



IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Eng.^a Informática – Computação Móvel

VIDEO GUICHÊ

MIGUEL ÂNGELO DOS SANTOS FERREIRA

Leiria, Setembro de 2021



IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Eng.^a Informática – Computação Móvel

VIDEO GUICHÊ

MIGUEL ÂNGELO DOS SANTOS FERREIRA
Número: 2192403

Projecto realizado sob orientação do Professor Doutor Luis Filipe Fernandes Silva
Marcelino (luis.marcelino@ipleiria.pt).

Leiria, Setembro de 2021

RESUMO

A pandemia Covid-19 veio mudar a forma e a proximidade com que comunicamos. Uma parte significativa da nossa comunicação, que anteriormente seria presencial, passou a ser digital, nomeadamente recorrendo a videochamadas. Contudo as videochamadas em muitas áreas de negócio é necessário exemplificar uma ação, ilustrar instruções ou até ter alguma forma mostrar ou demonstrar um produto que possa não estar fisicamente presente junto a um, ou a ambos os participantes numa conversação.

Neste projeto decidi escolher como caso de estudo o exercício físico, que permite o acompanhamento do treino por um profissional sem que tenham de estar num ginásio. Para isso, implementei algumas funcionalidades que possam completar uma videochamada no sentido de tornar o canal de comunicação mais rico. Para escolher que funcionalidades implementar realizei uma pesquisa sobre aplicações de exercício físico disponíveis nas lojas de aplicações e também foi realizado um questionário com profissionais da área para entender se essas funcionalidades fazem sentido.

Depois do levantamento de requisitos preparei o servidor e todos os seus componentes: para gestão das videochamadas, para comunicação síncrona e para o armazenamento e a disponibilização de ficheiros multimédia. A interação dos utilizadores é feita através de uma aplicação cliente que disponibiliza as funcionalidades previamente escolhidas. Neste projeto a aplicação cliente usa a plataforma Android e com a utilização de tecnologias como WebRTC e WebSockets foi possível implementar uma aplicação que consiga adicionar funcionalidades novas que conseguem ajudar numa videochamada, os treinadores pessoais.

Com base nos resultados e *feedback* obtidos dos testes com utilizadores posso afirmar que as funcionalidades identificadas e desenvolvidas nesta aplicação são um importante complemento para os treinadores pessoais realizarem treinos através de uma videochamada, sendo que todos os treinadores que realizaram os testes da aplicação referiram que seria uma grande ferramenta de auxílio aos treinos online.

ABSTRACT

The Covid-19 pandemic has changed the way and proximity with which we communicate. A significant part of our communication, which previously would have been face-to-face, has now become digital, namely using video calls. However, video calls in many business areas require an example of an action, illustrating instructions or even having some way of showing or demonstrating a product that may not be physically present with one or both participants in a conversation.

In this project I decided to choose physical exercise as a case study, which allows the monitoring of training by a professional without having to be in a gym. For this, I implemented some features that can complete a video call in order to make the communication channel richer. To choose which features to implement, I carried out a survey on physical exercise apps available in app stores and a questionnaire was also carried out with professionals in the area to understand if these features make sense.

After gathering the requirements, I prepared the server and all its components: for managing video calls, for synchronous communication and for the storage and availability of multimedia files. User interaction is done through a client application that provides the previously chosen features. In this project the client application uses the Android platform and with the use of technologies such as WebRTC and WebSockets it was possible to implement an application that can add new features that can help in a video call, the personal trainers.

Based on the results and *feedback* obtained from the tests with users, I can say that the features identified and developed in this application are an important complement for personal trainers to carry out training through a video call, and all the trainers who carried out the tests of the application mentioned that it would be a great tool to help online training.

ÍNDICE

Resumo	i
Abstract	iii
Índice	v
Lista de Figuras	vii
Lista de Tabelas	ix
Lista de Abreviaturas	xi
1 INTRODUÇÃO	1
1.1 Enquadramento	2
1.2 Objectivos	2
1.3 Metodologia	3
1.4 Ferramentas Utilizadas	4
1.5 Apresentação do Documento	4
2 TRABALHO RELACIONADO	5
2.1 Aplicações de treino pessoal	5
2.1.1 Home Workout	6
2.1.2 30 Day Fitness	7
2.1.3 7 Minute Workout	8
2.1.4 Adidas Training	9
2.1.5 Gym Trainer	10
2.1.6 Gym Coach	11
2.1.7 Fitness & BodyBuilding	12
2.1.8 BetterMe	13
2.1.9 Conclusão	14
2.2 Comparação de Tecnologias	15
2.2.1 Jitsi	15
2.2.2 Janus	16
2.2.3 OpenVidu	17
2.2.4 Ant Media	18
2.2.5 Conclusão	19
3 PROPOSTA DE SOLUÇÃO	21
3.1 Questionário	21
3.2 Especificação de requisitos	22

ÍNDICE

3.3	Arquitetura	24
3.3.1	Aplicação cliente	25
3.3.2	Servidor	26
3.4	Protótipos	26
4	IMPLEMENTAÇÃO	33
4.1	Arquitetura	33
4.2	DNS	34
4.3	Servidor	35
4.4	Jitsi	36
4.5	WebSockets	37
4.6	Aplicação Cliente	37
5	PROTOTIPO E TESTES	41
5.1	Prototipo	41
5.2	Testes	45
5.2.1	Guia de Teste	46
5.2.2	Resultados	46
5.2.3	Propostas de Alteração	47
5.3	Revisão da aplicação	48
6	CONCLUSÕES E TRABALHO FUTURO	51
6.1	Conclusões	51
6.2	Trabalho Futuro	51
	BIBLIOGRAFIA	53
	Apêndices	
A	APÊNCICE A	57
B	APÊNCICE B	59
	DECLARAÇÃO	63
	DECLARAÇÃO	65

LISTA DE FIGURAS

Figura 1	Capturas de ecrã Home Workout	7
Figura 2	Capturas de ecrã 30 Day Fitness	8
Figura 3	Capturas de ecrã 7 Minute Workout	9
Figura 4	Capturas de ecrã Adidas Training	10
Figura 5	Capturas de ecrã Gym Trainer	11
Figura 6	Capturas de ecrã Gym Coach	12
Figura 7	Capturas de ecrã Fitness & BodyBuilding	13
Figura 8	Capturas de ecrã BetterMe	14
Figura 9	Arquitetura Interna do Jitsi	16
Figura 10	Arquitetura Modular do Janus	17
Figura 11	Arquitetura do OpenVidu	18
Figura 12	Arquitetura funcional do Ant Media	19
Figura 13	Proposta de Arquitetura	25
Figura 14	Protótipo de login e registar	27
Figura 15	Protótipo ecrã Inicial e juntar a uma sala	28
Figura 16	Protótipo videochamada	29
Figura 17	Protótipo partilha de vídeos	30
Figura 18	Protótipo definir temporizador	31
Figura 19	Protótipo de consultar os planos de treino	32
Figura 20	Arquitetura	34
Figura 21	Capturas de ecrã principal e juntar a uma sala	41
Figura 22	Capturas de ecrã com possíveis acções	42
Figura 23	Capturas de ecrã a partilhar identificador da sala	43
Figura 24	Capturas de ecrã a definir temporizador	43
Figura 25	Capturas de ecrã da partilha de vídeos	44
Figura 26	Capturas de ecrã da partilha de vídeos	45
Figura 27	Treinadores a testar a aplicação	46
Figura 28	Alteração ao temporizador	49
Figura 29	Mostrar o exemplo do exercício	49

LISTA DE TABELAS

Tabela 1	Comparação de funcionalidade entre aplicações estudadas . . .	15
Tabela 2	Tabela de comparação de tecnologias	20
Tabela 3	Nível de importância de cada funcionalidade	23
Tabela 4	WebSockets disponíveis	37
Tabela 5	Resultados dos Testes: Pedidos de Ajuda para completar a tarefa	47

LISTA DE TABELAS

LISTA DE ABREVIATURAS

API	Application Programming Interface.
DNS	Domain Name System.
FQDN	Fully Qualified Domain Name.
HTTPS	Hyper Text Transfer Protocol Secure.
IDE	Integrated Development Environment.
IP	Internet Protocol.
RAM	Random Access Memory.
RTMP	Real Time Messaging Protocol.
SDK	Software development kit.
SSH	Secure Shell.
SSL	Secure Sockets Layer.
TLS	Transport Layer Security.
UC	Unidade Curricular.
WebRTC	Web Real-time Communication.

INTRODUÇÃO

A pandemia Covid-19, que limitou os contactos presenciais, levou a um crescimento elevado na utilização de videochamadas para comunicação, quer seja em empresas, em instituições públicas ou mesmo a nível pessoal. Devido a esta procura elevada o número de empresas que disponibilizam este serviço também aumentou.

No entanto, apesar de a quantidade de ofertas de ferramentas de comunicação aumentar, as funcionalidades destas soluções não diferem de forma significativa entre elas, resultando em ofertas genéricas e muito sobrepostas. Por exemplo, um utilizador de WhatsApp pode realizar uma videochamada idêntica a uma videochamada realizada com Google Meet ou Microsoft Teams.

O aumento da utilização da comunicação passou a ser não presencial mas, no entanto existem muitas atividades que não são fáceis de realizar à distancia e que necessitam de algo mais numa videochamada e não apenas a genérica ligação de vídeo onde podem apenas ver e falar com outra pessoa.

O número de pessoas que nos últimos anos começou a aderir a ginásio e ter uma atividade física regular aumentou. No entanto, durante o confinamento essas pessoas não se podiam deslocar para os ginásios e com as aplicações de vídeo chamadas existentes não existe uma grande interação entre o treinador e o participante.¹²

A adoção de vídeo chamadas para realização de exercício físico oferece várias oportunidades interessantes que permitem colmatar algumas das limitações de uma comunicação não presencial, integrando conteúdos multimédia e aumentando a informação de comunicação de vídeo, não se limitando a uma ligação de vídeo e voz.

Devido ao do uso de videochamadas e do exercício físico ter crescido muito nos últimos anos, o impacto que a adoção destas aplicações que associem estas duas áreas, como uma aplicação de videochamadas com mais funcionalidades apresenta um universo de utilizadores significativo.

A proliferação de aplicações de vídeo beneficiou do desenvolvimento e disponibilização do *open standard* para comunicação vídeo WebRTC, suportado na Web e em aplicações móveis. Este standard permite a comunicação multiplataforma e está presente num número significativo de ferramentas de comunicação.

1 <https://www.publico.pt/2020/05/04/sociedade/noticia/confinamento-pos-gente-pratica-r-atividade-fisica-1914691> consultado a 08/07/2021

2 <https://sicnoticias.pt/especiais/coronavirus/2021-02-21-Exercicio-fisico-em-confinamento.-Sem-as-caminhadas-era-muito-mais-complicado> consultado a 08/07/2021

1.1 ENQUADRAMENTO

Este projeto enquadra-se no Mestrado em Engenharia Informática - Computação Móvel e visa permitir aplicar os conhecimentos adquiridos nas várias unidades curriculares e o desenvolvimento de novas competências como o desenvolvimento de aplicação para dispositivos móveis, adaptação de tecnologias já existentes e capacidades de organização de trabalho.

Escolhi este projeto, pois tinha um grande interesse por trabalhar com uma aplicação de vídeo chamadas e conhecer o seu processo de desenvolvimento e também por acreditar que as aplicações de vídeo chamadas atuais não conseguem cumprir alguns requisitos específicos para algumas áreas de trabalho.

Este trabalho aplica os conhecimentos adquiridos nas UCs de Desenvolvimento para Dispositivos Móveis e Gestão de Projetos Informáticos.

Devido à pandemia, muitas das comunicações diárias que eram presencias passaram para o meio digital recorrendo principalmente a aplicações de vídeo. No entanto, algumas das necessidades de certas áreas não conseguem ser cumpridas por a maior parte destas aplicações existentes por isso este projeto vem exemplificar a possibilidade da adaptação de tecnologias já existentes a mercados não explorados no âmbito da comunicação digital.

É relevante também salientar que, a pandemia que motivou em muito os objetivos deste projeto, acabou por limitar a realização dos trabalhos: a dificuldade e limitação no contacto presencial com os profissionais de desporto acabou por reduzir significativamente os testes desejados para um estudo desta natureza.

1.2 OBJECTIVOS

O principal objetivo deste projeto é o estudo e aplicação das tecnologias de vídeo como uma forma de comunicação síncrona e aumentada com apresentação de informação formatada e adequada para o caso de estudo considerado, que poderá ser uma forma de comunicação alternativa à comunicação presencial.

O caso de estudo escolhido para este projeto é o treino físico pessoal, pois é uma parte essencial das nossas vidas e, devido à pandemia, muitas pessoas não tinham como praticar devido ao encerramento de ginásios.

Para a realização do objetivo principal recorreremos à criação de uma aplicação para videochamadas e também a criação de um servidor para gerir e controlar as videochamadas. Esta aplicação também inclui algumas funcionalidades específicas do

caso de estudo que foram previamente pensadas com um levantamento de requisitos com profissionais da área em estudo.

O resultado das avaliações da solução implementada permitirão responder aos objetivos definidos, tendo em conta as restrições adotadas neste projeto.

1.3 METODOLOGIA

Neste projeto foi adotada uma metodologia Ágil, que consiste em dividir tarefas maiores em pequenas parte e desenvolver essas pequenas partes individualmente para no final ter o projeto final completo.

O projecto foi dividido e calendarizado nas seguintes partes:

- Mês de setembro - Estudo das aplicações de comunicação já existentes
- Mês de outubro - Estudo de tecnologias de servidor e escolha do caso de estudo
- Mês de novembro e Dezembro - Implementação do Servidor, protótipo inicial da aplicação e levantamento de requisitos
- Mês de janeiro, fevereiro e março - implementação de funcionalidades relativas ao caso de estudo
- Mês de abril, maio e junho - realização de testes e algumas alterações á aplicação com o devido *feedback*
- Mês julho, agosto e setembro - análise dos resultados e conclusões, e escrita final do relatório.

Esta metodologia também permite uma adaptação rápida as alterações ao projeto por parte do cliente, alguma necessário, pois à medida que estava a desenvolver este projeto também fazia o levantamento de requisitos.

Para ser possível o acompanhamento do progresso do projeto foram realizadas reuniões a cada duas semanas, em que era discutido o trabalho que foi realizado nessas duas semanas e planeado o trabalho que iria ser realizado nas próximas duas semanas.

Estas reuniões eram realizados remotamente através da plataforma Microsoft Teams³, para a partilha do código foi utilizado o GitHub⁴ e para a realização do relatório foi utilizado o Overleaf⁵.

3 <https://www.microsoft.com/pt-pt/microsoft-teams/group-chat-software>

4 <https://github.com/>

5 <https://www.overleaf.com/>

1.4 FERRAMENTAS UTILIZADAS

No decorrer deste projecto foram utilizadas as seguintes ferramentas:

- Microsoft Teams - Ferramenta de vídeo para desktop e smartphones usada nas reuniões com o orientador
- GitHub - Ferramenta de controlo de versões de código
- Overleaf - Ferramenta online para elaboração do relatório
- Android Studio ⁶ - Ferramenta para desktop para apoio ao desenvolvimento
- bash ⁷ - Ferramenta para utilizada para aceder ao servidor por ssh

1.5 APRESENTAÇÃO DO DOCUMENTO

Este relatório está organizado em mais cinco capítulos.

No Capítulo 2 é descrito o estudo de aplicações já existentes no mercado, relacionadas com o nosso caso de estudo, e o estudo relacionado com as *frameworks* de WebRTC consideradas.

No Capítulo 3 temos a proposta de solução para o objetivo referido anteriormente com um questionário realizado a profissionais da área do desporto, os requisitos que a nossa aplicação necessita, uma proposta de arquitetura geral e os protótipos de como será a aplicação.

No Capítulo 4 é apresentada a implementação de toda a arquitetura, desde o servidor e os seus componentes até á aplicação cliente e todas as configurações necessárias para que a nossa aplicação possa funcionar corretamente.

No Capítulo 5 é apresentado o prototipo que resultou da implementação e também os testes realizados com os profissionais previamente questionados e as sugestões de alterações que resultaram desses mesmos testes.

No Capítulo 6 são apresentadas as conclusões finais e sugestões de trabalho futuro.

⁶ <https://developer.android.com/studio>

⁷ <https://www.gnu.org/software/bash/>

TRABALHO RELACIONADO

Para o nosso caso de estudo decidimos explorar de que forma as tecnologias de comunicação de vídeo podem ser usadas para substituir a comunicação presencial no exercício físico, por este ser uma parte essencial da vida de muitas pessoas e, também por não ter sido possível durante o confinamento as pessoas deslocarem-se a um ginásio para a prática do mesmo.

Para o nosso caso de estudo existem algumas particularidades que não são possíveis alcançar com a maior parte das aplicações de vídeo chamadas disponíveis no mercado atualmente.

O mercado de aplicações de vídeo chamadas é um mercado com várias empresas, por exemplo as seguintes:

- Zoom ¹
- Microsoft Teams ²
- Discord ³

Apesar destas aplicações terem sido as mais usadas no último ano, são aplicações de videochamada muito gerais, não têm nenhuma área de estudo como principal e não foi possível encontrar uma aplicação de vídeo chamada relacionada com o bem-estar físico. No entanto existem varias aplicações dedicadas ao treino pessoal em casa como as seguintes e as suas principais funcionalidades

2.1 APLICAÇÕES DE TREINO PESSOAL

Existem varias aplicações de treino pessoal nas lojas de aplicações Google Play e App Store, mesmo que existam tantas aplicações disponíveis a maior parte delas não são baseadas em provas, sendo que poucas são baseadas em relatórios de atividade física ou em experiência pessoal (E et al., 2015). Em relação às técnicas utilizadas por estas aplicações é possível afirmar que não existem diferenças em aplicações grátis ou pagas, sendo que apenas aplicam uma média de 5 de 23 possíveis (Anouk et al., 2014). Para o desenvolvimento destas aplicações de treino pessoal, existe uma

1 <https://zoom.us/>

2 <https://www.microsoft.com/pt-pt/microsoft-teams/group-chat-software>

3 <https://discord.com/>

falta de integração com especialistas no seu processo de desenvolvimento e pouco ou nenhum envolvimento com utilizadores. A falta de políticas de privacidade também é um fator preocupante (P et al., 2018).

2.1.1 *Home Workout*

O Home Workout⁴ desenvolvido por Leap Fitness Group com o objetivo fornecer rotinas de treino diárias para todos os grupos principais de músculos para um exercício em casa sem a necessidade de um treinador presente e apenas com o peso corporal.

O Home Workout fornece uma rotina de treino diárias, tem vários exercícios para diferentes grupos de músculos e com sequências de aquecimento.

Na Figura 1 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação Home Workout na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Desafios de treinos, várias semanas com treinos diferentes para uma certa zona do corpo
- Objetivos, em que é possível definir quantas vezes queremos treinar por semana
- Registo e histórico de peso
- Sincronização com o Google Fit

⁴ https://play.google.com/store/apps/details?id=homeworkout.homeworkouts.noequipment&hl=en_US&gl=US consultado a 20/09/2020

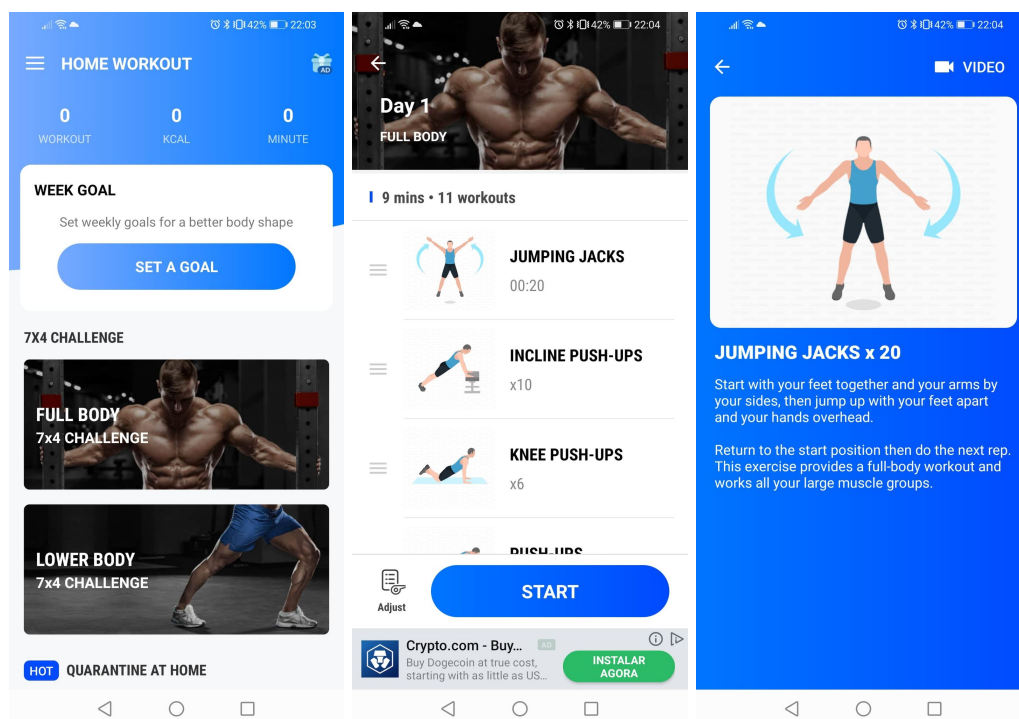


Figura 1: Capturas de ecrã Home Workout

2.1.2 30 Day Fitness

O 30 Day Fitness⁵ desenvolvido por Bending Spoons, tem como objetivo um aumento muscular em 30 dias de treino com apenas alguns minutos por dia, com a possibilidade de treinar em casa ou em qualquer outro lugar.

Esta aplicação permite ter um treino diferente diariamente e customizar o tempo do treino e os dias em que se realizar o treino.

Na Figura 2 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação 30 Day Fitness e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Ver todos os exercícios existentes na aplicação
- Históricos dos treinos realizados
- Desafios de 30 dias com treinos diários com exercícios focados numa zona do corpo
- Treino Customizado

⁵ <https://play.google.com/store/apps/details?id=com.bendingspoons.thirtydayfitness> consultado a 20/09/2020

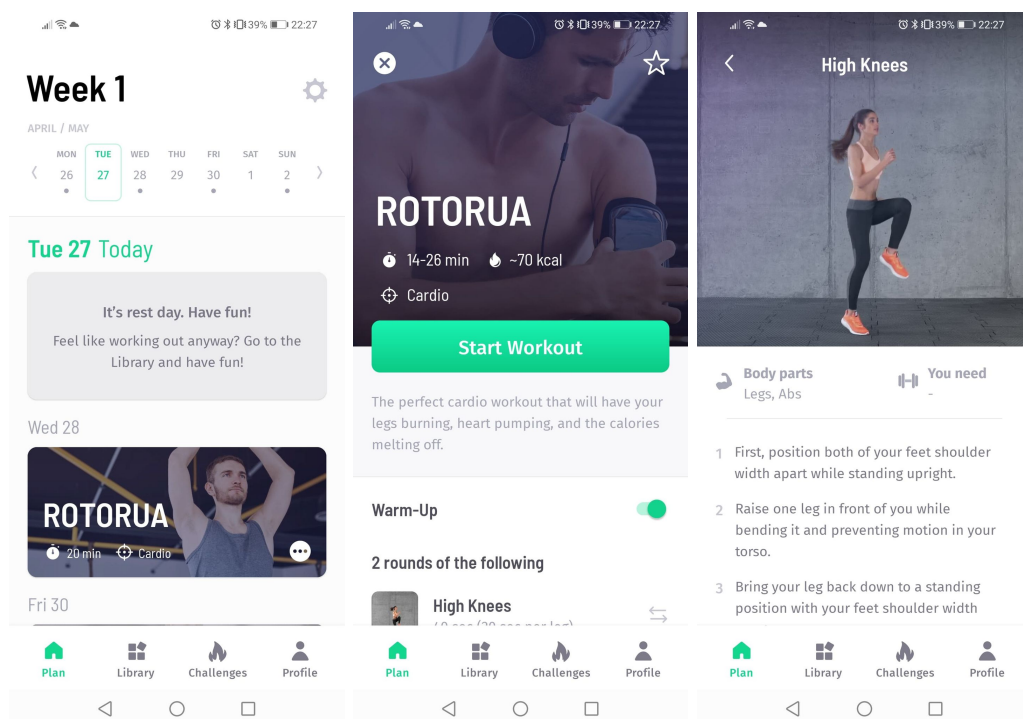


Figura 2: Capturas de ecrã 30 Day Fitness

2.1.3 7 Minute Workout

O 7 Minute Workout⁶ desenvolvida pela Simple Design Ltd. permite a realização de exercícios em qualquer lugar com, com pouco tempo necessário e sem material extra necessário.

Na Figura 3 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação 7 Minute Workout e na lista em baixo temos as funcionalidades mais relevantes e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Desafios de 30 dias com treinos de corpo inteiro com várias dificuldades possíveis
- Registo e histórico de peso
- Estatísticas do tempo de treino

⁶ <https://play.google.com/store/apps/details?id=com.popularapp.sevenmins> consultado a 20/09/2020

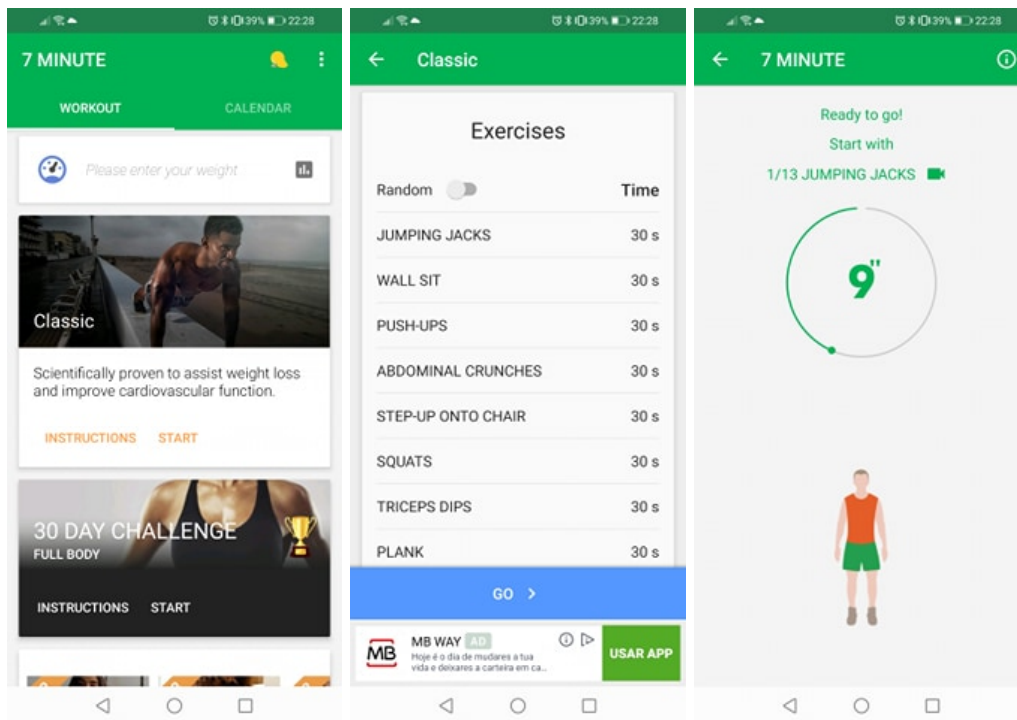


Figura 3: Capturas de ecrã 7 Minute Workout

2.1.4 *Adidas Training*

O Adidas Training⁷ desenvolvido por Adidas Runtastic o objetivo desta aplicação é fornecer exercícios de vários tipos e duração para ser possível treinar em qualquer altura do dia, permite a criação de treinos personalizados e focar numa área do corpo.

Na Figura 4 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação Adidas Training e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Estatísticas de treino, como numero de treinos realizados, tempo em treinos e calorías gastas
- Planos de treino com uma duração de várias semanas
- Conexão com Google Fit
- Planos de alimentação

⁷ <https://play.google.com/store/apps/details?id=com.runtastic.android.results.lite> consultado a 20/09/2020

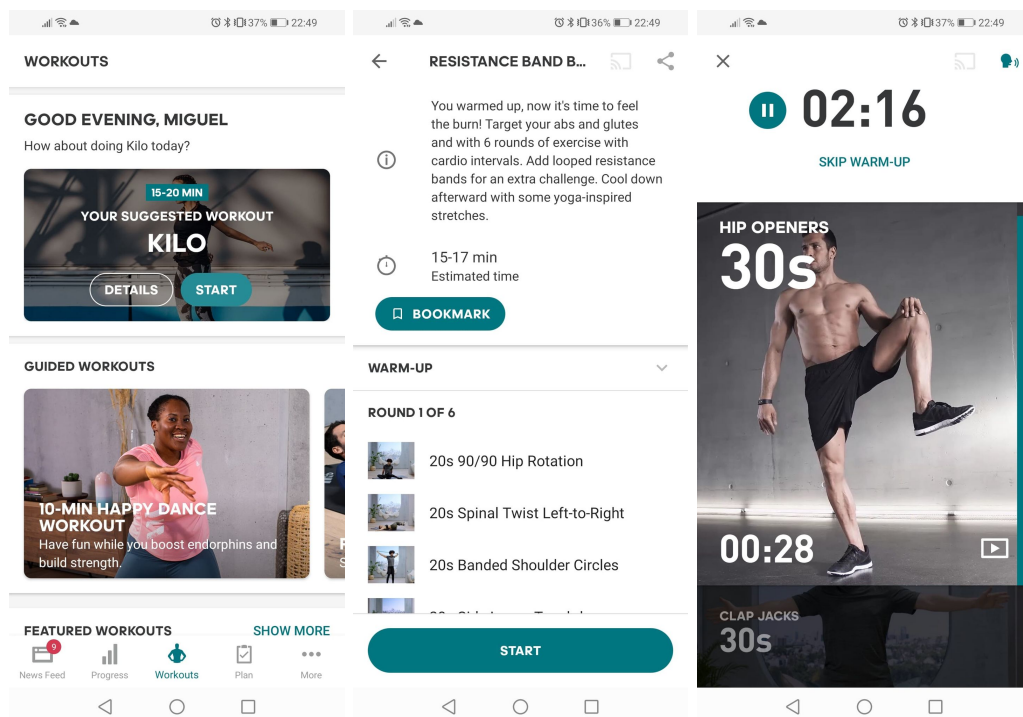


Figura 4: Capturas de ecrã Adidas Training

2.1.5 *Gym Trainer*

O Gym Trainer⁸ desenvolvido por TBG Apps, funciona como um treinador pessoal, ajudando a seguir o seu progresso e sugere conjunto de exercícios para maximizar os resultados. Ainda fornece desafios, dicas de alimentação, planos de treino para aumentar os músculos e para perder gordura.

Na Figura 5 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação Gym Trainer e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Exercícios para realizar no ginásio ou em casa
- Exercícios individuais sem plano de treino
- Desafios de 5 dias de treinos específicos para um exercício
- Ver todos os exercícios existentes na aplicação

⁸ <https://play.google.com/store/apps/details?id=com.gymtrainer.fitness.workout>, consultado a 20/09/2020

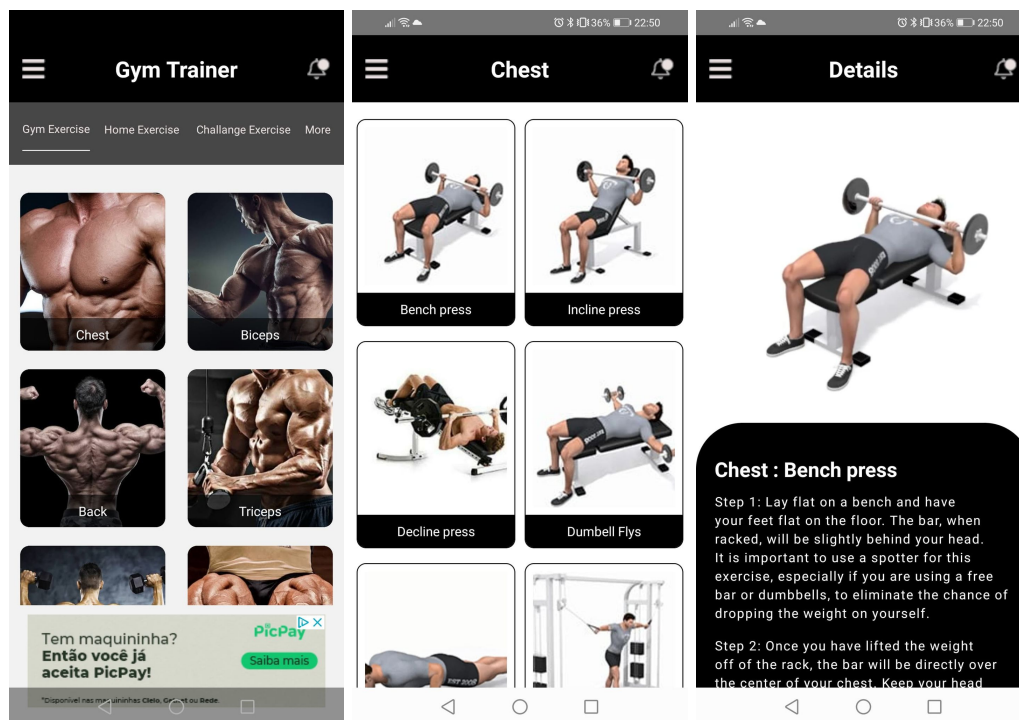


Figura 5: Capturas de ecrã Gym Trainer

2.1.6 Gym Coach

O Gym Coach⁹, desenvolvido por Ironclick, substitui um treinador pessoal como rotinas de treino construídas por especialistas, exemplos dos exercícios e a possibilidade de escolher o objetivo para qual treinamos e focar nesse objetivo.

Na Figura 6 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação Gym Coach e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Plano Alimentar
- Edição do numero de repetições por exercício
- Treinos diferentes por dia da semana

⁹ <https://play.google.com/store/apps/details?id=net.gym.coach> consultado a 20/09/2020

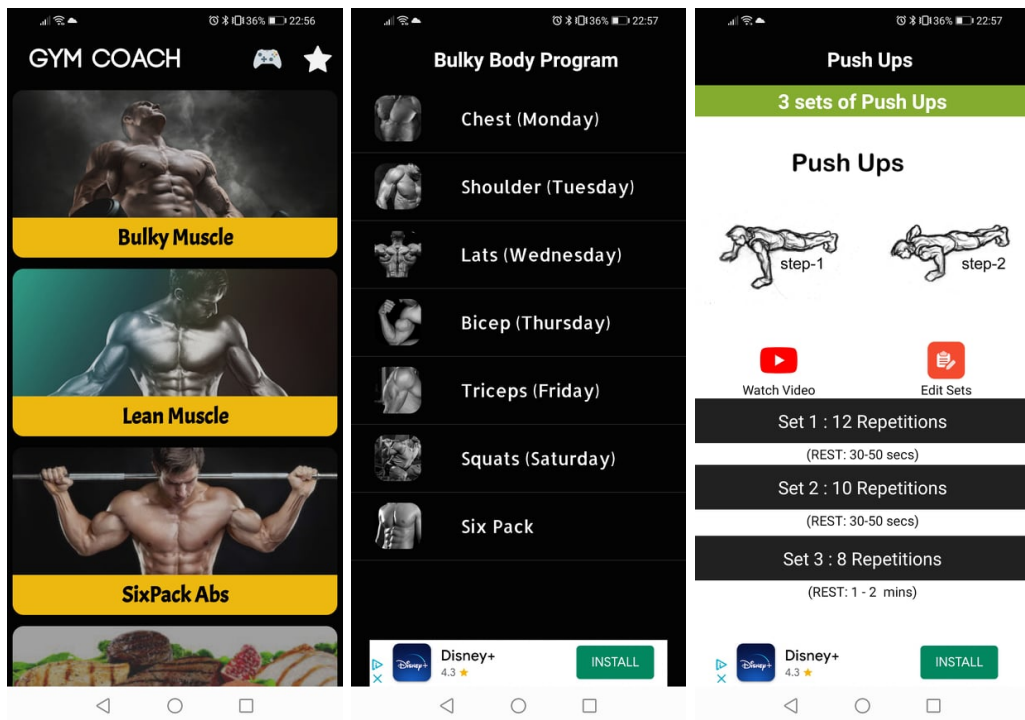


Figura 6: Capturas de ecrã Gym Coach

2.1.7 Fitness & BodyBuilding

O Fitness & BodyBuilding¹⁰, desenvolvido por VGFIT LLC, permite a criação de um treino com as necessidades de cada pessoa. Tem uma grande base de exercícios para cada músculo com uma descrição detalhada e com exemplificação de cada um.

Na Figura 7 podemos ver algumas capturas de ecrã do menu dos exercícios por grupo de músculo, vários exercícios desse grupo e o exemplo de um exercício da aplicação Fitness & BodyBuilding e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Plano de treinos por zona do corpo
- Criação de um plano de treino customizado
- Exercícios por zona do corpo
- Ver todos os exercícios existentes na aplicação
- Histórico de treinos
- Plano de alimentação

¹⁰ <https://play.google.com/store/apps/details?id=softin.my.fast.fitness> consultado a 20/09/2020

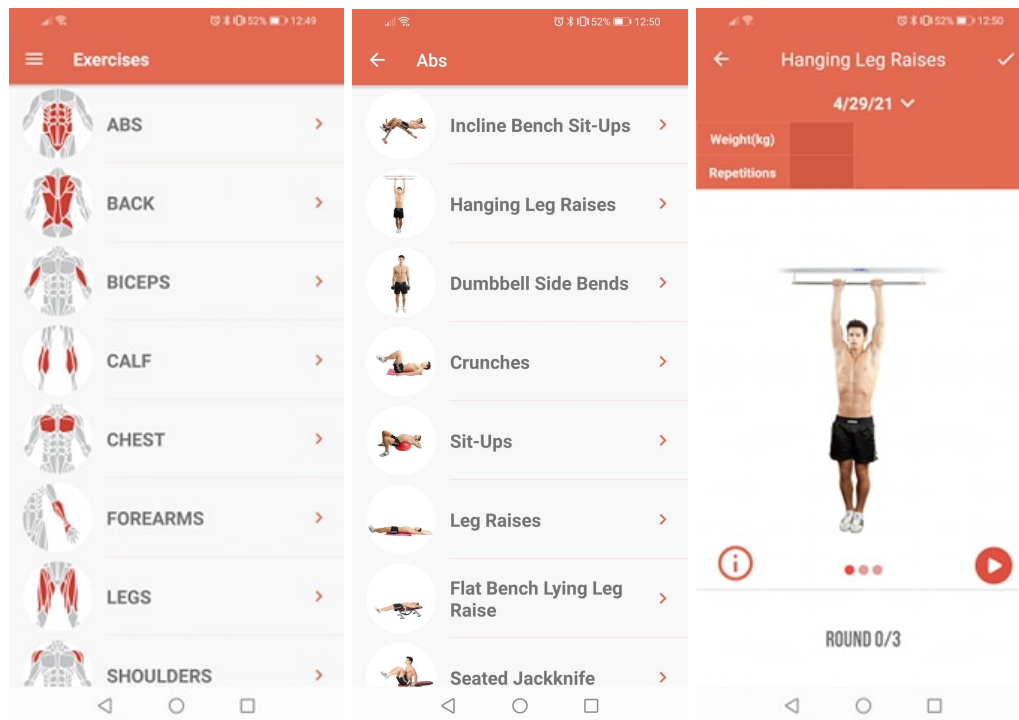


Figura 7: Capturas de ecrã Fitness & BodyBuilding

2.1.8 *BetterMe*

O BetterMe¹¹, desenvolvida por BetterMe Limited, junta um treinador pessoal com um nutricionista permitindo ter um plano de alimentação ajustado ao objetivo, treinos adaptados ao nível de experiência do utilizador e o acompanhamento da evolução do objetivo pretendido.

Na Figura 8 podemos ver algumas capturas de ecrã do menu principal, um dos planos de treino disponíveis e o exemplo de um exercício da aplicação BetterMe e na lista em baixo temos as funcionalidades mais relevantes.

- Vídeo/Imagem/Animação para exemplificar o exercício
- Conexão com Google Fit
- Contador de passos
- Acompanhamento do peso e de calorias ingeridas
- Tempo de Treino
- Vários níveis de treino (beginner, medium, advance)
- Plano de alimentação
- Lista de compras

¹¹ <https://play.google.com/store/apps/details?id=com.gen.workoutme> consultado a 20/09/2020

- Desafios de 7 a 30 dias com treinos, alimentação e estilo de vida

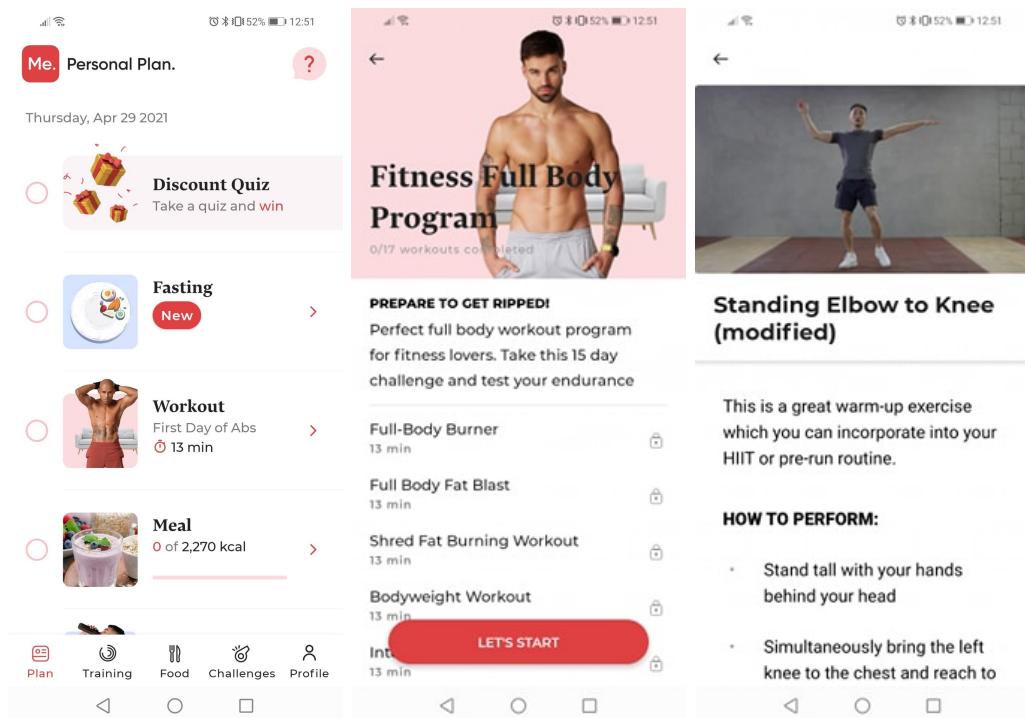


Figura 8: Capturas de ecrã BetterMe

2.1.9 Conclusão

Tendo em conta todas as aplicações descritas em cima e as suas funcionalidades podemos concluir que as funcionalidades mais comuns entre elas são o exemplo do exercício que se realiza seja com uma imagem, vídeo ou animação e a adaptação do nível do utilizador aos exercícios existentes.

Na Tabela 1 temos um resumo de todas as funcionalidades de todas as aplicações analisadas e podemos concluir que a exemplificação do exercício é uma funcionalidade presente em todas as aplicações, os planos de alimentação é a segunda funcionalidade mais frequente, a sincronização com Google Fit e desafios de treinos estão presentes em três das oito aplicações, as estatísticas, treino customizado, registo e histórico de peso e históricos estão presentes em apenas duas e contador de passos, objetivos, planos de treino, temporizador e lista de compras apenas num total de uma aplicação.

	Home Workout	30 Day Fitness	7 Minute Workout	Adidas Training	Gym Trainer	Gym Coach	Fitness & BodyBuilding	BetterMe
Exemplificação do exercício	X	X	X	X	X	X	X	X
Planos de alimentação				X		X	X	X
Sincronização com Google Fit	X			X				X
Desafios de treinos	X		X		X			
Estatísticas			X	X				
Treino customizado		X					X	
Registo e histórico de peso	X		X					
Históricos		X					X	
Contador de passos								X
Objectivos	X							
Planos de treino				X				
Temporizador								X
Lista de compras								X

Tabela 1: Tabela com comparação de funcionalidades entre aplicações estudadas

2.2 COMPARAÇÃO DE TECNOLOGIAS

O WebRTC¹² é um standard aberto de comunicação em tempo real, disponível através de APIs de JavaScript ou em bibliotecas no caso de aplicações Android e iOS com suporte para vídeo, voz e dados entre utilizadores permitindo desenvolver soluções de comunicação.

O projeto WebRTC é um projeto de código aberto e suportado por grandes empresas como Google, Apple, Microsoft e Mozilla.

O WebRTC é utilizado em algumas aplicações bastante conhecidas como Google Meet, Facebook Messenger, Discord, House Party, entre outros.

Para ser possível a realização de vídeo chamadas recorreremos à *framework* WebRTC, considerando as seguintes *frameworks* de alto nível.

2.2.1 Jitsi

O Jitsi¹³ é um conjunto de projetos de código aberto que permitem implementar soluções de vídeo chamadas seguras.

Ao implementar o nosso próprio servidor de Jitsi permite fornecer ao utilizadores uma melhor qualidade, menos latência e uma solução mais escalável e de baixo custo.

Para poder ser possível a utilização de vídeo durante uma chamada com vários participantes, o Jitsi implementa um VideoBridge, sendo um componente do servidor controlado remotamente (Ivov, 2013). Este componente é que é responsável pela organização e gestão das salas de videochamadas do Jitsi.

¹² <https://webrtc.org/>

¹³ <https://jitsi.github.io/handbook/docs/intro> consultado a 15/10/2020

É compatível com WebRTC e fornece bibliotecas para uma instalação mais fácil em Ubuntu e Debian.

O Jitsi também fornece um SDK para Android e iOS para uma integração fácil para dispositivos móveis e têm as seguintes características:

- Escalável
- Baixa latência
- Compatível com WebRTC
- Baixo Custo
- SDK para Android
- SDK para iOS

Na Figura 9 podemos também consultar a arquitetura interna do Jitsi que se encontra também disponível em *Jitsi Meet Handbook* ¹⁴.

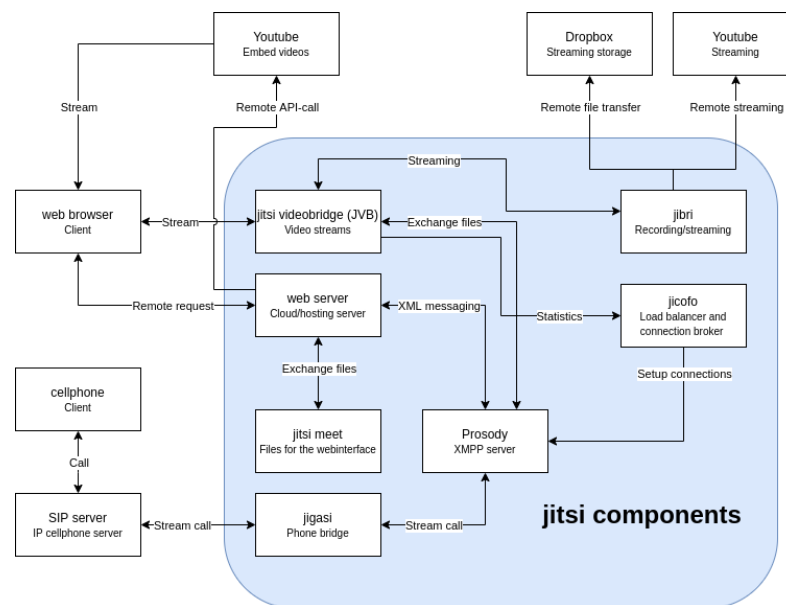


Figura 9: Arquitetura Interna do Jitsi *Jitsi Meet Handbook* ¹⁵

2.2.2 Janus

O Janus¹⁶ é uma *framework* com um propósito mais geral e não fornece alguma funcionalidade específica para além da comunicação de WebRTC com o *browser*.

¹⁴ <https://jitsi.github.io/handbook/docs/architecture#architecture-diagram> consultado a 15/10/2020

¹⁶ <https://janus.conf.meetecho.com/docs/> consultado a 15/10/2020

Para adicionar algumas funcionalidades específicas existem vários módulos disponíveis ou a possibilidade de criar os nossos próprios módulos para complementar a base do Janus.

Apesar de o Janus não fornecer diretamente SDK para Android e iOS existem uns SDK de terceiros disponíveis e têm as seguintes características:

- Escalável
- Baixa latência
- Compatível com WebRTC
- Baixo Custo
- Modular

Na Figura 10 podemos ver como funciona a arquitetura modular do Janus, e a mesma pode ser consultada no artigo *Performance analysis of the Janus WebRTC gateway* Amirante et al., 2015.

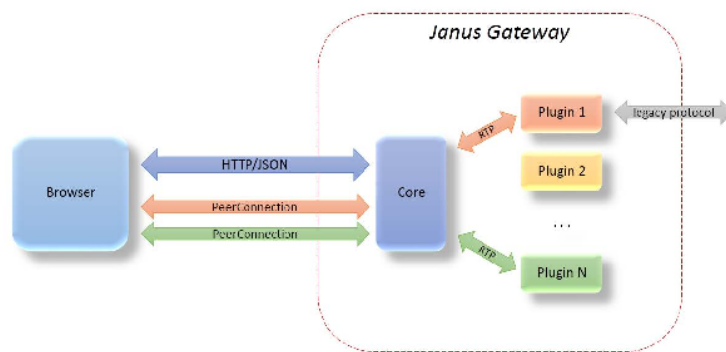


Figura 10: Arquitetura Modular do Janus

2.2.3 OpenVidu

O OpenVidu¹⁷ é uma plataforma que facilita a integração de vídeo chamadas na web ou em aplicações móveis.

É compatível com muitas tecnologias de *frontend* e com qualquer tecnologia de *backend*.

O OpenVidu fornece alguns tutoriais para uma fácil demonstração de como o OpenVidu funciona e de como implementar tanto no lado do cliente como no servidor.

O OpenVidu permite alterações ao *frontend* da sua *framework*, tanto que é possível adicionar funcionalidades novas como a conversão de voz para texto (Sally

¹⁷ <https://docs.openvidu.io/en/2.18.0/> consultado a 15/10/2020

et al., 2021). Isto é uma funcionalidade bastante útil devido à situação que vivemos no momento e o facto de existirem pessoas com dificuldades auditivas que não conseguiram entender uma videochamada por voz e têm as seguintes características:

- Escalável
- Baixa latência
- Compatível com WebRTC
- Baixo Custo
- SDK para Android
- SDK para iOS

Na Figura 11 podemos ver como a arquitetura do openVidu está implementada e a mesma está disponível em *OpenVidu Docs*¹⁸.

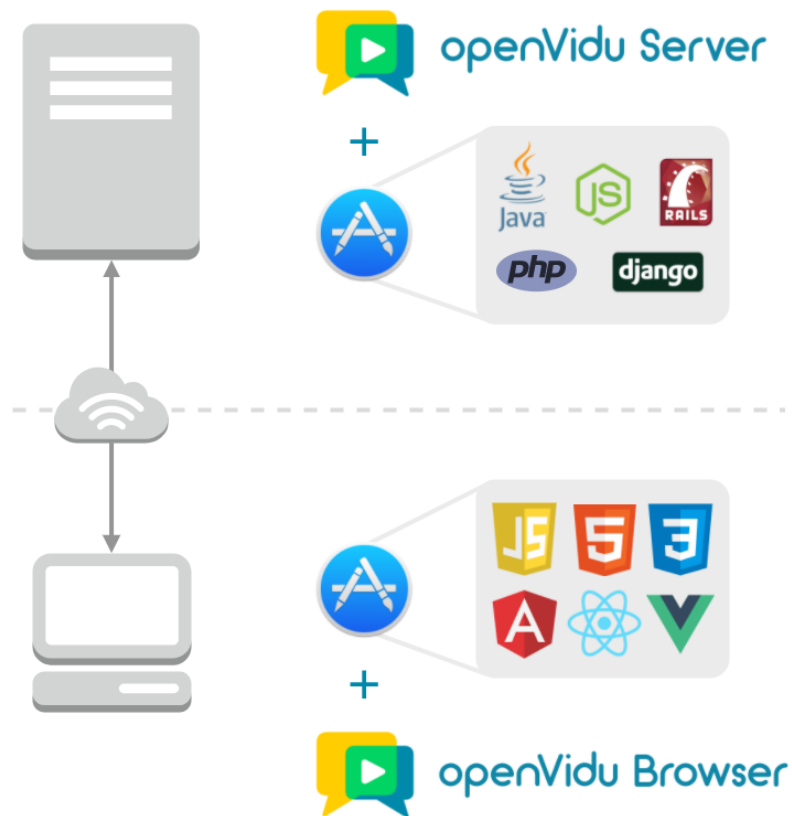


Figura 11: Arquitetura do OpenVidu *OpenVidu Docs*¹⁹

2.2.4 Ant Media

O Ant Media²⁰ é um *software de streaming* que fornece *streaming* com uma latência bastante baixa e um grande escalabilidade horizontal e vertical.

¹⁸ <https://docs.openvidu.io/en/2.18.0/#openvidu-architecture> consultado a 15/10/2020

²⁰ <https://antmedia.io/> consultado a 15/10/2020

No entanto, para ser possível essa latência baixa é necessário comprar uma licença, enquanto a versão *community* tem uma latência ainda considerável.

O Ant Media fornece um *deploy* mais fácil com imagens já criadas em vários fornecedores de *cloud* e também SDK para Android, iOS e JavaScript e têm as seguintes características:

- Escalável
- Baixa latência
- Compatível com WebRTC
- SDK para Android
- SDK para iOS

No entanto, o SDK para iOS e Android na versão *community* é apenas com a tecnologia RTMP e não com WebRTC que apenas está disponível na versão paga.

Como a pandemia Covid-19, a realização de testes, exames e até entrevistas de emprego passaram a ser realizados *online*, o Kumar et al., 2020 tem uma implementação com AntMedia e Inteligência Artificial para ser possível uma supervisão melhorada.

Na Figura 12 podemos ver a arquitetura funcional do Ant Media, e a mesma pode ser consultada na Wiki do repositório do Ant Media no GitHub²¹.

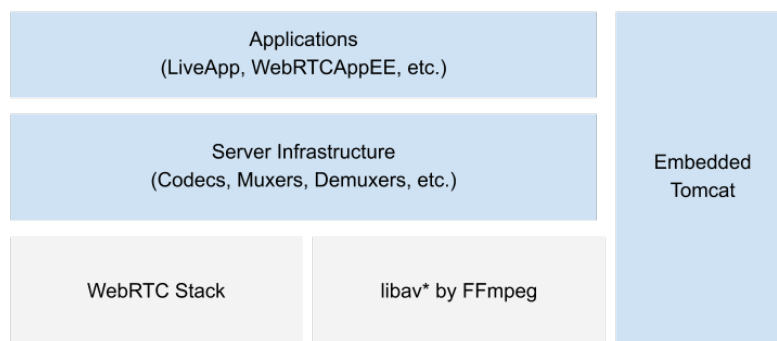


Figura 12: Arquitetura funcional do Ant Media²²

2.2.5 Conclusão

Tendo em conta as funcionalidades e descrição das tecnologias em cima descritas reparamos que todas implementam uma comunicação baseada em WebRTC, sendo que o Ant Media apenas numa versão paga.

²¹ <https://github.com/ant-media/Ant-Media-Server/wiki#functional-architecture> consultado a 15/10/2020

O Jitsi e o OpenVidu são as mais fáceis de implementar devido a tutorias que os mesmos fornecem. O Janus apesar de ser bastante completo apenas por módulos de terceiros é que é possível ter uma solução robusta de comunicação de vídeo.

O Ant Media numa versão gratuita tem uma latência considerável e não utiliza WebRTC.

O Janus é uma implementação que depende de vários módulos externos e depende um pouco de implementações de terceiros como no caso do SDK. O Jitsi e o OpenVidu foram então os que foram escolhidos para testar a sua implementação. Contudo ao testar a sua implementação o OpenVidu tem alguns problemas com o SDK em Android com a comunicação ao servidor, pelo que foi então implementado neste projeto o Jitsi.

Na Tabela 2 temos a comparação de todos os aspetos das tecnologias apresentadas previamente, todas elas são escaláveis, de baixa latência e compatível com WebRTC, sendo que o Ant Media é apenas possível numa versão paga. Os SDKs do AntMedia estão apenas disponíveis numa versão paga e o Janus não tem diretamente disponíveis esses SDKs, mas existem bibliotecas comunitárias. O Janus é a única tecnologia com uma arquitetura modular, apesar de esses módulos ser apenas desenvolvidos pela comunidade.

	Jitsi	Janus	OpenVidu	Ant Media
Escalável	X	X	X	X
Baixa latência	X	X	X	X
Compatível com WebRTC	X	X	X	X
Baixo Custo	X	X	X	
SDK para Android	X		X	X
SDK para iOS	X		X	X
Modular		X		

Tabela 2: Tabela de comparação de tecnologias

PROPOSTA DE SOLUÇÃO

Como referido nos capítulos anteriores o objetivo deste projeto é o estudo e aplicação da adoção de tecnologias de vídeo com outras formas de apresentação para uma comunicação síncrona com o nosso caso de estudo que é o exercício físico.

Com o estudo de algumas aplicações de treino pessoal já existentes no mercado foi possível fazer um levantamento de funcionalidades e elaborar um questionário com possíveis requisitos para ser realizado a vários treinadores para entender se nas suas realidades essas funcionalidades fazem sentido serem desenvolvidos.

3.1 QUESTIONÁRIO

Para o levantamento de requisitos foi realizado um questionário a vários treinadores que está disponível no anexo A e foram obtidas as respostas que estão disponíveis no anexo B.

Este questionário está dividido em 4 partes:

- Antes da pandemia
- Durante o confinamento
- Atualmente
- Aplicação de videochamadas para treino

Esta divisão permitiu nos três primeiros grupos conhecer um pouco a realidade de como funcionavam os treinos antes da pandemia, durante o confinamento e atualmente a que referimos a altura após do confinamento.

No último grupo referimos algumas funcionalidades retiradas das aplicações referidas no capítulo anterior e se estas fazem sentido na realidade dos treinadores que realizaram o questionário.

No primeiro grupo, relacionado com a realidade antes da pandemia, existia limite de participantes. No segundo grupo, durante a pandemia, os treinos eram dados por vídeo pré-gravados e iam começar a implementar um sistema de aulas por videochamada, mas depois ocorreu o desconfinamento. No terceiro grupo, que é atualmente, tem apenas uma aula *online* que é pré-gravada e irá continuar a ser pré-gravada.

No último grupo, que é mais específico para aplicações de videochamadas, as perguntas são mais direcionadas para os treinadores para perceber a experiência deles com aplicações de videochamadas e saber que funcionalidades que foram levantadas com o estudo prévio de aplicações de treino.

Em relação à experiência com aplicações de videochamadas temos metade com experiência e outra metade sem experiência em aplicações de videochamadas e em termos de expectativas para este tipo de aplicações são de ter qualidade de som e imagem, possibilidade de partilhar vídeos ou músicas, ver as pessoas e poder dar *feedback* a essas pessoas.

A maior parte dos treinadores preferem que o dispositivo utilizado para dar o treino seja o telemóvel sendo que para um seria indiferente ser por telemóvel ou tablet.

Todos consideram que é importante visualizar as pessoas a realizarem o treino, ter alguma forma de exemplificar o exercício a realizar, ter um limite de participantes por sala, apenas o treinador controlar os microfones dos participantes, ter o temporizador visível enquanto se está em videochamada e permitir o treinador poder retirar pessoas da videochamada.

A maior parte considera que é importante enviar mensagens às pessoas na sala e a gravação do treino, mas se for um atleta apenas com permissão para isso.

Nenhum dos treinadores consideram relevante a possibilidade de alterar o fundo, embelezamento da pessoa ou o uso de avatares.

Quando questionados sobre estatísticas importantes as mais comuns foram o numero de participantes por treino e por treinador e número de treinos por treinador.

3.2 ESPECIFICAÇÃO DE REQUISITOS

Como podemos verificar no capítulo anterior, nas várias funcionalidades das aplicações de exercício pessoal, todas estas aplicações têm em comum a possibilidade de ver um exemplo do exercício enquanto o realizamos.

Como anteriormente referido, foi realizado um questionário a vários treinadores para entender se as funcionalidades levantadas nas aplicações de exercício pessoal fazem sentido nas suas realidades e, considerando as respostas obtidas, podemos considerar que o temporizador e uma animação, vídeo ou imagem a exemplificar o exercício a ser realizado são das funcionalidades em que os treinadores questionados concordam como sendo das mais importantes.

Tendo em conta as respostas do questionário e as funcionalidades mais comuns das aplicações de exercício pessoal as funcionalidades mais importantes são o temporizador e a ter uma forma de exemplificar o exercício a realizar na altura.

	Pouco Importante	Importante	Muito Importante
Exemplificação do exercício			X
Planos de alimentação	X		
Sincronização com Google Fit	X		
Desafios de treinos	X		
Estatísticas		X	
Treino customizado		X	
Registo e histórico de peso	X		
Históricos	X		
Contador de passos	X		
Objectivos	X		
Planos de treino			X
Temporizador			X
Lista de compras	X		
Envio de mensagens			X
Apenas uma pessoa controlar os microfones		X	
Fundo customizado	X		
Avatars	X		
Embelezar a pessoa	X		
Permitir a gravação do treino		X	
Permitir retirar as pessoas da sala			X

Tabela 3: Nível de importância de cada funcionalidade

Segundo a Figura 3 temos as funcionalidades categorizadas por nível de importância como base no capítulo anterior e nas respostas do questionário, e tendo em conta esse nível de importância

Nesta aplicação iremos implementar:

- Videochamada
- Exemplificação do exercício através de uma imagem, vídeo ou animação
- Planos de treino
- Temporizador
- Envio de mensagens

A funcionalidade de permitir retirar as pessoas da sala apesar de importante não faz muito sentido em implementar, pois a aplicação está pensada como sendo uma chamada de duas pessoas.

3.3 ARQUITETURA

A arquitetura proposta consiste no cliente com um dispositivo móvel a comunicar com um servidor WebRTC e com alguma tecnologia de comunicação síncrona para ser possível a comunicação de alguns eventos que não estejam diretamente relacionados com a videochamada.

A aplicação tem de ter a possibilidade de comunicar com o servidor para a gestão das videochamadas, a comunicação síncrona para todos os eventos externos á videochamada e também tem de ter a possibilidade de reproduzir elementos media fornecidos pelo servidor.

O servidor terá de ter capacidades para a gestão das videochamadas, uma implementação com uma tecnologia de comunicação síncrona e a possibilidade de fornecer elementos media para poderem ser utilizados pela aplicação.

Na Figura 13 temos a posposta do que será a nossa arquitetura, temos um servidor com uma componente para a gestão das videochamadas, um sistema de comunicação síncrona e vídeos alojados no mesmo, a aplicação terá de comunicar com este através de WebRTC, com o protocolo de comunicação síncrona implementado e através de HTTPS. A aplicação em si, terá de ter o componente das videochamadas, leitor de media, planos de treinos, temporizador e um mecanismo de partilha para ser possível partilhar o id da sala. As ligações WebRTC permitem a comunicação com o componente de vídeo de chamadas no servidor, a comunicação síncrona será para existir uma possibilidade de ter funcionalidades adicionais e o HTTPS para poder utilizar os vídeos armazenados no servidor.

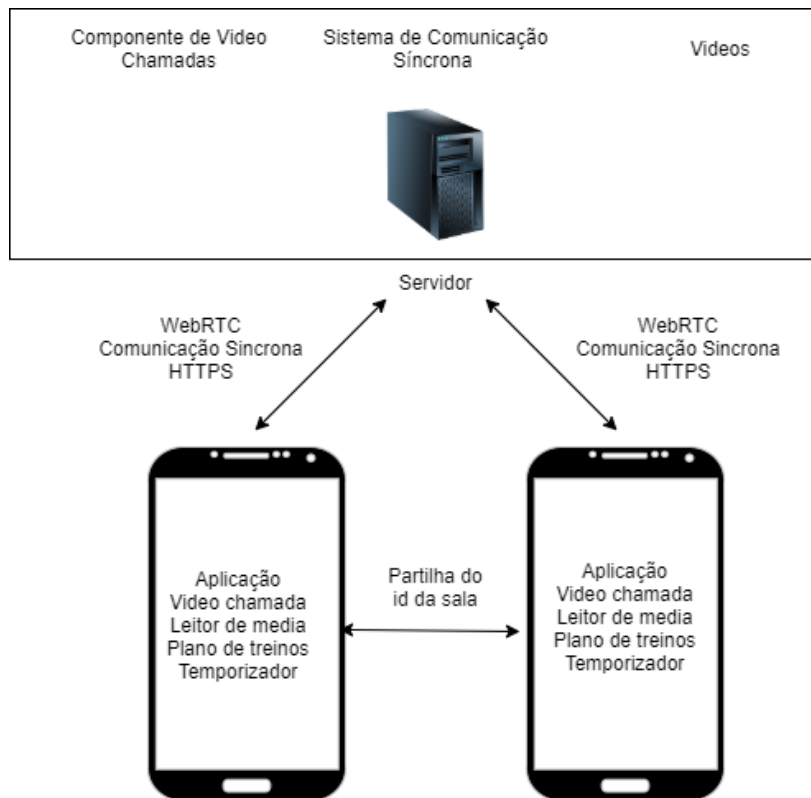


Figura 13: Proposta de Arquitetura

3.3.1 Aplicação cliente

A aplicação cliente tem de ter uma aplicação para um dispositivo móvel para que o seu acesso possa ser feito em qualquer lugar.

Para esta aplicação poder recorrer à *framework* de WebRTC previamente selecionada tem de integrar o SDK dessa mesma *framework* e, para implementar as outras funcionalidades que não fazem parte da base da *framework* escolhida, tem de poder integrar uma tecnologia de comunicação síncrona para ser possível realizar as funcionalidades extras.

O id da sala em que o utilizador se encontra é um aspeto importante, pois é através desse mesmo que outros utilizadores podem aceder à sala, para tal será necessária uma forma de partilhar esse id.

Para o uso dos vídeos, imagens ou animações a nossa aplicação tem de integrar um reprodutor de media para ser possível visualizar esses recursos, e também integrar um temporizador para seja possível o utilizador definir um tempo de treino e esteja visível para todos os utilizadores da sala.

A aplicação também deverá integrar um sistema de *chat* por mensagem para ser possível os utilizadores terem uma forma de comunicar sem ser obrigatoriamente

por voz e evitar ruídos enquanto o treinador estiver a explicar o que será para fazer nesse treino.

3.3.2 *Servidor*

Para a nossa aplicação poder recorrer à *framework* selecionada, o nosso servidor tem de integrar essa mesma *framework* e fazer a gestão das salas de videochamadas como criar e permitir a algum utilizador juntar-se a uma sala.

Para ser possível implementar as funcionalidades extras á base da *framework* selecionada o mesmo ou outro servidor tem de integrar uma tecnologia de comunicação síncrona para que a nossa aplicação possa fazer uso dessa comunicação.

Os vídeos, imagens ou animações têm de ser armazenados no nosso servidor para serem posteriormente utilizados pela nossa aplicação.

3.4 PROTÓTIPOS

Como qualquer aplicação tem de se pensar na experiência do utilizador, foram desenhados previamente os seguintes protótipos. Para os nossos protótipos, pensamos em manter as ações o mais simples possível para que não exista uma grande curva de aprendizagem nos utilizadores.

Na Figura 14 temos os formulários de login e de registo, para que os utilizadores possam criar conta e aceder a essa. Para a criação da conta é apenas necessário um email valido, e a confirmação desse mesmo email através de um link enviado por email.

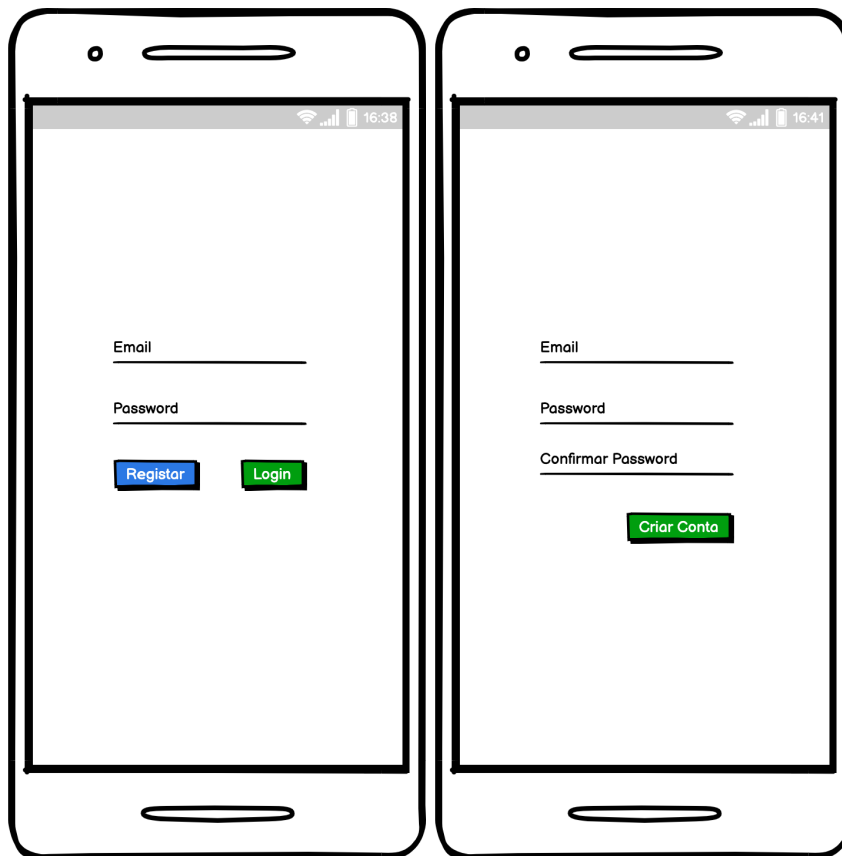


Figura 14: Protótipo de login e registrar

Na Figura 15 temos o menu principal que permite realizar três ações: criar uma sala, juntar a uma sala com o id da sala e consultar os planos de treino disponíveis. Ao criar uma sala o utilizador é logo redirecionado para o ecrã da videochamada, se escolher a ação de se juntar a uma sala é aberto um Modal com um input de texto para inserir o id da sala a qual se quer juntar, a ver os planos de treinos é redirecionado para o ecrã dos planos de treino.

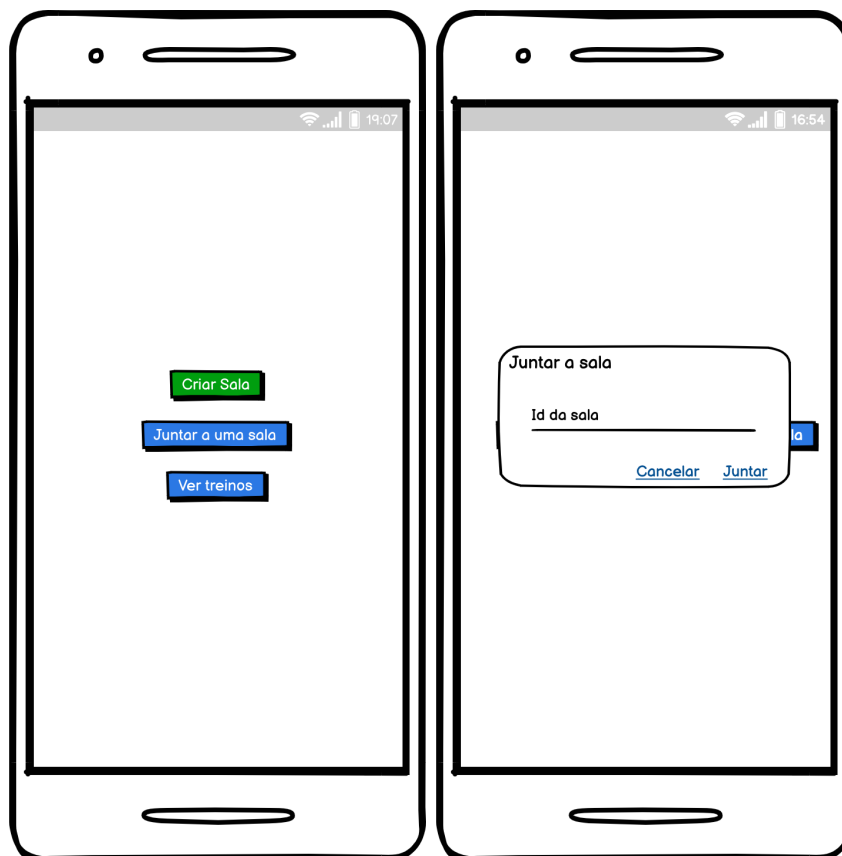


Figura 15: Protótipo ecrã Inicial e juntar a uma sala

Os próximos ecrãs, representados na Figura 16 exemplificam como será a sala da videochamada, tendo sempre uma parte do ecrã com a videochamada e um botão para ser possível realizar algumas ações. A videochamada será gerada e gerida pelo SDK da tecnologia previamente selecionada, como também todas as ações possíveis de realizar dentro da videochamada: desligar da mesma, abrir e escrever no *chat*, desligar a câmara e/ou microfone, escolha de dispositivo de som e trocar a câmara seccionada (frontal ou traseira). As ações de partilha de vídeo e temporizador serão comunicadas entre utilizadores presentes na sala através de um mecanismo de comunicação síncrona. A partilha do identificador da sala será realizada com um mecanismo externo que não terá de utilizar o servidor para o realizar.

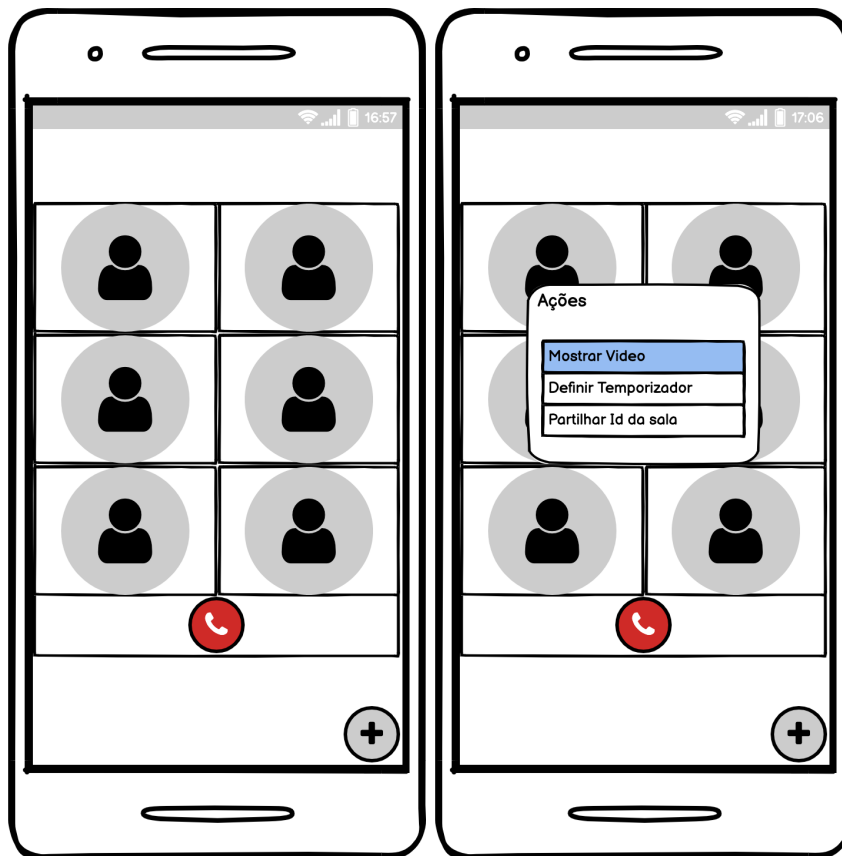


Figura 16: Protótipo videochamada

Na Figura 17 temos o prototipo de uma das ações possíveis de fazer que é a partilhar de um vídeo para tal ser possível tem de se escolher um dos vídeos presentes na lista e ao seleccionar o vídeo a partilhar abre um modal com o vídeo seleccionado. Estes vídeos tem de estar disponíveis no servidor para ser possível a visualização dos mesmos.

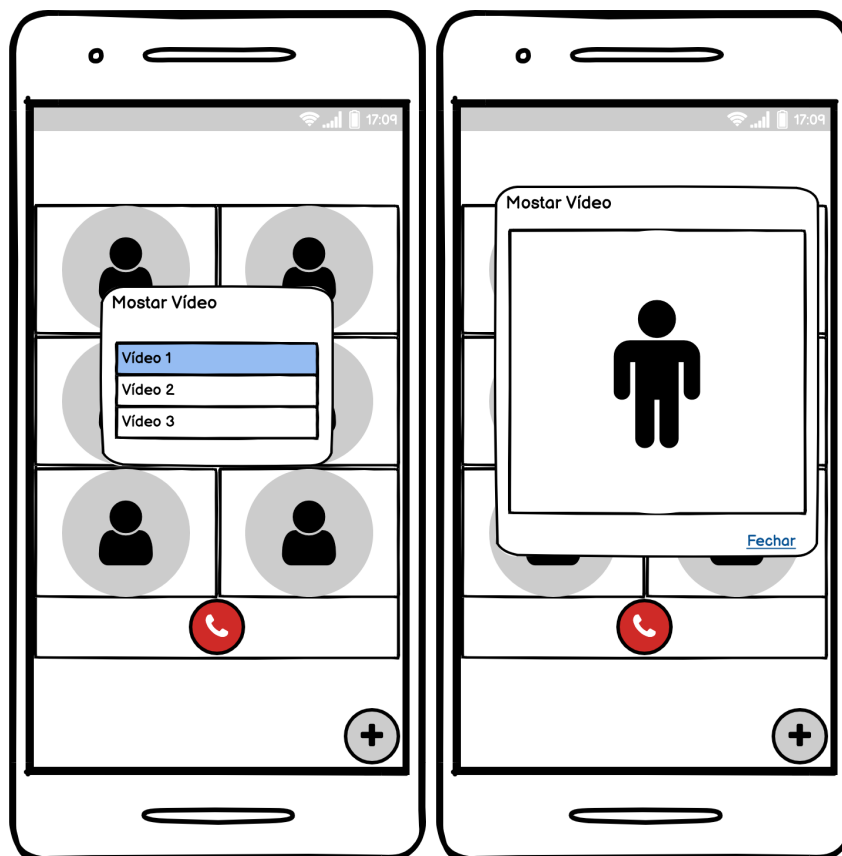


Figura 17: Protótipo partilha de vídeos

Na Figura 18 podemos ver como será para definir o temporizador, é usado um modal e esse temporizador aparece em cima da secção da videochamada. As ações deste temporizador são sempre comunicadas com todos os utilizadores presentes na sala com o recurso a um mecanismo de comunicação síncrona.

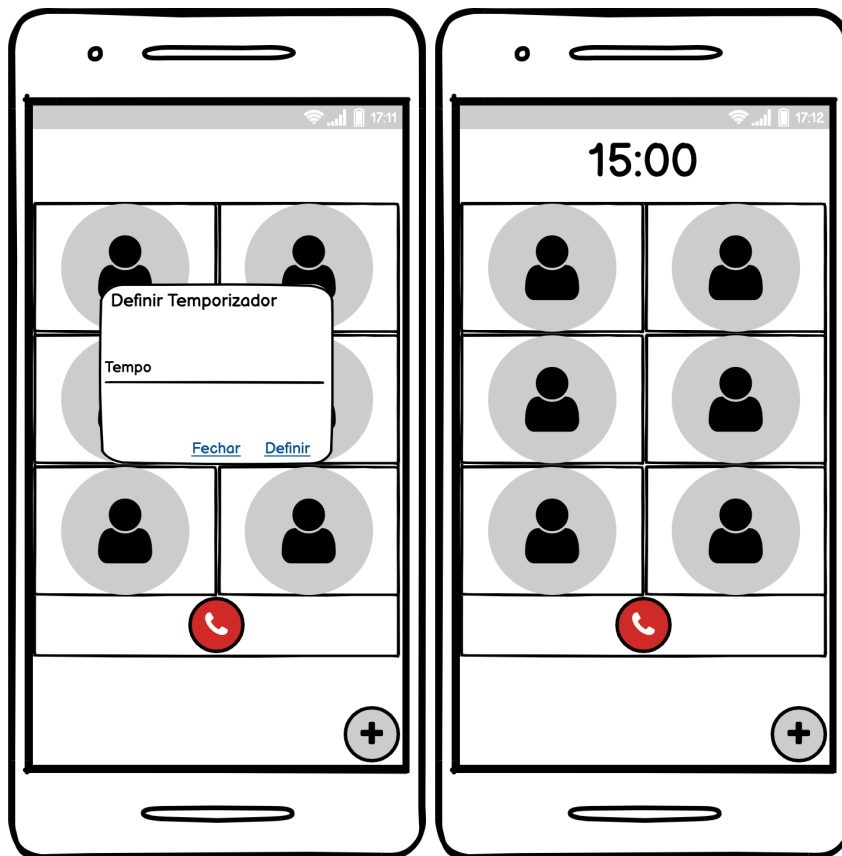


Figura 18: Protótipo definir temporizador

Na Figura 19 os planos de treino são apresentados vários planos disponíveis, e ao selecionar um dos planos são apresentados todos os exercícios desse plano com o número de repetições de cada um.



Figura 19: Protótipo de consultar os planos de treino

IMPLEMENTAÇÃO

4.1 ARQUITETURA

A arquitetura específica do nosso projeto consiste numa aplicação Android e um servidor de apoio com várias valências: uma instância Jitsi responsável pela gestão e criação de salas de videochamadas, com vídeos públicos para serem utilizados pela aplicação na demonstração de exercícios e também com uma instância de WebSockets implementado para facilitar comunicações que não estejam diretamente ligadas à videochamada.

A nossa aplicação utiliza o SDK do Jitsi para facilitar na comunicação com o servidor, VideoView para ser possível representar os recursos media disponíveis no servidor, Chronometer para ser possível a utilização do temporizador e o Socket.IO para poder comunicar com o servidor através de WebSockets.

O nosso servidor terá implementado um servidor Jitsi para fazer a gestão toda das videochamadas, um servidor WebSockets para ser possível a realização de comunicações síncronas na aplicação e terá de ter uma configuração para o que os vídeos estejam disponíveis para os utilizadores da aplicação.

Na Figura 20 temos a arquitetura da nossa aplicação, em que o nosso servidor tem uma integração Jitsi, um sistema de WebSockets para a comunicação síncrona, e vídeos para serem usados pela nossa aplicação. A aplicação comunica com o servidor através de WebRTC, WebSockets e HTTPS, também integra o SDK do Jitsi, uma VideoView, um Chronometer e implementa a comunicação síncrona com uso da biblioteca Socket.IO. A ligação WebRTC permite a comunicação entre o SDK do Jitsi da nossa aplicação e o componente Jitsi no servidor, os WebSockets serão para ser possível a partilha de vídeos e a gestão do temporizador para as salas das videochamadas e o HTTPS será para poder utilizar os vídeos armazenados no servidor.

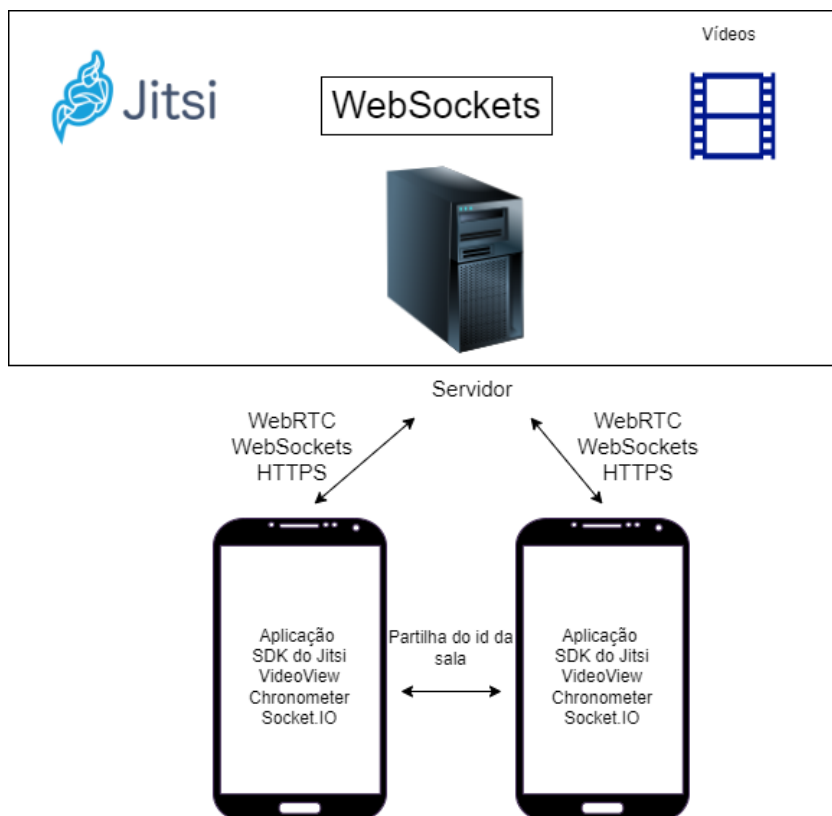


Figura 20: Arquitetura

Nas próximas secções irei descrever todos os passos necessários para as configurações necessárias para o servidor estar funcional, como a configuração do DNS, as configurações do servidor, a integração do Jitsi no nosso servidor e a implementação dos WebSockets e também a descrição de todo o desenvolvimento da aplicação.

4.2 DNS

Para ser possível ter o certificado TLS para que o nosso servidor seja reconhecido como um servidor seguro foi necessário registar um domínio.

O registo é um DNS do tipo A, o tipo de DNS mais básico que existe que vincula um domínio ou subdomínio a um endereço IP direto sendo que apenas funcionam para IPv4 enquanto o tipo AAAA é utilizado para IPv6.

Este certificado é necessário para que o nosso servidor possa servir por https e também para ser possível usar o SDK do Jitsi e o Android permitir a comunicação com o nosso servidor de Jitsi.

Para que o registo do nosso domínio não tivesse custos adicionais, o mesmo foi registado no Domain Names¹ com o pacote de estudante desenvolvedor do GitHub.

¹ <https://www.name.com/>

4.3 SERVIDOR

O servidor foi criado no Contabo² com Linux e nginx instalados.

O servidor tem a versão Debian 10 de 64 bits Little Edian, com 8 GB de RAM, 225 GB de espaço em disco e com um processador Intel Xeon com 2.20 GHz.

No servidor ainda temos na pasta `/var/www/media` temos todos os ficheiros media que estão disponíveis para os utilizadores da aplicação para poderem partilhar, esta pasta teve que ser configurada no ficheiro `default` localizado `/etc/nginx/site-available` para que todos os ficheiros dessa pasta possam ser visualizadas por qualquer utilizador.

```
location ~\textbackslash .mp4 {
    root /var/www/meida
}
```

O nosso servidor será utilizado para disponibilizar o Jitsi para permitir a criação de salas, os utilizadores juntarem-se as salas e ter todas as funcionalidades base do Jitsi.

Este servidor também implementa um sistema de comunicação síncrona para permitir aos utilizadores partilharem vídeos, imagens ou animações entre eles e para também permitir a utilização do temporizador. Estes vídeos, imagens ou animações estarão armazenados neste servidor.

Para a utilização e instalação do servidor Jitsi foi seguido o tutorial que está disponível na página web Self-Hosting Guide³ usando o domínio registado previamente.

Este tutorial apresenta todas as bibliotecas que devem ser instaladas previamente, qual o DNS que deve ser criado, como configurar o FQDN, adicionar o repositório da biblioteca Jitsi, configurar a `firewall`, configurar o certificado TLS e instalar a Jitsi Meet.

No servidor não existem portas restringidas, pois se trata de um ambiente de protótipos, então não houve uma necessidade de realizar uma limitação de portas abertas no servidor.

² <https://contabo.com/>

³ <https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-quickstart> consultado a 18/11/2020

4.4 JITSI

Para ser possível instalar o Jitsi no servidor é necessário ter uns pré-requisitos instalados. O servidor tem de ter um sistema operativo com uma versão Debian 9 ou mais recente ou Ubuntu 18.04 ou mais recente e ter as seguintes bibliotecas gnupg2⁴, nginx-full⁵, sudo⁶ e o OpenJDK 8 ou OpenJDK 11⁷. Se for um sistema Ubuntu, o Jitsi requisita dependências do repositório de bibliotecas universe⁸.

Ainda é necessário abrir as seguintes portas da *firewall*:

- 80 TCP - Para verificação do certificado SSL
- 443 TCP - Acesso geral ao Jitsi Meet
- 10000 UDP - Comunicações gerais de vídeo e áudio
- 22 TCP - Se o acesso ao servidor for por SSH
- 3478 UDP - Para consultar o servidor stun
- 5359 TCP - Como alternativa para as comunicações de vídeo e áudio

Depois destas configurações iniciais é necessário instalar o *jitsi-meet*. O instalador irá verificar se o Nginx ou Apache está presente no sistema e configurar um *host* virtual no servidor web para fornecer a Jitsi Meet. Neste instalador ainda é necessário definir o certificado SSL/TLS, que será para gerar um novo certificado ou usar um próprio. A opção escolhida foi a de usar um próprio que foi obtido previamente. Ainda será pedido para definir o *hostname* da instância do Jitsi Meet que será o que foi definido anteriormente no registo no DNS.

Esta instalação do Jitsi no servidor, inclui o Jitsi Videobridge que é um servidor compatível com WebRTC desenvolvido para distribuir as *streams* de vídeo entre os participantes de uma conferência. O Jitsi Conference Focus, um componente focado para servidor que gere as sessões de media e age como um *load balancer* entre cada um dos participantes e a videobridge. O Jitsi Gateway to SIP que permite clientes de SIP se juntarem a conferências. Jitsi Broadcasting Infrastructure que são um conjunto de ferramentas para gravar e/ou fazer *streaming* uma conferência.

4 <https://packages.debian.org/stretch/gnupg2> consultado a 14/01/2020

5 <https://packages.debian.org/sid/nginx-full> consultado a 14/01/2020

6 <https://www.sudo.ws/> consultado a 14/01/2020

7 <https://openjdk.java.net/> consultado a 14/01/2020

8 <https://help.ubuntu.com/community/Repositories/Ubuntu> consultado a 14/01/2020

join_room	Quando o utilizador se junta a uma sala
share_video	Quando o utilizador partilha um vídeo
start_timer	Quando o utilizador começa o temporizador
set_timer	Quando o utilizador define o temporizador
stop_timer	Quando o utilizador para o temporizador
video_shared	Emitir para utilizadores da sala quando o vídeo é partilhar
timer_started	Emitir para utilizadores da sala quando o temporizador é iniciado
timer_defined	Emitir para utilizadores da sala quando o temporizador é definido
timer_stoped	Emitir para utilizadores da sala quando o temporizador é parado

Tabela 4: WebSockets disponíveis

4.5 WEBSOCKETS

Para o utilizador poder partilhar vídeos de exercícios ou começar um temporizador e ser comunicado com todos os outros participantes da sala foram desenvolvidos WebSockets em JavaScript recorrendo a node.js e à biblioteca socket.io⁹.

O utilizador quando se liga à videochamada é também ligado ao grupo nos sockets com o id da sala e fica à escuta de partilha de um exemplo de exercício e das alterações ao temporizador.

O utilizador pode também enviar mensagens como partilhar um vídeo de um exemplo de exercício, definir o tempo do temporizador e gerir este mesmo temporizador.

Na Tabela 4 temos todos os canais de WebSockets disponíveis para poder ser possível a comunicação síncrona.

4.6 APLICAÇÃO CLIENTE

A aplicação desenvolvida é uma aplicação Android, foi desenvolvida na linguagem Kotlin¹⁰ e com recurso ao IDE Android Studio¹¹ e com gradle, a gradle utilizada é a versão 4.1.0 que é uma ferramenta de automação de *build* para ser então possível a construção da nossa aplicação.

Para a construção dos nossos layouts é utilizada a biblioteca com.google.android.material:material com a versão 1.2.1, esta biblioteca permite a utilização de Material Components para Android¹².

9 <https://socket.io/> consultado a 23/02/2021

10 <https://kotlinlang.org/>

11 <https://developer.android.com/studio>

12 <https://material.io/develop/android>

Existem quatro atividades principais na aplicação, a inicial onde temos as ações para criar uma sala, juntar a uma sala ou ver os planos de treino, a atividade da videochamada onde é possível estar na videochamada, partilhar exemplos dos exercícios a realizar e controlar o temporizador, a atividade dos planos de treino onde estão presentes os planos de treino disponíveis e a atividade de um plano de treino onde é possível visualizar os exercícios de um dos planos de treino.

Ao criar uma nova sala, a aplicação gera um identificador de 6 algarismos aleatórios para identificar a sala, este mesmo identificador tem de ser utilizado para ser possível o acesso a essa sala. Para ser possível a partilha do identificador dessa sala depois do utilizador criar uma sala tem uma ação para ser possível partilhar esse identificador.

A partilha do identificador da sala é feita através da aplicação WhatsApp com um mensagem pré-preenchida, em que o utilizar apenas tem de escolher o contacto a qual quer enviar a mensagem e também no momento antes do envio pode editar essa mesma mensagem.

O WebRTC é utilizado pelo SDK do Jitsi para a comunicação com o nosso servidor e ser possível a criação de salas, juntar a salas e ter todas as funcionalidades da videochamada na nossa aplicação. O SDK do Jitsi utilizado é a versão 2.10.0 sendo esta a biblioteca responsável pela comunicação com o servidor, pela vista da videochamada e todas as funcionalidades relacionadas com a mesma, como o *chat*, a gestão da câmara e do microfone, o desligar da videochamada actual, a criação de salas e a possibilidade de juntar a uma sala já existente. Este SDK fornece algumas classes de atividades e vistas, a atividade principal deste SDK é a *JitsiMeetActiviy* que mostra uma única vista, a *JitsiMeetView* que é a vista utilizada para mostrar a videochamada e realizar todas as ações relacionadas com a mesma. É possível usar a atividade directamente em que renderiza o ecrã total com essa atividade, ou então utilizar apenas a vista e adicionar elementos adicionais ao ecrã da videochamada. Nesta aplicação utilizamos apenas a vista, pois é necessário adicionar elementos externos á videochamada.

Para ser possível a reprodução de recursos de vídeos foram testadas várias hipóteses. Inicialmente foi testado a possibilidade de reproduzir vídeos de YouTube com o recurso à biblioteca oficial da Google¹³ e também foi testado com uma biblioteca desenvolvida pela comunidade¹⁴. Contudo, estes testes não foram bem sucedidos pela razão de quando o vídeo começa a reproduzir o mesmo para. Este problema será por já existir uma *stream* de vídeo ativa na aplicação. Como não foi possível a integração com vídeos do YouTube ainda foi explorada a opção de integrar com o Vimeo, mas sendo que para ter acesso à API deles é necessário ter uma conta com um plano pago essa opção foi descartada. Como não foi possível

13 <https://developers.google.com/youtube/android/player> consultado a 23/02/2021

14 <https://github.com/PierfrancescoSoffritti/android-youtube-player> consultado a 23/02/2021

integrar com uma plataforma externa de vídeos, foi implementado com a vista de vídeo do Android, sendo que os vídeos estão armazenados no nosso servidor. Estes vídeos são utilizados com um link HTTPS proveniente do servidor e esse url é utilizado como fonte na `VideoView` integrada.

Para implementação de um *chat* na aplicação, o SDK do Jitsi já tem como base esse *chat* implementado, portanto não foi necessário uma implementação externa. A vista de *chat* é acessada dentro da videochamada e a primeira vez iniciada é necessário configurar um nome a apresentar no *chat* e apenas depois desta configuração é que é possível enviar mensagens a todos os participantes da videochamada.

O temporizador foi desenvolvido recorrendo ao `Chronometer`¹⁵ que já tem implementado todas as funcionalidades de um temporizador simples, para o temporizador funcionar corretamente é necessário configura um tempo base em milissegundos e também que será uma contagem decrescente. Quando este temporizador começa também é necessário definir uma função que é chamada cada vez que este temporizador é alterado para quando este chega ao zero não continue a contar em tempo negativo.

Como a utilização das funcionalidades adicionais ao Jitsi foi desenvolvido usado `WebSockets`, para ser possível a utilização desses mesmo `WebSockets` na aplicação Android recorreremos à biblioteca `Socket.IO`¹⁶ que permite estará escuta e enviar mensagens através dos canais disponíveis. Para ser possível o envio de mensagens em formato JSON foi utilizada a biblioteca `JSONObject`¹⁷.

Para ser possível o envio e recepção de mensagens, é necessário conectar ao servidor, após se conectar ao servidor, é necessário definir a função a executar em cada mensagem recebido do servidor. Para enviar mensagens é necessário chamar o método para emitir mensagens da instância do `Socket.IO`.

Para a visualização dos planos de treino recorreremos a uma `ListView`¹⁸, sendo esta uma lista simples com apenas o nome do plano para representar o plano de treino e usado o `simple_list_item_1` do *layout* do Android.

Para representar a lista dos exercícios de cada plano de treino também foi utilizado uma `ListView`, mas desta vez foi criado um `Adapter`, o `ExerciseAdapter` que estende da class `BaseAdapter` para que possível a customização de cada linha da lista, para tal também foi criado um *layout* para representar cada linha da lista com duas `TextView` lado a lado.

15 <https://developer.android.com/reference/android/widget/Chronometer> consultado a 05/03/2021

16 <https://socket.io/blog/native-socket-io-and-android/> consultado a 23/02/2021

17 <https://developer.android.com/reference/org/json/JSONObject> consultado a 23/02/2021

18 <https://developer.android.com/reference/android/widget/ListView> consultado a 28/04/2021

Para a representação de cada exercício também foi criado a class *Exercise* que tem dois atributos, o *title* que será o nome do exercício a representar e *repetitions* que será o número de repetições a realizar desse exercício.

A aplicação foi disponibilizada através da geração da APK e partilha através de um link de WeTransfer¹⁹. Depois de o utilizador fazer o download tinha de permitir a instalação de aplicações de fontes desconhecidas através dos ficheiros.

¹⁹ <https://wetransfer.com/>

PROTOTIPO E TESTES

5.1 PROTOTIPO

Na Figura 21 temos a ecrã inicial da aplicação em que temos duas opções: a de criar uma sala e a de juntar a uma sala já existente. Ao criar a sala a aplicação gera um identificador aleatório para essa sala. Ao juntar a uma sala já existente é necessário ter o identificador dessa sala pelo que irá aparecer um modal para inserir esse mesmo identificador. Ao ver os planos de treino somos redirecionados para outro ecrã onde eles estão disponíveis. Na mesma figura é possível visualizar o menu principal nos dois modos existentes, o *light* e o *dark mode*, em que a preferência entre os dois é baseada na predefinição do *smartphone* do utilizador.

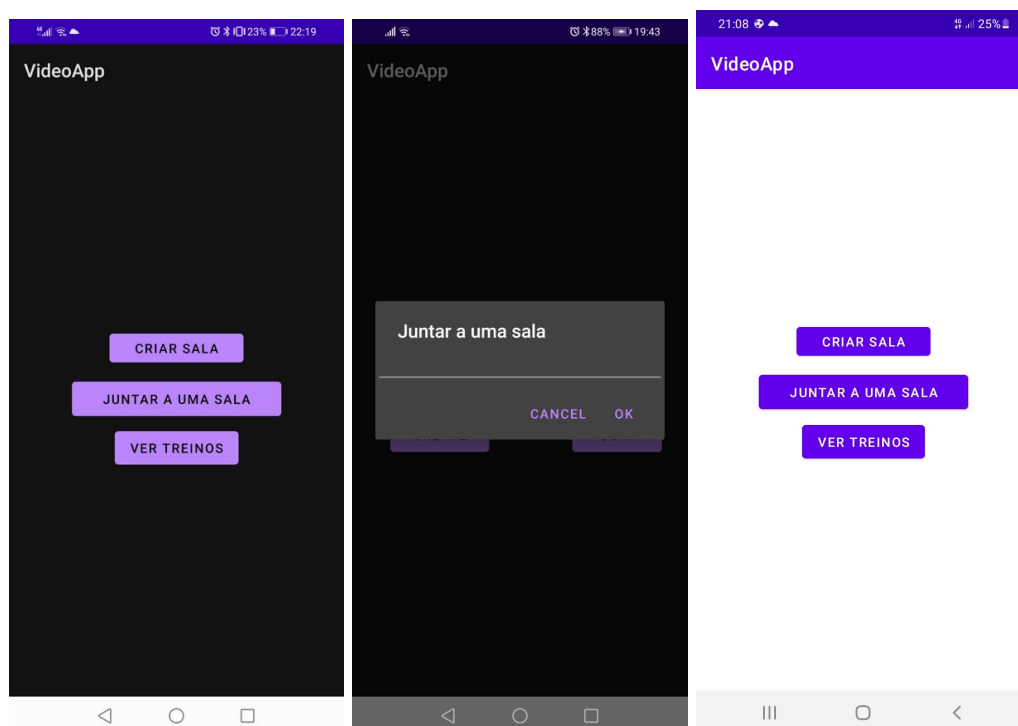


Figura 21: Capturas de ecrã principal e juntar a uma sala

Na Figura 22 temos o ecrã da videochamada, aplicação tem sempre a vista da videochamada Jitsi, esta vista é gerada pelo SDK, e também um botão no canto inferior direito para ser possível realizar algumas ações exteriores ao Jitsi. A vista da videochamada do Jitsi é a vista disponível no SDK, responsável pelas salas, desligar a chamada, o *chat*, o microfone, a câmara, escolha do dispositivo de som e

qual a câmara seleccionada (frontal ou traseira). As ações de partilhar vídeo e gerir o temporizador, são controladas pelos WebSockets. A partilha do identificador da sala é realizado através de uma mensagem WhatsApp.

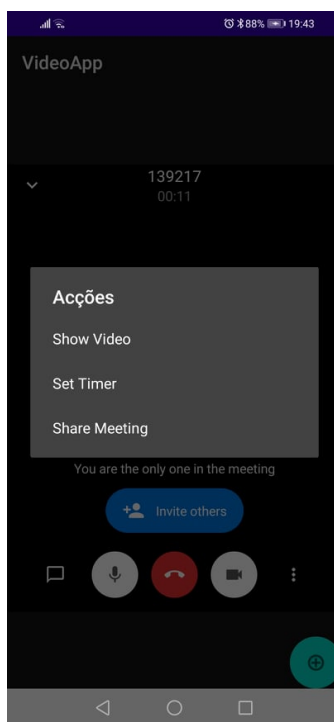


Figura 22: Capturas de ecrã com possíveis acções

Na Figura 23 temos o mecanismo utilizado para partilhar o identificador da sala em que nos encontramos, decidimos usar o WhatsApp com uma mensagem pré-criada com esse identificador. O utilizador apenas tem de seleccionar o contacto para qual pretende enviar esse identificador e a mensagem já irá aparecer pré-preenchida, se o utilizador o decidir pode sempre alterar essa mensagem.

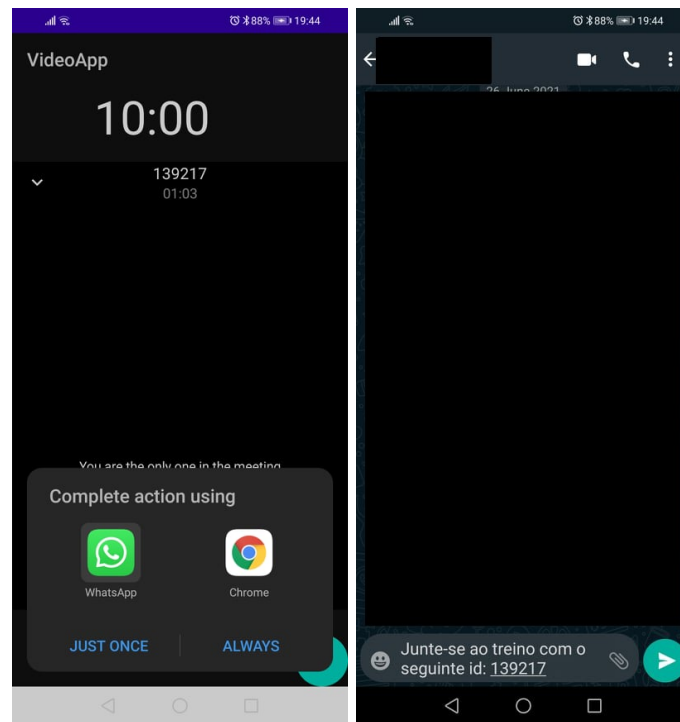


Figura 23: Capturas de ecrã a partilhar identificador da sala

Na Figura 24 temos como utilizar o temporizador o utilizador pode sempre definir o seu tempo através de um modal, e após de definido este fica sempre por cima da vista da videochamada do Jitsi. Este temporizador é sempre partilhado com todos os utilizadores através de WebSockets. Qualquer utilizador pode definir, começar ou para o temporizador.

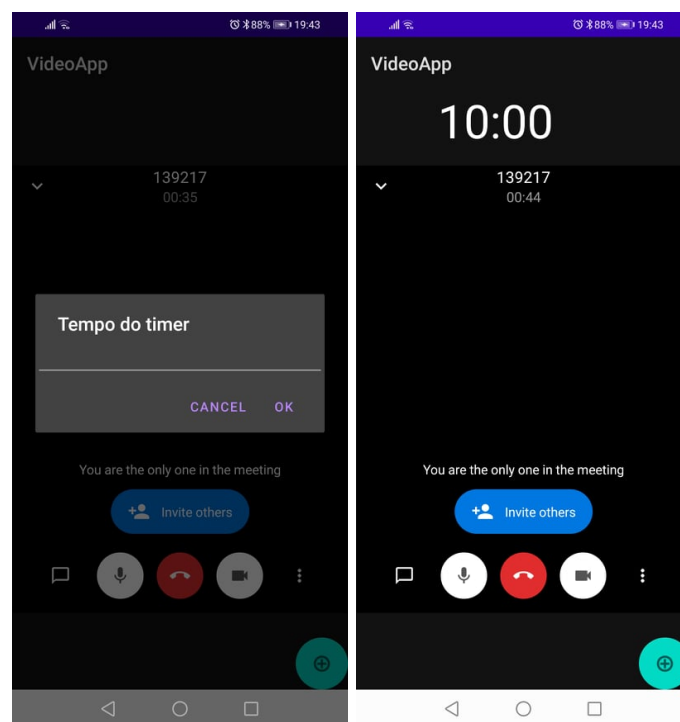


Figura 24: Capturas de ecrã a definir temporizador

Na Figura 25 temos a maneira de como partilhar um vídeo com os utilizadores presentes na sala, basta escolher um dos vídeos presentes na lista e ao selecionar um dos vídeos em todos os participantes é aberto um modal com o reprodutor de media a reproduzir o vídeo previamente selecionado. Após selecionar o vídeo a partilhar este é partilhado a partir de WebSockets, ao ver o vídeo no modal com o reprodutor de media cada utilizador pode fechar esse modal, ao fechar essa modal apenas esse utilizador é que ficaram sem o modal do reprodutor de media, ou seja, ao fechar um modal não é fechado para todos os utilizadores.

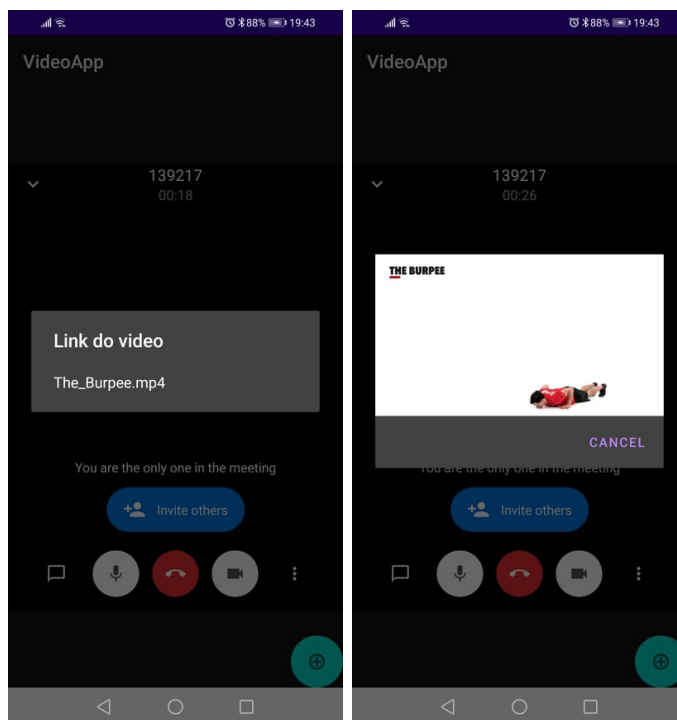


Figura 25: Capturas de ecrã da partilha de vídeos

Na Figura 26 temos a lista de treinos disponíveis e ao carregar num dos planos são apresentados todos os exercícios desse plano. Como se trata de um prototipo e o que é importante neste projeto é a prova de conceito da videochamada, estes exercícios foram pré-criados e não é possível criar, apenas consultar os pré-criados.

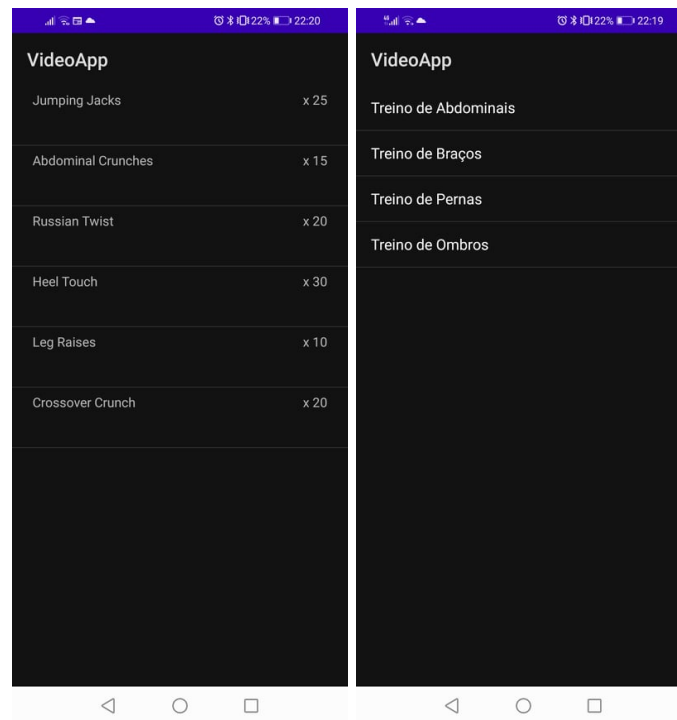


Figura 26: Capturas de ecrã da partilha de vídeos

5.2 TESTES

Para testarmos a usabilidade da nossa aplicação, recorreremos aos mesmo treinadores na Figura 27 que responderam ao nosso questionário, para qual foi construído uma guia com as respetivas condições de sucesso de cada tarefa que está disponível no subcapítulo 5.2.1 e os resultados no subcapítulo 5.2.2.

Cada treinador instalou a aplicação no seu telemóvel e foi completando cada etapa do guia individualmente, no final de completarem todos os passos, foram pedidas algumas opiniões e sugestões de melhoria. Os treinadores estavam organizados em pares para ser possível ter mais que um participante na sala da vídeochamada.

Depois da aplicação instalada foi cada treinador teve de realizar cada ponto do guia de teste, sem saber o que teria de realizar a seguir, pois apenas tinha conhecimento da tarefa a realizar quando lhe era dito e apenas era dito quando acabavam de realizar a tarefa que tinha sido indicada para fazer.



Figura 27: Treinadores a testar a aplicação

5.2.1 *Guia de Teste*

Em baixo estão numeradas as tarefas apresentadas e os seus respetivos objetivos:

1. Ver planos de treino - Chegar com sucesso ao ecrã com a lista de planos de treino
2. Ver exercícios do plano de treino - Chegar com sucesso ao ecrã com a lista de exercícios de um dos planos de Treino
3. Criar uma sala - Criar com sucesso uma sala e chegar ao ecrã da videochamada
4. Partilhar identificador da sala - Partilhar o identificador da sala com um dos seus contactos de WhatsApp
5. Partilhar vídeo - Apresentar com sucesso o vídeo na videochamada
6. Definir Temporizador - Conseguir definir um tempo inicial para o temporizador
7. Começar Temporizador - Conseguir que o temporizador comece a contar
8. Parar Temporizador - Conseguir que depois do temporizador esteja a contar consiga parar
9. Enviar Mensagem no *chat* - Enviar uma mensagem para o *chat* presente na videochamada
10. Desligar a chamada - Conseguir que a videochamada acabe

5.2.2 *Resultados*

As opiniões foram bastante positivas, sendo que é uma aplicação um pouco diferente ao que estavam habituados em que se torna um pouco mais fácil dar um treino,

todos referiram que é uma aplicação que daria muito jeito durante o confinamento para auxiliar nos treinos *online*. A métrica de avaliação utilizada foi se o treinador necessitou de ajuda a realizar algum ponto do guia ou não.

Na podemos observar os resultados dos testes realizados, sendo que o Não se refere ao não necessitar e ajuda e o Sim corresponde se o treinador necessitou de ajuda para realizar a tarefa, com base neste resultados podemos afirmar que a nossa aplicação tem uma utilização bastante intuitiva, sendo que existem alguns pontos que podem ser melhorados. Apesar de em algumas tarefas alguns dos treinadores necessitar de ajuda a realizar, todos realizaram todas as tarefas com sucesso.

	Treinador 1	Treinador 2	Treinador 3	Treinador 4
Ver planos de treino	Não	Não	Não	Não
Ver exercícios do plano de treino	Não	Não	Não	Não
Criar um sala	Não	Não	Não	Não
Partilhar id da sala	Não	Não	Sim	Sim
Partilhar video	Não	Não	Não	Não
Definir Temporizador	Não	Não	Não	Não
Começar Temporizador	Sim	Sim	Sim	Sim
Parar Temporizador	Não	Não	Não	Não
Enviar Mensagem no chat	Sim	Sim	Sim	Sim
Desligar a chamada	Não	Não	Não	Não

Tabela 5: Resultados dos Testes: Pedidos de Ajuda para completar a tarefa

5.2.3 Propostas de Alteração

Após todos os treinadores terem acabado de realizar as tarefas do guia, foram todos reunidos e perguntar se tinham algumas sugestões de melhoria a fazer na aplicação.

Na lista seguinte temos todas as sugestões que estes treinadores sugeriram alterar na aplicação, a ordem pela qual se apresentam é a apenas a ordem pela qual foram sugeridas e não pelo o nível de importância.

1. Temporizador sempre presente
2. Controlo do Temporizador com botões à frente do temporizador
3. Quando o temporizador parar emitir um som
4. Mostrar o exercício na lista de exercícios de cada plano de treino
5. Ver o plano de treino durante a videochamada
6. Várias opções de temporizador, por exemplo, ser a cada três

7. Cada participante poder para o temporizador e guardar esse registo como um recorde desse treino do participante
8. Falar directamente com uma pessoa
9. Só uma pessoa controlar os microfones

Os pontos 1, 2, 3 e 4 decidamos fazer essas alterações, pois são alterações bastante simples e conseguem alterar bastante a experiência do utilizador.

Enquanto os pontos 5, 6, 7, 8 e 9 decidimos deixar para uma implementação futura, pois envolve algumas necessidades de autenticação e permissões, e como o objectivo deste projeto é o estudo e aplicação da adoção de tecnologias de vídeo como uma forma de comunicação síncrona e não a implementação de um sistema de uma aplicação de videochamadas.

5.3 REVISÃO DA APLICAÇÃO

Como referido no capítulo anterior, depois da execução dos testes, os treinadores sugeriram algumas alterações à aplicação, sendo que consideramos algumas mais importantes que outra, como explicado na subsecção 5.2.3.

Como visto na Figura 24, o nosso temporizador não estava sempre presente na videochamada e apenas era visível após ser definido. Também as suas ações eram apenas acessíveis após carregar no botão situado no canto inferior direito, mas como sugerido pelos treinadores foi alterado e agora podemos visualizar sempre o temporizador e ter as suas ações mais acessíveis 28.

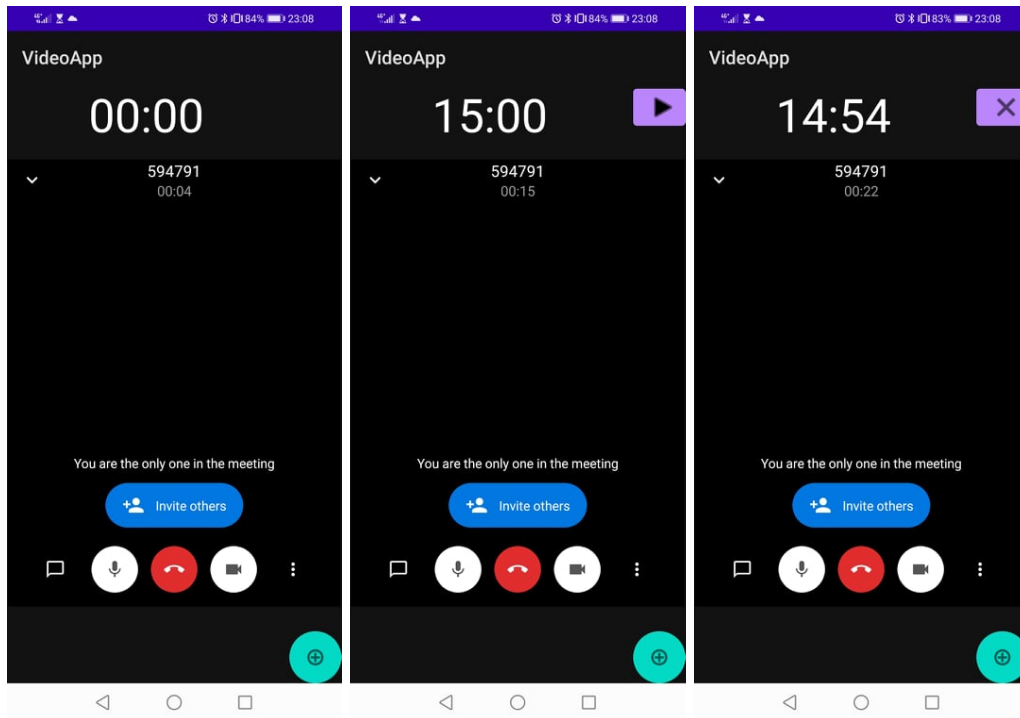


Figura 28: Alteração ao temporizador

Também a nossa lista de exercícios de um plano de treino, tinha apenas a informação do número de repetições de cada exercício. No entanto, como sugerido agora ao carregar num desses exercícios é possível ver a sua exemplificação [29](#).

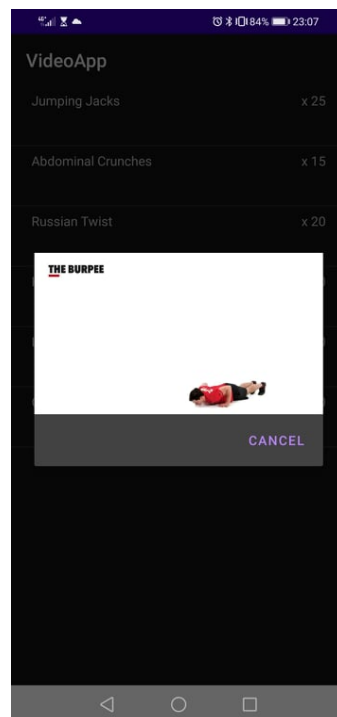


Figura 29: Mostrar o exemplo do exercício

CONCLUSÕES E TRABALHO FUTURO

6.1 CONCLUSÕES

Com a pandemia Covid-19 a alterar todas as nossas formas de comunicação e as soluções de mercado serem soluções muito gerais e muitas vezes algumas áreas de trabalho não conseguir providenciar um serviço completo com as mesma era necessário existir algo mais no mercado.

Sendo o objetivo deste projeto o estudo e aplicação de uma forma de comunicação síncrona a uma videochamada para que esta não seja apenas uma videochamada geral mas algo específico para um caso de estudo, com funcionalidades que maioritariamente apenas fazem sentido neste, considero que consegui comprovar com sucesso a aplicação de uma forma de comunicação síncrona em conjunto com uma videochamada para melhorar a qualidade de treino realizado á distancia.

Este sucesso não seria possível sem previamente realizar levantamentos de requisitos a aplicações já existentes e a profissionais da área.

Neste caso apenas foi possível com o recurso a tecnologias como WebRTC e WebSockets sendo que os WebSockets é a tecnologia mais importante para ser possível a implementação de uma comunicação síncrona para a partilha de informação entre utilizadores da mesma sala relativamente a funcionalidades mais específicas do nosso caso de estudo. O WebRTC é bastante importante, pois sem esta tecnologia não seria possível ter a videochamada em si.

Em relação aos testes e levantamento de requisitos existiram algumas limitações com o facto de não ser possível ter uma comunicação presencial regular e de os horários de os profissionais da área não ser compatível com os meus.

6.2 TRABALHO FUTURO

Com trabalho futuro penso que seria importante a implementação de um sistema de autenticação e de permissões para que nem as pessoas possam controlar as mesmas coisas durante a videochamada.

Também algumas melhorias a algumas das funcionalidades existentes como a possibilidade de guardar o registo do temporizador quando um dos participantes termina o treino e ter então um registo de tempos de treino.

Algo que também será necessário para esta aplicação passar a um estado de produção será a possibilidade de criação de planos de treino e a possibilidade de importação dos exemplos de exercícios.

BIBLIOGRAFIA

- Amirante, A. et al. (2015). «Performance analysis of the Janus WebRTC gateway». Em: *Proceedings of the 1st Workshop on All-Web Real-Time Systems*. <https://doi.org/10.1145/2749215.2749223>.
- Anouk, Middelweerd et al. (2014). «Apps to promote physical activity among adults: a review and content analysis». Em: <https://ijbnpa.biomedcentral.com/articles/10.1186/s12966-014-0097-9>.
- E, Knight et al. (2015). «Public Health Guidelines for Physical Activity: Is There an App for That? A Review of Android and Apple App Stores». Em: <https://mhealth.jmir.org/2015/2/e43>.
- Ivov, Emil (2013). «Hangout-like Video Conferences with Jitsi Videobridge and XMPP». Em: https://2013.jres.org/archives/169/paper169_article.pdf.
- Kumar, Pandey Anoop et al. (2020). «e-Parakh: Unsupervised Online Examination System». Em: <https://ieeexplore.ieee.org/abstract/document/9293792>.
- P, Bondaronek et al. (2018). «Quality of Publicly Available Physical Activity Apps: Review and Content Analysis». Em: <https://mhealth.jmir.org/2018/3/e53>.
- Sally, Eltenahy et al. (2021). «Conversion of Videoconference Speech into Text based on WebRTC and Web Speech APIs». Em: <https://ieeexplore.ieee.org/document/9513968>.

APÊNDICES



APÊNDICE A

1. Antes da pandemia
 - 1.1 Havia limite de participantes por treino?
 - 1.2 Como era controlado o numero limite de participantes, se este existir?
2. Durante o confinamento
 - 2.1 Percentagem de aulas pré-gravadas? Percentagem aulas ao vivo?
 - 2.2 Se foram aulas pré-gravadas, a razão pela qual não tentaram aplicações de vídeo chamada e se chegaram a tentar as aplicações de vídeo chamada
 - 2.3 Acham que aulas online é uma forma viável de ter aulas?
 - 2.4 Se havia limite de participantes, esse limite manteve-se ou não?
3. Actualmente
 - 3.1 Percentagem de aulas pré-gravadas? Percentagem aulas ao vivo?
 - 3.2 Se têm aulas online, acham que mesmo depois de passar a pandemia essas aulas vão se manter online?
 - 3.3 Se havia limite de participantes, esse número de participantes mudou?
 - 3.4 Número de aulas reduziu?
4. Aplicação de vídeo chamadas para treino
 - 4.1 Tem alguma experiência com este tipo de aplicações?
 - 4.2 Quais são as expectativas para uma aplicação deste tipo?
 - 4.3 Qual o dispositivo principal do coach a dar o treino?
 - 4.4 É importante ver os participantes a fazer o exercício?
 - 4.5 Ter a possibilidade de enviar mensagens para os participantes (todos/individualmente), as mensagens aparecerem no ecrã da pessoa?
 - 4.6 Seria útil ter uma imagem/animação para exemplificar o exercício?
 - 4.7 Faz sentido existir limite de participantes numa aula online?
 - 4.8 Apenas o coach controlar os microfones dos participantes? (Evitar ruído desnecessário)
 - 4.9 Importante ter timer no ecrã com o tempo restante do treino?

- 4.10 Atualmente muitas aplicações de chamadas de vídeo tem a possibilidade de ter um fundo customizado, embelezar a pessoa, e utilizar avatars. No conceito de treinos quais destas funcionalidades fazem sentido?
- 4.11 Permitir gravar o workout, tanto do lado do coach como do lado do participante?
- 4.12 Poder retirar pessoas da sala?
- 4.13 Existem estatísticas importantes a reter (número de participantes)?

APÊNDICE B

Treinador 1

1. Antes da pandemia
 - 1.1 22 participantes
 - 1.2 Controlado através da app do CrossFit
2. Durante o confinamento
 - 2.1 100
 - 2.2 Iam começar a implementar aulas por video chamada mas depois ocorreu os desconfinamento
 - 2.3 Para turmas pequenas sim
 - 2.4 Não havia limite
3. Actualmente
 - 3.1 Presenciais, e tem uma aula que é pré gravada
 - 3.2 Continuar online
 - 3.3 Baixou para 12 participantes
 - 3.4 Aumentou o número de aulas mas adicionaram com um novo espaço
4. Aplicação de vídeo chamadas para treino
 - 4.1 Não
 - 4.2 Ver as pessoas e por dar feedback específico
 - 4.3 Tablet
 - 4.4 Sim
 - 4.5 Sim ou então falar direta com uma pessoa
 - 4.6 Animação é importante
 - 4.7 Sim para controlo 10-15 pessoas
 - 4.8 Sim o coach controla os microfones
 - 4.9 Sim mostrar o timer
 - 4.10 Nenhuma é importante

4.11 Sim é importante mas o atleta tem de ter permissão

4.12 Sim

4.13 Nº de participantes por coach, Nº de aulas por coach, Nº de participantes por aula

Treinador 2

1. Antes da pandemia

1.1 22 participantes

1.2 Controlado através da app do CrossFit

2. Durante o confinamento

2.1 100

2.2 Iam começar a implementar aulas por video chamada mas depois ocorreu os desconfinamento

2.3 Para turmas pequenas sim

2.4 Não havia limite

3. Actualmente

3.1 Presenciais, e tem uma aula que é pré gravada

3.2 Continuar online

3.3 Baixou para 12 participantes

3.4 Aumentou o número de aulas mas adicionaram com um novo espaço

4. Aplicação de vídeo chamadas para treino

4.1 Não

4.2 Muita expectativa, nada existe nada no mercado e faz falta

4.3 Telemóvel

4.4 Sim (qualidade de imagem)

4.5 Sim

4.6 Sim (mas que seja possível retirar)

4.7 Sim para controlo máximo 18 pessoas

4.8 Não

4.9 Sim mostrar o timer

4.10 Fundo

4.11 Sim

4.12 Sim

4.13 Espaçamento entre aulas, assiduidade dos atletas

Treinador 3

1. Antes da pandemia

1.1 22 participantes

1.2 Controlado através da app do CrossFit

2. Durante o confinamento

2.1 100

2.2 Iam começar a implementar aulas por video chamada mas depois ocorreu os desconfinamento

2.3 Para turmas pequenas sim

2.4 Não havia limite

3. Actualmente

3.1 Presenciais, e tem uma aula que é pré gravada

3.2 Continuar online

3.3 Baixou para 12 participantes

3.4 Aumentou o número de aulas mas adicionaram com um novo espaço

4. Aplicação de vídeo chamadas para treino

4.1 Sim

4.2 Qualidade de som/imagem. Não ter falhas. Partilhar ecrã (vídeos/música)

4.3 Telemóvel/ Tablet

4.4 Sim, muito

4.5 Não acho relevante, visto que podemos falar

4.6 Sim, seria interessante

4.7 Sim para que seja possível ver todos os participantes

4.8 Sim

4.9 Sim

4.10 Não é necessário

4.11 Não

4.12 Sim

4.13 Não

Treinador 4

1. Antes da pandemia
 - 1.1 22 participantes
 - 1.2 Controlado através da app do CrossFit
2. Durante o confinamento
 - 2.1 100
 - 2.2 Iam começar a implementar aulas por video chamada mas depois ocorreu os desconfinamento
 - 2.3 Para turmas pequenas sim
 - 2.4 Não havia limite
3. Actualmente
 - 3.1 Presenciais, e tem uma aula que é pré gravada
 - 3.2 Continuar online
 - 3.3 Baixou para 12 participantes
 - 3.4 Aumentou o número de aulas mas adicionaram com um novo espaço
4. Aplicação de vídeo chamadas para treino
 - 4.1 Sim
 - 4.2 Não existir tantas
 - 4.3 Telemóvel
 - 4.4 Sim
 - 4.5 Era interessante
 - 4.6 Sim
 - 4.7 Sim
 - 4.8 Sim
 - 4.9 Sim
 - 4.10 O mais simples possível
 - 4.11 Sim
 - 4.12 Sim
 - 4.13 O número de participantes

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado neste projecto, com o título “*Video GUICHÊ*”, é original e foi realizado por Miguel Ângelo dos Santos Ferreira (2192403) sob orientação de Professor Doutor Luis Filipe Fernandes Silva Marcelino (luis.marcelino@ipleiria.pt).

Leiria, Setembro de 2021

Miguel Ângelo dos Santos FERREIRA

Miguel Ângelo dos Santos Ferreira

LINKS

Código Aplicação https://github.com/miguel4747/video_guiche

Código WebSockets https://github.com/miguel4747/video_guiche_sockets

Vídeo de Apresentação https://drive.google.com/file/d/15edS58mZ6AViNsGr_tFVavrvr2ci8p50/view?usp=sharing

APK https://drive.google.com/file/d/1NKZChd464XWzAc_e7fg0MImyUkT99Rd7/view?usp=sharing