

Rafael Claro (88860), Fábio Alves (108016), Bruno Pereira (112726), Rafael Dias (114258)

Versão deste relatório: **2022-05-14**, v1.0

RELATÓRIO – *ELABORATION & CONSTRUCTION*

Construção

Índice

1.1	Sumário executivo	2
1.2	Controlo de versões	2
1.3	Referências e recursos suplementares	2
2.	Arquitetura do sistema.....	3
2.1	Objetivos gerais	3
2.2	Requisitos com impacto na arquitetura	3
2.3	Decisões e justificação.....	4
2.4	Arquitetura do software.....	5
2.5	Arquitetura física de instalação	6
3.	Incremento 1	7
3.1	Casos de utilização no Incremento 1	7
3.2	Histórias de utilização selecionadas.....	8
3.3	Estratégia e estado da implementação	9

1. Introdução

1.1 Sumário executivo

Este relatório cobre a execução e o resultado do primeiro incremento de desenvolvimento, essencial para validar a arquitetura proposta e expandir as funcionalidades críticas do sistema.

O primeiro incremento teve como foco a validação da arquitetura do sistema. Ao longo deste período, foram implementadas funcionalidades essenciais que constituem o núcleo do negócio, o que garante que a base tecnológica seja sólida e eficaz. Esta etapa serviu como uma prova de conceito para as abordagens tecnológicas adotadas e preparou o terreno para expansões futuras.

1.2 Controlo de versões

Quando?	Responsável	Alterações significativas
11/5	Rafael Dias	2.1, 2.4, 2.5
12/05	Bruno Pereira	Sumário executivo, Referências e recursos suplementares, Casos de utilização no Incremento 1, Histórias de utilização selecionadas
12/05	Rafael Claro	Arquitetura de sistema
14/05	Rafael Claro	Apresentação - PowerPoint
14/05	Bruno Pereira	Apresentação - PowerPoint
22/05	Bruno Pereira	Estratégia e estado da implementação
22/05	Fábio Alves	APP WorkOut+

1.3 Referências e recursos suplementares

Para apoiar o desenvolvimento do sistema de informação da WorkOut foram utilizadas uma série de referências e recursos suplementares, garantindo assim que o projeto fosse baseado em informações sólidas e práticas do mercado. Documentos internos da empresa, tais como relatórios estratégicos e estudos de mercado prévios, ajudaram a alinhar o desenvolvimento do sistema com os objetivos de negócio da empresa.

Para entender melhor o ambiente de mercado e posicionar o sistema de forma competitiva foram realizadas pesquisas de mercado e análises competitivas detalhadas. Estes estudos ajudaram a identificar tendências do setor de fitness e bem-estar e a adaptar o sistema de forma a atender às expectativas do mercado.

Para além disso, os estudos académicos e técnicos sobre tecnologias emergentes e metodologias de desenvolvimento de sistemas influenciaram as decisões tecnológicas e garantiram que as soluções implementadas seguissem as práticas de vanguarda.

Por fim, também se teve em consideração os padrões e regulamentos pertinentes do setor de forma a garantir que o sistema cumprisse com todos os requisitos legais, especialmente em áreas críticas como segurança de dados e privacidade do utilizador.

2. Arquitetura do sistema

2.1 Objetivos gerais

1. Avaliar a necessidade de integração com sistemas legados existentes na WorkOut, como sistemas de gestão de membros ou de monitorização de desempenho físico;
2. Avaliar o contexto de uso do sistema para identificar possíveis problemas de desempenho, especialmente durante picos de atividade;
3. Priorizar a operação contínua e sem interrupções do sistema, garantindo alta disponibilidade e tolerância a falhas;
4. Definir as plataformas suportadas pelo sistema, incluindo web, dispositivos móveis (iOS e Android);
5. Identificar quaisquer restrições já conhecidas para a implementação, como suporte a sistemas legados específicos da WorkOut ou restrições de tecnologia impostas pela infraestrutura existente;
6. Avaliar o uso de tecnologias específicas que possam ser mais adequadas para atender aos requisitos do projeto;
7. Os utilizadores devem receber notificações sobre o começo de um novo desafio;
8. Os utilizadores deverão ter acesso à plataforma de comunicação com o PT a qualquer hora do dia.

2.2 Requisitos com impacto na arquitetura

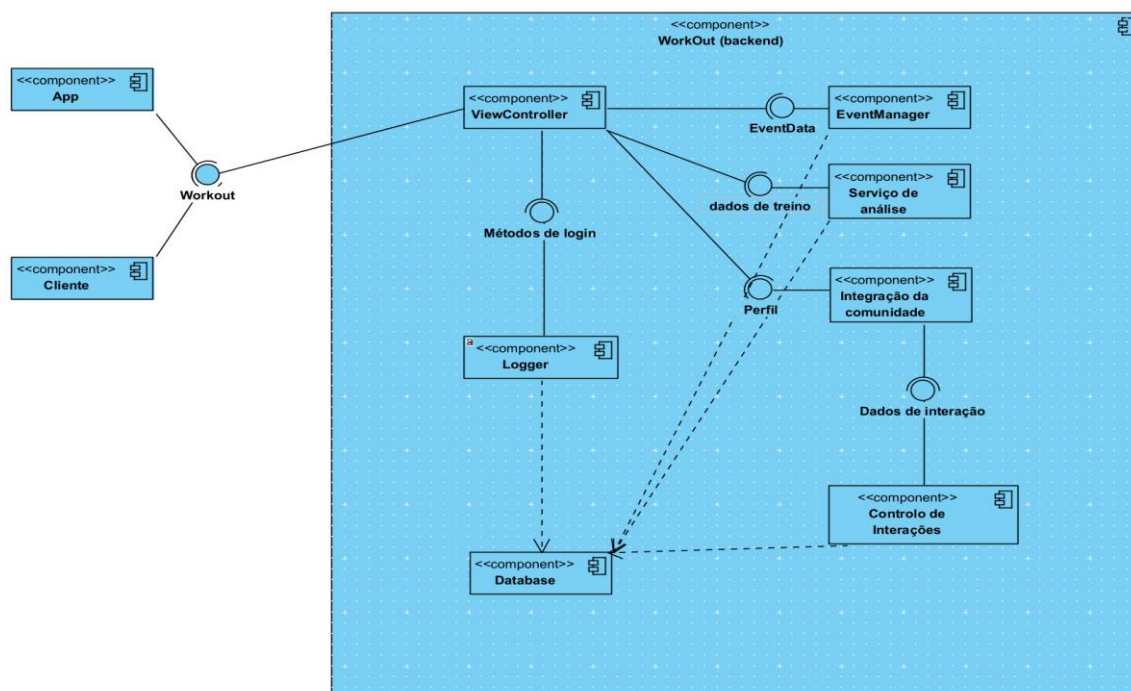
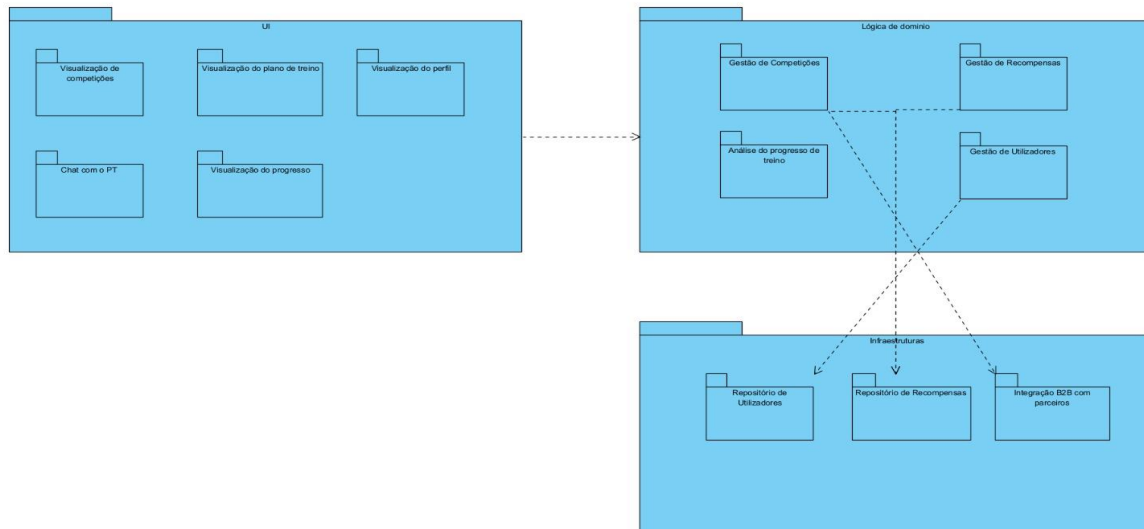
Referência	Descrição do Requisito Não-Funcional
RInt.1	O sistema deve apresentar uma interface amigável e intuitiva, adequada mesmo para utilizadores com pouca experiência tecnológica.
RInt.2	O sistema deve disponibilizar recursos que facilitem a visualização do progresso dos clientes, como gráficos.
RDes.1	O sistema deve ser responsivo e rápido, com tempos de carregamento reduzidos.

RDes.2	O sistema deve ser capaz de processar um grande volume de estatísticas dos clientes, sem perda de desempenho.
RSeg.1	Deve ser garantida a segurança e privacidade dos dados pessoais dos clientes e dos Personal Trainers.
RSeg.2	Deve ser assegurada a fiabilidade dos planos de treino e do registo das atividades físicas, prevenindo a adulteração ou falsificação.

2.3 Decisões e justificação

- O Android Studio foi a plataforma escolhida para a criação da app por ser uma IDE oficial para desenvolvimento Android, tendo por sua vez um bom suporte à linguagem Kotlin.
- A linguagem Kotlin foi escolhida por ser uma linguagem moderna e segura, que oferece clareza e concisão superior ao Java.
- Escolheu-se o uso de XML para a definição das interfaces dos utilizadores pois permite uma clara separação entre apresentação e lógica, facilitando desta maneira a manutenção e a realização de atualizações.
- A escolha do Firebase como solução de backend teve como objetivo o aproveitamento das suas capacidades de armazenamento de dados e autenticação.
- O Firebase suporta uma sincronização de dados em tempo real e facilita a implementação de funcionalidades mais complexas, como notificações push e análise de dados, também simplificando a manutenção da infraestrutura.

2.4 Arquitetura do software

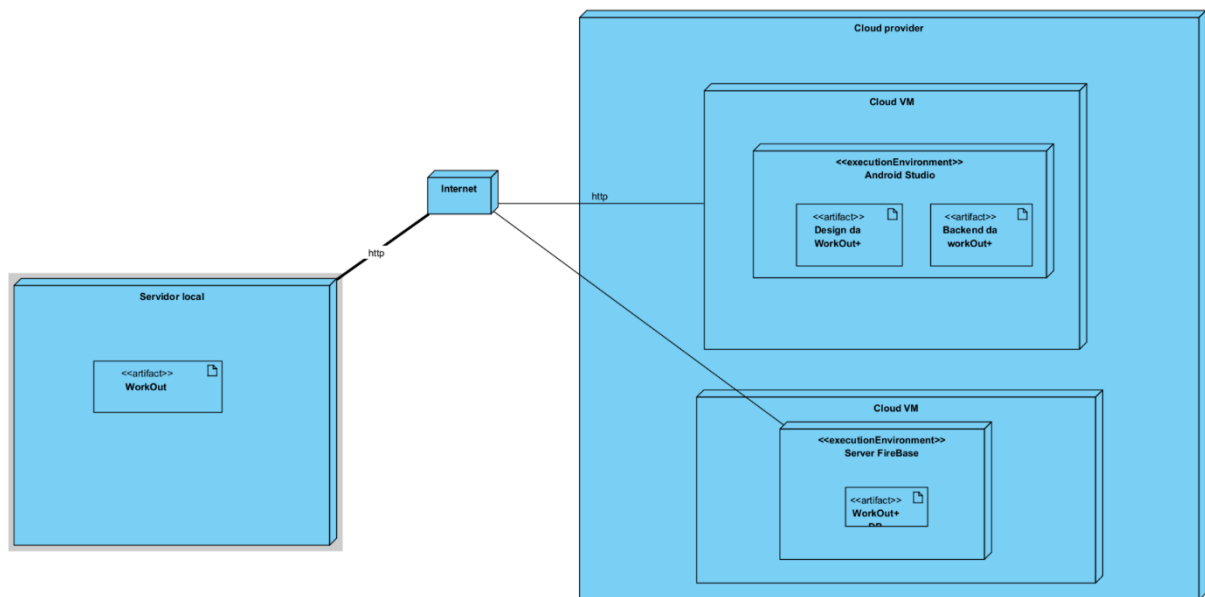


A articulação entre os módulos decorre da seguinte forma:

Os módulos de um sistema colaboram entre si através de interfaces bem definidas, que permitem a troca de informações e a execução de funções específicas. Aqui está uma explicação simplificada de como os módulos podem colaborar:

- **Comunicação:** Os módulos comunicam entre si através de chamadas de função, eventos ou mensagens. Por exemplo, o módulo "App" pode chamar uma função no módulo "Workout" para iniciar um treino.

- **Dependências:** Alguns módulos podem depender de outros para funcionar corretamente. Por exemplo, o módulo “Logger” pode depender do módulo “Db” para armazenar logs.
- **Fluxo de Dados:** Os módulos podem passar dados entre si. Por exemplo, o módulo “Métodos de login” pode passar os dados do utilizador para o módulo “Cliente” após um login bem-sucedido.
- **Orquestração:** Alguns módulos podem coordenar as ações de outros módulos. Por exemplo, o módulo “ViewController” pode orquestrar a interação entre o módulo “App” e o módulo “Cliente”.



2.5 Arquitetura física de instalação

O sistema é composto por um servidor local e um servidor na nuvem, ambos colaborando para suportar a aplicação “WorkOut”.

No servidor local, existe um componente chamado “WorkOut” que executa funções específicas do sistema. Este servidor local comunica-se com o servidor na nuvem através de uma conexão HTTP, permitindo a troca de informações e a coordenação de tarefas.

No servidor na nuvem, existem dois ambientes de execução distintos: o Android Studio e o Server FireBase. Cada um desses ambientes hospeda diferentes componentes da aplicação “WorkOut”.

O ambiente de execução do Android Studio contém dois artefatos: o “Design da WorkOut+” e o “Backend da workout+”. O “Design da WorkOut+” é responsável pela interface do usuário e a experiência do usuário, enquanto o “Backend da workout+” lida com a lógica de negócios e o processamento de dados.

Por outro lado, o ambiente de execução do Server FireBase hospeda um artefato chamado “WorkOut+”. Este componente pode ser responsável por funções como autenticação de usuários, gerenciamento de dados e integração com outros serviços.

Esta arquitetura permite que o sistema seja escalável e flexível, com diferentes componentes trabalhando juntos para fornecer um serviço eficiente e eficaz.

3. Incremento 1

3.1 Casos de utilização no Incremento 1

O primeiro incremento foca-se principalmente na validação da arquitetura do sistema. Este incremento foi escolhido de forma estratégica para implementar e testar funcionalidades chave que são essenciais para o núcleo do negócio da empresa.

Durante este incremento, foram selecionados casos de utilização com o intuito de verificar a integridade e a eficácia da arquitetura proposta, garantindo assim que ela atendesse aos requisitos funcionais e operacionais críticos do sistema. Os casos de utilização implementados abrangeram desde o gerenciamento dos perfis dos utilizadores até ao registo e monitorização das atividades físicas, fornecendo uma amostra representativa das interações dos utilizadores com o sistema.

Após a implementação, o sistema passou por uma série de testes detalhados, cujos quais incluíram testes unitários para verificar a lógica interna de cada componente, testes de integração para assegurar que todos os componentes do sistema funcionassem em conjunto de maneira eficiente, e testes de aceitação para avaliar como os utilizadores interagem com o sistema. Estes testes foram essenciais para identificar problemas e pontos de melhoria, permitindo desta forma que fossem feitas melhorias iterativas com base no feedback recebido.

Para além disso, os comentários dos utilizadores forneceram informações diretas sobre a experiência deste mesmo e como o sistema poderia ser melhorado de forma a atender melhor às suas necessidades. Este ciclo contínuo de feedback ajudou a melhorar a interface do utilizador e a funcionalidade geral do sistema.

A especificação detalhada dos casos de utilização encontra-se no anexo B do relatório de Análise (E3). A partir dessa análise, definiram-se as histórias de utilização a implementar.

3.2 Histórias de utilização selecionadas

As histórias (*user stories*) incluídas nesta interação fazem parte do *backlog* do projeto, acessíveis em

<https://projetoas1.atlassian.net/jira/software/projects/WOR/boards/1/backlog?atlOrigin=eyJpIjoiNDU1NzgwODUxNDgxNDQ4Y2E1NmQyNTgzMjQ0Y2UwMGUiLCJwIjoiajJ9>

Histórias incluídas nesta interação:

História/ <i>use case slice</i>	CrITÉrios de aceitação
A Joana (cliente) quer criar uma conta e definir objetivos.	<ul style="list-style-type: none">- A Joana acede ao aplicativo.- Ela é direcionada para a página de criação de conta.- Após a criação da conta a Joana insere as suas informações pessoais, tais como o nome, peso, altura, etc.- Depois disso é solicitado à Joana que defina os seus objetivos de fitness.- Ela escolhe “Perda de Peso” como o seu principal objetivo. e define metas específicas relacionadas com esse objetivos como, por exemplo, perder 5Kg em 3 meses.
A Joana (cliente) quer um Personal Trainer (PT) e quer ter acesso a um plano de treino.	<ul style="list-style-type: none">- A Joana acede à lista de PT’s disponíveis.- Ela pesquisa conforme as especialidades, avaliações e afinidades de treino.- A Joana escolhe então um PT que lhe pareça estar melhor alinhado com os seus objetivos e preferências de treino.- Uma vez que o PT já desenvolveu um plano de treino personalizado, a Joana já tem acesso a este mesmo, cujo qual inclui exercícios específicos, frequência e intensidade recomendada e informações sobre como acompanhar o seu progresso.
A Joana (cliente) quer acompanhar o seu progresso.	<ul style="list-style-type: none">- A Joana acompanha o seu progresso através de gráficos gerados a partir de dados que ela fornece sobre os seus treinos.- Ela visualiza o seu peso ao longo do tempo, a quantidade de calorias queimadas em cada treino e o cumprimento dos seus objetivos.- A Joana avalia o seu progresso e faz ajustes conforme necessário no seu plano de treino e estilo de vida.

3.3 Estratégia e estado da implementação

A estratégia de implementação adotada para este incremento focou-se no desenvolvimento da interface do cliente, utilizando diversas tecnologias de forma a assegurar uma boa experiência ao utilizador e um desenvolvimento eficiente. Optou-se pela utilização do Android Studio como IDE de desenvolvimento, pois esta é a plataforma oficial de desenvolvimento do Android, facilitando a integração dos recursos e serviços da Google. A linguagem Kotlin foi escolhida como linguagem de programação principal devido à sua eficácia e segurança. Para além disso, esta foi complementada com o uso de XML para definir as interfaces dos utilizadores, separando desta forma a lógica de negócios da apresentação.

Para o backend, escolhemos o Firebase devido às suas capacidades de autenticação e armazenamento de dados, bem como a sua sincronização de dados em tempo real, que é essencial para as atualizações do treino e do progresso dos utilizadores. A implementação foi concluída com sucesso, com todas as funcionalidades operacionais planeadas para esta implementação e correspondendo às expectativas iniciais. Os casos de uso, nomeadamente o login e registo de utilizadores, a monitorização do progresso através de gráficos e o acesso a planos de treino foram integrados e testados para garantir que funcionam corretamente.

Para garantir a qualidade e o desempenho do sistema foram realizados testes unitários, de integração e de aceitação. O feedback inicial dos utilizadores foi bastante positivo, demonstrando a eficácia da nossa abordagem.