pose estimation 2D*. A arquitetura baseia-se numa rede de poses 2D, num módulo de representação, um módulo de classificação de momentos e no módulo de contar e validar as repetições. Usou-se um

*dataset* criado pelos autores. Os resultados obtidos, tendo a confiabilidade como métrica, na fase de reconhecimento *per frame* usando *Multilayer Perceptron* (MLP) juntamente com um *encoder* foram os seguintes: 96,77% para os agachamentos, 91,87% para *burpees*, 94,82% para flexões e 98,17% para *Jumping Jacks*. A precisão dos resultados na validação e na contagem foi de 98,8% para agachamentos com uma exatidão de 67,22% em repetições inválidas. Para *burpees* foi 95,63% com uma veracidade de repetições inválidas de 52,45%. Nas flexões obteve-se 95,86% de precisão, com 71,33% de rigor quando inválidas. Por fim, os *Jumping Jacks* alcançaram 96,29% de precisão e 82,43% de confiabilidade nas repetições inválidas.

O que os autores desejaram melhorar foi a fidedignidade nas repetições inválidas, ou seja, aumentar este valor.

### 2.2.1. Discussão

Ao longo da secção anterior, foi possível observar que mesmo em *datasets* extensos, a técnica de *pose estimation* mantém uma precisão considerável. Além disso, demonstra a capacidade para ser aplicada a várias pessoas simultaneamente, mesmo quando essas pessoas assumem poses distintas. Conclui-se também que a precisão é aprimorada quando há apenas uma pessoa, em comparação com situações envolvendo múltiplas pessoas. A distinção entre esses artigos e o projeto OnTRACK reside no facto de, neste último, o objetivo ser a definição do exercício, enquanto nos artigos analisado, em geral, a ênfase é colocada na identificação das articulações. No entanto, esta identificação é crucial para o projeto, pois as articulações são utilizadas para a correta execução do exercício.

*Body pose estimation*, em retrospectiva à subsecção anterior, emerge como uma das tecnologias mais promissoras para este módulo de EEFV, visto que apresenta resultados altamente satisfatórios, tanto em 2D como em 3D, de uma maneira geral. Observa-se que o desempenho em 2D é mais satisfatório do que em 3D.

Comparativamente ao uso de sensores, podemos concluir que o *body pose estimation* consegue alcançar resultados superiores, mesmo em tarefas mais complexas.

## 2.3. Soluções envolvendo Augmented Reality

Sendo *Augmented Reality* (AR) uma tecnologia que, no contexto de exercício físico, é algo recente; não estão disponíveis muitos estudos relacionados com exercício físico, pois é uma tecnologia mais usada na medicina/tratamentos clínicos e na parte industrial.

Neste estudo, considerou-se pertinente compreender o estado da arte de AR pois um dos objetivos do projeto OnTRACK é a promoção da atividade física, e usando gamificação (abordagem que incorpora elementos de jogos em contextos não relacionados a jogos, com o objetivo de aumentar o envolvimento, motivar o comportamento desejado e melhorar a experiência do utilizador. Isso é feito através da aplicação de mecânicas de jogo, como pontos, desafios, recompensas e *rankings*, para tornar as atividades mais atrativas e

divertidas) no projeto, pode ser alcançado mais facilmente. Optou-se então pelo uso de AR para esta gamificação, sendo o objetivo de motivar os utilizadores a fazerem exercício, com qualidade.

Começando por descrever três aplicações, não académicas, que utilizam AR, grande parte como jogos, e começando com um bastante conhecida o *Pokémon GO* [102], em que os utilizadores têm de andar para concretizar os seus objetivos, tendo pelo caminho modelos tridimensionais animados em AR.

Do mesmo estilo do *Pokémon* temos o jogo *Harry Potter: Wizards Unite* em que o utilizador tem certos modelos tridimensionais animados em AR para libertar, em localizações diferentes.

Estas duas aplicações, assim como o projeto OnTRACK, empregam modelos tridimensionais animados em AR. No caso do projeto, esses elementos são utilizados para exemplificar os exercícios que o utilizador precisa de realizar.

Uma outra aplicação é o *The Journey* [103] tem como propósito ocultar pedras preciosas e proporcionar a busca por pistas, sendo estas reveladas de acordo com o número de passos dados pelo utilizador. Este conceito é relevante, uma vez que, de forma semelhante ao projeto, incentiva a atividade física dos utilizadores, neste caso, associada ao número de passos realizados.

Na revisão de Soltani e Morice [104] aborda-se um pouco de AR nos desportos e sobre como pode melhorar a experiência, não só para praticantes, mas como espetador, treinador e cliente. Faz até uma comparação gráfica do quanto pode ser usado entre estes.

Neste consideram AR benéfico pois num desporto consegue-se adicionar informação, receber *feedback* adicional, estimular treinos, até criar regras consoante a experiência do utilizador.

Para adicionar informação temos um exemplo de *speed skating* [105], que usa *Audio AR* e recebe *feedback* auditivo, com informação da relação entre o sujeito e o comportamento do modelo.

Este é benéfico pois facilita a aprendizagem dos movimentos do *skater* se este seguir os movimentos apresentados pelo modelo. A análise cinética apresenta a correção do movimento de treino em relação ao modelo.

Outro exemplo, mas desta vez para ténis, é o “*Laplacian vision: Augmenting motion prediction via optical see-Through head-mounted displays*” [106], usa-se *Optical See-Through AR* (OSTAR), um *overlay* da trajetória da bola e a cinética das bolas a saltar. O benefício aqui reside na capacidade de visualizar a trajetória da bola, prevendo-a com base no seu movimento.

No bilhar [107], temos também um exemplo que usa OSTAR e *overlays*, com tacadas sugeridas e ajuda na tacada; o que se revela bastante benéfico é a presença de um *overlay* que auxilie na tomada de decisões durante um jogo de bilhar.

Para escalada em rochas existe um exemplo de Kajastila e seus coautores [108], em que se usa *Projection AR* e um *overlay* com rotas de escalada. Conseguindo-se assim uma diversidade maior de movimentos, pois os utilizadores podem criar conteúdos para estas mesmas rotas.

AR pode até pode ajudar pessoas com falta de visão, como o estudo de Kálmán e seus colaboradores [109], que utiliza o *Haptic AR* com vibro-motores num cinto que, conforme a força desses motores, consegue indicar qual a direção correta da sua locomoção. Ou seja, os utilizadores invisuais conseguem correr, por exemplo, numa pista seguindo a vibração destes vibro motores.

Outro exemplo no contexto do ténis é o “*Visualisation of tennis swings for coaching*” [110], que se concentra principalmente em tenistas amadores. Este utiliza um *overlay* de vídeo em gravações existentes. O *overlay* consiste num tenista de elite representado por um avatar 3D, o qual é empregue para comparação com o vídeo do tenista amador. Isso permite que o utilizador visualize áreas de melhoria e compreenda o que poderia ter feito de forma mais eficaz. Essa abordagem facilita a identificação e correção de falhas, proporcionando uma comparação direta para cada caso de uso.

Mesmo em desportos menos praticados como o *Air Hockey* em “*Airhockey over a distance: A networked physical game to support social interactions*” [111], desenvolveram-se implementações para jogar *Air Hockey* à distância. Usou-se *Mechanical AR* em que a mecânica de cada mesa permitia que o disco fosse deslocado de uma localização para outra, criando uma perceção de espaço físico dividido. O benefício notável é a capacidade de promover interação social durante o jogo, através de uma videoconferência.

Algo tão simples como matraquilhos pode ser melhorado quando implementado AR, como em “*KickAR: Exploring game balancing through boosts and handicaps in augmented reality table football*” [112] em que se implementou um método para equilibrar a *skill* dos jogadores, tornando assim numa melhor experiência. O utilizado foi *Projection AR* e utilizava um *overlay* com ícones, em que estes permitiam mudar as regras do jogo em tempo real, entre outros aspetos para "embeleazar" o jogo.

Para basquete temos ainda um artigo de Page e Vande Moere [113] em que se implementou uma *scoreboard* individual e nos *jerseys* dos jogadores, foram incorporadas informações sobre faltas, pontos e alertas de tempo. Isso é benéfico, pois os jogadores estão mais conscientes de determinadas informações durante o jogo.

Falando do desporto mais conhecido no mundo, o futebol, temos o exemplo em “*Sports Support System: Augmented Ball Game for Filling Gap between Player Skill Levels*” [114] que utiliza *Projection AR* e um *overlay* visual que prevê a trajetória e velocidade da bola e

de onde os jogadores deveriam estar (posição recomendada). Extremamente benéfico para jogadores iniciantes, pois torna-se mais fácil perceber o que estão a fazer de errado.

Um exemplo para técnicas de judo, “*Microsoft kinect as a tool to support training in professional sports: Augmented reality application to tachi-waza techniques in judo*” [115] referencia o treino de técnicas de judo, em pé, em que usa *Mirror AR*. Este durante o treino indica qual a postura ótima, com indicadores 2D a representar os desvios espaciais do modelo.

Para as artes marciais, o autor Sodhi e seus coautores de [116] usaram *Projection AR* e um *overlay* para "guiar" as mãos com os movimentos mais precisos. Segundo uma análise cinética, as trajetórias das mãos demonstraram melhor *performance* quando usado *projected AR* em comparação a um vídeo. Facilitando assim o treino e melhorando os movimentos, sendo amador ou não.

Um outro grande desporto, o ciclismo, pode ser melhorado com AR; segundo [117], os resultados de ciclistas melhoraram quando treinavam com "competição". Daí usou-se o *OSTAR* usando *wearables*, em que tinham informação da pista, mapa e notificações (se o ciclista estava atrasado em relação ao tempo planeado, por exemplo).

Num contexto de escalada em rochas, algo semelhante ao mencionado anteriormente, “*BetaCube - Enhancing training for climbing by a self-calibrating camera-projection unit*” [118] utilizou *Projection AR* em que tinha *overlay* com um *replay* de um outro escalador e criação de rotas, estabelecendo a semelhança com o exemplo anterior. Isso torna a escalada mais acessível, pois os praticantes não precisam pensar tanto sobre como abordar a escalada.

Ainda no judo, num estudo de Witkowski e seus colaboradores [119], estudou-se treinar judo sem oponente. Foi usado o *OSTAR* e *Projection AR*, em que utilizava um *overlay* de modelos virtuais, uma imagem real do atleta a fazer a técnica. O que beneficia bastante os atletas pois não estão dependentes de ninguém.

A revisão de Flora Ma e partícipes [120] refere que a tecnologia AR deveria ser mais amplamente considerada para atividades físicas, pois esta poderá melhorar a *performance* dos exercícios efetuados uma vez que tem o potencial de melhorar o desempenho dos exercícios realizados. Além disso, a revisão destaca que a prática de exercícios num contexto de AR motiva determinados utilizadores, especialmente quando é aplicada a gamificação [121].

Passando a uma explicação mais ao detalhe dos seguintes artigos, de notar que foram escolhidos estes artigos pois acharam-se mais relevantes, em relação ao módulo EEfV.

Começando então pelo documento de Lee e seus colaboradores [122] cujo objetivo foi de determinar o efeito de exercícios baseados em AR em pessoas idosas, neste caso do sexo feminino, sendo focado na força dos músculos.

Esta experiência teve 30 sujeitos, sendo estes separados em grupos, AR, *yoga* e *self-exercise*, teve a duração de 12 semanas, em que tinham 3 sessões de 60 minutos cada por semana.

Os exercícios para o grupo de AR foram exercícios de flexão dos joelhos, andar para trás e para a frente, de saltos, em pontas de pés, manter-se só numa perna, subir escadas. Para o grupo de *yoga* os exercícios eram posições, sendo a posição gato, árvore, cobra, triângulo, *twist*, alongar de perna e posição pombo. No grupo de *self exercise*, foi apenas dada uma formação de prevenção de queda e foi sugerido que seguissem o programa de exercícios uma vez todas as segundas.

Obtiveram-se os resultados qualitativos, sendo no geral uma melhoria, em 2 dos grupos, do risco de queda. Usou-se a métrica diferença de *Morse Fall Scale* [123], em que menor pontuação significa menor risco de queda, indo de 0 até 125, em que 0 até 24 é baixo risco, de 24-45 risco moderado e 46 até 125 alto risco. Os resultados quantitativos obtidos por grupo, foram de -10 para o grupo AR (de 23 para 10), -3 para o grupo de *yoga* (de 15,5 para 12,5) e 2,5 para o grupo de *self exercise* (10 para 12,5). Avaliou-se também, a força muscular e verificou-se que este parâmetro melhorou nos dois grupos, sendo maior para o grupo de AR com 15,21 kg no tendão da perna direita.

O artigo de Ellis Louise e coautores [124] em que o objetivo foi examinar o impacto das restrições sociais durante COVID-19 e seu impacto tanto a nível físico como psicológico dos jogadores de jogos AR.

Fez-se um inquérito de 40 perguntas, em que se recolheu informação de jogadores de *Pokémon GO* e *Harry Potter: Wizards United*. No final foram consideradas 2004 respostas.

Os resultados indicam que o tempo dedicado ao exercício físico diminuiu, em média, de 7,5 para 6,5 horas por semana. No entanto, neste estudo, foi possível concluir que esses jogos funcionavam como uma motivação para a prática de exercícios. Para alguns participantes, a única razão para se exercitarem era a presença do jogo, chegando mesmo a mencionar que só saíam de casa devido ao jogo.

### 2.3.1. Discussão

O AR não pode ser comparado com as secções anteriores, pois a sua finalidade é diferente. Neste caso, o AR não é suposto avaliar se as tarefas foram bem executadas, mas sim ajudar o utilizador a realizar essas tarefas da melhor maneira.

Não apenas isso, mas também por meio da gamificação, procura-se instigar o utilizador a executar tarefas que, possivelmente, não teria a motivação para realizar, caso não fossem apresentadas sob uma abordagem ludificada.

Com AR melhora-se também a qualidade do exercício, pois é possível comparar-se ao *overlay* visual ou redefinir a posição devido à AR da trajetória da bola, por exemplo.

## 3. Desenvolvimento

Neste capítulo serão retratados conceitos e tecnologias relevantes, a metodologia de desenvolvimento, a arquitetura e a descrição da solução desenvolvida.

### 3.1. Conceitos e tecnologias relevantes

Nesta secção estão representadas todas as tecnologias utilizadas para a realização deste projeto de investigação. Para cada tecnologia, uma subsecção e sua breve explicação.

#### 3.1.1. Realidade aumentada

Antes de referir as tecnologias que se utilizaram, explicar sucintamente o que é AR.

AR é uma versão melhorada do mundo real, conseguindo-se através do uso de elementos visuais, sonoros, e outros estimulantes sensoriais via tecnológica ([125], [126]). O objetivo da AR é, então, melhorar o mundo real, através de objetos 3D para ajudar na compreensão de um outro objeto real, podendo até ser possível interagir com o objeto virtual para mais informações.

Esta tecnologia utiliza ferramentas que permitam a AR; é necessário algum poder de processamento, pois utiliza, geralmente, o sistema operativo do dispositivo e câmaras ou lentes, para visualizar o que foi colocado no ambiente e sensores para alinhar o mundo digital com o real.

#### 3.1.2. ARCore

ARCore ([127]) é uma plataforma da Google utilizada para criar várias experiências em AR. Usa diferentes API para permitir a perceção do ambiente em volta e com isso haver uma melhor interação e mais informação disponível.

Utilizou-se esta plataforma pois é a indicada para *Android*; uma vez que a plataforma OnTRACK já existente foi desenvolvida em *Android*, decidiu-se então utilizar ARCore.

As três capacidades mais importantes para o ARCore são *Motion Tracking*, a compreensão do meio envolvente e a estimação da luz presente.

##### 3.1.2.1. Conceitos fundamentais do ARCore

Para conseguir realizar as ações necessárias, o que o ARCore faz, principalmente, é obter a posição do utilizador enquanto se move, através da câmara, e construir o seu próprio conhecimento do mundo real. Usando assim, como referido anteriormente, o *Motion Tracking*.

*Motion tracking* é um processo de localização e mapeamento simultâneo, ou SLAM, e serve para entender onde está o dispositivo móvel em relação ao mundo. Para isto necessita de capturar imagens, chamadas de *feature points*, que são pontos com características

visualmente distintas. Usa estes pontos para computar a sua mudança de local, com informação visual e com as medidas de inércia consegue-se a pose, posição e orientação do dispositivo, podendo assim alinhar a câmara virtual e adicionar objetos 3D na perspetiva correta, parecendo-se assim um objeto da vida real.

A tecnologia ARCore procura sempre para *clusters* de *feature points* (conjunto de *feature points*) conseguindo assim distinguir os planos horizontais dos verticais e usar a informação para colocar objetos virtuais nos planos reais. Como também inclui perceção de profundidade e estimação de luz, o utilizador consegue colocar objetos 3D e estes parecerem estar mais próximos, e/ou mais escuros. [128]

### 3.1.2.2. ARCore Geospatial

A tecnologia ARCore tem várias API, como já referido, a que se utilizou no módulo EEfV é a *Geospatial* ([5]), esta tecnologia consiste em criar experiências de AR numa certa coordenada. Usa o sensor e GPS do dispositivo, deteta o seu ambiente, compara-o com o modelo disponibilizado pelo *Visual Positioning System* (VPS), pertencente à Google, e adquire assim a localização do dispositivo. Este procedimento é feito para não haver discrepância entre as coordenadas do dispositivo e as do VPS.

As imagens do *Street View*, que são a base do VPS, juntamente com redes neuronais, criam partes de imagens para serem utilizadas na computação de pontos 3D do ambiente global. Quando o utilizador faz o pedido este processa os pixéis e tenta reconhecer parte do ambiente comparando-o com o modelo do VPS, conseguindo assim a posição e orientação do dispositivo e a localização do VPS.

Com esta tecnologia é possível criar objetos virtuais, para o qual também se utilizam pontos âncora. Resumidamente, os pontos âncora são pontos que asseguram a posição do objeto virtual ao longo do tempo. Estes utilizam a posição segundo o *World Geodetic System* (WGS) (mais informações em [129]), tornando assim possível criar um ponto âncora em quase todo o mundo. Os pontos âncora e as poses estão interligados, sendo que as poses podem variar de acordo com a sua posição e ambiente. Estes pontos âncora podem ser criados com base na pose, que por sua vez devolve uma posição (x,y). Os planos são um tipo de pose, porém são especiais, uma vez que são *trackables* e, quando utilizados com os pontos âncora, criam um objeto virtual. Mesmo que o utilizador mude a sua posição, o objeto permanece no mesmo plano. Um exemplo seria criar um objeto virtual numa parede, no canto inferior esquerdo, quando o ARCore atualizar a posição da pose, este ficaria na parede, no mesmo canto.

Os dispositivos que podem utilizar esta tecnologia estão listados em [130].

### 3.1.3. TensorFlow Lite

TFLite ([131]) é uma *framework* de ML, focada nos dispositivos móveis, microcontroladores e outros dispositivos *edge*. Tendo esta *framework* bastante otimização e uma boa *performance*, comparada com a TensorFlow, é então utilizada para este tipo de dispositivos.

A *framework* TensorFlow para reconhecimento e predição, utiliza uma biblioteca denominada de TFLite *Task Library* [132], que dispõe de modelos e *interfaces* para tarefas de ML, podendo estes estar relacionados com classificação de vídeo, áudio, texto ou imagem.

Para este projeto o que se utilizou foi *pose estimation* e *video classification*.

#### 3.1.3.1. Pose estimation

Na *pose estimation* o objetivo é, usar-se um modelo ML, estimar a pose de uma pessoa, a partir de uma imagem ou vídeo. Basicamente esta é uma tarefa de ML que estima onde estão os *keypoints* do corpo; *keypoints* são, no fundo, as articulações, com um ID indexado. Com essa informação o modelo tenta processar a imagem com *inputs* e *outputs* dos *keypoints*, com um *score* entre 0 e 1, com uma confiança associada, sendo que esta é a probabilidade desse *keypoint* existir naquela posição. Um dos modelos "*state of the art*" é o *Movenet* [133] que consegue detetar 17 *keypoints*.

#### 3.1.3.2. Video classification

Na classificação de vídeo, que também é uma tarefa de ML, não se estima apenas a pose de uma pessoa, mas tenta-se sim identificar o que o vídeo representa, seja uma ação ou movimento. O modelo deste tipo necessita de ser treinado com um conjunto de classes únicas, sendo usados, no modelo, os *frames* como *input*; os *outputs* são a probabilidade de essa classe única estar representada.

A classificação de vídeo utiliza as relações espaço-temporais entre os *frames* adjacentes, processando-os e reconhecendo assim as ações no vídeo.

Os tipos de modelos que usa são os *Mobile Video Networks* (MoViNets), estes são treinados com um excerto de vídeo com uma etiqueta associada, sendo cada etiqueta um conceito, ou classe, para reconhecer. O modelo é um modelo de *streaming*, este recebe vídeo continuamente e responde em tempo real, identificando as classes do *dataset* de treino no vídeo, *frame* por *frame*.

#### 3.1.4. Sensores do dispositivo móvel

Os dispositivos móveis dispõem de vários sensores, que podem ser desde acelerómetros a sensores de luz ambiente e até sensores de som. Os sensores variam de dispositivo para dispositivo, mas no geral, muitos estão presentes; como por exemplo os sensores de impressão digital e o de luz ambiente (sensor que muda a luz do ecrã automaticamente).

#### 3.1.5. Google Maps API

A API que a linguagem *Angular* usa para os mapas é a API da Google, logo necessita de uma API *key* do Google *maps*, podendo, ou não, usar a API de *places* [134], para que no mapa se verifique os pontos de interesse devolvidos pela API. No módulo EEfV decidiu-se usar a API de *places* pois poderiam ser usados como pontos de referência.

Utiliza também a API de elevação da Google [135] para quando se criam marcadores estes tenham uma elevação do terreno, esta será utilizada pelos modelos tridimensionais animados para estes estarem o mais próximo do chão quanto a API devolver.

Esta API (Google *Maps*), como o nome indica, é uma API que, quando se faz um pedido, devolve um mapa, podendo este mapa ser personalizado dentro de certos parâmetros. Neste mapa, é possível acrescentar marcadores de diferentes tipos e utilizar estilos distintos, como o estilo de satélite, que exhibe o terreno com imagens de satélite, ou o estilo padrão, que mostra apenas os tipos de terreno com cores diferentes.

## 3.2. Metodologia de desenvolvimento

Nesta secção descreve-se a metodologia que se usou e os requisitos para este projeto em específico, além disso também uma breve explicação da metodologia.

### 3.2.1. Prototipagem evolutiva

A Prototipagem Evolutiva é uma abordagem de desenvolvimento de *software* que envolve a construção rápida de uma versão básica e funcional de um sistema de *software* e, em seguida, aprimorá-la continuamente por meio de múltiplas iterações. O principal objetivo da prototipagem evolutiva é obter *feedback* cedo no processo de desenvolvimento e adaptar rapidamente o *software* com base no retorno dos utilizadores e nas mudanças nos requisitos [136]. Segundo [137] esta abordagem é dividida em 4 fases.

1. Conceito inicial
2. Design e Implementação do protótipo inicial
3. Refinamento e melhoria
4. Convergência para o produto final

O protótipo inicial é geralmente uma versão simples do produto final, sendo o foco recursos e funcionalidades críticas (necessárias). Depois com base no *feedback* recebido o protótipo vai ser refinado, seja com correção de *bugs*, a novas *features* e/ou refazer certas *features*. Continua-se este processo até que o *software* esteja de acordo com os requisitos finais.

Neste projeto, o conceito inicial era, uma aplicação com funcionalidades diferenciadoras para a plataforma OnTRACK, nomeadamente que permitissem: validar a correta execução dos exercícios prescritos de forma automatizada com recurso a técnicas de ML; criar “estações de exercício virtual” em locais reais com recurso a mecanismos de AR; utilizar ferramentas de localização e *context awareness* para proactivamente sugerir exercícios.

O design e implementação do protótipo inicial realizaram-se através de atividades de desenvolvimento e validação do protótipo inicial. Sendo “*spikes*” - investigações rápidas para explorar soluções técnicas – apresentaram-se estas à equipa do projeto OnTRACK para avaliação. Posteriormente, avançou-se para a definição da arquitetura, conforme descrita na secção 3.3 e para a implementação do protótipo, que estava dividido em “*spikes*”. Durante

este processo, o protótipo aprimorou-se continuamente com base no *feedback* recebido da equipa OnTRACK, principalmente dos orientadores do trabalho.

Conforme mencionado anteriormente, a fase de refinamento e melhoramento conduziu-se com base no *feedback* recebido, principalmente dos orientadores do projeto, mas, também da equipa envolvida. Durante essa etapa, introduziram-se as denominadas “*spikes*” no âmbito do projeto OnTRACK, resultando na criação do protótipo inicial.

Submeteu-se o protótipo a várias iterações de refinamento e melhoramento, incluindo a correção de bugs, a incorporação de novas funcionalidades e a revisão dos recursos existentes. Esse processo iterativo possibilitou uma evolução gradual do *software*, assegurando que atendesse aos requisitos e expectativas da equipa.

A convergência para o produto final ocorreu após diversas iterações de refinamento, quando o protótipo atingiu um estado considerado estável e funcional. Esta fase representou o resultado, final, deste trabalho, descrito na secção 3.4. Posteriormente, o protótipo final foi submetido a um processo de validação, descrito na secção 4.

### 3.2.2. Levantamento de requisitos

Lista de requisitos, final, do projeto:

- *Body pose estimation*
  - Implementação de um sistema de estimativa de pose corporal para identificar e monitorar os movimentos do utilizador durante os exercícios.
- Contagem de repetições
  - Desenvolvimento de um algoritmo para contagem automática de repetições com base nos dados de pose estimados.
- Integração de AR
  - Utilização de tecnologia de AR para criar modelos tridimensionais animados que demonstrem visualmente os exercícios aos utilizadores.
- Exemplificação Visual
  - Apresentação clara e intuitiva dos exercícios, oferecendo aos utilizadores uma referência visual detalhada por meio dos modelos tridimensionais animados em 3D.
- Mapa Interativo
  - Implementação de um mapa interativo que mostra as localizações dos modelos tridimensionais animados AR como estações de exercícios, proporcionando uma visão geral do ambiente virtual de treino.
- Personalização de Exercícios
  - Capacidade de personalizar os exercícios com diferentes níveis de dificuldade e objetivos, permitindo uma experiência adaptada às necessidades individuais dos utilizadores.
- *Feedback* em tempo real

- Fornecimento de *feedback* em tempo real sobre a postura e a execução correta dos exercícios para maximizar a eficácia do treino.
- Registo e Histórico
  - Implementação de funcionalidades para o registo e acompanhamento do progresso do utilizador ao longo do tempo.
- Compatibilidade de multiplataforma
  - Desenvolvimento da aplicação para ser compatível com diversas plataformas, como dispositivos móveis, *tablets* e, se possível, dispositivos de AR.
- Segurança e Privacidade
  - Implementação de medidas robustas de segurança e privacidade dos dados do utilizador, especialmente considerando a natureza sensível da informação de localizações.

### 3.3.Arquitetura

Nesta secção representa-se a arquitetura, através de figuras, acompanhadas de uma breve explicação do seu conteúdo. Esta arquitetura abrange a plataforma de forma geral, bem como a aplicação *web* específica.

#### 3.3.1. Arquitetura da plataforma

A seguinte figura, Figura 7, demonstra a arquitetura da plataforma, não só do módulo EEV. Certas partes do módulo EEV não utilizam o servidor do IPL, conseqüentemente do ataque informático que ocorreu durante a realização do trabalho. Este poderia colocar em causa a disponibilidade da plataforma no período de submissão deste trabalho, pelo que se decidiu utilizar um servidor local.

Refira-se que apenas se trabalharam as partes *Web* e *mobile*, no módulo EEV; e que o servidor é um servidor local. No entanto, o servidor local utiliza as mesmas tecnologias: *kubernetes* e PostgreSQL. Para deixar bem claro, o que diferencia da Figura 7 para o módulo EEV é o servidor; ao invés de Servidor IPL trata-se de um servidor local.

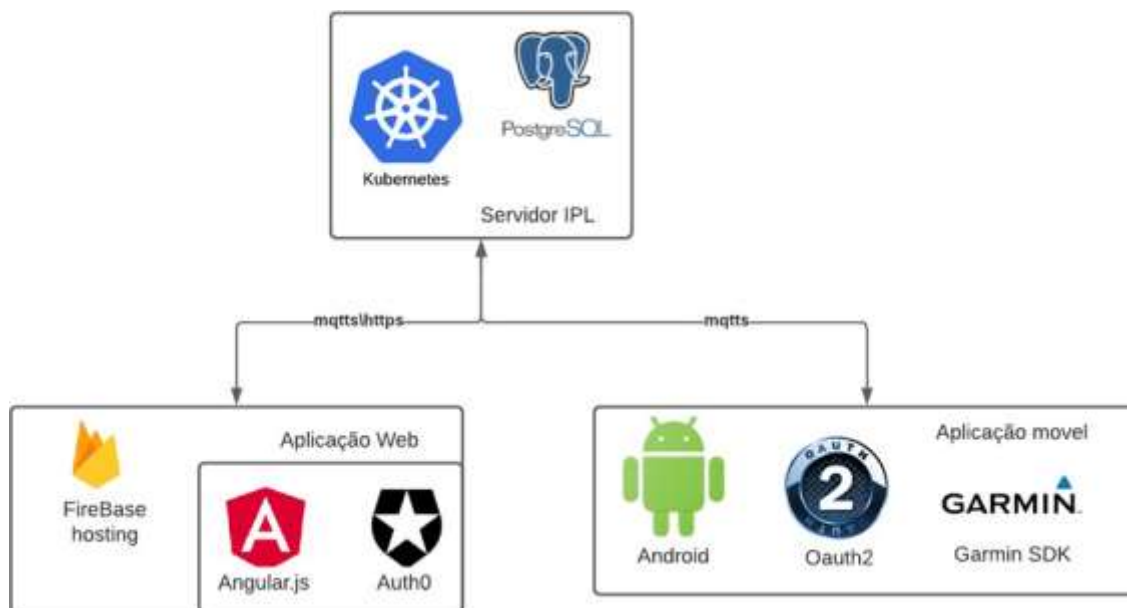


Figura 7: Arquitetura da plataforma

### 3.3.2. Arquitetura do módulo desenvolvido – Aplicação Web

Nesta subsecção referencia-se o que foi utilizado do projeto OnTRACK previamente realizado, e o que se adicionou no módulo EEJV. Quando mencionada a aplicação web, referencia-se tanto a vertente de *frontend* como a de *backend*.

Desenvolveu-se o *backend* em kubernetes, onde cada *pod* corresponde a um micro-serviço. No entanto, ao longo do projeto, devido a algumas questões de desempenho, optou-se por adicionar a nova funcionalidade a um *pod* já existente. A componente de *backend*, neste caso, expõe exclusivamente os serviços MQTT e MQTT-WSS, sendo este o único ponto de entrada dos clientes para o *backend*, destacando novamente que esta estrutura já estava previamente desenvolvida.

O cluster é composto por dois servidores que sustentam os micro-serviços da plataforma: PostgreSQL, responsável pelos serviços de base de dados; e VerneMQ, encarregado de fornecer a camada de transporte através de MQTTS e MQTT-WSS.

Os micro serviços têm uma ampla utilização nos dias de hoje devido à sua fácil escalabilidade e capacidade de otimizar a relação custo/desempenho. Outra vantagem dos micro-serviços é que, se um deles falhar, apenas a funcionalidade associada fica inativa, não afetando toda a aplicação.

Para a componente de *frontend* como representado na Figura 7, recorreu-se ao *Firebase* para hospedar a aplicação, ou seja, para disponibilizá-la num domínio específico.

Para a autenticação baseada em e-mail/palavra-chave, optou-se pelo Auth0. Escolheu-se a *framework* Angular JS.

No *frontend*, não houve alterações significativas na arquitetura do módulo EEJV, exceto pela não utilização do *Firebase*, devido à hospedagem local.

### **3.4. Descrição da solução desenvolvida**

Nesta secção aborda-se as funcionalidades da plataforma, não no seu todo, mas as realizadas para o módulo EEFFV, tanto aplicação móvel, como *web* e *backend*.

#### **3.4.1. Funcionalidades da app móvel**

As próximas subsecções serão exclusivamente dedicadas à aplicação móvel, abordando apenas o desenvolvimento realizado no módulo EEFFV, uma vez que é o componente com maior ênfase e, portanto, será mais detalhado.

#### **3.4.2. Funcionalidade de realidade aumentada**

Esta funcionalidade consiste em adicionar ao mundo real estações de exercícios virtuais, para os utilizadores saberem que tipo de exercícios existem naquele espaço, através da colocação de um modelo 3D animado virtual a realizar o exercício desejado.

Para esta funcionalidade usou-se o ARCore *Geospatial* para Unity (AR *Foundation* [26]). Existiu a possibilidade de se usar a API para *Kotlin/Android*, no entanto um dos objetivos era colocar objetos animados para os exercícios, por essa razão usou-se a API para Unity pois tornava a modelação mais simples.

Esta funcionalidade é algo simples, pois apenas transmite o que a câmara recebe e, quando encontrada uma superfície, dispõe então os objetos virtuais (modelos tridimensionais animados a exemplificar o exercício), demonstrado na Figura 8.



Figura 8: Captura de ecrã do modelo animado AR 3D

Para integrar esta funcionalidade com a aplicação móvel, foi necessário estabelecer comunicação entre elas. Para isso, decidiu-se utilizar um botão na aplicação móvel para iniciar o exercício. Este botão é exibido apenas quando o utilizador aponta para uma estação de exercício. Ao clicar no botão, é enviado um *Intent* correspondente, com o objeto da *main activity* (disponibilizado como *companion object* na classe *MainActivity* da aplicação *android*) e com isso usar uma função disponível nesse objeto. Para se usar classes de *android* em Unity é preciso usar *AndroidJavaObject*.

Explicando sucintamente o controlador e o que faz; em Unity existem funções que se dedicam à parte de UI e ao fazer *update* deste mesmo tendo o *Awake*, *Update*, *OnEnable*, *OnDisable*, sendo que *Awake* chama-se ao iniciar a instância de *script*, mas apenas depois de iniciar a cena. O *Update* chama-se todos os *frames*, *OnEnable* chama-se quando o

*GameObject*, neste caso o próprio *script*, é ativado; o *OnDisable* chama-se quando este é desativo ou o objeto é destruído (exemplo: sair da aplicação).

Relativamente à lógica do projeto, começando pelo pedido ao *backend* na função *GetExercisesAsync*: esta é uma função assíncrona em que se recebe um cliente HTTP como parâmetro, e faz-se um pedido através de um URL, sendo a resposta um JSON; este JSON transforma-se em *string* e a função *GetExercisesAsync* (Listagem 1) devolve este JSON. Quando os exercícios são devolvidos, estes consomem-se e são adicionados ao ecrã pela função *PlaceGeospatialAnchor* (Listagem 2).

```
static async Task<string> GetExercisesAsync(HttpClient client) {
//allows for validation of SSL certificates
ServicePointManager.ServerCertificateValidationCallback = ((sender, certificate, chain,
sslPolicyErrors) => true);
var json = await client.GetStringAsync("https://ontrack.social:30008/api/exercises");
    return json.ToString();
}
```

Listagem 1: Código função *getExercisesAsync*

```
private ARGeospatialAnchor PlaceGeospatialAnchor(GeospatialAnchorHistory history) {
var pose = EarthManager.CameraGeospatialPose;
Quaternion quaternion = Quaternion.AngleAxis((float)poseHeading, Vector3.up);
var anchor =
AnchorManager.AddAnchor(history.Latitude,history.Longitude,history.Altitude,quaterni
on);
if (anchor != null) {
    GameObject anchorGO;
    if (!history.IsDone) {
        anchorGO = Instantiate(GeospatialPrefab, anchor.transform);
        anchorGO.GetComponentInChildren<TextMeshPro>()[0].text =
        $"Do{history.Amount}{history.Type}s";
    } else {
        anchorGO = Instantiate(GeospatialPrefabWhenDone, anchor.transform);
        anchorGO.GetComponentInChildren<TextMeshPro>()[0].text = "Exercise Completed!";
    }
    anchorGO.GetComponentInChildren<TextMeshPro>()[1].text = $"{history.Id}";
    anchorGO.GetComponentInChildren<TextMeshPro>()[1].enabled = false;
    anchorGO.SetActive(true);
    _anchorObjects.Add(anchorGO);
    SnackbarText.text = $"{_anchorObjects.Count} Anchor(s) Set!";
} else {
    SnackbarText.text = string.Format("Failed to set {0}!", "an anchor");
}
return anchor;}

```

Listagem 2: Código função *PlaceGeospatialAnchor*

Explicando sucintamente a função *PlaceGeospatialAnchor*: recebe-se o objeto do *backend* começando-se pela pose, pose esta que retira um objeto da biblioteca fonte ARCore para determinar a orientação na qual o utilizador está voltado. É possível calcular um *quaternion* correspondente e, em seguida, aplicá-lo ao objeto 3D, para este ser criado virado para o utilizador. Com esses dados transforma-se então o objeto *anchor* para ser possível adicionar no ecrã. Se o *anchor* não for *null*, ou seja, se não houver erros, gera-se uma instância do objeto e muda-se o texto conforme o necessário. A lógica que fez mais sentido foi, se este exercício ainda não estiver completo, demonstrar o texto do exercício, sua quantidade e tipo; se o exercício estiver completo demonstrar um outro objeto 3D e o texto ser "*Exercise*

*Completed!*". Acabando esta função com ativar o objeto na vista e adicionar ao vetor de *anchors*.

Outras funções pertinentes neste módulo, EEFV, são a *IsTargetVisible*, a *RotateAnchor* e a *StartExercise* (Listagem 3). Demonstra-se o botão de *StartExercise* devido à função *IsTargetVisible*, função esta que demonstra o nome associado ao objeto 3D, sendo que este só é exibido quando o objeto se encontra na câmara. Carrega-se no botão para iniciar o fragmento de *Pose Estimation*. O objetivo da função *RotateAnchor* é aplicar-se a rotação de maneira que o objeto fique voltado para o utilizador.

```
bool IsTargetVisible(Câmara c, GameObject go) {
var planes = GeometryUtility.CalculateFrustumPlanes(c);
var point = go.transform.position;
foreach (var plane in planes) {
    if (plane.GetDistanceToPoint(point) < 0)
        return false;
    }
    return true;
}
private void RotateAnchor(GeospatialPose pose, GameObject anchorGO) {
    if (IsTargetVisible(Câmara.current, anchorGO)) {
        goVisibleText = anchorGO.GetComponentInChildren<TextMeshPro>()[0].text;
        goVisibleIdText = anchorGO.GetComponentInChildren<TextMeshPro>()[1].text;
        StartExerciseButton.gameObject.SetActive(true);
    } else {
        StartExerciseButton.gameObject.SetActive(false);
    }
    anchorGO.transform.rotation = Quaternion.AngleAxis((float)pose Heading, Vector3.up);
}
public void StartExercise() {
    AndroidJavaClass main = new("pt.ipleiria.citechcare.ontrack.ui.MainActivity");
    AndroidJavaObject mainObject = main.GetStatic<AndroidJavaObject>("activity");
    mainObject.Call("launchPose", goVisibleText, goVisibleIdText);
}
```

Listagem 3: Código das funções *IsTargetVisible*, *RotateAnchor* e *StartExercise*

### 3.4.3. Funcionalidade do mapa

Esta funcionalidade consiste na representação de um mapa, no qual exibem-se os exercícios existentes na base de dados, por meio de marcadores. Além disso, representa-se a localização do utilizador através de um marcador, caso este permita o uso da localização, conforme ilustrado na Figura 9. Na representação do mapa, marca-se a posição do utilizador a roxo e dos exercícios em amarelo.

Demonstra-se no título deste marcador de exercício a estação e seu estado, podendo este estado ser *Done* ou *To Do*; no caso da Figura 9 encontra-se no estado *To Do*, na sua descrição demonstra as suas coordenadas para o utilizador a encontrar facilmente.

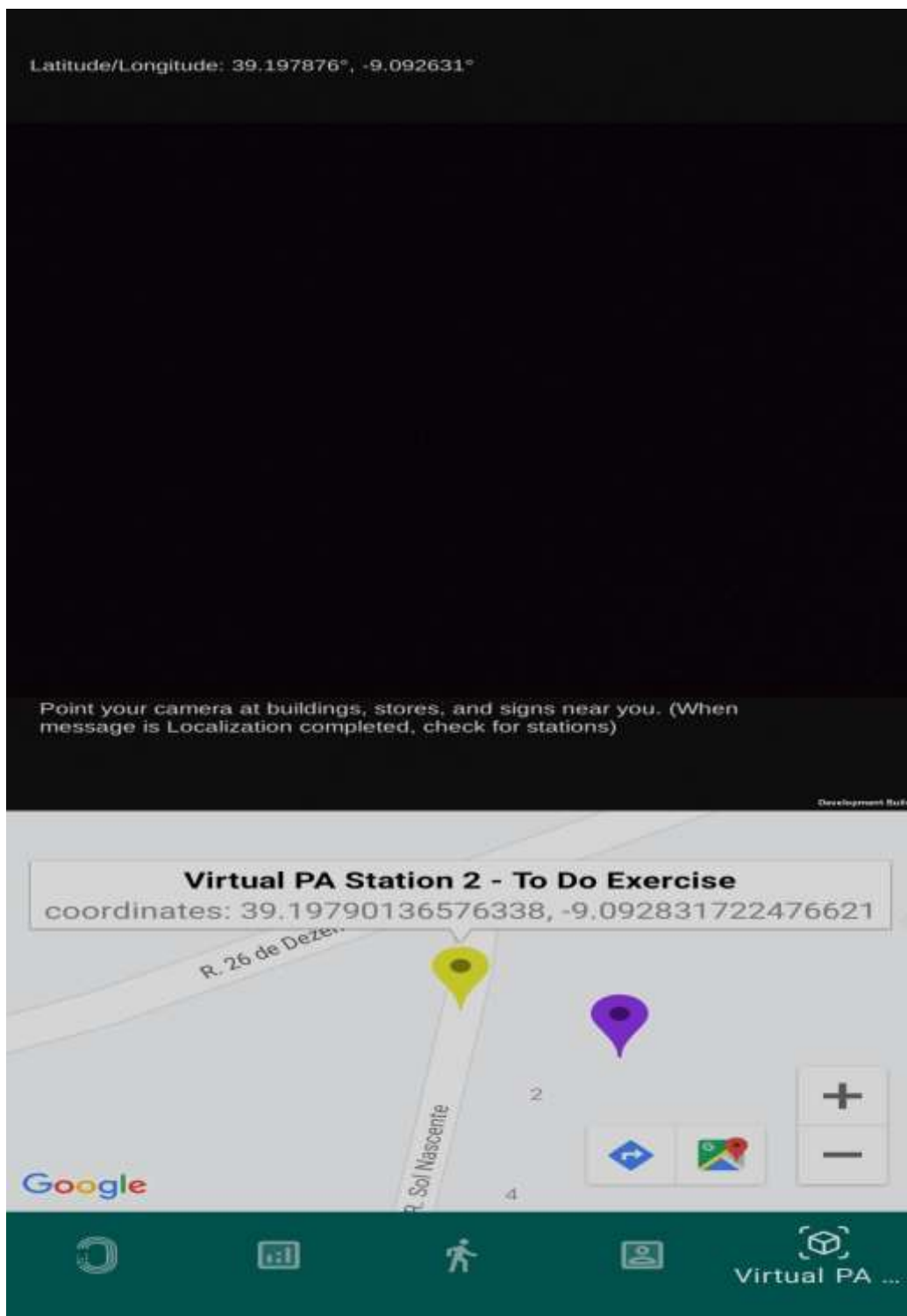


Figura 9: Captura de ecrã do mapa na plataforma de mobile

Usou-se a biblioteca *Jetpack Compose* para desenvolver esta *feature*, juntamente com a API da Google para tratar dos dados de localização e demonstrar o mapa. Faz-se um pedido, através do *viewModel* para o *backend*; este *endpoint* devolve a lista dos exercícios, de seguida, acede à localização do utilizador, se assim o permitir, com a variável *fusedLocationProviderClient*.

O mapa só é demonstrado quando a *flag show* se verifica e a lista de exercícios não é vazia. A posição *default* do mapa, caso o utilizador não tenha dado autorização para a localização, é o primeiro exercício.

Assim, cria-se o *Composable Google Map*, que desenha o mapa na tela. Cria-se marcadores de acordo com o número de exercícios e um marcador para a localização do utilizador, se disponível. Se não houver informações sobre os exercícios, apenas se exibe uma mensagem a demonstrar que não existem quaisquer exercícios.

#### 3.4.4. Sensores de acelerómetro

Usaram-se sensores no estudo deste projeto para uma tentativa de reforço na confirmação dos exercícios, nomeadamente dos agachamentos; através do acelerómetro no eixo dos Y, pois o utilizador deve usar o telemóvel verticalmente para fazer os agachamentos, consegue-se obter uma aceleração quando o utilizador sobe e desce. Para este sensor, os valores considerados fidedignos encontravam-se num intervalo de aceleração maior de 14 m/s<sup>2</sup> ou menor do que 6,5 m/s<sup>2</sup>. Para chegar a estes valores usou-se uma aplicação como teste, *Arduino Science Journal* [138], utilizou-se tal aplicação para se gravar os resultados, a aplicação permitia sem grande esforço adicional, e com isso fazer-se a comparação dos valores nos exercícios desejados.

Na parte de testes da verossimilidade destes sensores, utilizou-se a biblioteca *SensorManager*, que estende da *interface SensorEventListener*, tendo esta *interface* duas funções, *onSensorChanged* e *onAccuracyChanged*, em que a primeira referida é chamada sempre que o sensor deteta uma nova leitura, mesmo que o dispositivo não se movimente este faz uma nova leitura (com um novo *timestamp*); a segunda apenas é chamada quando o valor da precisão muda.

A conclusão alcançada foi que a implementação desta funcionalidade na nossa aplicação não seria vantajosa, devido a um intervalo de precisão considerado "fácil" de ser comprometido. Fatores como a orientação do dispositivo móvel (não paralelo ao utilizador) podem resultar em leituras incorretas, assim como a execução dos agachamentos em uma velocidade mais lenta por parte do utilizador. Além disso era impossível, usar-se os sensores e *body pose estimation* em simultâneo pois *body pose* requer a câmara para visualizar a pose do utilizador, enquanto os sensores necessitam que o utilizador tenha o dispositivo na sua posse.

#### 3.4.5. Funcionalidade do *body pose estimation*

Na pesquisa sobre qual tecnologia, na parte de *body pose estimation*, se deveria usar descobriu-se que existiam 2 grandes soluções viáveis: MLKit e TensorFlow. Aquando da pesquisa, o MLKit infere 33 *keypoints* com o modelo BlazePose [139] e usa o algoritmo *k-nearest neighbors*, que basicamente classifica a classe correta, pela classe mais comum entre k vizinhos próximos (em que k é designado pelo *developer*). No entanto, aquando da resolução do projeto, em abril de 2023, prevê-se uma atualização que poderá resultar na falta de compatibilidade com versões anteriores do sistema, para não correr o risco de a aplicação

deixar de funcionar, optou-se por usar o TensorFlow. No que se necessita para o módulo EEFFV, tinham *performances* equiparadas.

TensorFlow infere 17 *keypoints*, utiliza o modelo *MoveNet*, [133], em que este tem duas vertentes, *Lightning* e *Thunder* sendo o primeiro mais pequeno, mais rápido, mas menos preciso e o segundo sendo a versão contrária, mais lenta, maior e mais precisa.

Analisou-se, também, a possibilidade de usar *video classification* [140] invés de *pose estimation*, no entanto a *video classification* utiliza-se para reconhecer ações, como por exemplo jogar futebol, e não para fazer repetições.

Uma questão pertinente, foi usar-se apenas um modelo para todas as posições, ou um por cada. Segundo os exemplos, encontrados em [141], para fazer classes múltiplas existem dois métodos principais: *One-Vs-All* e *One-Vs-One*. No método *One-Vs-All* são criados  $n$  modelos classificadores binários, onde  $n$  é o número de classes. Já no método *One-Vs-One* utiliza-se  $n * n - 1/2$  modelos classificadores binários, onde  $n$  é o número de classes. Conclui-se que, para este projeto, a abordagem mais apropriada seria a utilização do método *One-Vs-All*. Após avaliação do número de poses envolvidas, determinou-se que essa abordagem seria mais adequada, uma vez que cada pose seria associada a um modelo específico, eliminando possíveis ambiguidades e simplificando o processo de classificação.

Para usar *body pose estimation* em TFLite, analisou-se também TensorFlow e como funcionava com ML, pois TFLite é uma versão otimizada para aparelhos *edge*. Para facilitar este estudo seguiu-se a documentação [4] e para entender melhor como se resolvia o problema em *Android*, usou-se o exemplo já criado (encontrado em [142]).

Para se fazer o modelo do protótipo, com *Jumping Jacks*, flexões e agachamentos, usou-se o caderno na plataforma Colab da Google [143].

Para desenvolver o modelo em TFLite criou-se o modelo em TensorFlow, e seguidamente converteu-se para TFLite. O Colab utilizado é dividido em três etapas principais: pré-processamento das imagens, treinar a classificação de poses e converter o modelo para TFLite.

Começando por explicar que este modelo de classificação usa o modelo TFLite, modelo este que aceita o *output* do *MoveNet* como *input*, e devolve a classificação de poses, como os exercícios realizados ("Flexões, agachamentos").

Passando então ao primeiro passo, o processo foi gerar o *dataset* de treino com as imagens etiquetadas, e correr com o *MoveNet* conseguindo assim capturar todas as *landmarks* e suas etiquetas, guardando num ficheiro CSV. Para isso, necessitou-se de fazer *upload* de um arquivo compactado (*zip*) contendo uma estrutura de diretórios organizada da seguinte forma: uma pasta base, dentro da qual estão localizadas várias pastas, correspondentes a cada pose necessária. Cada uma dessas pastas de poses contém várias imagens representando a pose correspondente.

Este *zip* vai se dividir em diretório de treino e de teste.

Após a divisão, o próximo passo é realizar o pré-processamento do *dataset* de treino e teste. Este arquivo contém as coordenadas das *landmarks* como "X", as etiquetas corretas como "Y" em formato (N, *label\_count*), a lista de todas as classes encontradas no *dataset* como "classes" e um *dataframe* chamado "*dataframe*", que armazena o CSV como um *dataframe* do *Pandas*, incluindo as *features* e as etiquetas corretas para uso posterior na classificação do modelo.

Passando ao passo 2, a criação do modelo de classificação de poses. Este é dividido em 2 submodelos, sendo eles o vetor de *features* para as *landmarks* detetadas, o submodelo 2 que utiliza esse vetor numa camada densa para prever as poses e, por fim, treina-se o modelo baseado no *dataset* pré-processado anteriormente. Convertem-se as coordenadas que são transformadas de um vetor de *features*, vetor este que é uma transformação das *landmarks*; primeiramente move o centro da pose para a origem, escala a pose para o tamanho de 1, por fim, nivela as coordenadas construindo um vetor de *features*. Depois de detetar as *landmarks* calcula-se então as poses usando-se um modelo *Keras*.

Por último, o terceiro passo, converte-se o modelo TensorFlow (*Keras*) para TFLite. É necessário converter para TFLite pois os dispositivos móveis requerem esse formato. Para se converter usa-se *dynamic range quantization* [144] para reduzir o tamanho do modelo, aproximadamente 4 vezes, sem perder quantidades significativas de rigor. Na conclusão deste passo cria-se também um ficheiro que mapeia os índices das classes para o nome, para ser perceptível para humanos.

Usaram-se imagens para treinar e testar o modelo podem-se encontrar em [145] para as flexões, [146] para os agachamentos; deste *dataset* usou-se apenas as imagens que mais se adequavam ao modelo, tendo sido o processo de escolha feito manualmente. Por último, para os *Jumping Jacks* usaram-se dois vídeos em que se retiraram *frames*, pois não foi impossível encontrar *datasets* com alguma qualidade; os vídeos foram [147] e [148]. Tanto para o *dataset* de agachamentos como para o de *Jumping Jacks* as imagens dividiram-se manualmente, entre os dois passos do exercício (considerado *up/down*) para se conseguir treinar o modelo e assim fazer o contador de repetições.

O Colab devolve dois ficheiros, o modelo em si "*Pose\_classifier.tflite*" e "*Pose\_labels.txt*". O primeiro é o modelo em TFLite e o segundo as *labels* das poses transformadas em linguagem humana, para poder haver perceção do resultado. Usa-se este modelo, ficheiro TFLite, na aplicação, no diretório *assets* que posteriormente se utiliza na classe *Pose\_classifier* no diretório "ML". Usa-se este juntamente com um *Interpreter* (biblioteca do Tensor Flow) para fazer a coligação entre pose e a sua *label*.

Depois de alguns testes, concluiu-se que poderia haver alguns problemas a fazer a contagem dos exercícios, por estes terem algumas poses parecidas (*Jumping Jacks down* e *squats up*). Deste modo, optou-se por separar os modelos e assim ser uma classificação mais simples e com menos chance de confusão. Apenas se separou os modelos pois achou-se que faria sentido neste contexto.

Para demonstrar os *keypoints* do modelo, utilizaram-se pontos vermelhos nas articulações, com linhas a interligá-los, como demonstrado na Figura 10. Como é possível observar na figura, esta também demonstra o número de repetições, o *score* que representa a confiança atual da posição, seguido do nível de confiança necessário para a posição estar correta. Demonstra também quais as posições que acha corretas, e o seu nível de confiança (*score*), representado em maior detalhe na Figura 11.

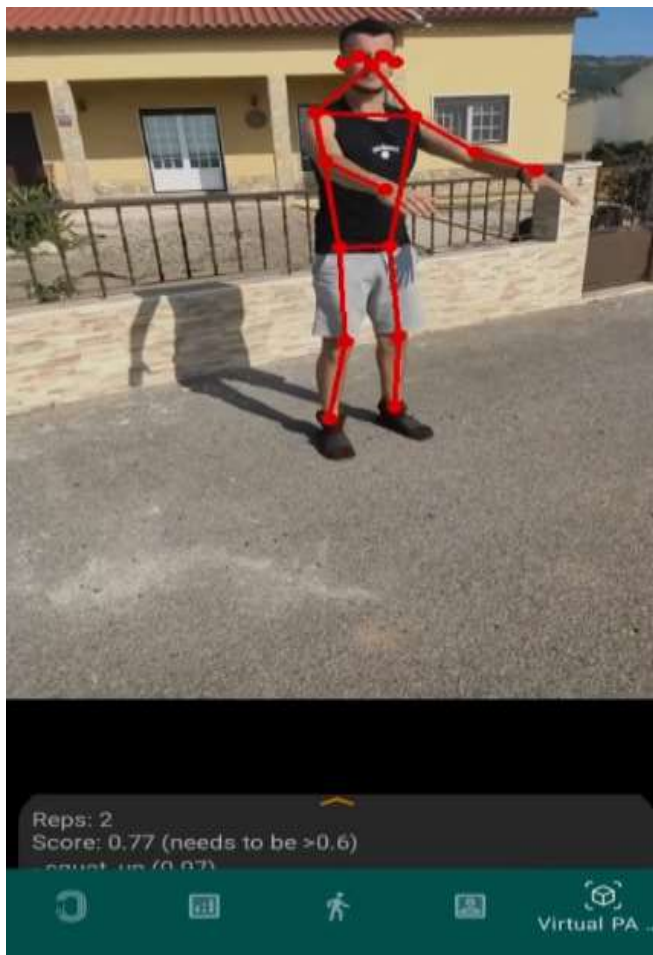


Figura 10: Captura de ecrã do body pose na plataforma mobile

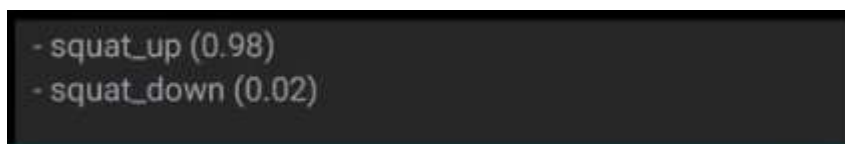


Figura 11: Captura de ecrã do *score* na plataforma mobile

### 3.4.6. Funcionalidades da app web

Na aplicação *Web*, adicionou-se uma opção no menu lateral, denominada “Exercícios”; representa-se nesta página um mapa mundo e uma lista de exercícios.

Como é uma funcionalidade apenas para profissionais de saúde, o mapa representa todos os pontos (exercícios) presentes na base de dados – conforme demonstrado na Figura 12, onde se encontra o mapa mundo na parte superior da página e a tabela na inferior.

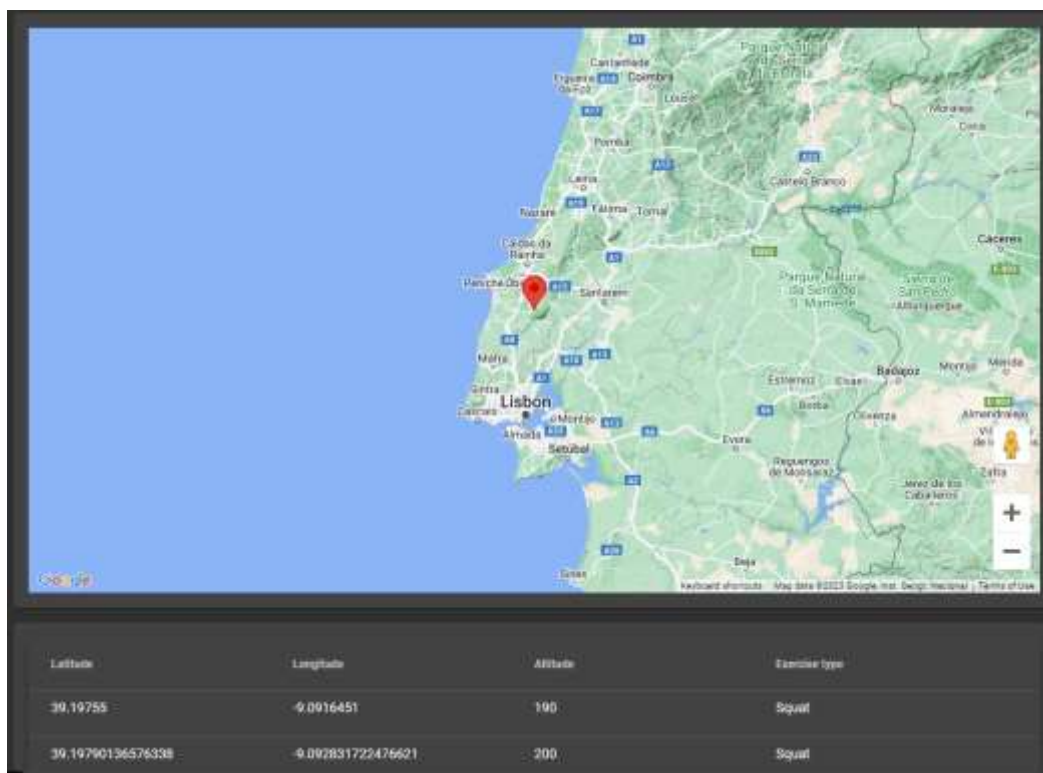


Figura 12: Captura de ecrã do mapa na plataforma web

Para se adicionar um exercício à lista, precisa-se de um clique no mapa; no clique recolhe-se a latitude e longitude, faz-se um pedido à API, posteriormente, para a altitude, pois necessita-se da altitude para adicionar o exercício no AR da aplicação móvel. Depois do clique no mapa, apresenta um *pop up* em que se pede ao utilizador para inserir o tipo de exercício, conforme retratado na Figura 13.

Na adição do exercício é obrigatório colocar os dados: tipo de exercício e a quantidade de repetições, pois estes são utilizados na aplicação móvel.

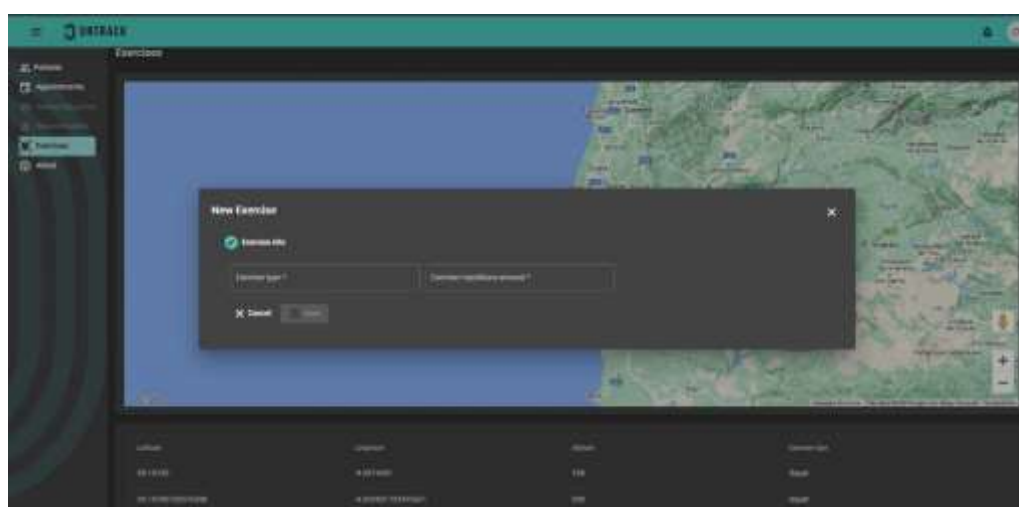


Figura 13: Captura de ecrã adicionar exercício na plataforma web

Mais detalhadamente, chama-se a função *addExercise*, seguidamente ao clique no mapa, existe um modelo pré-definido no código referido na Listagem 4. Representam-se na Listagem 5, as funções que utilizam o modelo. Explicando sucintamente as funções do ficheiro, tem-se a *getARExercises* que consiste em fazer-se um pedido GET à base de dados *user* tabela *ARExercise*, que recebe todos os exercícios na base de dados, transformando-os então num vetor de *ARExerciseModel*; exibe-se na consola uma mensagem quando ocorre um erro. Faz-se um pedido na função *addExercise* para a mesma base de dados e tabela do *getARExercises*; no entanto, esta faz um pedido POST com o exercício que pretende adicionar, demonstra-se, também, uma mensagem na consola se der erro. Já na função *getAltitude* faz-se um pedido direto à API da Google, recebe-se, nessa função, a latitude e longitude como parâmetros, e recebe-se, do pedido à API da Google, a altitude daquela localização, sendo que se ocorrer um erro também este é representado na consola através de uma mensagem. Os restantes ficheiros da parte de exercícios são alguns ajustes ao modelo e UI.

```
export class ARExerciseModel {
  public readonly date: Date;
  public readonly latitude: number
  public readonly longitude: number
  public readonly altitude: number
  public type: string
  public readonly amount: number

  constructor(
    latitude: number,
    longitude: number,
    altitude: number,
    date: Date,
    type: string,
    amount: number
  ) {
    this.date = date
    this.latitude = latitude
    this.longitude = longitude
    this.altitude = altitude
    this.type = type
    this.amount = amount
  }
}
```

Listagem 4: Código modelo ARExercise

```

export interface ARExercise {
  exerciseUrl: string;
  textfile: string;
  date: any;
}

const httpOptions = {
  headers: new HttpHeaders({
    'Content-Type': 'application/json',
    'Access-Control-Allow-Origin': '*'
  })
};

@Injectable()
export class ARExerciseService {
  arExercisesUrl =
`https://${environment.http.url}:${environment.http.port}/api/exercises`;

  constructor(private http: HttpClient, private auth0: AuthService) {
    this.auth0.getAccessTokenSilently().subscribe(token => {
      httpOptions.headers = httpOptions.headers.set('token', `${token}`);
    });
  }

  getARExercises(): Observable<ARExerciseModel[]> {
    console.log('getARExercises');
    return this.http.get<ARExerciseModel[]>(this.arExercisesUrl)
      .pipe(
        catchError(this.handleError) // then handle the error
      );
  }

  addExercise(ARExercise: ARExerciseModel): Observable<any> {
    return this.http.post<ARExerciseModel>(this.arExercisesUrl, ARExercise)
      .pipe(
        catchError(this.handleError)
      );
  }

  getAltitude(lat: number, lng: number) {
    return
    this.http.get<ApiResponse>(`https://maps.googleapis.com/maps/api/elevation/json?l
ocations=${lat}%2C${lng}&key=AIzaSyANV2qm3QdZ1ZTKDhs5fUNagvHjC3-s6ug`).pipe(
      catchError(this.handleError) // then handle the error
    );
  }
}

```

Listagem 5: Código das funções do serviço de exercício

### 3.4.7. Funcionalidades do backend

A secção de *backend*, que incorpora os micro serviços, desempenha a função de registar e disponibilizar dados relacionados com os exercícios (*AR Exercise*). Optou-se por integrá-la com o micro serviço do *http\_get\_measures*, evitando assim a necessidade de criar micro serviços adicionais. Para isso, necessitou-se, também, de incluir no ficheiro *provision.yaml* do *database\_user* código que permitisse a criação automática da tabela de exercícios, a qual inclui latitude, longitude, altitude, tipo de exercício, quantidade, data de criação e se foi realizado previamente ou não.

Quando mencionamos o registo e a disponibilização de dados, este comportamento assemelha-se ao de uma API, onde, ao seguir um determinado caminho, '/api/exercises', através de solicitações POST e GET, efetua-se o registo ou a obtenção de dados.

## 4. Verificação e validação

O processo de verificação e validação da plataforma ocorreu em dois formatos, um deles realizou-se nas reuniões com os professores orientadores e/ou os vários elementos da equipa, no qual estes apresentavam as suas opiniões/sugestões acerca do trabalho realizado e propunham alterações, se necessárias. Fez-se também um tipo de verificação e validação mais formal, sendo este denominado testes de usabilidade.

Testes de usabilidade são testes onde é pedido aos utilizadores para realizarem certas tarefas previamente definidas e após as realizarem, responder a um questionário. No caso deste módulo o escolheu-se o questionário SUS [149], sendo este composto por 10 questões que podem ser identificadas na lista da citação mencionada. Escolheu-se este questionário devido a ter sido utilizado anteriormente no âmbito deste Projeto, pelo aluno Rúben Caceiro [150].

No questionário das tarefas utilizou-se, para cada, uma escala de 7 níveis denominada *Single Ease Question* (SEQ) [151], em que consiste numa simples pergunta, "No geral quão fácil foi esta tarefa", em que o nível 1 é "Muito difícil" e o nível 7 "Muito Fácil".

O público-alvo destes testes são utilizadores com acesso à rede local, visto que houve a limitação do servidor do Politécnico de Leiria. Escolheram-se então 10 utilizadores para responder aos questionários. Sendo um número reduzido de utilizadores não se considerou estatisticamente significativo, no entanto, conseguiu-se um melhor *feedback* da aplicação.

### 4.1. Aplicação Web

Nos testes da aplicação *web* foi dada uma conta com o *login* já realizado na plataforma, na página inicial, e a partir daí foram propostas as três seguintes tarefas:

- Tarefa 1 - Aceder à *tab* de exercícios
- Tarefa 2 - Adicionar um exercício
- Tarefa 3 - Observar informações sobre um exercício em concreto

Na tarefa 1, conseguiu-se concluir que aceder à *tab* de exercícios é algo bastante explícito na aplicação pois todos os utilizadores a consideraram extremamente fácil de realizar, como se pode observar pela Figura 14. Daí concluir-se que não seria necessário adicionar nada para tornar esta tarefa mais fácil.

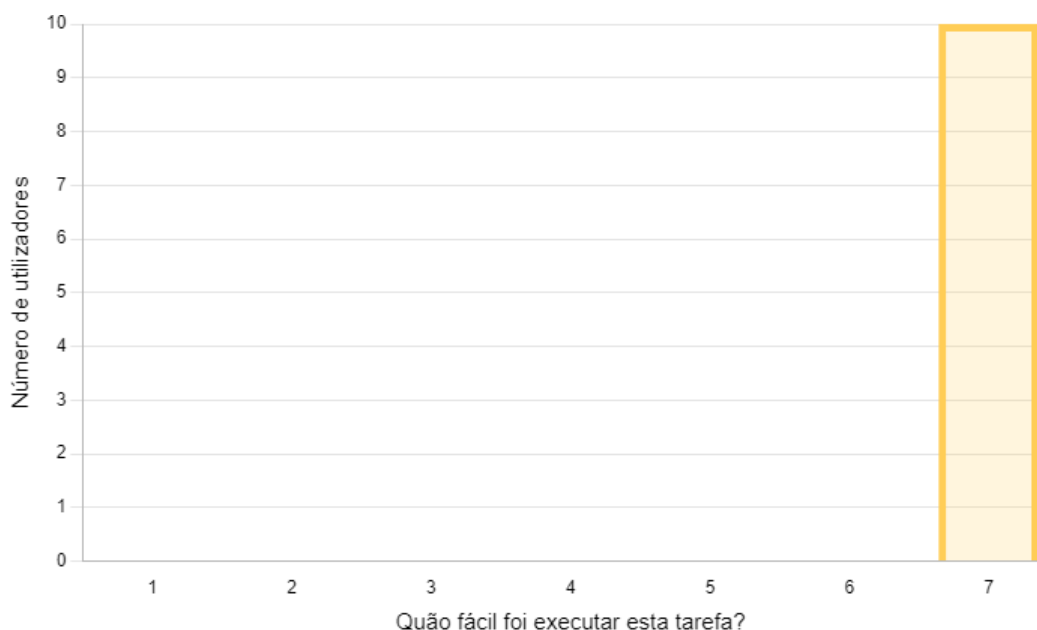


Figura 14: Questionário tarefa 1

Na tarefa 2 os utilizadores continuaram a achar fácil executar o procedimento solicitado, no entanto com alguns comentários, conforme os resultados dispostos na Figura 15.

Os comentários dos utilizadores que votaram 4 e 5 foram no sentido de que era necessário algum *scroll* para fazer o *zoom* indicado, e que o mapa inicializava longe da localização destes mesmos.

Sendo algo que se poderá realizar nos trabalhos futuros deste projeto.

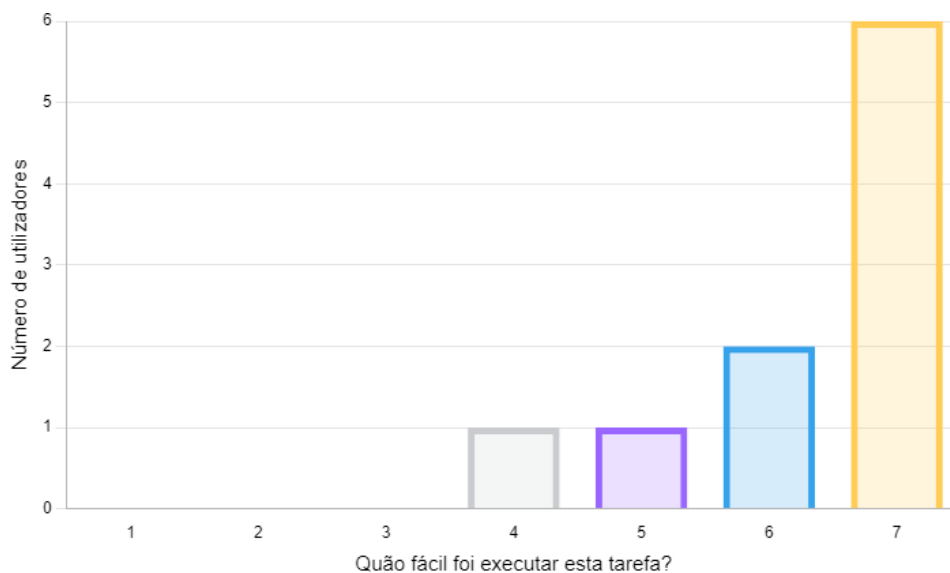
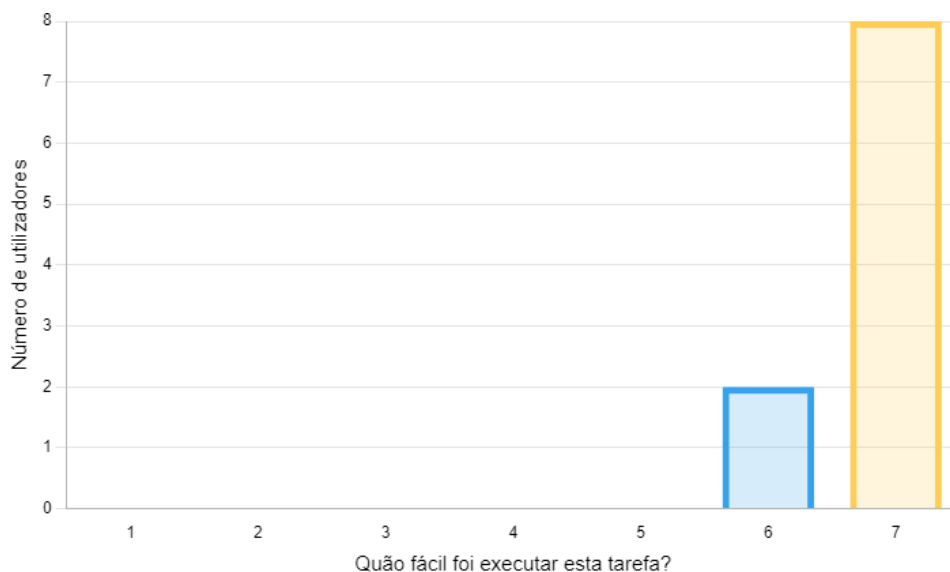


Figura 15: Questionário tarefa 2

Na tarefa 3, e para facilitar o procedimento, foi dito, ao utilizador, que era necessário aceder à *tab* dos exercícios para que este conseguisse observar a informação. No geral foi um resultado positivo, como representado na Figura 16; fez-se uma observação pelo utilizador que selecionou a opção 6, de que se não tivesse a informação inicial de ir para o *tab* de

exercícios, talvez não fosse lá diretamente, procuraria por alguma informação na página inicial. No entanto, não se achou relevante mudar a maneira como é feita a apresentação desta informação.



**Figura 16: Questionário tarefa 3**

O questionário SUS foi feito no final a todos os utilizadores, em que a métrica foi de 1- discordo totalmente e 5 - concordo totalmente.

O questionário SUS [152] gera um único número que representa uma medida composta da usabilidade geral do sistema em estudo. O *score* vai de 0 a 100, sendo que valores mais altos representam melhor usabilidade. Para os itens 1, 3, 5, 7 e 9, a contribuição de pontuação é a posição na escala menos 1. Para os itens 2, 4, 6, 8 e 10, a contribuição é 5 menos a posição na escala. E, por último, multiplica-se a soma das pontuações por 2,5 para obter o valor geral da usabilidade do sistema.

Para concluir a análise aos testes da *web* pode-se referir que no geral, vendo os resultados das várias figuras apresentadas, que é uma aplicação bastante intuitiva, com uma boa *performance* e , tirando o pormenor do mapa, algo fácil de utilizar para um utilizador sem uma experiência imensa nas plataformas *web*.

Pelo questionário SUS consegue-se, também, induzir que os utilizadores não tiveram muitas dificuldades na aplicação *web* pois no geral foi um resultado positivo. Um questionário SUS com uma pontuação acima de 68 é considerado acima da média. Neste caso específico a média foi de 94.

## 4.2. Aplicação móvel

Tal como na aplicação *web*, forneceu-se ao utilizador, antes das tarefas, uma conta com o *login* realizado. Com isto as tarefas pedidas aos utilizadores foram as seguintes.

- Tarefa 1 - Proceder à vista de exercícios

- Tarefa 2 - Observar informações sobre a sua localização
- Tarefa 3 - Observar informações sobre um exercício em concreto
- Tarefa 4 - Começar um exercício
- Tarefa 5 - Realizar um exercício

Na tarefa 1 consegue-se entender que a aplicação é intuitiva, pois todos os utilizadores acharam muito fácil encontrar a vista pretendida, tal como representado na Figura 17. Com isto conclui-se que não é preciso considerar melhorar a navegação dentro do menu principal da *app*.

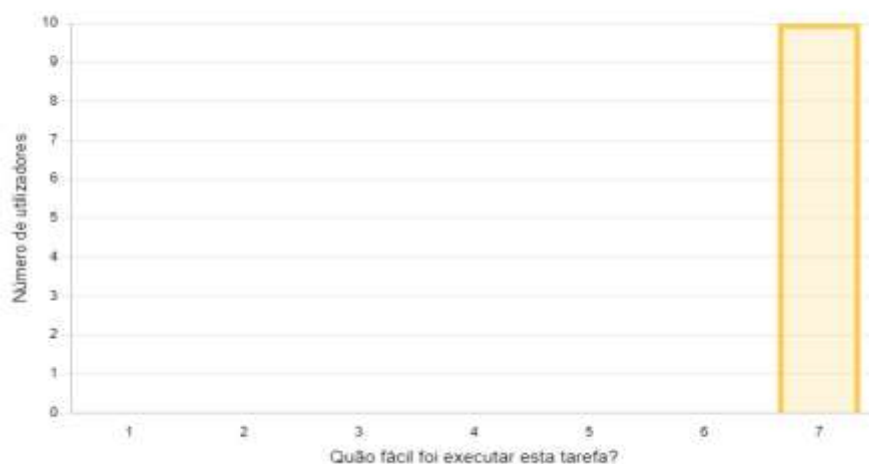


Figura 17: Questionário tarefa 1 aplicação móvel

A tarefa 2 teve, também, um resultado bastante positivo (como representado na Figura 18), no entanto dois dos utilizadores comentaram que se apenas tiverem um exercício prescrito, torna-se difícil, sem informação prévia, saber qual é o marcador da localização ou qual o do exercício. Com isto percebeu-se que talvez se pudesse adicionar uma legenda ao mapa para elucidar todos os utilizadores (algo a fazer no futuro).

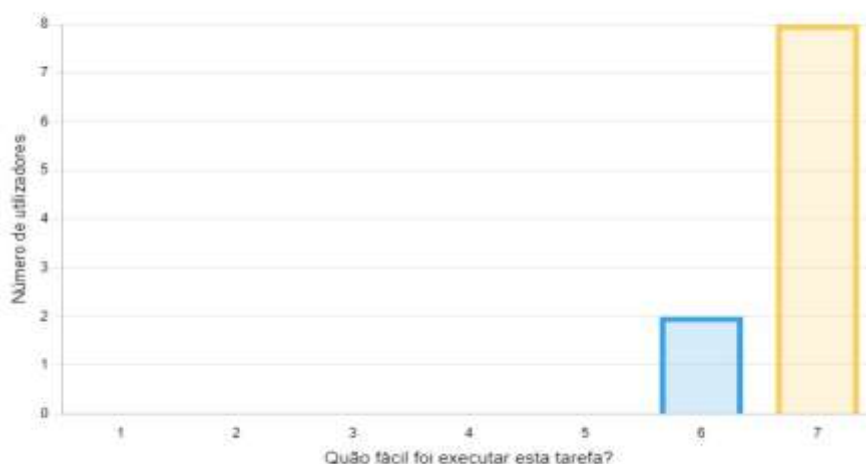


Figura 18: Questionário tarefa 2 aplicação móvel

A tarefa 3, tal como a tarefa 2, teve um resultado positivo, visível na Figura 19, no entanto houve 3 utilizadores que se aperceberam que se apenas tiverem 1 exercício torna-se complicado de distinguir com a localização antes do clique. A solução para esta tarefa é a mesma da tarefa 2.

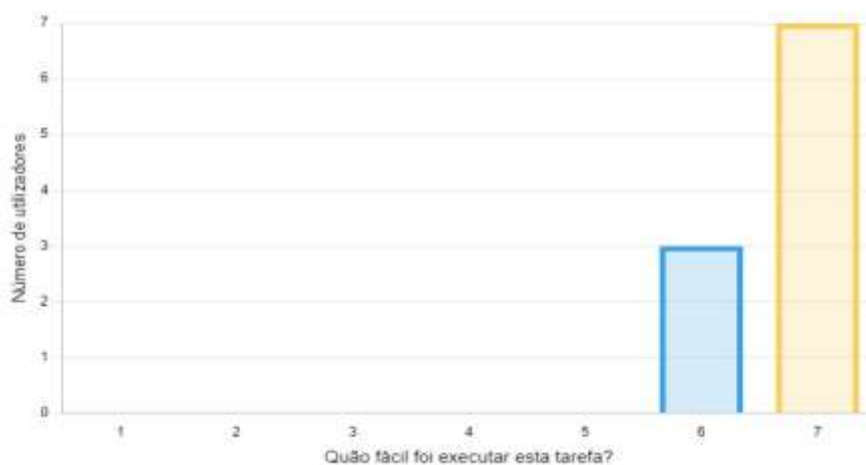


Figura 19: Questionário tarefa 3 aplicação móvel

Na tarefa 4 os resultados já foram um pouco dispersos, como é possível visualizar na Figura 20. As observações desta tarefa foram que por vezes os modelos tridimensionais animados não se encontravam no chão, daí ter de se rodar bastante o telemóvel para conseguir encontrar o /modelo 3D, além disso tem de se estar a uma certa distância para o botão de começar exercício aparecer.

Com isto consegue-se concluir que talvez não seja assim tão intuitivo começar um exercício. A solução que se propõe é, no início da vista, colocar um tutorial para demonstrar a explicação de como iniciar o exercício. No que respeita ao modelo 3D não estar no chão, ou

à altura do utilizador, não se pode resolver apenas no módulo EEV pois a altura é devolvida pela API da Google.

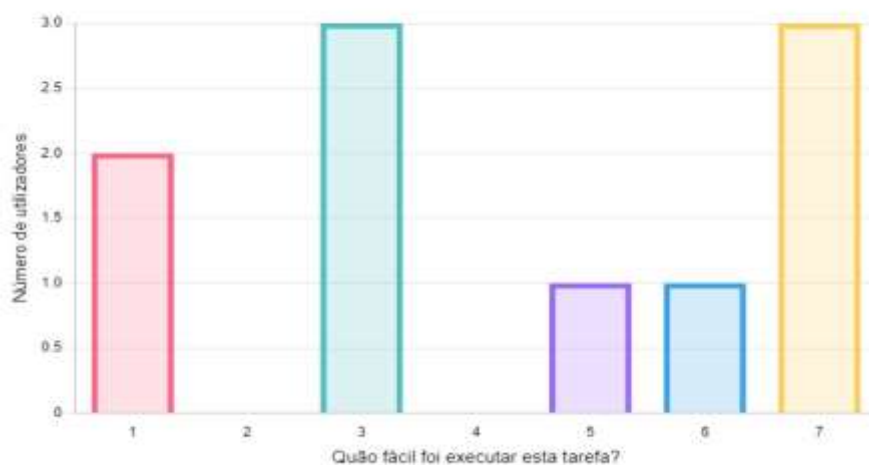


Figura 20: Questionário tarefa 4 aplicação móvel

Na tarefa 5 os utilizadores já se encontravam na parte de *body pose estimation* e os resultados foram positivos, como é possível observar na Figura 21. No entanto, houve observações, sendo estas que algumas das poses tinham de ser feitas de um ângulo em específico para estas serem contadas. Conclui-se então que, para resolver esta situação, teria de se melhorar o modelo utilizado.

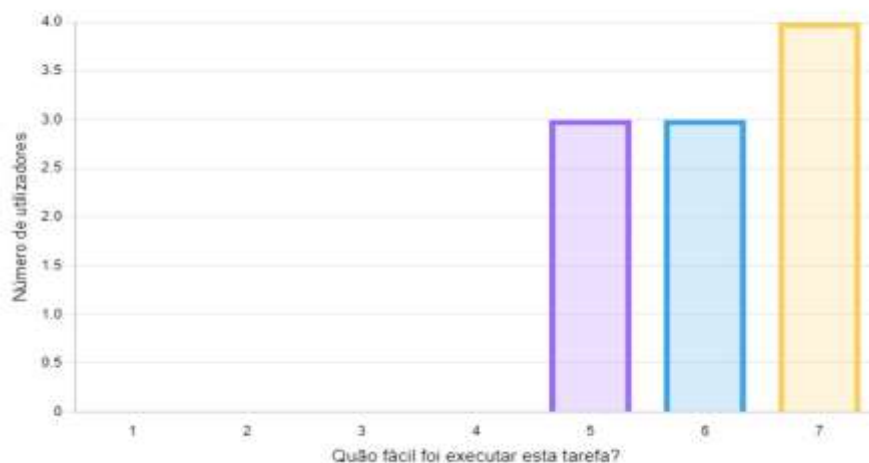


Figura 21: Questionário tarefa 5 aplicação móvel

Tal como para a aplicação *web* na aplicação móvel demonstra-se, na próxima figura, o gráfico para o questionário SUS feito no final a todos os utilizadores, em que, tal como no questionário SUS anterior, 1 é discordo totalmente, e 5 concordo totalmente.

Para concluir a análise aos testes da aplicação móvel pode-se referir que no geral, observando os resultados das várias figuras apresentadas, é uma aplicação intuitiva, no entanto, necessita

de alguns ajustes, principalmente na parte do mapa/ AR. Consegue-se concluir que é de fácil utilização para um utilizador sem grande experiência em aplicações móveis.

Pelo questionário SUS consegue-se, também, induzir que os utilizadores não tiveram muitas dificuldades na aplicação móvel pois no geral foi um resultado positivo, ainda que, em relação à aplicação *web*, menos positivo. A média de resultados foi de 88,5 que é consideravelmente acima da média.

## 5. Conclusão

O projeto OnTRACK, sendo um projeto focado em pacientes com DPOC, oferece uma solução personalizada para aumentar os níveis de atividade física, solução esta que promove a troca de informação entre o profissional de saúde e o paciente.

Realizou-se um sumário sobre o estado de arte para analisar que aplicações das tecnologias na saúde existem e se, as que já feitas, se obtiveram sucesso, de forma a obter uma melhor percepção sobre quais as tecnologias mais relevantes e/ou quais as com maior potencial de dar resultado.

Sendo OnTRACK uma plataforma já previamente definida e estruturada, foi necessário despender algum tempo para aprender a sua arquitetura e qual a melhor maneira de implementar novas *features*. Sendo esta aprendizagem pouco referida no documento, pois achou-se relevante manter este mesmo mais técnico.

Em última análise, no módulo EEJV, as principais contribuições para o projeto OnTRACK foram, a parte de AR desenvolvida em Unity, em que se utilizou os modelos tridimensionais animados, o *body pose estimation* para verificar e validar os exercícios, utilizando o *Colab* para o TFLite e *Kotlin* para a implementação de *mobile*, a utilização do Google Maps para a criação de marcadores (*web*) e para a sua visualização (*web* e *mobile*), sendo estes marcadores, os exercícios tanto *web* como *mobile* e a localização do utilizador, apenas *web*.

A metodologia utilizada, como referido anteriormente, foi a prototipagem evolutiva. Começou-se com *spikes* da aplicação, com funcionalidades separadas para testar, conforme o *feedback*, se algo prosseguia para a aplicação, ou se era necessário refinar a funcionalidade para ir de encontro ao desejado, ou até mesmo não prosseguir para a aplicação, como o caso dos sensores.

A funcionalidade de AR tinha como objetivo inicial de retratar os exercícios, para exemplificar aos pacientes o exercício a fazer, no entanto, no protótipo ficou apenas um modelo *default* para todos os exercícios, com duas instâncias, quando o exercício está concluído e quando está por concluir, sendo a ação “saltar” quando o exercício está por concluir e “correr” quando o exercício está concluído. Esta funcionalidade instancia os modelos conforme as coordenadas devolvidas pela funcionalidade do mapa, pois é guardado no *backend*.

Para os modelos de *body pose*, houve um trabalho prévio de encontrar imagens para fazer um *dataset* fidedigno para a avaliação das poses. Colocou-se o modelo na plataforma *Colab*, todas as imagens separadas em pastas, para se poder fazer o pré-processamento das imagens, treinar a classificação de poses, usando o *MoveNet*, e converter para TFLite, devido ao modelo ter sido criado em *TensorFlow*. Fizeram-se várias versões, principalmente do *dataset*, pois não atingia valores considerados relevantes.

O aspeto visual da funcionalidade móvel, baseou-se bastante no exemplo dado pela própria *TensorFlow*. Tendo informação sobre a pose a fazer, e também qual o *score* necessário mínimo da pose para a contagem aumentar.

Na funcionalidade do mapa, foi usado *Kotlin* com *Jetpack Compose* para *mobile* e *Angular JS* para a *web*. Para esta funcionalidade necessitou-se das API, *places* para demonstrar os pontos de interesse no mapa e também a API de elevação para colocar os modelos tridimensionais animados o mais próximo do chão possível.

Fizeram-se testes de usabilidade para todas estas funcionalidades, em que o resultado SUS foi bastante positivo, tendo uma média de resultados de 88,5 na aplicação *mobile* e na *web* de 94. Ainda que sendo poucos utilizadores para fins estatísticos, conseguiu-se através destes um *feedback* mais imparcial para a aplicação.

Conclui-se assim com a ideia de que as *features* com ML podem ser utilizadas, e com bastante sucesso, na realização de exercício físico. Por outro lado, podemos concluir que os sensores internos do dispositivo móvel, se não usados com sensores externos, têm uma margem de erro bastante grande. Conseguiu-se também perceber que, cada vez mais, o ML está a ser utilizado em conjunto com aplicações relacionadas com saúde, e a ter cada vez mais sucesso.

O desejado para trabalhos futuros, e começando pela parte de AR, seria a utilização de objetos 3D com os exercícios em questão, e não apenas um modelo 3D animado *default*. Adicionar também um modelo 3D para quando o exercício está completo seria outro aspeto útil.

Na parte de *body pose estimation*, seria importante adicionar mais exercícios e talvez ter melhores modelos.

No mapa, ao invés de ter apenas as localizações dos exercícios, seria útil ter também uma opção para selecionar a localização do exercício e obter as direções, da localização do utilizador até ao exercício, para este conseguir iniciar o exercício.

Relativamente à aplicação *Web*, seria importante reconfigurar o zoom inicial do mapa, melhorar a estabilidade, principalmente do *backend*, e dar *refresh* automático quando adicionado um novo exercício.

Na vertente de segurança e privacidade, há oportunidade substancial para melhorias. Nomeadamente, recomenda-se a inclusão de medidas adicionais, como a utilização de uma chave de API ou um mecanismo de autenticação, para garantir o acesso autorizado à API no *backend*.

Estas salvaguardas são essenciais para reforçar a segurança do sistema e proteger a privacidade dos utilizadores, mitigando potenciais riscos associados à manipulação indevida de dados e acessos não autorizados.

A implementação destas medidas alinha-se com as melhores práticas de segurança da informação e contribui significativamente para a integridade e confidencialidade dos dados manipulados pela aplicação.

Por fim, referir que o objetivo de criar estas *features* foi concluído com sucesso, no entanto, ainda com pormenores para melhorar a experiência do utilizador.

## 6. Bibliografia

- [1] ‘Chronic obstructive pulmonary disease’. [Online]. Available: <https://platform.who.int/mortality/themes/theme-details/topics/indicator-groups/indicator-group-details/MDB/chronic-obstructive-pulmonary-disease>
- [2] ‘2020 Gold Reports - Global Initiative for Chronic Obstructive Lung Disease - GOLD’. Accessed: Mar. 07, 2024. [Online]. Available: <https://goldcopd.org/gold-reports/>
- [3] Y. Murata *et al.*, ‘Computed Tomography Lung Density Analysis: An Imaging Biomarker Predicting Physical Inactivity in Chronic Obstructive Pulmonary Disease: A Pilot Study’, *J Clin Med*, vol. 12, no. 8, 2023, doi: 10.3390/jcm12082959.
- [4] ‘Pose estimation | TensorFlow Lite’. [Online]. Available: [https://www.tensorflow.org/lite/examples/pose\\_estimation/overview](https://www.tensorflow.org/lite/examples/pose_estimation/overview)
- [5] ‘Build global-scale, immersive, location-based AR experiences with the ARCore Geospatial API | Google Developers’. [Online]. Available: <https://developers.google.com/ar/develop/geospatial>
- [6] ‘Location and Context APIs | Google for Developers’. [Online]. Available: <https://developers.google.com/location-context>
- [7] S. K. Chaurasia and S. R. N. Reddy, ‘State-of-the-art survey on activity recognition and classification using smartphones and wearable sensors’, *Multimedia Tools and Applications 2021 81:1*, vol. 81, no. 1, pp. 1077–1108, Jan. 2021, doi: 10.1007/S11042-021-11410-0.
- [8] M. Rabbi, A. Pfammatter, I. Zhang, B. Spring, and T. Choudhury, ‘Automated Personalized Feedback for Physical Activity and Dietary Behavior Change With Mobile Phones: A Randomized Controlled Trial on Adults’, *JMIR Mhealth Uhealth* 2015;3(2):e42 <https://mhealth.jmir.org/2015/2/e42>, vol. 3, no. 2, p. e4160, Jan. 2015, doi: 10.2196/MHEALTH.4160.
- [9] M. Zhou *et al.*, ‘Evaluating Machine Learning–Based Automated Personalized Daily Step Goals Delivered Through a Mobile Phone App: Randomized Controlled Trial’, *JMIR Mhealth Uhealth* 2018;6(1):e28 <https://mhealth.jmir.org/2018/1/e28>, vol. 6, no. 1, p. e9117, Jan. 2018, doi: 10.2196/MHEALTH.9117.

- [10] ‘Tailored Daily Activity: An Adaptive Physical Activity Smartphone Intervention’, *https://home.liebertpub.com/tmj*, vol. 26, no. 4, pp. 426–437, Jan. 2020, doi: 10.1089/TMJ.2019.0034.
- [11] D. and V. K. and V. E. and M. R. and F. H. Ugulino Wallace and Cardador, ‘Wearable Computing: Accelerometers’ Data Classification of Body Postures and Movements’, *Advances in Artificial Intelligence - SBIA 2012*, pp. 52–61, 2012.
- [12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, ‘Activity recognition using cell phone accelerometers’, *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, Jan. 2011, doi: 10.1145/1964897.1964918.
- [13] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, ‘Transition-Aware Human Activity Recognition Using Smartphones’, *Neurocomputing*, vol. 171, pp. 754–767, 2016, doi: <https://doi.org/10.1016/j.neucom.2015.07.085>.
- [14] Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, and Y. Liu, ‘Towards unsupervised physical activity recognition using smartphone accelerometers’, *Multimed Tools Appl*, vol. 76, no. 8, pp. 10701–10719, Jan. 2017, doi: 10.1007/S11042-015-3188-Y/TABLES/4.
- [15] A. P. Dempster, N. M. Laird, and D. B. Rubin, ‘Maximum Likelihood from Incomplete Data Via the EM Algorithm’, *J R Stat Soc Series B Stat Methodol*, vol. 39, no. 1, pp. 1–22, Sep. 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.
- [16] W. Revelle, ‘Hierarchical Cluster Analysis And The Internal Structure Of Tests.’, *Multivariate Behav Res*, vol. 14 1, pp. 57–74, 1979.
- [17] D. Arthur and S. Vassilvitskii, ‘K-Means++: The Advantages of Careful Seeding’, *Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms*, vol. 8, pp. 1027–1035, Jan. 2007, doi: 10.1145/1283383.1283494.
- [18] L. Kaufmann and P. Rousseeuw, ‘Clustering by Means of Medoids’, *Data Analysis based on the L1-Norm and Related Methods*, pp. 405–416, Jan. 1987.
- [19] U. Von Luxburg, ‘A Tutorial on Spectral Clustering’, *Stat Comput*, vol. 17, no. 4, pp. 395–416, Jan. 2007, doi: 10.48550/arxiv.0711.0189.
- [20] C. Avilés-Cruz, A. Ferreyra-Ramírez, A. Zúñiga-López, and J. Villegas-Cortéz, ‘Coarse-Fine Convolutional Deep-Learning Strategy for Human Activity Recognition’, *Sensors 2019, Vol. 19, Page 1556*, vol. 19, no. 7, p. 1556, Jan. 2019, doi: 10.3390/S19071556.
- [21] T. Y.-H. Chen, A. Sivaraman, S. Das, L. Ravindranath, and H. Balakrishnan, ‘Designing a Context-Sensitive Context Detection Service for Mobile

- Devices’, Jan. 2015, [Online]. Available: <https://dspace.mit.edu/handle/1721.1/98905>
- [22] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, ‘A robust human activity recognition system using smartphone sensors and deep learning’, *Future Generation Computer Systems*, vol. 81, pp. 307–313, Jan. 2018, doi: 10.1016/J.FUTURE.2017.11.029.
- [23] ‘UCI Machine Learning Repository’. [Online]. Available: <http://archive.ics.uci.edu/ml/index.php>
- [24] L. Barrett, ‘Home and Community Preferences of the 45+ Population 2014’, Jan. 2015, doi: 10.26419/RES.00105.001.
- [25] A. Moller *et al.*, ‘GymSkill: A personal trainer for physical exercises’, *2012 IEEE International Conference on Pervasive Computing and Communications, PerCom 2012*, pp. 213–220, 2012, doi: 10.1109/PERCOM.2012.6199869.
- [26] Y. Chen, Y. Tian, and M. He, ‘Monocular human pose estimation: A survey of deep learning-based methods’, *Computer Vision and Image Understanding*, vol. 192, p. 102897, Jan. 2020, doi: 10.1016/J.CVIU.2019.102897.
- [27] A. Toshev and C. Szegedy, ‘DeepPose: Human pose estimation via deep neural networks’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, Jan. 2014, doi: 10.1109/CVPR.2014.214.
- [28] B. Sapp and B. Taskar, ‘MODEC: Multimodal decomposable models for human pose estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3674–3681, 2013, doi: 10.1109/CVPR.2013.471.
- [29] S. Johnson and M. Everingham, ‘Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation’, in *Proceedings of the British Machine Vision Conference 2010*, British Machine Vision Association, 2010, pp. 12.1-12.11. doi: 10.5244/C.24.12.
- [30] S. Johnson and M. Everingham, ‘Learning effective human pose estimation from inaccurate annotation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1465–1472, 2011, doi: 10.1109/CVPR.2011.5995318.
- [31] D. Ramanan, ‘Learning to parse images of articulated bodies’, *Adv Neural Inf Process Syst*, vol. 19, 2006, [Online]. Available: <https://proceedings.neurips.cc/paper/2006/file/a209ca7b50dcaab2db7c2d4d1223d4d5-Paper.pdf>

- [32] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, ‘Human pose estimation with iterative error feedback’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 4733–4742, Jan. 2016, doi: 10.1109/CVPR.2016.512.
- [33] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, ‘2D human pose estimation: New benchmark and state of the art analysis’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3686–3693, Jan. 2014, doi: 10.1109/CVPR.2014.471.
- [34] X. Sun, J. Shang, S. Liang, and Y. Wei, ‘Compositional Human Pose Regression’, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 2621–2630, Jan. 2017, doi: 10.1109/ICCV.2017.284.
- [35] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, ‘Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments’, *IEEE Trans Pattern Anal Mach Intell*, vol. 36, no. 7, pp. 1325–1339, 2014, doi: 10.1109/TPAMI.2013.248.
- [36] D. C. Luvizon, H. Tabia, and D. Picard, ‘Human pose regression by combining indirect part detection and contextual information’, *Comput Graph*, vol. 85, pp. 15–22, Jan. 2019, doi: 10.1016/J.CAG.2019.09.002.
- [37] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, ‘Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation’, *CoRR*, vol. abs/1406.2984, 2014, [Online]. Available: <http://arxiv.org/abs/1406.2984>
- [38] W. Yang, W. Ouyang, H. Li, and X. Wang, ‘End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 3073–3082, Jan. 2016, doi: 10.1109/CVPR.2016.335.
- [39] K. Simonyan and A. Zisserman, ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’, *CoRR*, vol. abs/1409.1556, 2014.
- [40] A. Newell, K. Yang, and J. Deng, ‘Stacked Hourglass Networks for Human Pose Estimation’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9912 LNCS, pp. 483–499, Jan. 2016, doi: 10.48550/arxiv.1603.06937.
- [41] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, ‘Convolutional Pose Machines’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 4724–4732, Jan. 2016, doi: 10.48550/arxiv.1602.00134.

- [42] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, ‘Multi-context attention for human pose estimation’, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 5669–5678, Jan. 2017, doi: 10.1109/CVPR.2017.601.
- [43] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, ‘Learning Feature Pyramids for Human Pose Estimation’, *CoRR*, vol. abs/1708.01101, 2017, [Online]. Available: <http://arxiv.org/abs/1708.01101>
- [44] Y. Chen, C. Shen, X. S. Wei, L. Liu, and J. Yang, ‘Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation’, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 1221–1230, Jan. 2017, doi: 10.1109/ICCV.2017.137.
- [45] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas, ‘Jointly Optimize Data Augmentation and Network Training: Adversarial Data Augmentation in Human Pose Estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2226–2234, Jan. 2018, doi: 10.1109/CVPR.2018.00237.
- [46] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, ‘Multi-Scale Structure-Aware Network for Human Pose Estimation’, *CoRR*, vol. abs/1803.09894, 2018, [Online]. Available: <http://arxiv.org/abs/1803.09894>
- [47] W. Tang, P. Yu, and Y. Wu, ‘Deeply Learned Compositional Models for Human Pose Estimation’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11207 LNCS, pp. 197–214, 2018, doi: 10.1007/978-3-030-01219-9\_12/FIGURES/8.
- [48] K. Sun, B. Xiao, D. Liu, and J. Wang, ‘Deep high-resolution representation learning for human pose estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 5686–5696, Jan. 2019, doi: 10.1109/CVPR.2019.00584.
- [49] W. Tang and Y. Wu, ‘Does learning specific features for related parts help human pose estimation?’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 1107–1116, Jan. 2019, doi: 10.1109/CVPR.2019.00120.
- [50] U. Iqbal and J. Gall, ‘Multi-person pose estimation with local joint-to-person associations’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9914 LNCS, pp. 627–642, 2016, doi: 10.1007/978-3-319-48881-3\_44/FIGURES/7.
- [51] H. S. Fang, S. Xie, Y. W. Tai, and C. Lu, ‘RMPE: Regional Multi-person Pose Estimation’, *Proceedings of the IEEE International Conference on Computer*

- Vision*, vol. 2017-October, pp. 2353–2362, Jan. 2017, doi: 10.1109/ICCV.2017.256.
- [52] T. Y. Lin *et al.*, ‘Microsoft COCO: Common objects in context’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi: 10.1007/978-3-319-10602-1\_48/COVER.
- [53] G. Papandreou *et al.*, ‘Towards Accurate Multi-person Pose Estimation in the Wild’, *CoRR*, vol. abs/1701.01779, 2017, [Online]. Available: <http://arxiv.org/abs/1701.01779>
- [54] S. Huang, M. Gong, and D. Tao, ‘A Coarse-Fine Network for Keypoint Localization’, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 3047–3056, Jan. 2017, doi: 10.1109/ICCV.2017.329.
- [55] K. He, G. Gkioxari, P. Dollar, and R. Girshick, ‘Mask R-CNN’, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 2980–2988, Jan. 2017, doi: 10.1109/ICCV.2017.322.
- [56] B. Xiao, H. Wu, and Y. Wei, ‘Simple Baselines for Human Pose Estimation and Tracking’, *CoRR*, vol. abs/1804.06208, 2018, [Online]. Available: <http://arxiv.org/abs/1804.06208>
- [57] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep residual learning for image recognition’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, Jan. 2016, doi: 10.1109/CVPR.2016.90.
- [58] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, ‘Cascaded Pyramid Network for Multi-person Pose Estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7103–7112, Jan. 2018, doi: 10.1109/CVPR.2018.00742.
- [59] G. Moon, J. Y. Chang, and K. M. Lee, ‘PoseFix: Model-agnostic General Human Pose Refinement Network’, *CoRR*, vol. abs/1812.03595, 2018, [Online]. Available: <http://arxiv.org/abs/1812.03595>
- [60] L. Pishchulin *et al.*, ‘DeepCut: Joint subset partition and labeling for multi person pose estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 4929–4937, Jan. 2016, doi: 10.1109/CVPR.2016.533.
- [61] M. Eichner and V. Ferrari, ‘We are family: Joint pose estimation of multiple persons’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6311

- LNCS, no. PART 1, pp. 228–242, 2010, doi: 10.1007/978-3-642-15549-9\_17/COVER.
- [62] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, ‘Deepercut: A deeper, stronger, and faster multi-person pose estimation model’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9910 LNCS, pp. 34–50, 2016, doi: 10.1007/978-3-319-46466-4\_3/TABLES/9.
- [63] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, ‘Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields’, *CoRR*, vol. abs/1611.08050, 2016, [Online]. Available: <http://arxiv.org/abs/1611.08050>
- [64] A. Newell and J. Deng, ‘Associative Embedding: End-to-End Learning for Joint Detection and Grouping’, *CoRR*, vol. abs/1611.05424, 2016, [Online]. Available: <http://arxiv.org/abs/1611.05424>
- [65] X. Nie, J. Feng, J. Xing, and S. Yan, ‘Pose Partition Networks for Multi-person Pose Estimation’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11209 LNCS, pp. 705–720, 2018, doi: 10.1007/978-3-030-01228-1\_42/FIGURES/4.
- [66] F. Xia, P. Wang, X. Chen, and A. Yuille, ‘Joint multi-person pose estimation and semantic part segmentation’, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6080–6089, Jan. 2017, doi: 10.1109/CVPR.2017.644.
- [67] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, ‘PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model’, *CoRR*, vol. abs/1803.08225, 2018, [Online]. Available: <http://arxiv.org/abs/1803.08225>
- [68] M. Kocabas, S. Karagoz, and E. Akbas, ‘MultiPoseNet: Fast Multi-Person Pose Estimation Using Pose Residual Network’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11215 LNCS, pp. 437–453, 2018, doi: 10.1007/978-3-030-01252-6\_26/FIGURES/6.
- [69] S. Kreiss, L. Bertoni, and A. Alahi, ‘PifPaf: Composite fields for human pose estimation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 11969–11978, Jan. 2019, doi: 10.1109/CVPR.2019.01225.
- [70] S. Li and A. B. Chan, ‘3D human pose estimation from monocular images with deep convolutional neural network’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes*

- in Bioinformatics*), vol. 9004, pp. 332–347, 2015, doi: 10.1007/978-3-319-16808-1\_23/COVER.
- [71] S. Li, W. Zhang, and A. B. Chan, ‘Maximum-Margin Structured Learning with Deep Networks for 3D Human Pose Estimation’, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2848–2856, 2015, doi: 10.1109/ICCV.2015.326.
- [72] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, ‘Structured Prediction of 3D Human Pose with Deep Neural Networks’, *CoRR*, vol. abs/1605.05180, 2016, [Online]. Available: <http://arxiv.org/abs/1605.05180>
- [73] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua, ‘Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation’, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 3961–3970, Jan. 2017, doi: 10.1109/ICCV.2017.425.
- [74] C.-H. Chen and D. Ramanan, ‘3D Human Pose Estimation = 2D Pose Estimation + Matching’, *CoRR*, vol. abs/1612.06524, 2016, [Online]. Available: <http://arxiv.org/abs/1612.06524>
- [75] F. Moreno-Noguer, ‘3D Human Pose Estimation from a Single Image via Distance Matrix Regression’, *CoRR*, vol. abs/1611.09010, 2016, [Online]. Available: <http://arxiv.org/abs/1611.09010>
- [76] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, ‘Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose’, *CoRR*, vol. abs/1611.07828, 2016, [Online]. Available: <http://arxiv.org/abs/1611.07828>
- [77] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, ‘Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach’, Jan. 2017, [Online]. Available: <http://arxiv.org/abs/1704.02447>
- [78] J. Martinez, R. Hossain, J. Romero, and J. J. Little, ‘A simple yet effective baseline for 3d human pose estimation’, *CoRR*, vol. abs/1705.03098, 2017, [Online]. Available: <http://arxiv.org/abs/1705.03098>
- [79] W. Yang, W. Ouyang, X. Wang, J. S. J. Ren, H. Li, and X. Wang, ‘3D Human Pose Estimation in the Wild by Adversarial Learning’, *CoRR*, vol. abs/1803.09722, 2018, [Online]. Available: <http://arxiv.org/abs/1803.09722>
- [80] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, ‘Learning to Estimate 3D Human Pose and Shape from a Single Color Image’, *CoRR*, vol. abs/1805.04092, 2018, [Online]. Available: <http://arxiv.org/abs/1805.04092>
- [81] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, ‘Integral human pose regression’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol.

- 11210 LNCS, pp. 536–553, 2018, doi: 10.1007/978-3-030-01231-1\_33/TABLES/11.
- [82] C. Li and G. H. Lee, ‘Generating multiple hypotheses for 3D human pose estimation with mixture density network’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9879–9887, Jan. 2019, doi: 10.1109/CVPR.2019.01012.
- [83] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, ‘Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9909 LNCS, pp. 561–578, 2016, doi: 10.1007/978-3-319-46454-1\_34/FIGURES/8.
- [84] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, ‘Deep Kinematic Pose Regression’, *CoRR*, vol. abs/1609.05317, 2016, [Online]. Available: <http://arxiv.org/abs/1609.05317>
- [85] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt, ‘Monocular 3D Human Pose Estimation Using Transfer Learning and Improved CNN Supervision’, *CoRR*, vol. abs/1611.09813, 2016, [Online]. Available: <http://arxiv.org/abs/1611.09813>
- [86] B. X. Nie, P. Wei, and S. C. Zhu, ‘Monocular 3D Human Pose Estimation by Predicting Depth on Joints’, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 3467–3475, Jan. 2017, doi: 10.1109/ICCV.2017.373.
- [87] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, ‘End-to-end Recovery of Human Shape and Pose’, *CoRR*, vol. abs/1712.06584, 2017, [Online]. Available: <http://arxiv.org/abs/1712.06584>
- [88] G. Pavlakos, X. Zhou, and K. Daniilidis, ‘Ordinal Depth Supervision for 3D Human Pose Estimation’, *CoRR*, vol. abs/1805.04095, 2018, [Online]. Available: <http://arxiv.org/abs/1805.04095>
- [89] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, ‘Neural body fitting: Unifying deep learning and model based human pose and shape estimation’, *Proceedings - 2018 International Conference on 3D Vision, 3DV 2018*, pp. 484–494, Jan. 2018, doi: 10.1109/3DV.2018.00062.
- [90] G. Varol *et al.*, ‘BodyNet: Volumetric Inference of 3D Human Body Shapes’, *CoRR*, vol. abs/1804.04875, 2018, [Online]. Available: <http://arxiv.org/abs/1804.04875>
- [91] G. Varol *et al.*, ‘Learning from Synthetic Humans’, *CoRR*, vol. abs/1701.01370, 2017, [Online]. Available: <http://arxiv.org/abs/1701.01370>

- [92] A. Arnab, C. Doersch, and A. Zisserman, ‘Exploiting temporal context for 3D human pose estimation in the wild’, *CoRR*, vol. abs/1905.04266, 2019, [Online]. Available: <http://arxiv.org/abs/1905.04266>
- [93] D. Tomè, C. Russell, and L. Agapito, ‘Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image’, *CoRR*, vol. abs/1701.00295, 2017, [Online]. Available: <http://arxiv.org/abs/1701.00295>
- [94] H. Rhodin, M. Salzmann, and P. Fua, ‘Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation’, *CoRR*, vol. abs/1804.01110, 2018, [Online]. Available: <http://arxiv.org/abs/1804.01110>
- [95] D. Mehta *et al.*, ‘Single-Shot Multi-Person 3D Body Pose Estimation From Monocular RGB Input’, *CoRR*, vol. abs/1712.03453, 2017, [Online]. Available: <http://arxiv.org/abs/1712.03453>
- [96] G. Rogez, P. Weinzaepfel, and C. Schmid, ‘LCR-Net: Localization-classification-regression for human pose’, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 1216–1224, Jan. 2017, doi: 10.1109/CVPR.2017.134.
- [97] A. Zanfir, E. Marinoiu, and C. Sminchisescu, ‘Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes: The Importance of Multiple Scene Constraints’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2148–2157, Jan. 2018, doi: 10.1109/CVPR.2018.00229.
- [98] H. Joo *et al.*, ‘Panoptic Studio: A Massively Multiview System for Social Interaction Capture’, *CoRR*, vol. abs/1612.03153, 2016, [Online]. Available: <http://arxiv.org/abs/1612.03153>
- [99] D. Mehta *et al.*, ‘XNect: Real-time Multi-person 3D Human Pose Estimation with a Single RGB Camera’, *CoRR*, vol. abs/1907.00837, 2019, [Online]. Available: <http://arxiv.org/abs/1907.00837>
- [100] D. Mehta *et al.*, ‘VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera’, *CoRR*, vol. abs/1705.01583, 2017, [Online]. Available: <http://arxiv.org/abs/1705.01583>
- [101] B. Ferreira *et al.*, ‘Deep learning approaches for workout repetition counting and validation’, *Pattern Recognit Lett*, vol. 151, pp. 259–266, Jan. 2021, doi: 10.1016/J.PATREC.2021.09.006.
- [102] ‘Pokémon GO - Apps on Google Play’. [Online]. Available: <https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo&gl=us>

- [103] I. Odenigbo, J. K. Reen, C. Eneze, A. Friday, and R. Orji, ‘The Journey: An AR Gamified Mobile Application for Promoting Physical Activity in Young Adults’, Jan. 2022, doi: 10.1145/3511047.3537652.
- [104] P. Soltani and A. H. P. Morice, ‘Augmented reality tools for sports education and training’, *Comput Educ*, vol. 155, p. 103923, 2020, doi: <https://doi.org/10.1016/j.compedu.2020.103923>.
- [105] A. Godbout and J. E. Boyd, ‘Rhythmic sonic feedback for speed skating by real-time movement synchronization’, *Int J Comput Sci Sport*, vol. 11, pp. 37–51, Jan. 2012.
- [106] Y. Itoh, J. Orlosky, K. Kiyokawa, and G. Klinker, ‘Laplacian vision: Augmenting motion prediction via optical see- Through head-mounted displays’, *ACM International Conference Proceeding Series*, vol. 25-27-February-2016, Jan. 2016, doi: 10.1145/2875194.2875227.
- [107] T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland, ‘Stochasticks: Augmenting the billiards experience with probabilistic vision and wearable computers’, *International Symposium on Wearable Computers, Digest of Papers*, pp. 138–145, 1997, doi: 10.1109/ISWC.1997.629930.
- [108] R. Kajastila, L. Holsti, and P. Hamalainen, ‘The augmented Climbing wall: High-exertion proximity interaction on a wall-sized interactive surface’, *Conference on Human Factors in Computing Systems - Proceedings*, pp. 758–769, Jan. 2016, doi: 10.1145/2858036.2858450.
- [109] V. Kálmán, C. Baczó, M. Livadas, and T. Csielka, ‘Wearable technology to help with visual challenges-two case studies’, *Stud Health Technol Inform*, vol. 217, pp. 526–532, 2015, doi: 10.3233/978-1-61499-566-1-526.
- [110] P. Kelly and N. E. O’Connor, ‘Visualisation of tennis swings for coaching’, *International Workshop on Image Analysis for Multimedia Interactive Services*, 2012, doi: 10.1109/WIAMIS.2012.6226750.
- [111] F. F. Mueller, L. Cole, S. O’Brien, and W. Walmink, ‘Airhockey over a distance: A networked physical game to support social interactions’, *International Conference on Advances in Computer Entertainment Technology 2006*, 2006, doi: 10.1145/1178823.1178906.
- [112] K. Rogers *et al.*, ‘KickAR: Exploring game balancing through boosts and handicaps in augmented reality table football’, *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, Jan. 2018, doi: 10.1145/3173574.3173740.

- [113] M. Page and A. Vande Moere, ‘Evaluating a Wearable Display Jersey for Augmenting Team Sports Awareness’, *Proceedings of the 5th International Conference on Pervasive Computing*, pp. 91–108, 2007.
- [114] Y. Sano, K. Sato, R. Shiraishi, and M. Otsuki, ‘Sports Support System: Augmented Ball Game for Filling Gap between Player Skill Levels’, *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, pp. 361–366, 2016, doi: 10.1145/2992154.2996781.
- [115] C. Sieluzycki, P. Kaczmarczyk, J. Sobiecki, K. Witkowski, J. Maslinski, and W. Cieslinski, ‘Microsoft kinect as a tool to support training in professional sports: Augmented reality application to tachi-waza techniques in judo’, *Proceedings - 2016 3rd European Network Intelligence Conference, ENIC 2016*, pp. 153–158, 2016, doi: 10.1109/ENIC.2016.030.
- [116] R. Sodhi, H. Benko, and A. D. Wilson, ‘LightGuide: Projected visualizations for hand movement guidance’, *Conference on Human Factors in Computing Systems - Proceedings*, pp. 179–188, 2012, doi: 10.1145/2207676.2207702.
- [117] G. Sörös, F. Daiber, and T. Weller, ‘Cyclo: A Personal Bike Coach through the Glass’, *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, 2013, doi: 10.1145/2543651.2543660.
- [118] F. Wiehr, F. Daiber, F. Kosmalla, and A. Krüger, ‘BetaCube - Enhancing training for climbing by a self-calibrating camera-projection unit’, *Conference on Human Factors in Computing Systems - Proceedings*, vol. 07-12-May-2016, pp. 1998–2004, Jan. 2016, doi: 10.1145/2851581.2892393.
- [119] K. Witkowski, J. Sobiecki, J. Maslinski, W. Cieslinski, A. Rokita, and R. M. Kalina, ‘The use of augmented-reality technology to improve judo techniques. Premises, assumptions, methodology, research tools, preliminary scenarios – the first stage of the study’, *Archives of Budo*, vol. 12, 2016.
- [120] Y.-L. Ng, F. Ma, F. K. Ho, P. Ip, and K. Fu, ‘Effectiveness of virtual and augmented reality-enhanced exercise on physical activity, psychological outcomes, and physical performance: A systematic review and meta-analysis of randomized controlled trials’, *Comput Human Behav*, vol. 99, pp. 278–291, 2019, doi: <https://doi.org/10.1016/j.chb.2019.05.026>.
- [121] ‘Gamification Definition & Meaning - Merriam-Webster’. [Online]. Available: <https://www.merriam-webster.com/dictionary/gamification>
- [122] J. Lee, H. Yoo, and B.-H. Lee, ‘Effects of augmented reality-based Otago exercise on balance, gait, and physical factors in elderly women to prevent falls: a randomized controlled trial’, *J Phys Ther Sci*, vol. 29, no. 9, pp. 1586–1589, 2017, doi: 10.1589/jpts.29.1586.

- [123] ‘Morse Fall Risk Scale - NurseAssess by Nurse Tech, Inc.’ [Online]. Available: <https://nurseassess.com/assessment-descriptions/morse-fall-risk-scale/>
- [124] M. D. and I. K. and S. J. and B. J. and Y. K. Ellis Louise A and Lee, ‘COVID-19 as ‘Game Changer’ for the Physical Activity and Mental Well-Being of Augmented Reality Game Players During the Pandemic: Mixed Methods Survey Study’, *J Med Internet Res*, vol. 22, no. 12, p. e25117, Dec. 2020, doi: 10.2196/25117.
- [125] ‘What is Augmented Reality (AR) | Microsoft Dynamics 365’. [Online]. Available: <https://dynamics.microsoft.com/en-us/mixed-reality/guides/what-is-augmented-reality-ar/>
- [126] ‘Augmented Reality (AR) Defined, with Examples and Uses’. [Online]. Available: <https://www.investopedia.com/terms/a/augmented-reality.asp>
- [127] ‘Overview of ARCore and supported development environments | Google Developers’. [Online]. Available: <https://developers.google.com/ar/develop>
- [128] ‘Fundamental concepts | ARCore | Google Developers’. [Online]. Available: <https://developers.google.com/ar/develop/fundamentals>
- [129] ‘World Geodetic System 1984 (WGS 84)’, *Office of Geomatics, National Geospatial-Intelligence Agency*, [Online]. Available: <https://earth-info.nga.mil/?dir=wgs84&action=wgs84>
- [130] ‘ARCore supported devices | Google Developers’. [Online]. Available: [https://developers.google.com/ar/devices#google\\_play\\_devices](https://developers.google.com/ar/devices#google_play_devices)
- [131] ‘TensorFlow Lite’. [Online]. Available: <https://www.tensorflow.org/lite/guide>
- [132] ‘TensorFlow Lite Task Library’. [Online]. Available: [https://www.tensorflow.org/lite/inference\\_with\\_metadata/task\\_library/overview](https://www.tensorflow.org/lite/inference_with_metadata/task_library/overview)
- [133] ‘Pose estimation and classification on edge devices with MoveNet and TensorFlow Lite — The TensorFlow Blog’. [Online]. Available: <https://blog.tensorflow.org/2021/08/pose-estimation-and-classification-on-edge-devices-with-MoveNet-and-TensorFlow-Lite.html>
- [134] ‘Overview | Places API | Google for Developers’. Accessed: Feb. 26, 2024. [Online]. Available: <https://developers.google.com/maps/documentation/places/web-service/overview>
- [135] ‘Get Started | Elevation API | Google for Developers’. Accessed: Mar. 10, 2024. [Online]. Available: <https://developers.google.com/maps/documentation/elevation/start>

- [136] S. Hekmatpour, 'Experience with Evolutionary Prototyping in a Large Software Project', *SIGSOFT Softw. Eng. Notes*, vol. 12, no. 1, pp. 38–41, Jan. 1987, doi: 10.1145/24574.24577.
- [137] L. Sherrell, 'Evolutionary Prototyping', *Encyclopedia of Sciences and Religions*, p. 803, 2013, doi: 10.1007/978-1-4020-8265-8\_201039.
- [138] 'Arduino Science Journal - Apps on Google Play'. [Online]. Available: <https://play.google.com/store/apps/details?id=cc.arduino.sciencejournal>
- [139] 'Pose Classification | mediapipe'. [Online]. Available: [https://google.github.io/mediapipe/solutions/pose\\_classification.html](https://google.github.io/mediapipe/solutions/pose_classification.html)
- [140] 'Video classification | TensorFlow Lite'. [Online]. Available: [https://www.tensorflow.org/lite/examples/video\\_classification/overview](https://www.tensorflow.org/lite/examples/video_classification/overview)
- [141] Z. Zhang, B. Krawczyk, S. García, A. Rosales-Pérez, and F. Herrera, 'Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data', *Knowl Based Syst*, vol. 106, pp. 251–263, Jan. 2016, doi: 10.1016/j.knosys.2016.05.048.
- [142] 'examples/lite/examples/pose\_estimation/android at master · tensorflow/examples'. [Online]. Available: [https://github.com/tensorflow/examples/tree/master/lite/examples/pose\\_estimation/android](https://github.com/tensorflow/examples/tree/master/lite/examples/pose_estimation/android)
- [143] 'pose\_classification.ipynb - Colaboratory'. [Online]. Available: [https://colab.research.google.com/github/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/tutorials/pose\\_classification.ipynb](https://colab.research.google.com/github/tensorflow/tensorflow/blob/master/tensorflow/lite/g3doc/tutorials/pose_classification.ipynb)
- [144] 'Post-training dynamic range quantization | TensorFlow Lite'. [Online]. Available: [https://www.tensorflow.org/lite/performance/post\\_training\\_quant](https://www.tensorflow.org/lite/performance/post_training_quant)
- [145] 'Pushups Dataset'. [Online]. Available: <https://universe.roboflow.com/adrian-lachowicz/push-ups-counter/dataset/1>
- [146] 'RU AI Team 3 Squat Frames | Kaggle'. [Online]. Available: <https://www.kaggle.com/datasets/jeremydeane/ru-ai-team-3-squat-frames?resource=download>
- [147] 'Jumping Jacks - YouTube'. [Online]. Available: <https://www.youtube.com/watch?v=c4DAnQ6DtF8>
- [148] 'Weight loss exercise jumping jacks shorts jumpingjacks - YouTube'. [Online]. Available: [https://www.youtube.com/watch?v=Tv\\_3AqhdN2A](https://www.youtube.com/watch?v=Tv_3AqhdN2A)
- [149] 'Can You Use The SUS For Websites? – MeasuringU'. [Online]. Available: <https://measuringu.com/sus-websites/>

- [150] ‘DESENVOLVIMENTO DE UMA PLATAFORMA PERSONALIZADA MHEALTH DE TREINO PARA COMBATER A INATIVIDADE FÍSICA NA DPOC Mestrado em Engenharia Informática-Computação Móvel Rúben Daniel Neves Caceiro Leiria, março de 2022’.
- [151] ‘If You Could Only Ask One Question, Use This One. – MeasuringU’. [Online]. Available: <https://measuringu.com/single-question/>
- [152] ‘SUS Calculation - Make a copy’. Accessed: Mar. 04, 2024. [Online]. Available: <https://docs.google.com/spreadsheets/u/0/d/1zblyZlvFNdu5vFzLfiHJpdSX8JDhUm7ezWGRyy0uBI/copy?pli=1>