

## **Entrega Trabalho Prático 4**

**Projeto Aplicado**

**Aluno/os:**

21140 - Pedro Vieira Simões  
21145 – Gonçalo Moreira da Cunha  
21152 – João Carlos da Costa Apresentação

**Professor/es: Eduardo Peixoto**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, janeiro de 2023

# IPCA GYM

### Resumo

Este trabalho prático, relativo à unidade curricular de **Projeto Aplicado**, propende a melhorar a performance de trabalho em equipa num desafio que irá explorar as necessidades de um smart campus, no IPCA e demonstrar técnicas e conceitos abordados inter e extracurricular.

A ideia do projeto será um sistema para um ginásio e uma aplicação para os utilizadores. Ambas serão aplicações mobile e ainda terá um sistema de gestão de acessos carregado por um hardware com uso de cartões para o ginásio.

## Índice

Resumo .....	3
Índice de Imagens.....	6
1. Introdução.....	7
1.1. Contextualização .....	7
1.2. Motivação e Objetivos.....	7
1.3. Estrutura do Documento .....	7
2. Produto .....	8
2.1. Visão do Produto.....	8
3. Processos gerais do funcionamento do produto.....	9
3.1. Requisitos Funcionais.....	9
3.1.1. Ginásio (Sistema).....	9
3.1.2. Cliente.....	9
3.2. Requisitos Não Funcionais.....	10
4. Organização do Grupo .....	10
5. Stakeholders.....	11
6. Partes não interessadas.....	11
7. Benefícios .....	11
8. Impacto Diário.....	12
9. Cliente .....	12
10. Utilizadores .....	12
11. Modelo Canvas .....	12
11.4.1. Ginásio e seus funcionários.....	13
11.4.2. Utilizadores/Clientes do ginásio .....	13
12. Grelha de autoavaliação.....	14
13. Atas .....	15
13.1. Entrega #1 .....	15
13.1.1. Ata #1 .....	15
13.2. Entrega #2.....	15
13.2.1. Ata #1 .....	15
13.2.2. Ata #2 .....	16
13.2.3. Ata #3 .....	16
13.2.4. Ata #4 .....	17
13.3. Entrega #3.....	17
13.3.1. Ata #1 .....	17
13.3.2. Ata #2 .....	17
13.3.3. Ata #3 .....	18
13.4. Entrega #4.....	18
13.4.1. Ata #1.....	18
13.4.2. Ata #2.....	18
13.4.3. Ata #3.....	19
13.4.4. Ata #4.....	19
14. Diagramas .....	20
14.1. Diagrama Entidade-Relação .....	20

14.2.	Diagrama de Casos de Uso .....	22
14.2.1.	Caso de uso 1 – Gestão de Conta .....	22
14.2.2.	Caso de uso 2 – Loja online.....	23
14.2.3.	Caso de uso 3 - Aplicação .....	24
14.2.4.	Caso de uso 4 – Gestão de lotação .....	25
14.3.	Diagrama de atividades.....	26
14.3.1.	Cancelamento de serviço .....	26
14.3.2.	Reagendamento/Cancelamento de serviço.....	26
14.4.	Diagrama de estados .....	27
14.5.	Diagrama de sequência por ecrã.....	27
15.	Mockups .....	28
15.1.	Cliente .....	28
15.1.1.	Login .....	28
15.1.2.	Menu Principal.....	29
15.1.3.	Clique na foto .....	30
15.1.4.	Conta do utilizador.....	31
15.1.5.	Definições .....	32
15.1.6.	Editar perfil.....	33
15.1.7.	Atividade .....	34
15.1.8.	Treinos.....	35
15.1.9.	Nutrição.....	35
15.1.10.	Loja online .....	36
15.2.	Sistema do Ginásio .....	37
15.2.1.	Capacidade.....	37
15.2.2.	Gestão dos clientes .....	38
15.2.3.	Página inicial .....	39
15.2.4.	Loja online.....	40
15.2.5.	Atividade .....	41
16.	Cronograma.....	42
17.	Código implementado para o projeto .....	43
17.1.	Base de Dados .....	43
17.2.	Back-end .....	43
17.2.1.	Programação por camadas.....	44
17.2.2.	JWT Tokens e Authentication .....	45
17.2.3.	Funcionalidade de Login (Encriptação de password).....	47
17.2.4.	Documentação de código com OpenAPI.....	48
17.3.	Front-end.....	50
17.3.1.	Activities.....	50
17.3.2.	Resources.....	52
17.4.	Componentes físicos.....	53
18.	Avaliação .....	54
19.	Conclusão.....	54
20.	Bibliografia .....	54
21.	Assinaturas .....	55

## Índice de Imagens

Figura 1 - Diagrama de Entidade-Relação .....	20
Figura 2 - Diagrama Casos de Uso 1 (Gestão de Conta).....	22
Figura 3 - Diagrama de Casos de Uso 2 (Loja Online) .....	23
Figura 4 - Diagrama de Casos de Uso 3 (Aplicação) .....	24
Figura 5 - Diagrama de Casos de Uso 4 (Gestão de lotação) .....	25
Figura 6 - Diagrama de Atividades - Cancelar Serviço .....	26
Figura 7 - Diagrama de Atividades - Reagendamento de Serviço.....	26
Figura 8 - Diagrama de Estados - Marcação de Consulta .....	27
Figura 9 - Login do Utilizador .....	28
Figura 10 - Utilizador - Página inicial .....	29
Figura 11 - Utilizador - Menu .....	30
Figura 12 - Utilizador - Conta .....	31
Figura 13 - Utilizador - Definições .....	32
Figura 14 - Utilizador - Editar perfil .....	33
Figura 15 - Utilizador - Atividade .....	34
Figura 16 - Utilizador - Treino.....	35
Figura 17 - Utilizador - Nutrição .....	35
Figura 18 - Utilizador - Loja Online.....	36
Figura 19 - Sistema - Capacidade .....	37
Figura 20 - Sistema - Gestão de clientes.....	38
Figura 21 - Sistema - Página inicial .....	39
Figura 22 - Sistema - Loja online .....	40
Figura 23 - Sistema - Atividade .....	41
Figura 24 - Diagrama de Gantt .....	42
Figura 25 - Microsoft SQL Server Management Studio (SSMS) .....	43
Figura 26 - Dependências API Layer .....	44
Figura 27 - Dependências BLL Layer .....	44
Figura 28 - Dependências DAL Layer .....	44
Figura 29 - Login de um funcionário (back-end).....	45
Figura 30 - Método auxiliar CreateTokenFuncionário() .....	46
Figura 31 - Headers de Authorize num Request HttpPatch .....	46
Figura 32 - Métodos auxiliares CreatePasswordHash() e VerifyPasswordHash().....	47
Figura 33 - Tratamento de password no Request HttpPost de um Funcionário.....	47
Figura 34 - Verificação de palavra-passe no Login de um Funcionário.....	47
Figura 35 - Geração documentação no Swagger .....	48
Figura 36 - Propriedades da Solução do projeto .....	48
Figura 37 - Documentação de métodos .....	49
Figura 38 - Documentação de modelos de dados .....	49
Figura 39 - Resultado documentação no Swagger .....	49
Figura 40 - Organização do Front End (Activities) .....	51
Figura 41 - Organização do Front-End (Resources) .....	52
Figura 43 - Autoavaliação.....	54

## 1. Introdução

### 1.1. Contextualização

Este projeto visa implementar soluções e desafios propostos pelo docente Eduardo Peixoto, consolidando assim os conhecimentos não só adquiridos durante a unidade curricular do projeto, mas também no percurso académico atual.

### 1.2. Motivação e Objetivos

A ideia de um sistema para o ginásio foi originada pela ideia de futuramente o IPCA vir a ter mais instalações à medida que este vai crescendo e desta forma existir uma forma de gerir o mesmo e ainda ajudar os clientes.

Temos por objetivos pessoais:

- Cimentar conhecimentos obtidos ao longo do percurso académico;

Objetivos do projeto:

- Criar um sistema que ajude a gerir um ginásio, ao nível de:
  - Lotação;
  - Utilizadores;
  - Artigos de venda;
  - Fluxo de entradas/saídas;
  - Etc;
- Criar uma aplicação que ajude os utilizadores, ao nível de:
  - Procurar as melhores horas de utilização do ginásio;
  - Facilitar o treino;
  - Entregar uma gama de produtos e planos (nutrição e treino);
  - Etc;

### 1.3. Estrutura do Documento

O documento está estruturado de forma que seja de simples leitura. Existe recurso a referências de material fornecido pelo professor Eduardo Peixoto e/ou referências a excertos de Web grafia.

Este trabalho encontra-se dividido em grupos, de forma a facilitar a procura e associação face ao material fornecido pelo docente.

## 2. Produto

### 2.1. Visão do Produto

Dentro dos subtópicos possíveis encaixados no Smart Campus vai ser abordado a Saúde. Foi decidido toda uma construção em torno do desenvolvimento android que visa à nossa universidade acompanhar a vida saudável e atlética dos estudantes.

O IPKA GYM nasce após notar-se a necessidade desse mesmo acompanhamento e a falta de um setor que permita a atividade aos jovens, no sentido de incentivar aos estudantes a realizar um estilo de vida saudável.

Será então possível aos estudantes terem um acompanhamento mobile da sua atividade física, tal como os diferentes exercícios que pode fazer ao longo do seu treino.

Os gestores do ginásio conseguirão fazer uma monitorização de todas as pessoas inscritas no ginásio, já que, em conjunto com outra unidade curricular, irá ser implementado um sistema externo para gestão de acesso através de um chip/cartão eletrónico.

Este projeto visa alcançar este objetivo através da implementação de uma aplicação Mobile e de hardware de gestão de acessos para que se torne mais cómoda a utilização da mesma.



## 3. Processos gerais do funcionamento do produto

### 3.1. Requisitos Funcionais

#### 3.1.1. Ginásio (Sistema)

- Gestão da lotação atual
  - Quantidade atual (avisos caso esteja perto do limite)
  - Quantidade de pessoas que foram nesse dia
  - Quantidade de pessoas que foram nesse mês
  - Quantidade de pessoas que já saíram
- Gestão de clientes
  - Registo
  - cancelar "contrato"
  - alterar dados biométricos
- Gestão de avaliações
- Gestão de pedidos da loja online
- Histórico de entradas e saídas de cada user (data passagem, dizer se é entrada ou saída, id da pessoa que entrou ou saiu, nome dessa pessoa)

#### 3.1.2. Cliente

- Login
- Dados biométricos
- Planos de treino
  - planos
    - emagrecer
    - massa muscular
    - definir
    - pernas, braços, peito, ...
- Planos de nutrição
- Pode avaliar o ginásio/aplicação
- Marcar consulta de medições/nutrição
- Ver evolução ao nível das medidas musculares, ...
- Gestão de conta
  - editar alguns dados
  - cancelar "contrato"
- Loja online

## 3.2. Requisitos Não Funcionais

- **Implementação:** linguagens Kotlin e C#, com recurso a base de dados SQLServer e hardware Arduino; (é provável que sofra alterações tendo em conta que estamos a ter cadeiras em simultâneo com esta)
- **Usabilidade:** aplicação de uso simples bastante didática;
- **Interface:** aplicação mobile para uma utilização mais flexível quando comparada com desktop, etc;
- **Compatibilidade:** o sistema deverá ser compatível com a versão atual ou mais recente do sistema operativo Android;
- **Eficiência no software:** o software será eficaz sem causar muito tempo de espera entre ações;
- **Interoperabilidade:** sistema terá comunicação com a base de dados.

## 4. Organização do Grupo

O grupo 7 da cadeira de Projeto Aplicado da Licenciatura de Engenharia de Sistemas Informáticos pós-laboral (LESI-PL) é constituído pelos seguintes membros de grupo: Pedro Simões, Gonçalo Cunha e João Apresentação.

O grupo tem como orientador o professor Eduardo Peixoto, que poderá participar nas reuniões de grupo, para auxiliar na concretização dos objetivos do projeto.

A nível de cargos distribuídos pelo grupo, temos:

- Gonçalo Cunha - Product Owner, Programador
- João Apresentação - Scrum Master, Programador
- Pedro Simões - Programador

**Product Owner** tem o papel de fazer a definição da visão do projeto, supervisionamento e avaliação das etapas de desenvolvimento.

**Scrum Master** tem o papel de facilitar as sprints, ajudar a equipa a manter-se concentrada nas sprints e fazer o planeamento e organização da mesma.

**Programador** tem a função desenvolver o software em si, obedecendo às tarefas fornecidas pelo Scrum master e de realizar as suas tarefas até ao fim de cada sprint.

Não está projetado a alteração de cargos entre membros de grupo, exceto em ocasiões especiais, das quais serão devidamente registadas com os membros que sofreram a alteração, data e razões da mesma.

### 5. Stakeholders

- **Ginásio (IPCA):** interesse em oferecer uma plataforma aos seus membros e um sistema de gestão desses aos seus funcionários;
- **Cliente/Membro:** interesse em usar a aplicação e usufruir do ginásio;
- **Funcionário (Personal Trainers/Gerente):** interesse em gerir o estado e propriedades do ginásio;
- **Nutricionista:** interesse pelo fornecimento de dados nutricionais e marcação de serviços relacionados com o mesmo;
- **Fornecedores Comerciais:** interesse pelo fornecimento de roupa e alimentos a serem disponibilizados na loja online;
- **IPCA (ou outro estabelecimento académico):** interesse pela atração de novos estudantes e progressão do estabelecimento.

### 6. Partes não interessadas

Todo o tipo de empresas e estabelecimentos não conectados à indústria da saúde, desporto, alimentar (alimentos de treino) e têxtil (roupa de treino).

### 7. Benefícios

De uma forma generalizada o projeto a ser desenvolvido pretende fundamentar a ideia de uma vida saudável em cooperativo com campus universitário, que não só irá atrair os estudantes como irá tornar este politécnico mais destacado e importado com a saúde dos seus alunos.

De uma forma mais ao nível do que será desenvolvido serão beneficiados os stakeholders pelos seguintes aspetos:

- Aplicação mobile com o intuito de:
  - Promoção de uma vida saudável;
  - Facilitar na ida ao ginásio através da visualização de lotação do estabelecimento;
  - Ajudar no treino através dos planos fornecidos;
  - Ajudar na alimentação/nutrição;
  - Fornecer uma loja online de roupa e alimentação;
  - Sistema de ginásio que facilita a gestão de:
    - Lotação;
    - Experiência dos clientes;
    - Contratos;
    - Dados dos clientes;

### 8. Impacto Diário

Atualmente muitas das vezes consome-se tempo e dinheiro a ir a determinado estabelecimento sendo que este poderá ter atingido a sua lotação máxima ou mesmo ter um número elevado de utilizadores, o que pode ser resolvido através de uma gestão desse tipo de dados. De certa forma este sistema vem como forma de resolver e criar um impacto positivo na sociedade ao nível de desperdício de tempo e recursos. Além disso com a existência de uma App mobile é possível visualizar todo esse acesso de forma simples e eficaz. O cliente ainda terá facilidade nos seus treinos e progressos o que torna toda esta experiência bastante satisfatória. Apesar de tudo isto, o projeto tem como maior impacto a aposta numa vida diariamente saudável de cada estudante.

### 9. Cliente

O cliente do nosso projeto será o ginásio pertencente ao instituto académico (IPCA). Irá lhe ser disposto um sistema de gestão de acessos (cartões e leitor de cartões) juntamente com uma aplicação a ser oferecida aos seus membros/utilizadores.

### 10. Utilizadores

Os utilizadores serão única e exclusivamente aqueles que sejam membros do ginásio e seus clientes. Estes terão um cartão de acesso ao recinto e ainda uma conta para uso da aplicação.

### 11. Modelo Canvas

#### 11.1. Parcerias-Chave

- Ginásios;
- Personal Trainers;
- Empresas para fornecimento de roupa e alimentos desportivos;
- Institutos académicos (no caso IPCA);

#### 11.2. Atividades-Chave

- Marketing;
- Desenvolvimento e manutenção do software;
- Desenvolvimento e manutenção do hardware;
- Suporte.

#### 11.3. Recursos-Chave

- Desenvolvedores de software;
- Desenvolvedores de hardware;
- Marcas;
- Contratos com os parceiros-chave.

## 11.4. Proposta de valor

### 11.4.1. Ginásio e seus funcionários

- Experiência facilitada na gestão de lotação, dados biométricos e avaliações dos seus clientes.

### 11.4.2. Utilizadores/Clientes do ginásio

- Forma mais eficiente e fácil de verificar a lotação do ginásio;
- Experiência personalizada da sua alimentação e treino;
- Existência de loja online de pesquisa simples.

## 11.5. Relacionamento com clientes

- Sistema de avaliações;
- Suporte técnico.

## 11.6. Canais

- Aplicação mobile;
- Parcerias;
- Marketing;
- Mídia social.

## 11.7. Segmentos de clientes

- Ginásios;
- Funcionários do ginásio;
- Clientes do ginásio.

## 11.8. Custos

- Desenvolvedores de software;
- Desenvolvedores de hardware;
- Equipa de instalação do sistema;
- Hardware;
- Marketing.

## 11.9. Receitas

- Percentagem de cada contrato individual;
- Percentagem de vendas da loja online.

### 12. Grelha de autoavaliação

A autoavaliação do grupo será feita em cada entrega, até ao final deste projeto.

As notas atribuídas para cada membro de grupo serão discutidas entre os mesmo em reuniões semanais, estas serão calculadas tendo em conta um sistema de pontuação que apresenta diversos parâmetros e os seus pesos. O documento oficial que contém as avaliações encontra-se num ficheiro externo ao relatório.

Parâmetros de avaliação de cada membro de grupo:

- Assiduidade (**5** pontos)
- Participação (**2.5** pontos)
- Comunicação (**1** ponto)
- Respeito entre colegas (**1** ponto)
- Cumprimento de prazos (**2.5** pontos)
- Organização (**2** pontos)
- Iniciativa (**3** pontos)
- Qualidade de trabalho (**2** pontos)
- Autonomia (**1** ponto)

### 13. Atas

Esta seção apresentará todas as atas feitas até à data atual, identificando o que foi feito ao longo de cada sprint em uma reunião, indicando o local da reunião e hora, todos os membros de grupo devem estar presentes em cada reunião. Neste documento estarão apresentadas as atas de forma mais compacta.

As atas apresentam-se também em documentos externos, estando estas devidamente separadas.

#### 13.1. Entrega #1

##### 13.1.1. Ata #1

**Local da Reunião:** Online (via Discord)

**Hora de Início:** 22:30      **Hora de Fim:** 23:45

##### **Ordem de trabalhos:**

1. Fez-se um brainstorming sobre a ideia do projeto que envolvesse o tema Smart Campus, que foi decidida nesta reunião.
2. Escolha de um nome para a aplicação a ser desenvolvida, o grupo escolheu o nome IPCA GYM
3. Foi feita a distribuição de tarefas para a 1ª entrega do projeto, nos quais foi tratado o documento do mesmo, Mockups, business Model Canvas e levantamento de requisitos

#### 13.2. Entrega #2

##### 13.2.1. Ata #1

**Local da Reunião:** Online (via Discord)

**Data:** 22 de outubro de 2022

**Hora de Início:** 23:30      **Hora de Fim:** 01:30

##### **Ordem de trabalhos:**

1. Foi efetuada a distribuição de tarefas para a próxima entrega, nos quais estão incluídos a resolução de:
  - a. Diagrama de Componentes;
  - b. Diagrama de casos de uso;
  - c. Diagrama de classes;
  - d. Diagrama de Business Process Model and Notation (BPMN);
  - e. Diagrama de Atividades;
  - f. Diagrama de Estados;
  - g. Diagrama de Entidade-Relação.
2. Discussão de ideias para o futuro diagrama Entidade-Relação, que seria a base do grupo para fazer a Datava-se para a aplicação, foi usado como referência o Diagrama de Classes.
3. Foi pensado também ideias de logótipos para o IPCA GYM.

### 13.2.2. Ata #2

**Local da Reunião:** Online (via Discord)

**Data:** 31 de outubro de 2022

**Hora de Início:** 23:30      **Hora de Fim:** 01:30

#### **Ordem de trabalhos:**

1. Foi efetuada a revisão e análise dos diagramas feitos até à data presente, nos quais estão incluídos:
  - a. Diagrama de Componentes;
  - b. Diagrama de casos de uso;
  - c. Diagrama de classes;
  - d. Diagrama de Business Process Model and Notation (BPMN);
  - e. Diagrama de Atividades.

Ficaram alguns diagramas em falta, que ficou decidido previamente que iriam ser feitos mais tarde.

2. Início do diagrama de Entidade-Relação.
3. Foi distribuída as tarefas para a semana seguinte de forma que todos os membros consigam tirar o melhor das suas capacidades para as mesmas.

### 13.2.3. Ata #3

**Local da Reunião:** Online (via Discord)

**Data:** 08 de novembro de 2022

**Hora de Início:** 23:30      **Hora de Fim:** 01:30

#### **Ordem de trabalhos:**

1. Foi efetuada a revisão e análise dos diagramas feitos até à data presente, com objetivo de melhorar os mesmos, nos quais estão incluídos:
  - a. Diagrama de Componentes;
  - b. Diagrama de casos de uso;
  - c. Diagrama de classes;
  - d. Diagrama de Business Process Model and Notation (BPMN);
  - e. Diagrama de Atividades;

Após esta ação, foi inicializada a discussão de ideias para os seguintes diagramas:

- a. Diagrama de Estados;
  - b. Diagrama de Entidade-Relação.
2. Início de implementação do programa no qual foi feito apenas a página de Login, sem qualquer funcionalidade.
3. Foi feita também uma procura de tutoriais e documentação para servirem de apoio para o desenvolvimento da aplicação.



### 13.2.4. Ata #4

**Local da Reunião:** Online (via Discord)

**Data:** 12 de novembro de 2022

**Hora de Início:** 23:30      **Hora de Fim:** 01:30

#### **Ordem de trabalhos:**

1. Foi efetuada a revisão e análise dos diagramas feitos até à data presente, com objetivo de melhorar os mesmo, nos quais estão incluídos:
  - a. Diagrama de Componentes;
  - b. Diagrama de casos de uso;
  - c. Diagrama de classes;
  - d. Diagrama de Business Process Model and Notation (BPMN);
  - e. Diagrama de Atividades;
  - f. Diagrama de Estados;
  - g. Diagrama de Entidade-Relação.
2. Tendo em conta o adiantamento de tarefas por parte dos membros de grupo, não foi realizado mais nada nesta reunião.

### 13.3. Entrega #3

#### 13.3.1. Ata #1

**Local da Reunião:** Online (via Discord)

**Data:** 19 de novembro de 2022

**Hora de Início:** 16:30      **Hora de Fim:** 17:00

#### **Ordem de trabalhos:**

1. Tendo em conta a semana ocupada com avaliações de outras cadeiras, não houve muita coisa a ser tratada nesta sprint;
2. Análise e possível melhoramento de diagramas previamente feitos;
3. Discussão de grupo em relação ainda às linguagens a serem utilizadas, tendo em conta o decorrer de outras cadeiras.

#### 13.3.2. Ata #2

**Local da Reunião:** Online (via Discord)

**Data:** 26 de novembro de 2022

**Hora de Início:** 15:30      **Hora de Fim:** 16:30

#### **Ordem de trabalhos:**

1. Análise crítica e construção dos diagramas de atividades e estados;
2. Melhoria e ajustes nos Mockups de acordo com alterações sugeridas pelo grupo;
3. Discussão de grupo em relação a como fazer o diagrama de sequência por ecrã.

## 13.3.3. Ata #3

**Local da Reunião:** IPCA (com docente)

**Data:** 3 de dezembro de 2022

**Hora de Início:** 21:00 **Hora de Fim:** 21:30

### **Ordem de trabalhos:**

1. Conclusão dos diagramas de sequência por ecrã;
2. Melhoria e alterações do relatório face a observações feitas pelo docente (diagrama de Gantt, legendas e introduções a tópicos);
3. Realização do dossier de organização do grupo;
4. Discussão em grupo para planeamento da próxima entrega e aspetos a melhorar.

## 13.4. Entrega #4

### 13.4.1. Ata #1

**Local da Reunião:** Online (via Discord)

**Data:** 16 de Dezembro de 2022

**Hora de Início:** 15:30 **Hora de Fim:** 16:00

### **Ordem de trabalhos:**

1. Planeamento da construção do projeto
2. Análise da entrega passada e melhorias do projeto
3. Discussão de grupo sobre tarefas individuais a realizar
4. Planeamento do Poster

### 13.4.2. Ata #2

**Local da Reunião:** Online (via Discord)

**Data:** 23 de Dezembro de 2022

**Hora de Início:** 15:30 **Hora de Fim:** 16:30

### **Ordem de trabalhos:**

1. Parte da aplicação Back-End\* concluída
2. Análise crítica sobre aspetos a melhorar no programa para futuras sprints
3. Reorganização do calendário e ajustes de tarefas individuais

\*Back-End (em contexto de desenvolvimento de software) é a parte de um sistema que gere a lógica e o armazenamento de dados, é normalmente responsável por tarefas como armazenar e recuperar dados, processar pedidos e implementar a lógica de negócio. O Back-End é geralmente executado num servidor e pode ser construído com uma variedade de linguagens de programação e tecnologias

## 13.4.3. Ata #3

**Local da Reunião:** Online (via Discord)

**Data:** 30 de Dezembro de 2022

**Hora de Início:** 15:30 **Hora de Fim:** 16:30

### **Ordem de trabalhos:**

1. Revisão e análise crítica sobre o que foi implementado no Back-End
2. Começo da implementação do Front-End \*
3. Construção do Poster publicitário
4. Ajustes e correção de problemas

\*Front-End em contexto de desenvolvimento de software, o Front-End é a parte de um sistema que apresenta a interface de utilizador e gere a interação do utilizador. O Front-End é normalmente responsável por tarefas como renderizar a interface de utilizador, processar a entrada do utilizador e apresentar dados ao utilizador.

## 13.4.4. Ata #4

**Local da Reunião:** Online (via Discord)

**Data:** 5 de Janeiro de 2022

**Hora de Início:** 21:00 **Hora de Fim:** 22:30

### **Ordem de trabalhos:**

1. Ponto de situação do que foi implementado até agora (em função do tempo restante).
2. Implementação do Front-End quase concluído
3. Revisão e finalização do poster publicitário
4. Começo da ligação entre o Front-End e o Back-End
5. Atualização do relatório

## 14. Diagramas

### 14.1. Diagrama Entidade-Relação

Segue-se abaixo o diagrama de entidade-relação da base de dados do IPCA GYM:

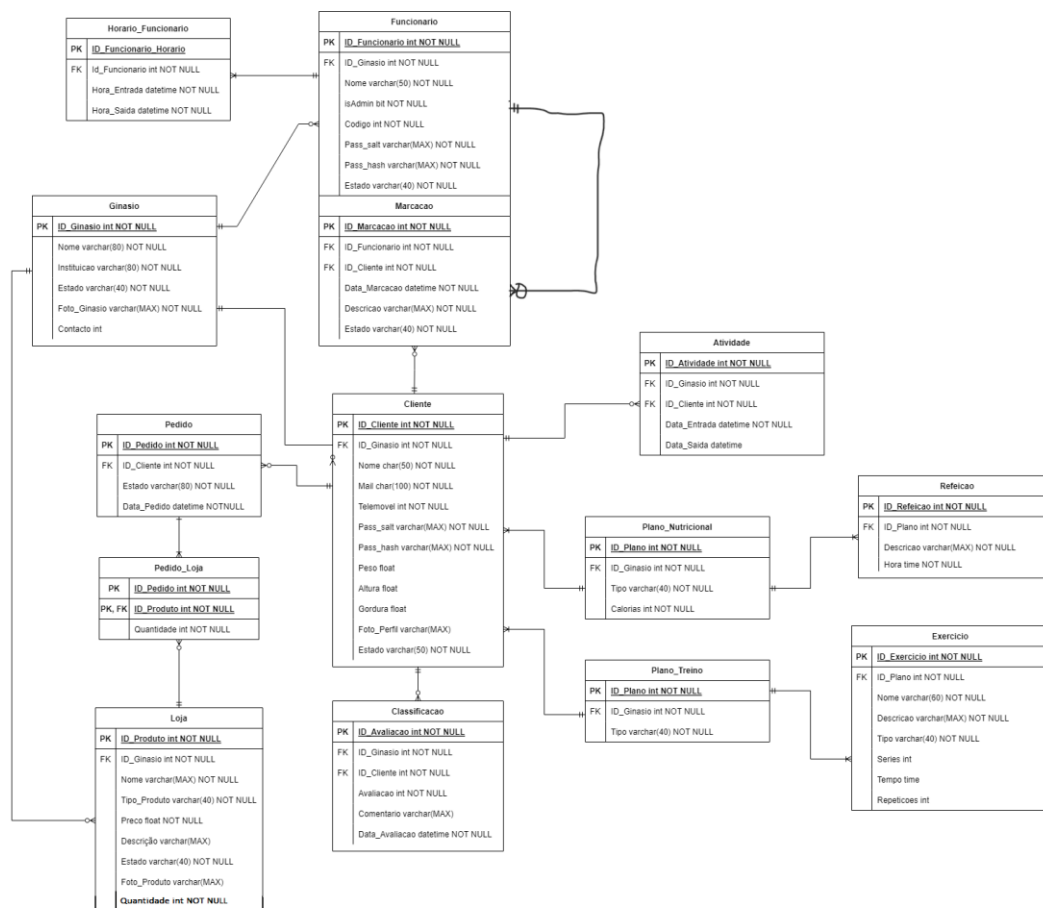


Figura 1 - Diagrama de Entidade-Relação

Como entidades principais este diagrama possui:

- **Cliente** – dados de um cliente que está a utilizar a aplicação;
- **Funcionário** – dados de um funcionário do ginásio em causa, possui o atributo “isAdmin” para determinar se este tem como role Gerente ou não;
- **Ginásio** – dados do ginásio em causa, entidade criada de forma que o projeto, mais tarde, tenha suporte para várias instituições académicas
- **Loja** – possui dados de todos os produtos disponíveis e indisponíveis na loja de cada ginásio
- **Atividade** – entidade criada com o propósito de analisar as entradas e saídas de cada cliente no ginásio (recebe informação do Arduino)

De forma que fosse possível suportar alguns dados sobre outras funcionalidades, foram adicionadas as seguintes entidades:

- **Plano\_Nutricional** e Refeição – entidades que possuem dados sobre diferentes refeições e seus horários, cada ginásio define o seu plano nutricional;
- **Plano\_Treino** e Exercício – entidades que possuem dados sobre diferentes exercícios e suas descrições, cada ginásio define o seu plano de treino;
- Pedido e **Pedido\_Loja** – entidade que possui dados de cada encomenda feita pelo utilizador na loja do ginásio no qual este está inscrito;
- **Horario\_Funcionario** – regista o horário de cada funcionário, de forma a verificar a sua disponibilidade para as diferentes marcações;
- **Marcação** – possui a informação de todas as marcações marcadas pelo cliente com o funcionário, associadas a cada ginásio;
- **Classificação** – contém todas as avaliações feitas pelos clientes a cada ginásio.

### 14.2. Diagrama de Casos de Uso

Segue-se abaixo os diferentes casos de usos criados, os diagramas demonstram de uma forma generalizada o uso da aplicação.

#### 14.2.1. Caso de uso 1 – Gestão de Conta

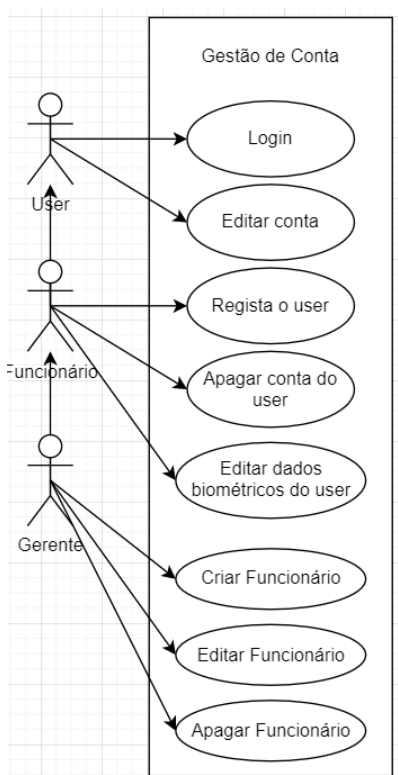


Figura 2 - Diagrama Casos de Uso 1 (Gestão de Conta)

- **CdU 1.1 Login** - O utilizador introduz os seus dados para entrar na nossa aplicação;
- **CdU 1.2 Editar Conta** - O utilizador é capaz de editar o seu perfil;
- **CdU 1.3 Regista o user** - O funcionário do ginásio é capaz de registar um utilizador na aplicação;
- **CdU 1.4 Apagar conta do user** - O funcionário é capaz de apagar a conta de um utilizador e respetivamente todos os dados associados a ele;
- **CdU 1.5 Editar dados biométricos do user** - O funcionário, durante uma consulta, é capaz de alterar os dados do utilizador (altura, peso, massa muscular gorda/magra);
- **CdU 1.6 Criar funcionário** - O gerente é capaz de criar um utilizador com a propriedade de ser funcionário também;
- **CdU 1.7 Editar Funcionário** - O gerente é capaz de editar todos os dados relativos a um funcionário;
- **CdU 1.8 Apagar Funcionário** - O gerente é capaz de apagar a conta de um funcionário, eliminando também todos os dados a ele.

### 14.2.2. Caso de uso 2 – Loja online

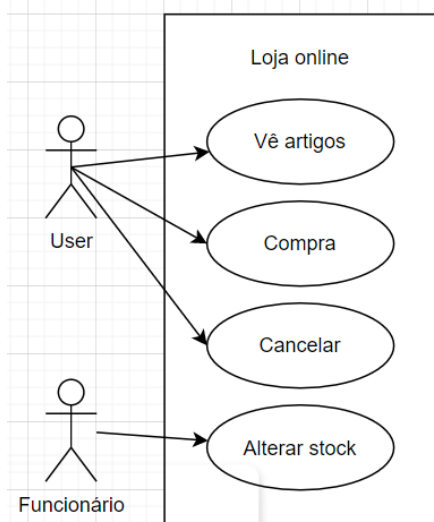


Figura 3 - Diagrama de Casos de Uso 2 (Loja Online)

- **CdU 2.1 Vê artigos** - O utilizador é capaz de ver os artigos na nossa loja online;
- **CdU 2.2 Compra** - O utilizador é capaz de comprar artigos da nossa loja online;
- **CdU 2.3 Cancelar** - O utilizador é capaz de cancelar uma encomenda (desde que essa já esteja registada na nossa loja online);
- **CdU 2.4 Alterar Stock** - O Funcionário é capaz de alterar o stock existente na nossa loja.

## 14.2.3. Caso de uso 3 - Aplicação

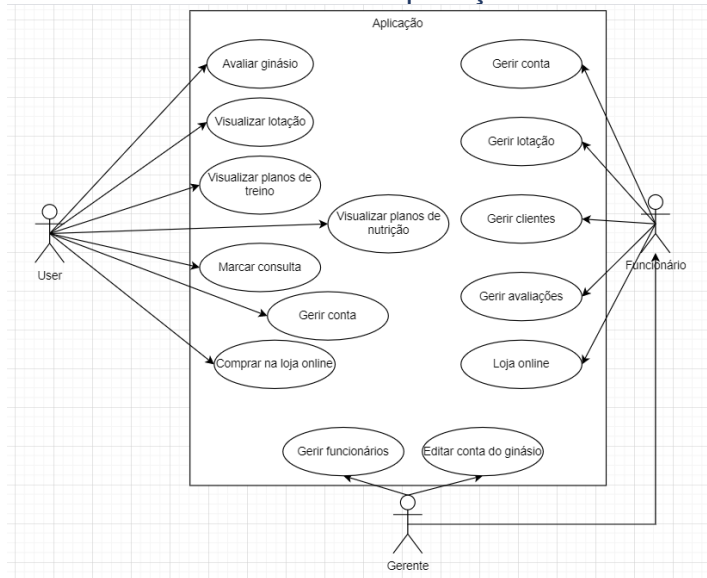


Figura 4 - Diagrama de Casos de Uso 3 (Aplicação)

- **CdU 3.1 Avaliar ginásio** - O utilizador da aplicação poderá usufruir de um sistema de avaliações relativamente ao ginásio/aplicação;
- **CdU 3.2 Visualização lotação** - O utilizador poderá através da página inicial visualizar a lotação atual do ginásio;
- **CdU 3.3 Visualizar planos de treino** - O utilizador poderá visualizar todos os planos de treino que o ginásio tem para oferecer;
- **CdU 3.4 Visualizar planos de nutrição** - O utilizador poderá visualizar todos os planos de nutrição que o ginásio tem para oferecer;
- **CdU 3.5 Marcar consulta** - O utilizador poderá marcar uma consulta nutricional ou física;
- **CdU 3.6 Gerir Conta** - O utilizador poderá alterar os dados da sua conta;
- **CdU 3.7 Comprar na loja online** - O utilizador poderá realizar compras na loja online;
- **CdU 3.8 Gerir funcionários** - O gerente poderá gerir os funcionários (remover, criar, editar);
- **CdU 3.9 Editar conta do ginásio** - O gerente poderá alterar os dados do ginásio;
- **CdU 3.10 Gerir conta** - O funcionário poderá editar os seus dados de conta;
- **CdU 3.11 Gerir lotação** - O funcionário poderá monitorizar a lotação de uma forma mais detalhada;
- **CdU 3.12 Gerir clientes** - O funcionário poderá gerir dados dos seus clientes;
- **CdU 3.13 Gerir avaliações** - O funcionário poderá monitorizar as avaliações feitas á sua ginásio/aplicação;
- **CdU 3.14 Loja online** - O funcionário poderá adicionar/remover/editar produtos á loja online.



### 14.2.4. Caso de uso 4 – Gestão de lotação

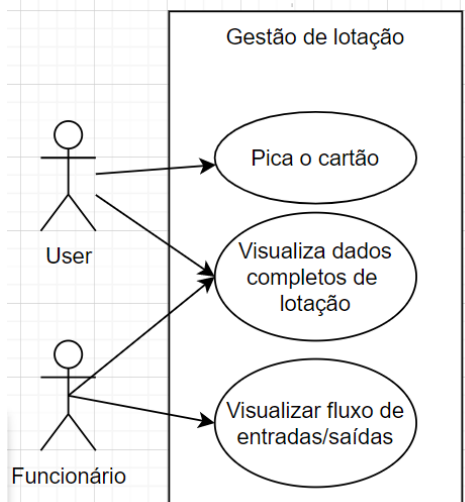


Figura 5 - Diagrama de Casos de Uso 4 (Gestão de lotação)

- **CdU 4.1 Picar o cartão** - O utilizador terá de picar o cartão no leitor do ginásio;
- **CdU 4.2 Visualiza dados completos de lotação** - O utilizador e o funcionário poderão visualizar a lotação detalhada do ginásio;
- **CdU 4.3 Visualizar fluxo de entradas/saídas** - O funcionário poderá monitorizar todas as entradas e saídas do ginásio desde o início da sua existência.

## 14.3. Diagrama de atividades

Foram desenvolvidos 3 diagramas de atividades.

### 14.3.1. Cancelamento de serviço

Para o cancelamento de um serviço por parte de um funcionário, é envolvida a participação do cliente, gerente e funcionário. O funcionário então cancela e notifica o gerente do incidente. O gerente recebe essa notificação e toma uma decisão. Rejeitando esse pedido, o funcionário é notificado e termina a atividade.

No caso de aceitar, o cliente é notificado e aí esse decide remarcar ou cancelar. Caso este não aceite termina a tarefa. Caso aceite, é remarcada e termina com o funcionário a ser notificado.

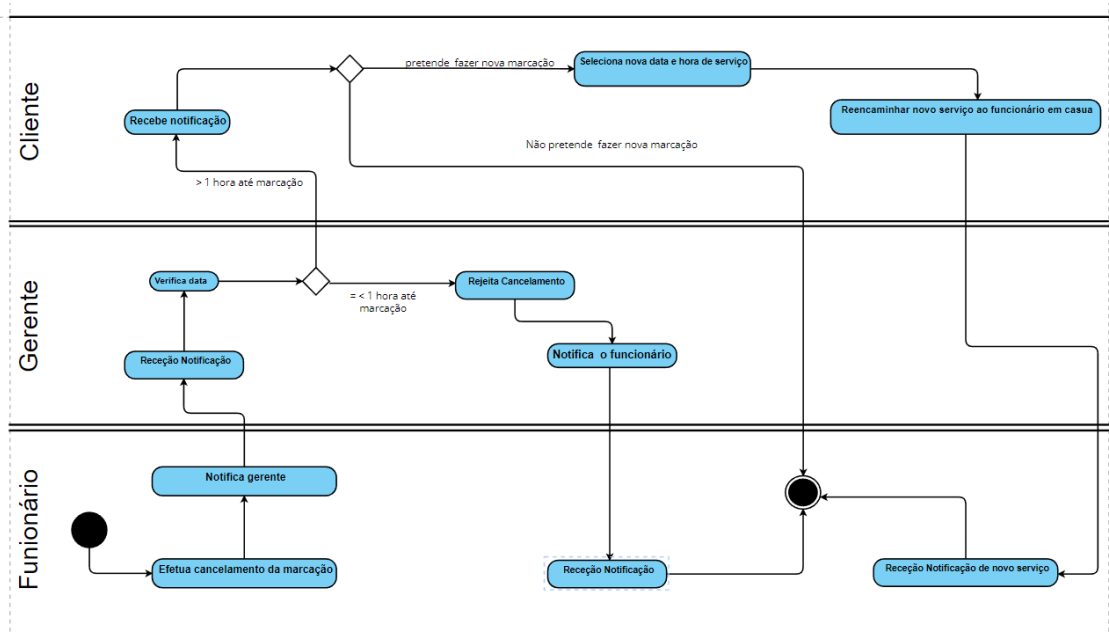


Figura 6 - Diagrama de Atividades - Cancelar Serviço

### 14.3.2. Reagendamento/Cancelamento de serviço

Para o reagendamento/cancelamento de um serviço é envolvida a participação do cliente e do funcionário. O cliente altera os dados da sua marcação, onde este decide se pretende remarcar ou cancelar. No caso desta ser cancelada, o funcionário é notificado e termina a atividade. Por outro lado, o reagendamento leva a definição de uma nova data e dependendo da disponibilidade esta pode ou não ser aceite e desta forma termina com o funcionário sendo notificado.

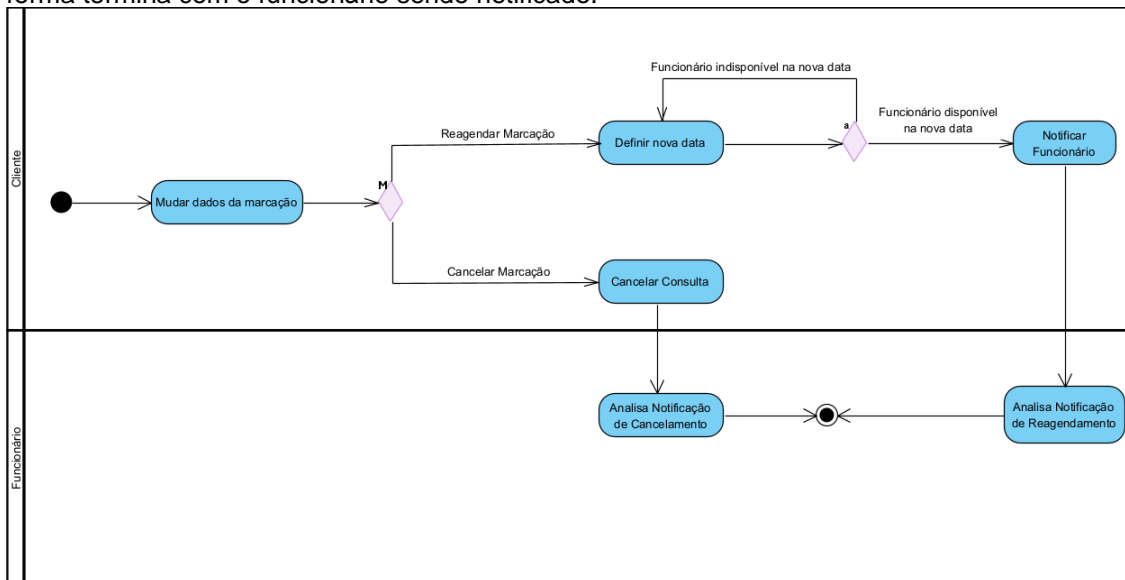


Figura 7 - Diagrama de Atividades - Reagendamento de Serviço

## 14.4. Diagrama de estados

Foi desenvolvido apenas um diagrama de estados para analisar o estado da marcação de uma consulta. Dentre os estados possível este pode ser dado como:

- **Pendente:** quando este se encontra a espera de uma resposta;
- **Cancelada:** quando não é possível marcar para uma determinada data, ou cancelado pelo próprio cliente que a marcou;
- **Agendada:** quando esta é marcada definitivamente.

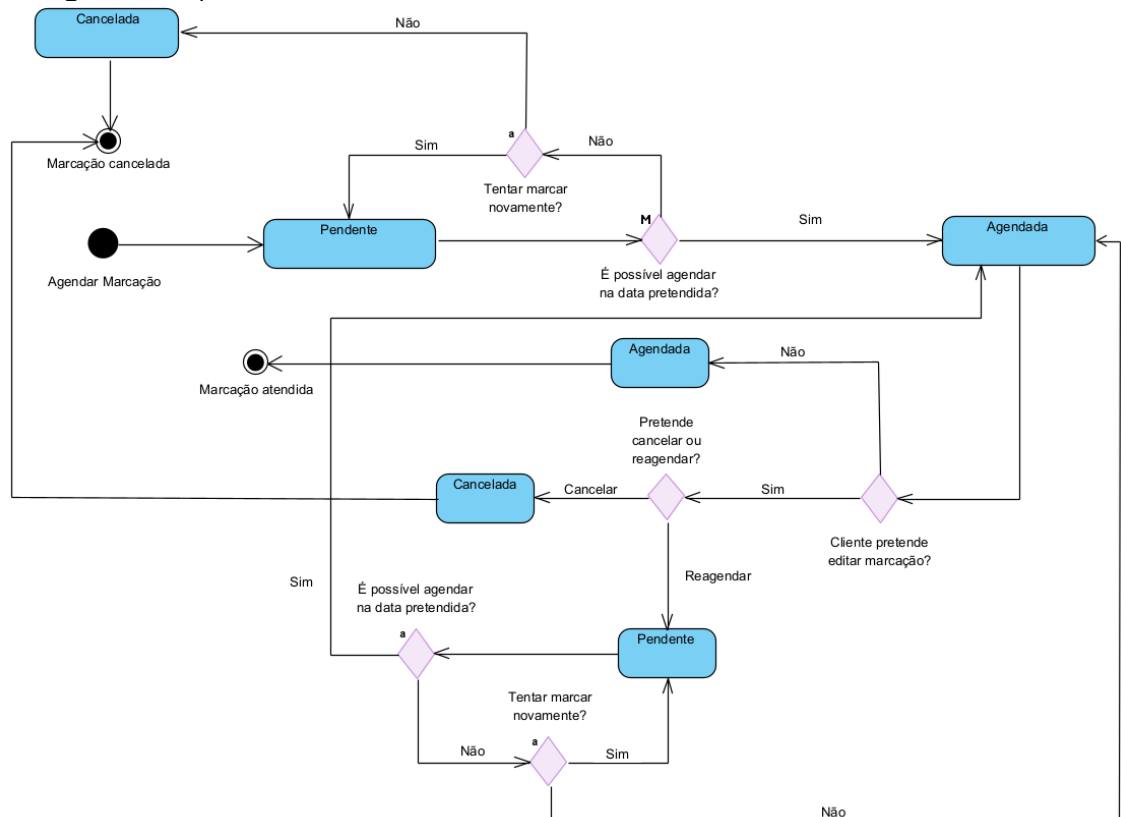


Figura 8 - Diagrama de Estados - Marcação de Consulta

## 14.5. Diagrama de sequência por ecrã

O diagrama de sequência por ecrã referente aos Mockups realizados recorrendo ao Figma encontram-se no seguinte link:

<https://www.figma.com/file/Q4tM34gl91b9fhrGvUeXR/MileriuPT's-team-library?node-id=0%3A1&t=a7h8lpVPLF9ISzXo-1>

## 15. Mockups

### 15.1. Cliente

#### 15.1.1. Login

Login do utilizador no qual é inserido o código de utilizador (inteiro) e password. Caso este não saiba a password terá de recorrer ao link para recuperar a conta.

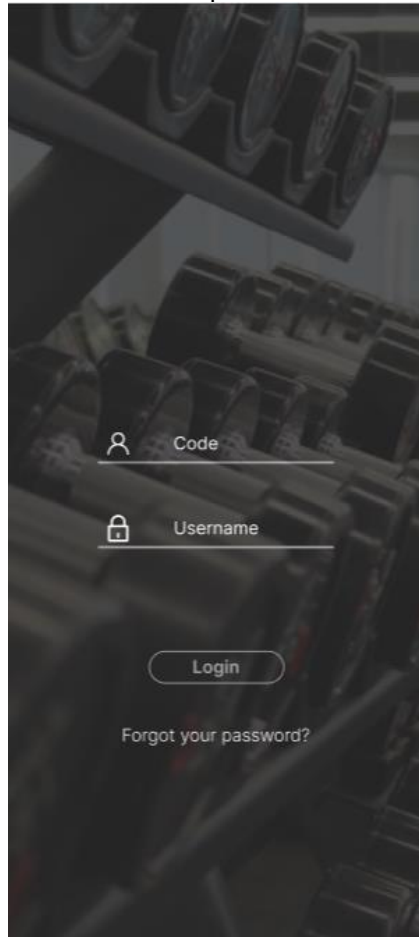


Figura 9 - Login do Utilizador

### 15.1.2. Menu Principal

Menu de entrada do utilizador, onde este poderá saber mais informações sobre o ginásio, informações de nutrição, equipa pertencente ao ginásio além de dados sobre a capacidade do estabelecimento. Este ainda poderá aceder a um menu através do clique da foto e tem outras funcionalidades na barra inferior horizontal.

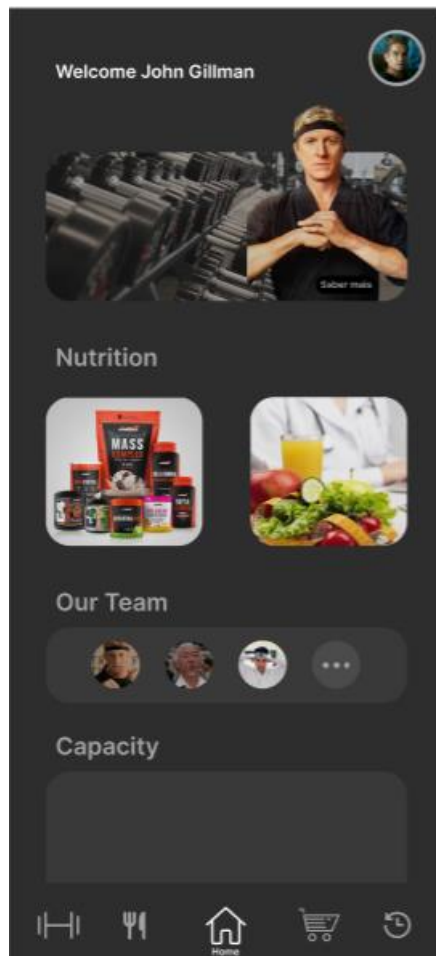


Figura 10 - Utilizador - Página inicial

### 15.1.3. Clique na foto

Clicando na foto aparece um menu com algumas opções. Pode aceder a conta, definições da app e terminar a sessão.

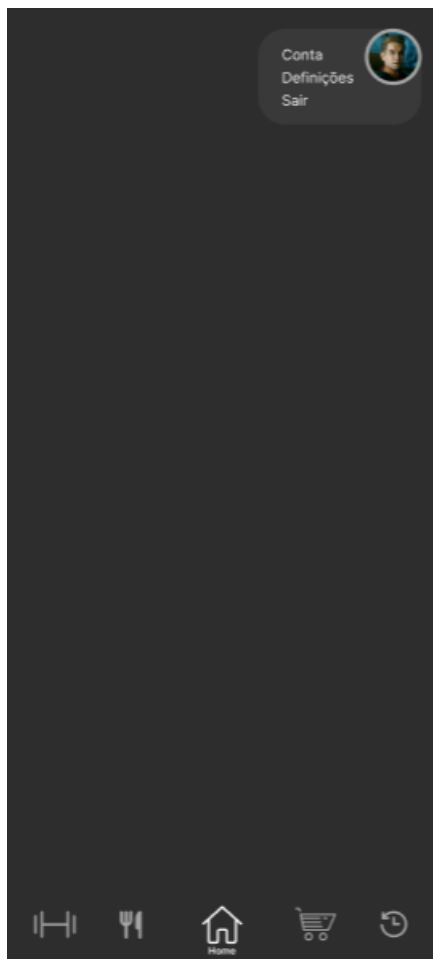


Figura 11 - Utilizador - Menu

## 15.1.4. Conta do utilizador

Acedendo a conta do utilizador, este pode editar a conta, marcar uma consulta, ver informações acerca das suas medidas, estatísticas, detalhes dos seus treinos e contrato.



Figura 12 - Utilizador - Conta

## 15.1.5. Definições

Nas definições este poderá ver algumas das informações acerca da app.

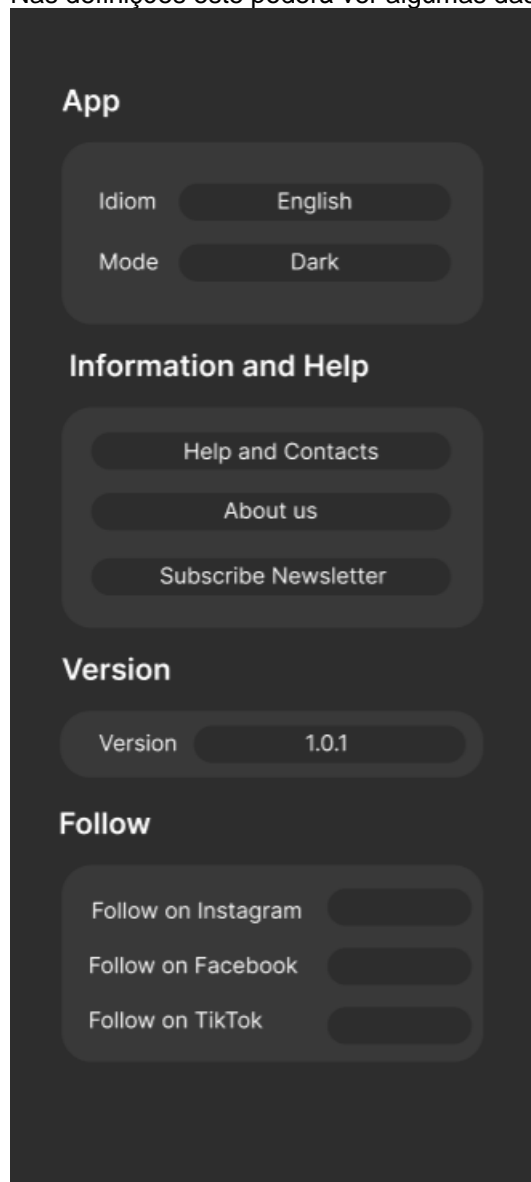


Figura 13 - Utilizador - Definições



### 15.1.6. Editar perfil

O utilizador poderá alterar os seus dados principais da conta (nome, email, contacto e data de nascimento).

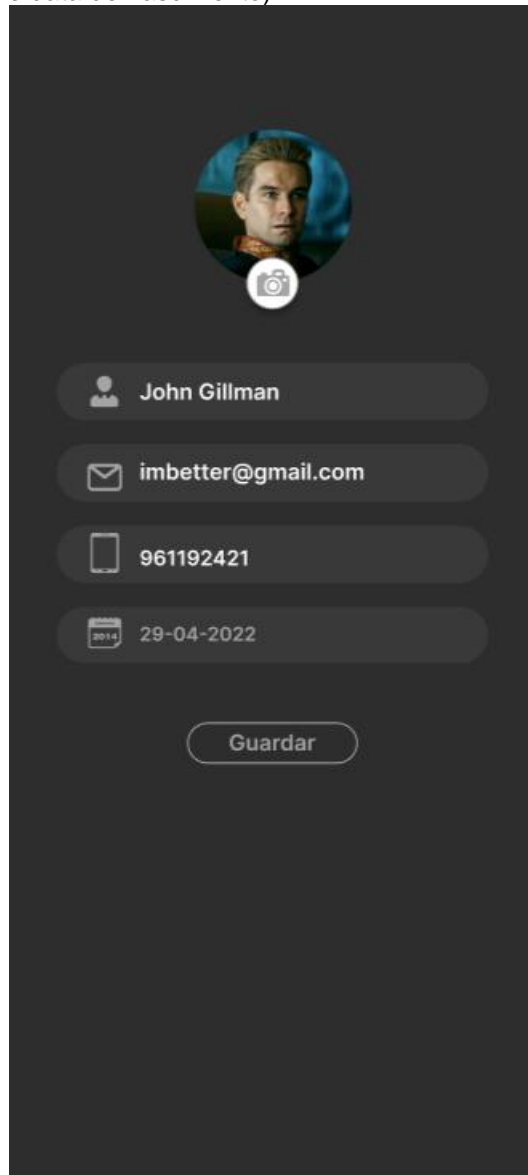
The screenshot shows a dark-themed mobile application interface for editing a user profile. At the top, there is a circular profile picture of a man with a camera icon overlay. Below this, there are four horizontal input fields, each with an icon on the left: a person icon for the name 'John Gillman', an envelope icon for the email 'imbetter@gmail.com', a mobile phone icon for the contact number '961192421', and a calendar icon for the birth date '29-04-2022'. At the bottom of the form is a rounded button labeled 'Guardar' (Save).

Figura 14 - Utilizador - Editar perfil

## 15.1.7. Atividade

Aqui ele poderá ver toda a sua atividade ao nível de treinos executados, além de os poder ver detalhadamente.

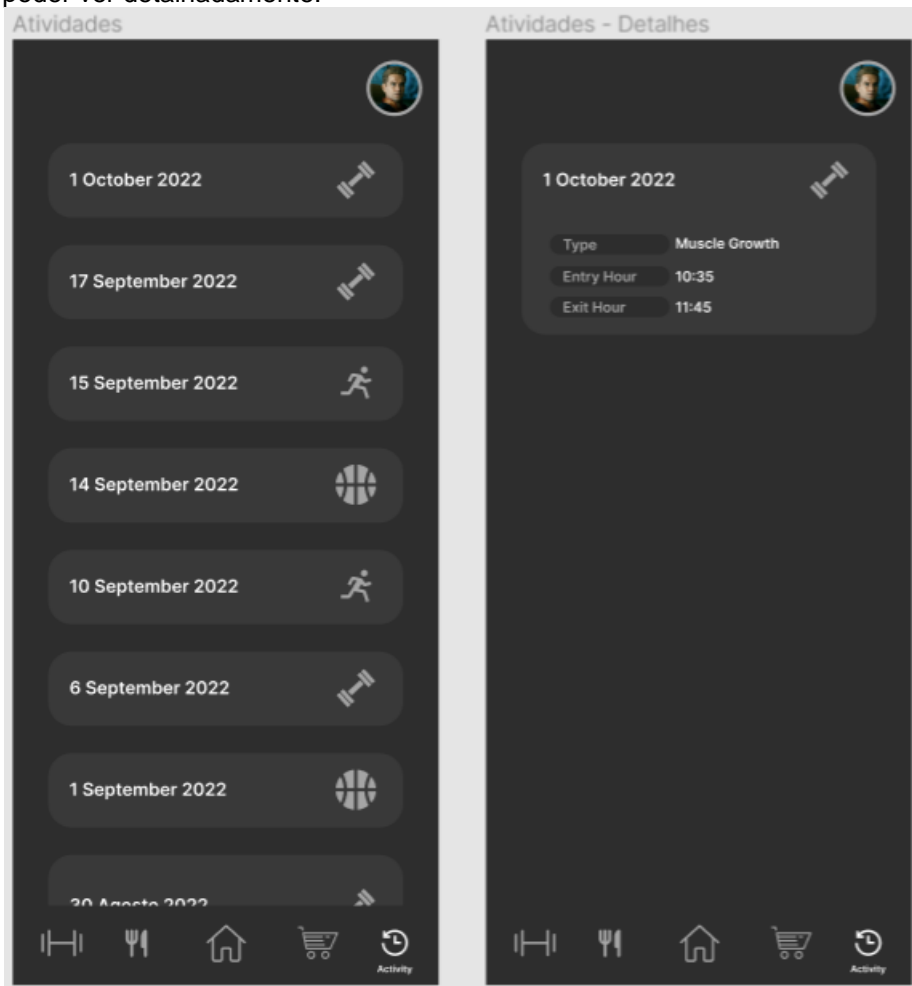


Figura 15 - Utilizador - Atividade

## 15.1.8. Treinos

A app oferece ao utilizador os planos de treino que se seguem. Um desses tipos de plano é treinar separadamente cada membro que nos leva ao 2º Mockups. Para o caso de um desporto segue para o 3º Mockups.

O 4º Mockups representa um treino selecionado com os seus exercícios.

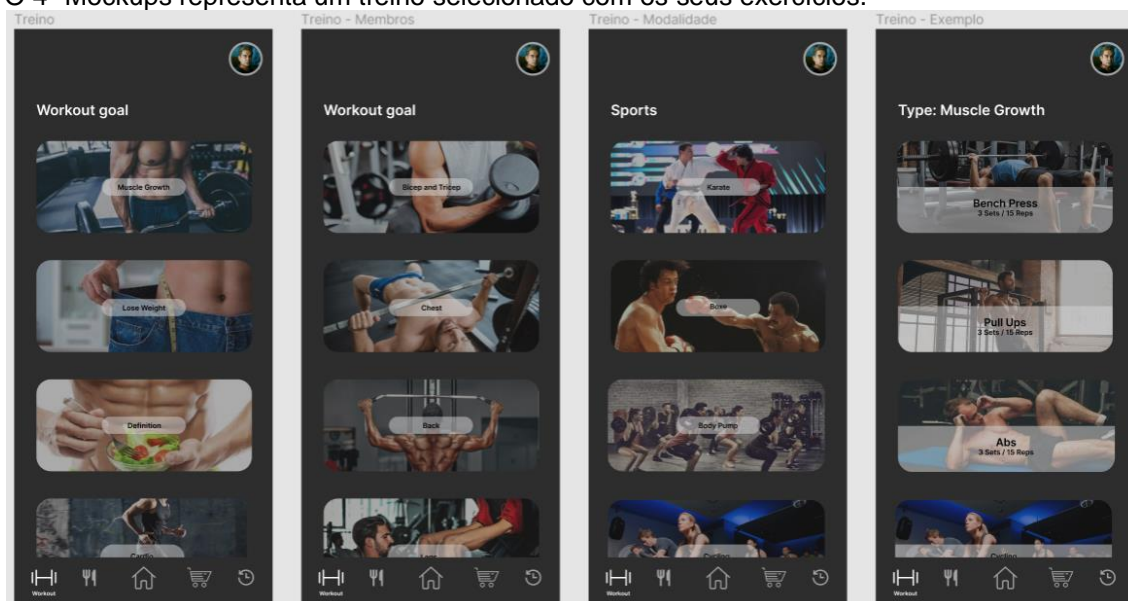


Figura 16 - Utilizador - Treino

## 15.1.9. Nutrição

O utilizador poderá também aceder aos seus planos de nutrição onde poderá consultar o que deve comer a uma determinada hora e alterar os seus objetivos de nutrição.

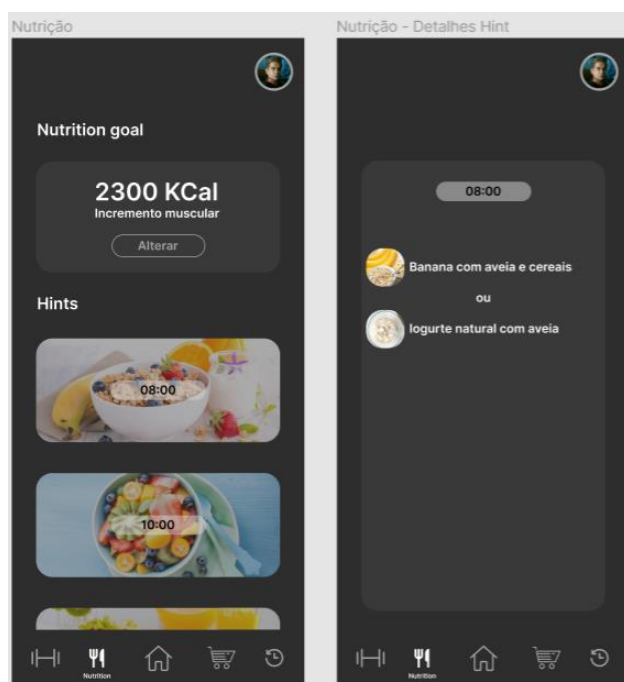


Figura 17 - Utilizador - Nutrição

## 15.1.10. Loja online

Loja online para compra de equipamento de treino, alimentos e outros recursos.

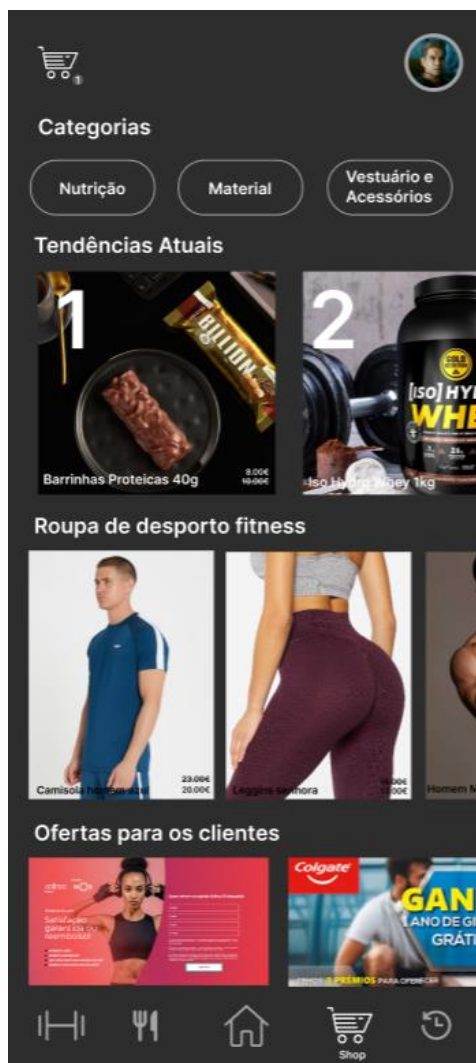


Figura 18 - Utilizador - Loja Online

## 15.2. Sistema do Ginásio

### 15.2.1. Capacidade

O funcionário do ginásio poderá gerir todos os dados de capacidade, estatísticas e outros recursos de monitorização através da secção de capacidade da sua app.

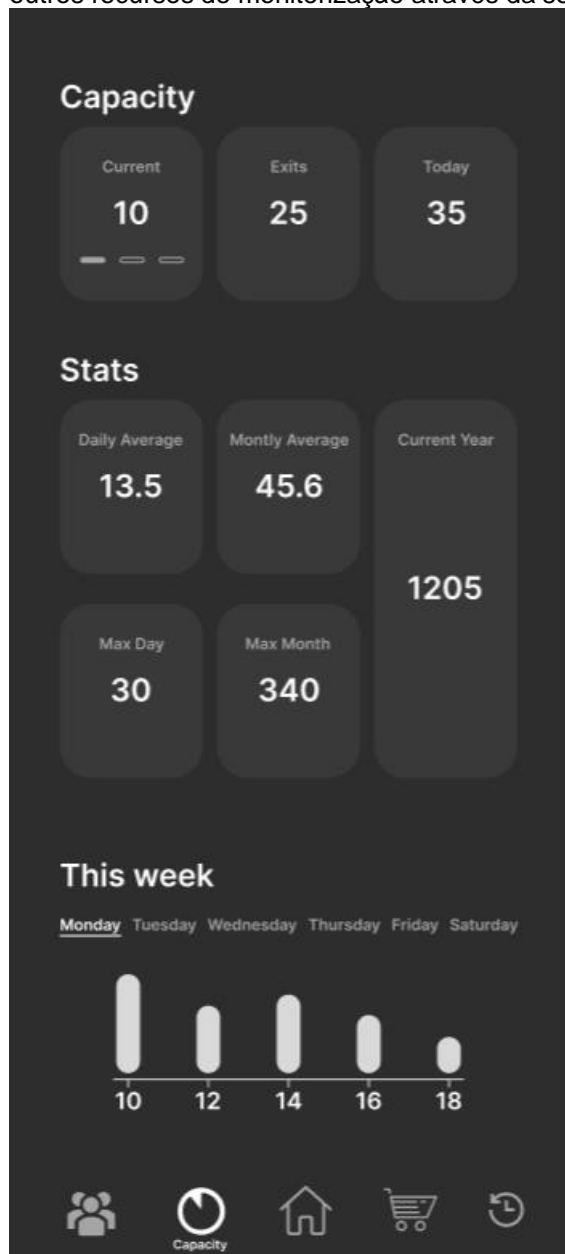


Figura 19 - Sistema - Capacidade

## 15.2.2. Gestão dos clientes

Este poderá consultar os clientes, além de visualizar e editar os seus dados métricos. Ainda poderá inserir novos clientes ao sistema.

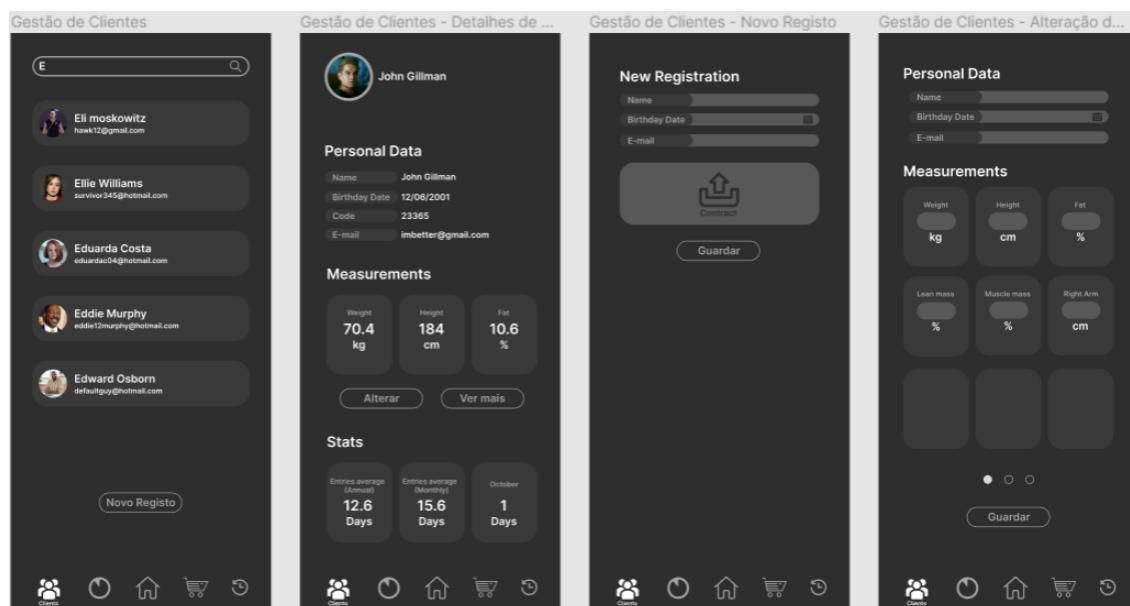


Figura 20 - Sistema - Gestão de clientes

### 15.2.3. Página inicial

Através da página inicial este poderá ver a avaliação atual do seu ginásio, além de comentários.

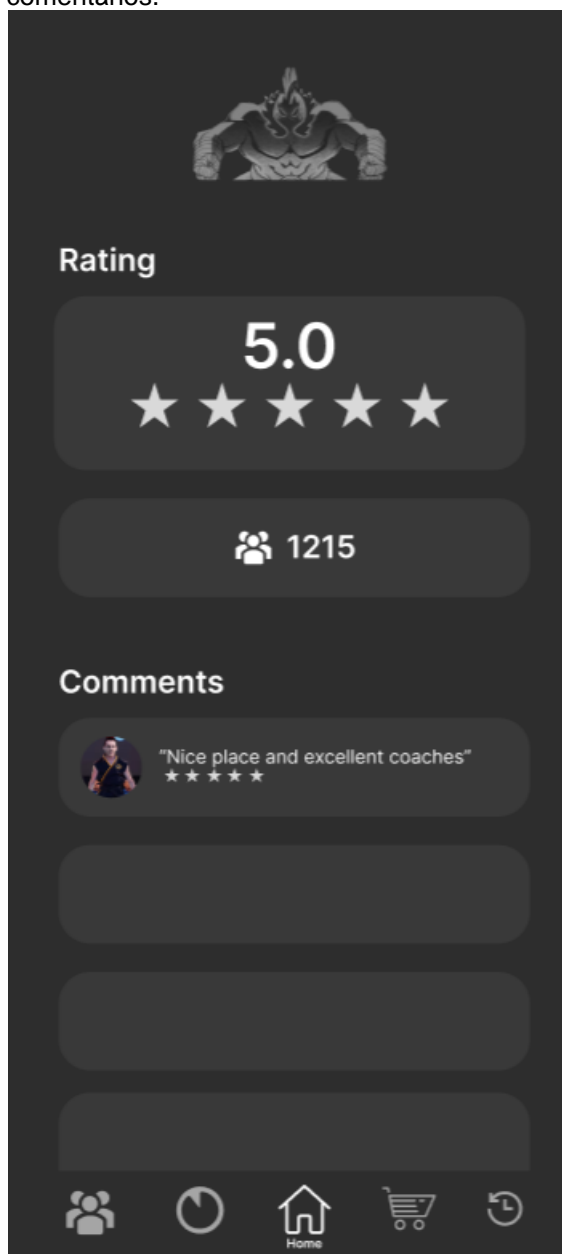


Figura 21 - Sistema - Página inicial

## 15.2.4. Loja online

Página para gerir os pedidos de material da loja online por parte dos utilizadores.

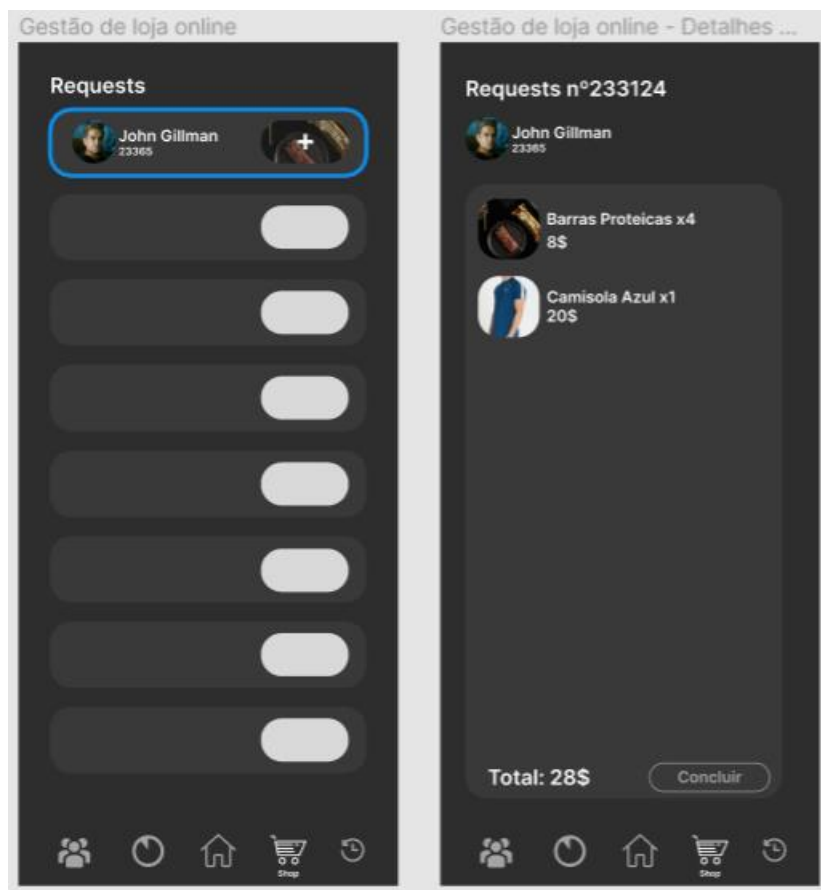


Figura 22 - Sistema - Loja online



## 15.2.5. Atividade

Aqui poderá gerir os fluxos de entrada e saída do ginásio ao longo de todo o tempo.

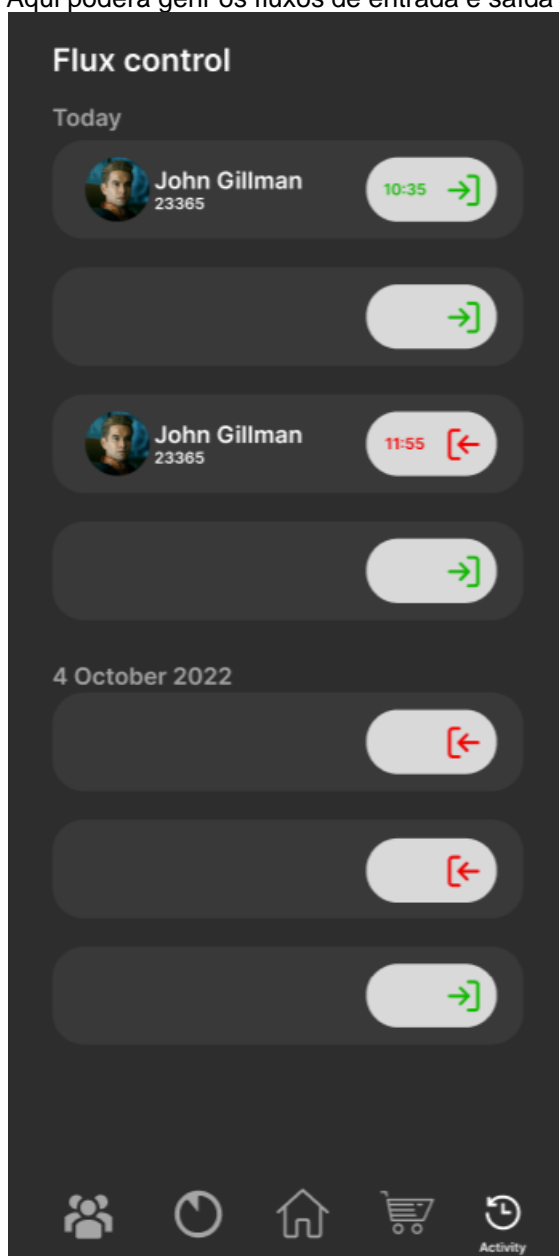


Figura 23 - Sistema - Atividade

## 16. Cronograma

O cronograma do projeto encontra-se num documento externo chamado “Cronograma.xlsx” juntamente com o planeamento das sprints até ao final do projeto. (estas podem sofrer alterações ao longo do desenvolvimento do mesmo, sendo essas alterações devidamente documentadas).

Este mesmo cronograma também foi apresentado em um diagrama de Gantt, identificando assim as tarefas e subtarefas que serão feitas ao longo do projeto, identificando também assim a data e duração das mesmas. Este diagrama será atualizado a cada entrega.

Segue-se o Diagrama de Gantt:

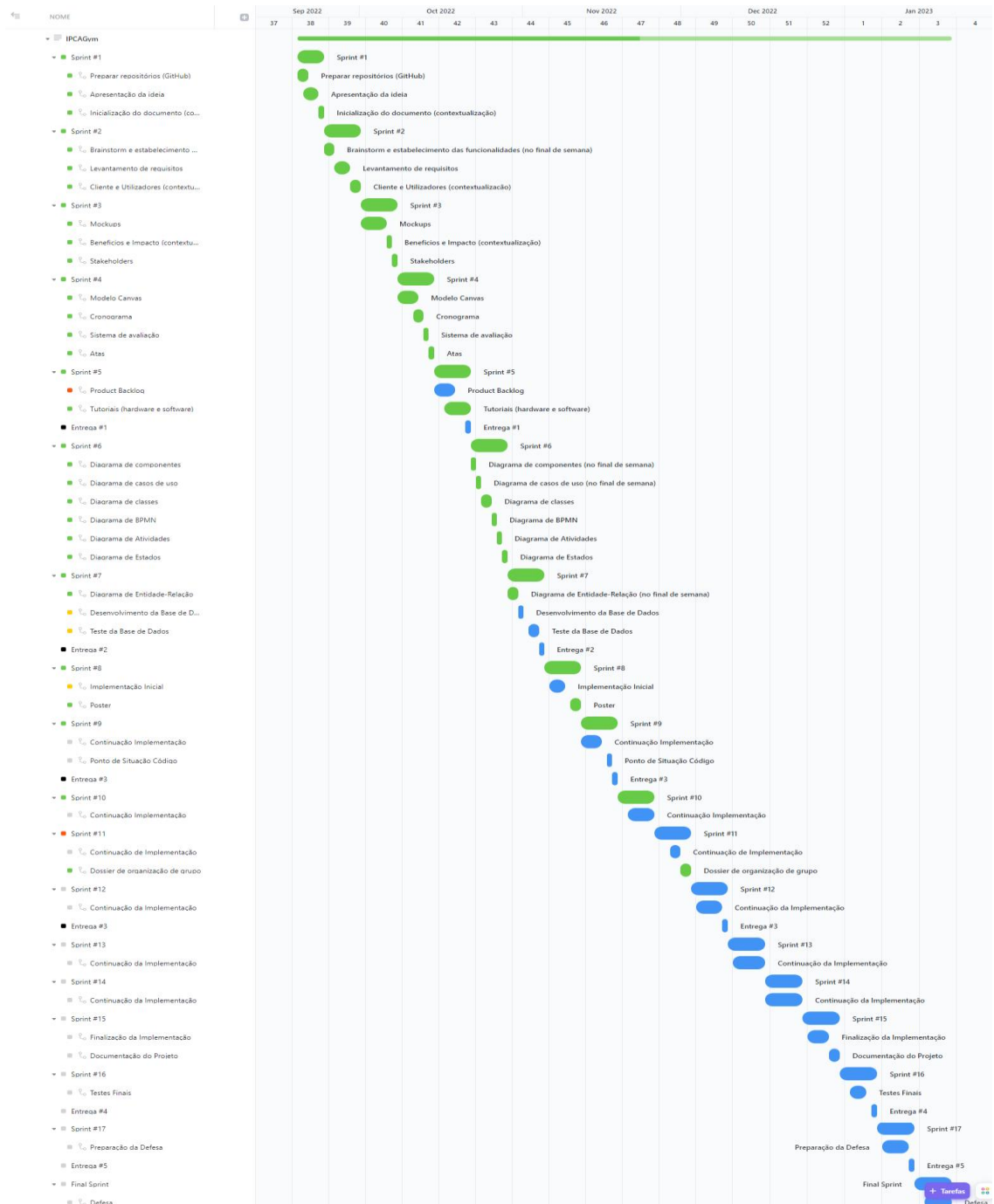


Figura 24 - Diagrama de Gantt

17. Código implementado para o projeto

Este projeto foi implementado com o suporte de um front-end, back-end e uma base de dados para guardar a informação.

De forma, a haver uma interação fora da aplicação com o cliente, será implementado também um Arduino com um sensor de cartões elétricos.

## 17.1. Base de Dados

Como base de dados foi usado o SQL Server, linguagem esta da qual o grupo já tem experiência tanto a nível de comandos (queries) como a nível de IDE, neste caso foi utilizado o SSMS 18 (Microsoft SQL Server Management Studio).

O projeto foi iniciado com uma base de dados local na máquina de cada membro de grupo, com a previsão de no final ser escolhida a base de dados mais povoada e com dados mais coerentes para fazer deployment.

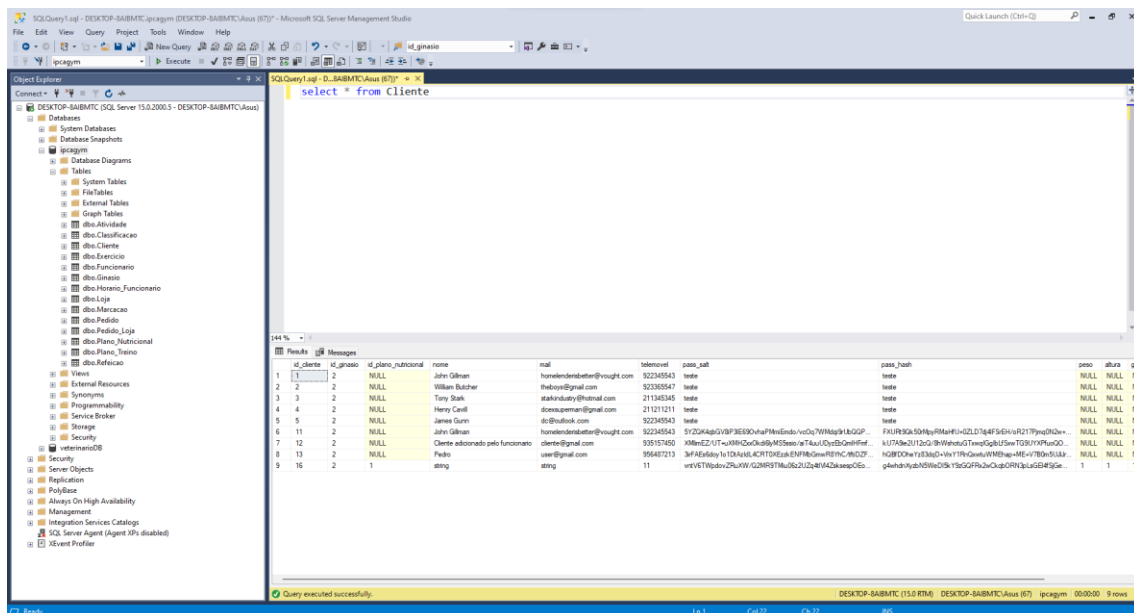


Figura 25 - Microsoft SQL Server Management Studio (SSMS)

## 17.2. Back-end

A nível de back-end, de forma ao grupo seguir a metodologia da cadeira de ISI e também de forma a trabalhar com algo que tivesse mais experiência, foi escolhido trabalhar com .NET Framework (C#), utilizado o IDE Visual Studio 2022.

Segue-se a abordagem dos pontos principais do back-end, onde será especificado o que feito em cada um:

- Programação por camadas
- JWT Tokens e Authentication
- Funcionalidade de Login (Encriptação de passwords)
- Documentação de código com OpenAPI

## 17.2.1. Programação por camadas

De forma a organizar melhor a solução do back-end e também contemplar os objetivos propostos para a entrega final de ISI, esta ficou dividida por diferentes camadas.

A atribuição de dependências foi a seguinte:

- Camada do Backend\_IPCA\_Gym (camada da API) depende da camada de Business Logic (BLL)
- Camada do Business Logic depende da camada de Data Access (DAL)
- Camada de Data Access depende da camada de Business Object (BOL)

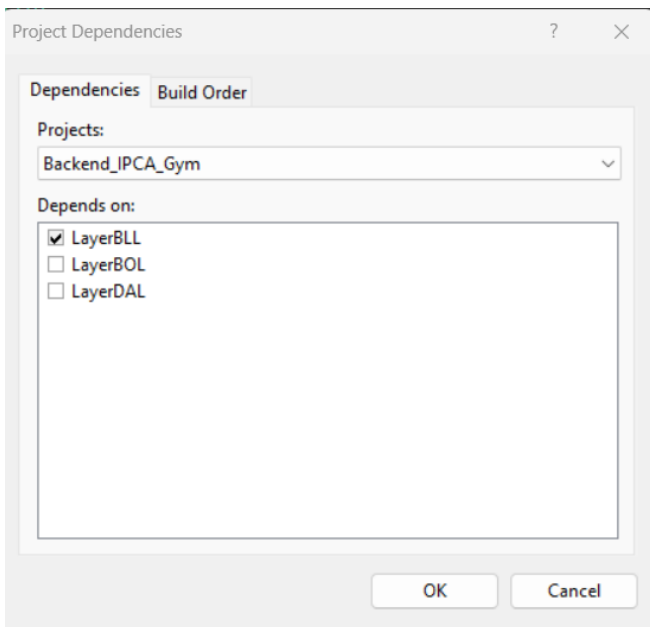


Figura 26 - Dependências API Layer

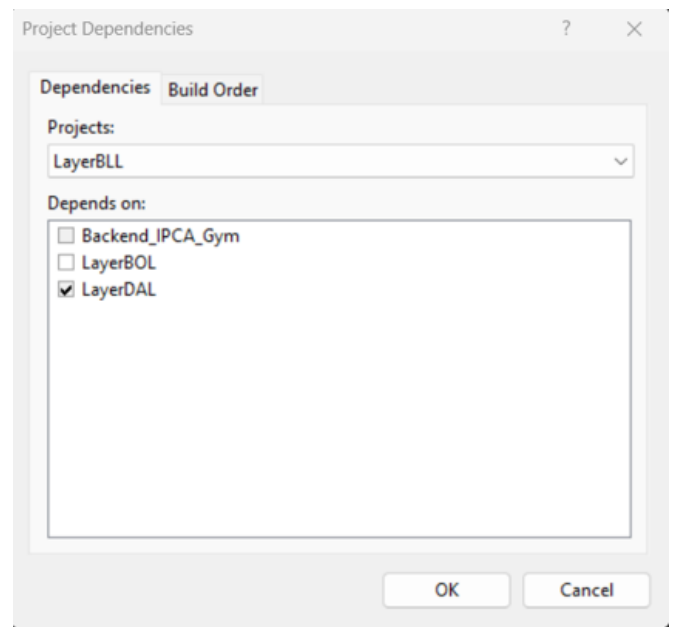


Figura 27 - Dependências BLL Layer

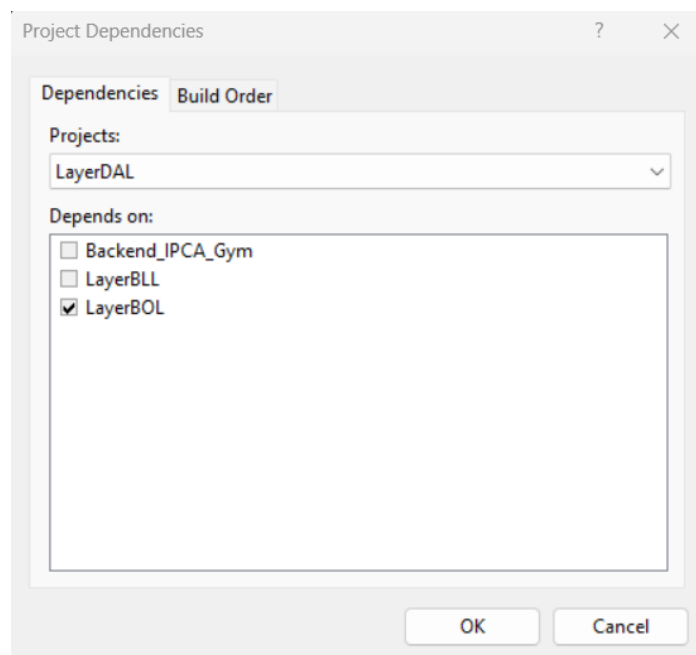


Figura 28 - Dependências DAL Layer

## 17.2.2. JWT Tokens e Authentication

De forma a manter o programa mais seguro e fazer a verificação de quem tem acesso a cada request, foram implementadas tokens de sessão com JWT Tokens, fazendo assim a autenticação do utilizador.

Com isto é possível obter informações sobre o utilizador que está a utilizar a aplicação e se este tem as permissões necessárias para aceder a dados específicos ou não.

A token é atribuída ao utilizador dependendo dos dados de Login, segue-se um exemplo para um login de um funcionário:

```
public static async Task<string> LoginService(string sqlDataSource, LoginFuncionario conta, IConfiguration _configuration)
{
    string query = @"
        select * from dbo.Funcionario
        where codigo = @codigo and estado != 'Inativo';
    ";

    try
    {
        using (SqlConnection databaseConnection = new SqlConnection(sqlDataSource))
        {
            databaseConnection.Open();
            using (SqlCommand myCommand = new SqlCommand(query, databaseConnection))
            {
                myCommand.Parameters.AddWithValue("@codigo", conta.codigo);
                using (SqlDataReader reader = myCommand.ExecuteReader())
                {
                    reader.Read();

                    Funcionario targetFuncionario = new Funcionario();
                    targetFuncionario.id_funcionario = reader.GetInt32(0);
                    targetFuncionario.id_ginasio = reader.GetInt32(1);
                    targetFuncionario.nome = reader.GetString(2);
                    targetFuncionario.is_admin = reader.GetBoolean(3);
                    targetFuncionario.codigo = reader.GetInt32(4);
                    targetFuncionario.pass_salt = reader.GetString(5);
                    targetFuncionario.pass_hash = reader.GetString(6);
                    targetFuncionario.estado = reader.GetString(7);

                    reader.Close();
                    databaseConnection.Close();

                    if (!PasswordEncryption.VerifyPasswordHash(conta.password, Convert.FromBase64String(targetFuncionario.pass_hash), Convert.FromBase64String(targetFuncionario.pass_salt)))
                    {
                        throw new ArgumentException("Password Errada.", "conta");
                    }

                    string token = Token.CreateTokenFuncionario(targetFuncionario, _configuration);

                    return token;
                }
            }
        }
    }
}
```

Figura 29 - Login de um funcionário (back-end)

Como é possível visualizar, o funcionário insere os seus dados e estes são mandados por parâmetro, é obtida a informação da query de busca do funcionário e é verificada a palavra-passe, no caso de sucesso é atribuída a token de sessão a partir do seguinte método auxiliar:

```
/// <summary>
/// Método auxiliar que faz a criação de um Token de Sessão
/// </summary>
/// <param name="funcionario">Funcionario que esta a efetuar login</param>
/// <param name="_configuration">Dependency Injection</param>
/// <returns>Token jwt de sessão</returns>
1 reference
public static string CreateTokenFuncionario(Funcionario funcionario, IConfiguration _configuration)
{
    string role = string.Empty;

    if (funcionario.is_admin) role = "Gerente";
    else role = "Funcionario";

    if (funcionario.nome == "adminaccount") role = "Admin";

    List<Claim> claims = new List<Claim>
    {
        new Claim(ClaimTypes.Email, funcionario.codigo.ToString()),
        new Claim(ClaimTypes.Role, role),
        new Claim(ClaimTypes.SerialNumber, funcionario.id_ginasio.ToString())
    };

    var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuration.GetSection("AppSettings:Token").Value));
    var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha512Signature);

    var token = new JwtSecurityToken(
        claims: claims,
        expires: DateTime.Now.AddDays(1),
        signingCredentials: creds);

    var jwt = new JwtSecurityTokenHandler().WriteToken(token);

    return jwt;
}
```

Figura 30 - Método auxiliar CreateTokenFuncionario()

Neste método o que irá diferenciar entre os utilizador a sua token é a Role atribuída, há a possibilidade do funcionário ser um “Funcionário” ou um “Gerente”, foi criada uma Role à parte para testes e para efetuar qualquer operação sem problema chamada de “Admin”, que é atribuída a funcionários com o nome “adminaccount”

Para fazer a Autenticação dos requests, o utilizador que solicita um serviço precisa de ter uma token de sessão em que a Role corresponda com as Roles permitidas no serviço, segue-se a seguir um exemplo:

```
[HttpPatch("{targetID}"), Authorize(Roles = "Admin, Gerente, Funcionario, Cliente")]
0 references
public async Task<IActionResult> Patch([FromBody] Cliente cliente, int targetID)
```

Figura 31 - Headers de Authorize num Request HttpPatch

## 17.2.3. Funcionalidade de Login (Encriptação de password)

Como já foi mencionado previamente, o programa apresenta uma funcionalidade de login e registo, da qual a palavra passe do utilizador será encriptada a partir do namespace fornecido pelo System, System.Security.Cryptography.

Para fazer a encriptação foram criados dois métodos auxiliares, utilizados no login e registo do utilizador:

```
6 references
public class PasswordEncryption
{
    4 references
    public static void CreatePasswordHash(string password, out byte[] passwordHash, out byte[] passwordSalt)
    {
        using (var hmac = new HMACSHA512())
        {
            passwordSalt = hmac.Key;
            passwordHash = hmac.ComputeHash(Encoding.UTF8.GetBytes(password));
        }
    }

    2 references
    public static bool VerifyPasswordHash(string password, byte[] passwordHash, byte[] passwordSalt)
    {
        using (var hmac = new HMACSHA512(passwordSalt))
        {
            var computeHash = hmac.ComputeHash(Encoding.UTF8.GetBytes(password));
            return computeHash.SequenceEqual(passwordHash);
        }
    }
}
```

Figura 32 - Métodos auxiliares CreatePasswordHash() e VerifyPasswordHash()

O CreatePasswordHash() é utilizado no registo de um cliente ou funcionário para encriptar a palavra-passe inserida pelo utilizador:

```
PasswordEncryption.CreatePasswordHash(newFuncionario.pass_salt, out byte[] passwordHash, out byte[] passwordSalt);
newFuncionario.pass_hash = Convert.ToBase64String(passwordHash);
newFuncionario.pass_salt = Convert.ToBase64String(passwordSalt);
```

Figura 33 - Tratamento de password no Request HttpPost de um Funcionário

Já o VerifyPasswordHash, como já foi possível ver no Login do ponto 17.2.2. deste projeto, é utilizado para verificar se uma palavra-passe inserida corresponde à palavra-passe registada na base de dados desencriptada:

```
if (!PasswordEncryption.VerifyPasswordHash(conta.password, Convert.FromBase64String(targetFuncionario.pass_hash), Convert.FromBase64String(targetFuncionario.pass_salt)))
{
    throw new ArgumentException("Password Errada.", "conta");
}
```

Figura 34 - Verificação de palavra-passe no Login de um Funcionário

### 17.2.4. Documentação de código com OpenAPI

De forma a fazer a documentação correta da API, foi utilizado o SwaggerUI, ferramenta que foi sugerida pelo professor de ISI (Óscar Ribeiro).

Começou-se por fazer a implementação do gerador do Swagger no Program.cs (que funciona como Startup da API):

```
c.SwaggerDoc("Alpha", new OpenApiInfo
{
    Title = "IPCAGym",
    Version = "Alpha",
    Description = "Web API para o aplicativo IPCAGym"
});
```

Figura 35 - Geração documentação no Swagger

Foi também exportada toda a documentação para um ficheiro .xml externo que se encontra no seguinte diretório:

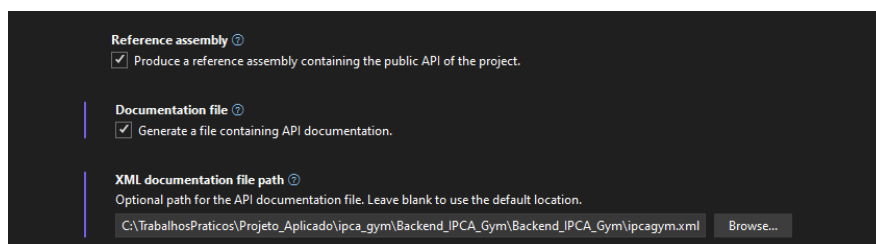


Figura 36 - Propriedades da Solução do projeto



## Entrega 4 de Projeto Aplicado



ESCOLA  
SUPERIOR  
DE TECNOLOGIA

A forma que foi usada para a documentação de métodos e classes foi a seguinte:

```
/// <summary>
/// Método http post para inserção de um novo ginásio
/// </summary>
/// <param name="newGinasio">Dados do novo ginásio a ser inserido</param>
/// <returns>Resposta do request que contém a sua mensagem e seu código em formato json</returns>
[HttpPost, Authorize(Roles = "Admin")]
public async Task<IActionResult> Post([FromBody] Ginasio newGinasio)
{
    string sqlDataSource = _configuration.GetConnectionString("DatabaseLink");
    Response response = await GinasioLogic.PostLogic(sqlDataSource, newGinasio);

    if (response.StatusCode != LayerBLL.Utils.StatusCodes.SUCCESS) return StatusCode((int)response.StatusCode);

    return new JsonResult(response);
}
```

Figura 37 - Documentação de métodos

```
namespace LayerBOL.Models
{
    19 references
    public class Ginasio
    {
        /// <summary>
        /// Id do ginásio
        /// </summary>
        /// <example>1</example>
        5 references
        public int id_ginasio { get; set; }
        /// <summary>
        /// Nome da instituição/estabelecimento
        /// </summary>
        /// <example>UMinhos</example>
        6 references
        public string instituicao { get; set; }
        /// <summary>
        /// Estado do ginásio (Ativo ou Inativo)
        /// </summary>
        /// <example>Ativo</example>
        6 references
        public string estado { get; set; }
        /// <summary>
        /// Foto do ginásio que poderá ser nula
        /// </summary>
        /// <example>C:\OneDrive\ginasio.png</example>
        11 references
        public string? foto_ginasio { get; set; }
        /// <summary>
        /// Contacto do ginásio
        /// </summary>
        /// <example>911922933</example>
        10 references
        public int contacto { get; set; }
    }
}
```

Figura 38 - Documentação de modelos de dados

Sendo este o resultado no Swagger:

Figura 39 - Resultado documentação no Swagger

### 17.3. Front-end

O front-end do projeto foi implementado na linguagem Kotlin, utilizando o IDE Android Studio, seguindo assim a metodologia da cadeira de Programação de Dispositivos Móveis (PDM).

Este foi o novo desafio para os membros de grupo deste projeto, tendo em conta que o mesmo apresentava experiência nula na linguagem e no funcionamento do IDE.

O projeto ficou dividido em duas partes:

- Activities – tratam-se das classes que fazem o funcionamento do front-end de forma interativa e também o tratamento de dados recebidos e/ou enviados para a API externa (back-end falado no ponto 17.2)
- Resources – tratam-se de ficheiros XML compostos pelo código necessário para a construção do visual do software a nível físico, estes apresentam Layouts, Menus e Drawables (package com todas as imagens e ícones utilizados na aplicação)

#### 17.3.1. Activities

As activities estão divididas por 3 partes distintas:

- Classes de tratamento de dados e funcionamento do front-end (cliente\_classes e funcionários\_classes): cada um desses packages possui as activities necessárias para cada funcionalidade da aplicação, sendo que cada uma está associada a um layout
- Modelos de dados (data\_classes e body\_models): possui os formatos de cada objeto a ser enviado e/ou recebido da API
- Requests (requests): possui objetos que tratam de fazer os requests todos do front-end para a API e espera uma resposta

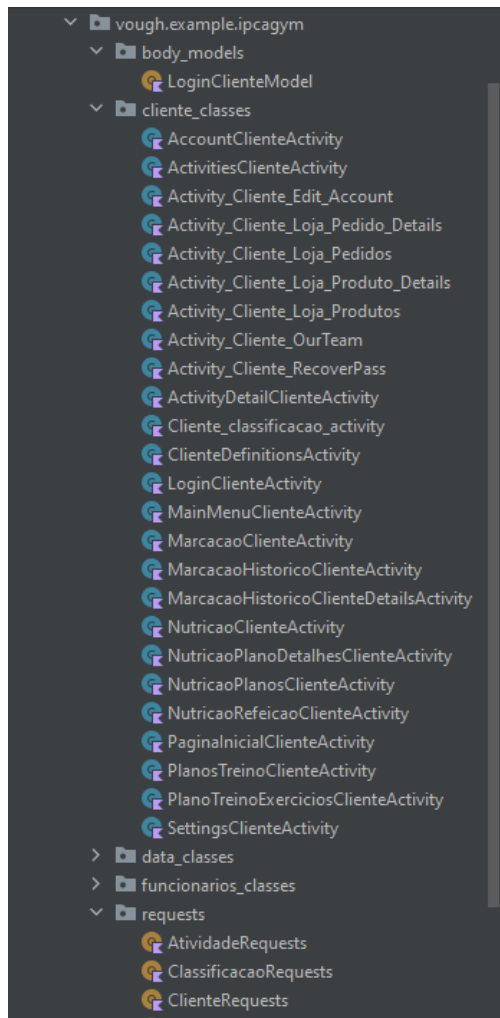


Figura 40 - Organização do Front End (Activities)

### 17.3.2. Resources

Esta parte contém a parte física do software, composto por vários packages, no entanto, apenas 3 estão a ser utilizadas:

- Menus: possuí os menus utilizados na aplicação, neste caso foram apenas utilizados duas barras de navegação inferiores que têm o funcionamento de menu
- Layouts: possuí os ficheiros responsáveis pela montagem da parte física e interativa do utilizador com o software, cada layout está associado a uma activity apenas
- Drawables: possuí todos os recursos utilizados na aplicação, essencialmente imagens e ícones

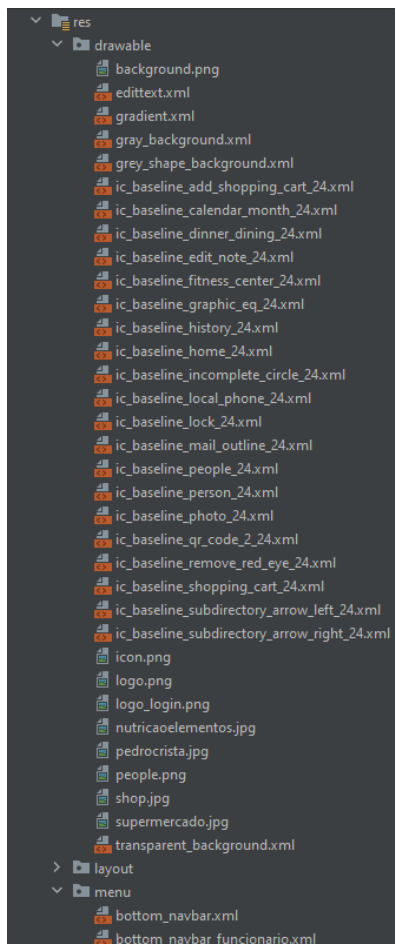


Figura 41 - Organização do Front-End (Resources)

### 17.4. Componentes físicos

O projeto apresenta apenas um componente físico externo à aplicação, que se trata de um leitor de cartões em Arduino (Sensor RFID).

O objeto deste aparelho seria fazer o registo de uma entrada e saída do ginásio IPCAGym a partir de cartões/pins eletrónicos, sendo assim registado na base de dados cada entrada e saída.

Para fazer a implementação deste projeto foi utilizado o como IDE o Arduino IDE e Visual Studio Code, o mesmo foi desenvolvido na linguagem C, de forma a consolidar as metodologias da cadeira de Sistemas Embebidos e de Tempo Real (SETR)

## 18. Avaliação

Segue-se a avaliação da entrega 4 do grupo de trabalho:

Aluno	Assiduidade	Participação	Comunicação	Respeito entre	Cumprimento de prazos	Organização	Iniciativa	Qualidade de trabalho	Autonomia	Total
Pedro Simões	20	20	20	20	20	20	20	20	20	20
Gonçalo Cunha	20	20	20	20	20	20	20	20	20	20
João Apresentação	20	20	20	20	20	20	20	20	20	20

Figura 42 - Autoavaliação

## 19. Conclusão

O progresso atual do projeto foi possível com o recurso a ferramentas de suporte, com uma distribuição por sprints semanais e organização de reuniões para estruturação das tarefas.

Com esta 4ª entrega foi possível terminar todo o Back-End e parte do Front-End, estando agora a ser inicializado a ligação entre as 2 componentes e também o projeto relativamente ao componente externo de leitor de cartões (Sensor RFID)

## 20. Bibliografia

### Repositório GitHub

[https://github.com/Presentation12/lpca\\_Gym](https://github.com/Presentation12/lpca_Gym)

### Figma

<https://www.figma.com/file/Q4tM34gl91b9fhrGvUeXRr/MileriuPT's-teamlibrary?node-id=0%3A1&t=p9cWit1VJHUNwf2m-1>

### Sites para produção dos diagramas:

<https://app.diagrams.net>

<https://online.visual-paradigm.com/pt/login.jsp>

### Sites para pesquisa e suporte no desenvolvimento de código:

<https://stackoverflow.com>

<https://dev.to>

<https://www.youtube.com>

### Material solicitado pelos docentes:

- ISI:

- [https://elearning2.ipca.pt/2223/pluginfile.php/601521/mod\\_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2002%20-%20Web%20API%20Exemplo%20-%20ASP.NET%20Core.pdf](https://elearning2.ipca.pt/2223/pluginfile.php/601521/mod_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2002%20-%20Web%20API%20Exemplo%20-%20ASP.NET%20Core.pdf)
- [https://elearning2.ipca.pt/2223/pluginfile.php/604923/mod\\_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2005b%20-%20OAuth%20RESTful%20Services\\_v2.pdf](https://elearning2.ipca.pt/2223/pluginfile.php/604923/mod_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2005b%20-%20OAuth%20RESTful%20Services_v2.pdf)
- [https://elearning2.ipca.pt/2223/pluginfile.php/604543/mod\\_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2006%20-%20Web%20API%20Documentação%20API%20.pdf](https://elearning2.ipca.pt/2223/pluginfile.php/604543/mod_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2006%20-%20Web%20API%20Documentação%20API%20.pdf)
- [https://elearning2.ipca.pt/2223/pluginfile.php/604542/mod\\_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2003%20-%20Cliente%20HTTP%20-%20ASP.NET%20Core.pdf](https://elearning2.ipca.pt/2223/pluginfile.php/604542/mod_resource/content/0/ESI-ISI%202022-23%20-%2005%20-%20Cloud%20-%2003%20-%20Cliente%20HTTP%20-%20ASP.NET%20Core.pdf)

- PDM:

- <https://github.com/lgleto?tab=repositories>

21. Assinaturas

*Pedro Simões*

---

**(Pedro Simões)**

*Gonçalo Cunha*

---

**(Gonçalo Cunha)**

*João Apresentação*

---

**(João Apresentação)**