

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  
**PUC Minas Virtual**  
**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

PUCfit

Lucas Ribeiro Cardoso

Belo Horizonte  
Maio de 2023.



# Projeto Integrado

## *Sumário*

Projeto Integrado	3
1. Cronograma de Trabalho	4
2. Introdução	5
2.1 Objetivos do Trabalho	5
2.2 Apresentação do problema	5
2.3 Descrição geral do software	5
3. Definição Conceitual da Solução	7
3.1 Diagrama de Casos de Uso	7
3.2 Requisitos Funcionais	7
3.3 Requisitos Não-funcionais	9
4. Protótipo Navegável do Sistema	9
5. Diagrama de Classes de Domínio	10
6. Arquitetura da Solução	10
6.1 Padrão Arquitetural	10
6.2 C4 model - Diagrama de Contexto	11
7. Frameworks de Trabalho	11
8. Estrutura Base do Front End	112
9. Modelo Relacional ou Projeto de Banco de Dados NoSQL	14
10. Plano de Testes	155
11. Apropriação de Horas no Projeto	166
12. Código da Aplicação	16
13. Avaliação Retrospectiva	177
13.1 Objetivos Estimados	17
13.2 Objetivos Alcançados	17
13.2 Lições aprendidas	17
14. Referências	18

## 1. Cronograma de Trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
03/04/2023	05/04/2023	1. Revisão de todo o material disponibilizado na disciplina do Projeto Integrado (textos e vídeos).	Entendimento geral do Projeto Integrado.
06/04/2023	07/04/2023	2. Mapeamento da documentação de escopo para identificação de atores, requisitos e outras informações necessárias para o início do Projeto.	Lista de atores, requisitos e outras informações importantes.
07/04/2023	08/04/2023	3. Desenvolvimento dos Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Software.	Definição conceitual / introdução.
08/04/2023	09/04/2023	4. Identificação de ferramenta e desenvolvimento de diagrama de Caso de Uso.	Diagrama de Caso de Uso.
11/04/2023	13/04/2023	5. Criação do Protótipo de Interface navegável e seus itens relacionados.	Protótipo navegável.
24/04/2023	25/04/2023	6. Identificação de ferramenta e desenvolvimento de diagrama de Classes de Domínio.	Diagrama de Classes de Domínio.
26/04/2023	26/04/2023	7. Definição do Padrão Arquitetural do Projeto.	Descrição do Padrão Arquitetural definido para o Projeto.
27/04/2023	28/04/2023	8. Identificação de ferramenta e desenvolvimento de diagrama de Contexto.	Diagrama de Contexto.
29/04/2023	30/04/2023	9. Descrever Frameworks e Estrutura Base do Front End.	Descrição dos Frameworks e apresentação do Layout (do Front-End).
02/05/2023	04/05/2023	10. Identificação de ferramenta e desenvolvimento do Projeto de Banco de Dados.	Projeto de Banco de Dados.
05/05/2023	05/05/2023	11. Definição do Plano de Testes.	Plano de Testes.
05/05/2023	15/05/2023	12. Implementação / codificação.	Codificação (Back-End e Front-End).
18/05/2023	18/05/2023	13. Armazenamento da codificação em repositório de código (GitHub).	Link do repositório do código.
18/05/2023	18/05/2023	14. Finalização do Projeto com as descrições Finais.	Descrições finais, como: retrospectiva, objetivos estimados e alcançados e lições aprendidas.

## **2. Introdução**

### **2.1. Objetivos do trabalho**

O objetivo geral deste Trabalho é desenvolver uma aplicação que forneça um serviço completo às academias - mais especificamente às academias de CrossFit.

Inicialmente, a aplicação será desenvolvida somente para mobile (Android), onde tem-se uma única interface que pode ser acessada, pelos Empregados do estabelecimento e seus Clientes (Aluno), essa interface fornecerá todos os recursos necessários para ambos.

### **2.2. Apresentação do problema**

Com o avanço tecnológico, a popularização dos Smartphones e o acesso à Internet, aplicativos e softwares vêm ganhando espaço para facilitar o cotidiano de empresas e usuários.

Pensando nessas resoluções criadas para facilitar o dia a dia, foi identificada a necessidade da criação de uma solução para facilitar tanto os processos internos de quanto o atendimento ao público de academias de CrossFit. A aplicação tem em vista facilitar as rotinas de trabalho, especialmente dos Professores, no quesito de ter um controle de seus alunos e aulas de maneira mais ágil, possibilitando assim, um maior contato com seus alunos (Clientes) durante as aulas.

Pelo lado do cliente, a disponibilização de uma plataforma simples e objetiva, onde é possível visualizar as aulas que serão ministradas para ele e, caso não tenha tempo disponível para se deslocar à academia e realizar as atividades, possa reproduzir os exercícios mais simples em sua casa. Será possível, também, visualizar sua presença, realizar pagamentos, consultas etc.

### **2.3. Descrição geral do software**

Esse software tem por objetivo fornecer uma aplicação que possa conectar o estabelecimento ao Empregado (mais especificamente, Professor) e a um Cliente (Aluno), facilitando assim todo o processo.

O Empregado (Recepcionista) será o responsável pelo cadastramento do Empregado (Professor) em eventual contratação e Cliente (aluno) quando aderir ao plano da academia. Registrará os períodos de férias dos funcionários, pagamentos etc.

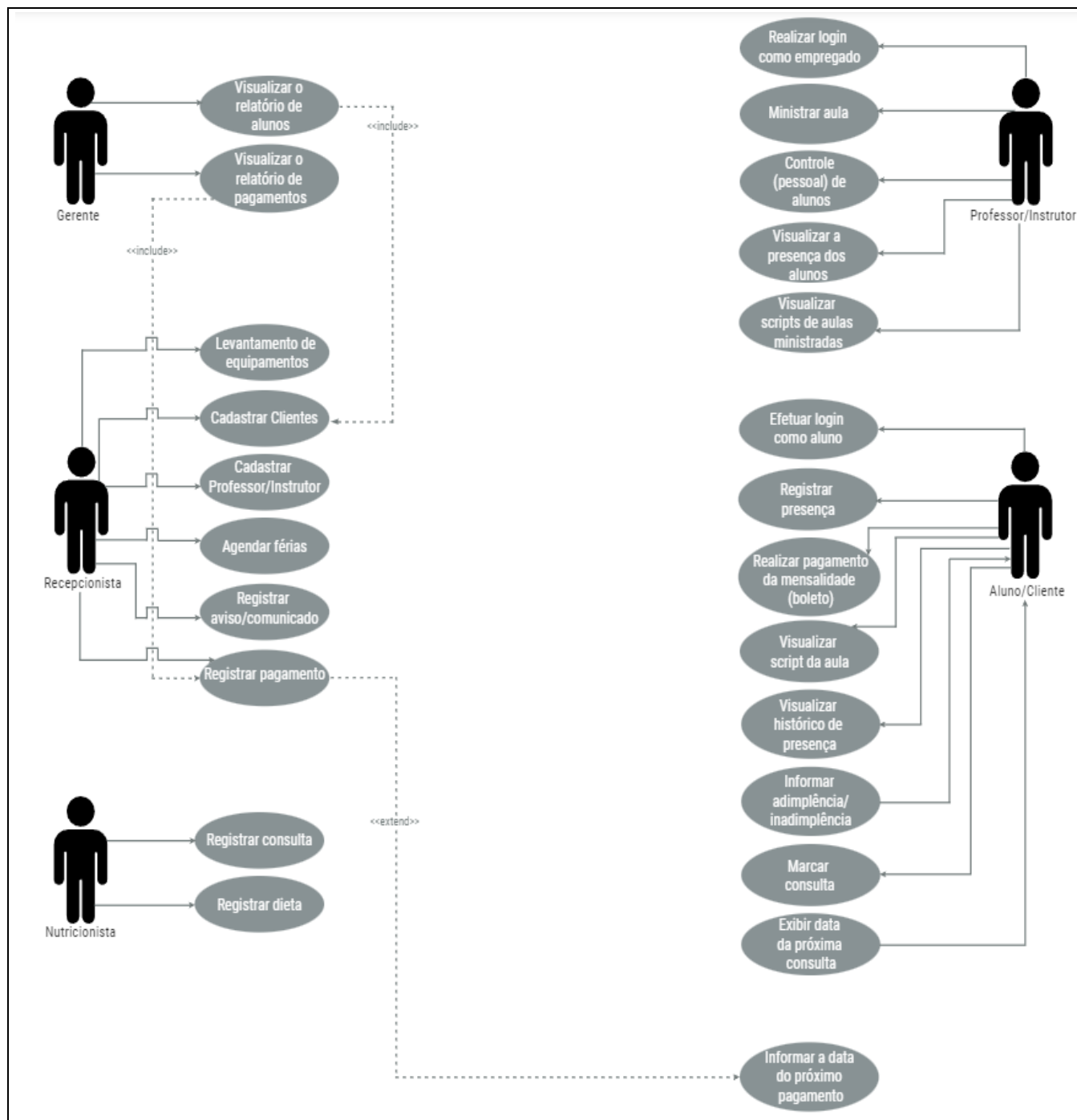
## PUCfit

Os Empregado (Professor), além de outras funcionalidades, terá o foco mantido em: realizar o cadastro de alunos (para o controle pessoal) e disponibilizar os scripts de suas aulas.

O Cliente (Aluno) poderá fazer o cadastro na plataforma, criando seu perfil e poderá fazer consultas de suas aulas, presenças, pagamentos etc.

### 3. Definição Conceitual da Solução

#### 3.1 Diagrama de Casos de Uso



#### 3.2 Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	Um usuário (Recepcionista) deve ser capaz de cadastrar um novo usuário (Aluno, Professor)	B	A

RF02	Um usuário deve ser capaz de realizar o login em seu perfil como (Empregado**) ou (Aluno)	B	A
RF03	Um usuário (Recepcionista) deve ser capaz de registrar o período de férias dos funcionários	A	A
RF04	Um usuário (Aluno) deve ser capaz de visualizar seus pagamentos	M	M
RF05	Um usuário (Professor) deve ser capaz de cadastrar novos alunos para seu controle	B	A
RF06	Um usuário (Professor) deve preencher todas as informações para cadastrar alunos em seu controle	B	M
RF07	Um usuário (Professor) deve ser capaz de listar todos os alunos em seu controle	B	M
RF08	Um usuário (Professor) deve ser capaz de excluir um aluno em seu controle	M	A
RF09	Um usuário (Professor) deve ser capaz de ver todas as informações de seus alunos em seu controle	M	B
RF10	Um usuário (Professor) deve ser capaz de cadastrar um script da aula	M	A
RF11	Um usuário (Professor) deve ser capaz de listar todos os scripts de suas aulas	B	M
RF12	Um usuário (Aluno) deve ser capaz confirmar a presença na aula	M	M
RF13	Um usuário (Aluno) deve ser capaz de cancelar a presença na aula	B	B
RF14	Um usuário (Professor) deve ser capaz de verificar a presença dos alunos por aula	A	M
RF15	Um usuário (Aluno) deve ser capaz de ver suas presenças	B	M
RF16	Um usuário (Aluno) deve ser capaz de ver os scripts de aula	B	M
RF17	Um usuário (Recepcionista), ao receber a confirmação de pagamento de mensalidade do usuário (Aluno), deve ser capaz de registrar o pagamento no sistema	M	A
RF18	Um usuário (Aluno) deve ser capaz de marcar consultas	A	A
RF19	Um usuário (Aluno) deve ser capaz de ver a data da próxima consulta que está agendada	B	B
RF20	Um usuário (Recepcionista) deve ser capaz de gerar um aviso/comunicado	M	M

\* B = Baixa, M = Média, A = Alta.

\*\* Empregado = Recepcionista, Instrutor/Professor e Gerente.



### 3.3 *Requisitos Não-funcionais*

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve apresentar um tempo de resposta abaixo de 2 segundos no processamento de todas as operações, com exceção do login	M
RNF02	O sistema deverá executar em plataforma Android	A
RNF03	O sistema deve estar disponível em qualquer período, funcionando em um regime 24/7	A
RNF04	O sistema deve ser dimensionado para suportar, pelo menos, 70 usuários conectados simultaneamente	M
RNF05	O sistema deve garantir a segurança das senhas dos usuários (Alunos), criptografando-as ao serem inseridas no Banco de Dados	A
RNF06	O sistema deve ser de fácil usabilidade, possibilitando assim a agilidade do manuseio	B

## 4. *Protótipo Navegável do Sistema*

**Link do vídeo (Google Drive):**

[https://drive.google.com/drive/folders/1AI3\\_vGEiu525WaY6OCDKVuE8e\\_iQsTYy?usp=sharing](https://drive.google.com/drive/folders/1AI3_vGEiu525WaY6OCDKVuE8e_iQsTYy?usp=sharing)

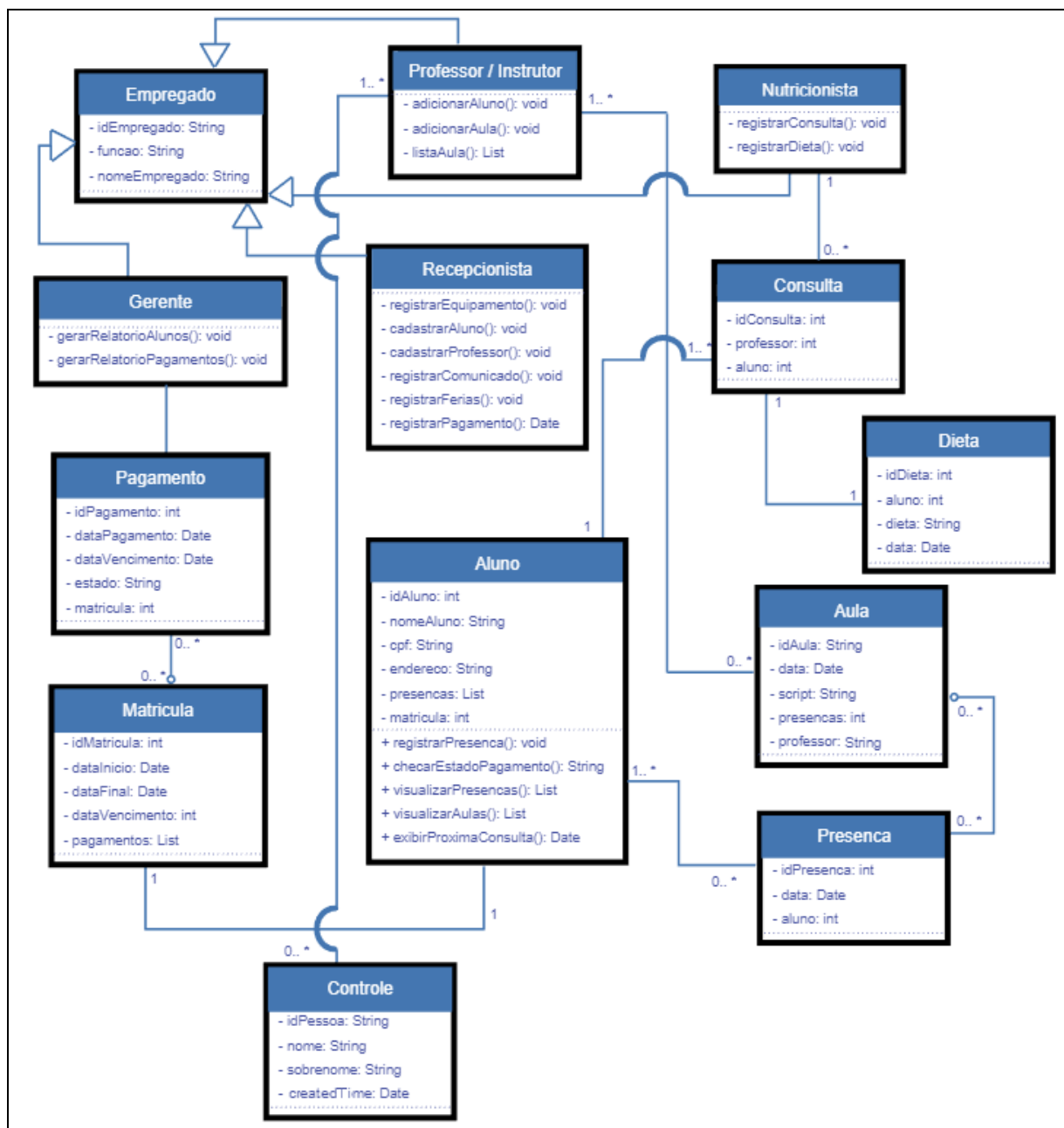
**Link do protótipo navegável:**

<https://www.figma.com/proto/DcBsEr2SjKyRykNklY3QS2/PUCfit?node-id=27-1341&starting-point-node-id=27%3A1341>

**Link das telas (draft):**

<https://www.figma.com/file/DcBsEr2SjKyRykNklY3QS2/PUCfit?type=design&node-id=0-1&t=8lD2FCYrwpvoSYR6-0>

## 5. Diagrama de Classes de Domínio



## 6. Arquitetura da Solução

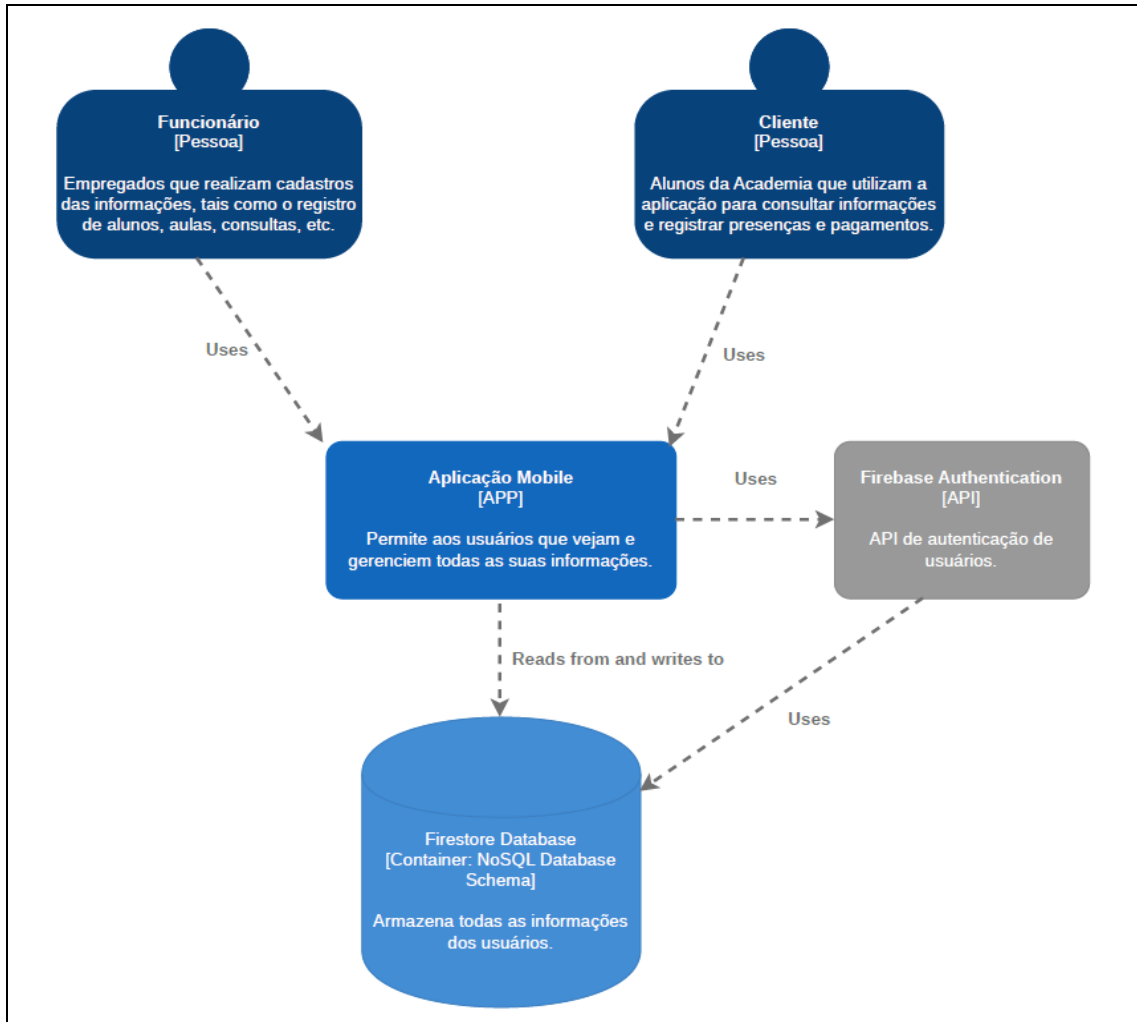
### 6.1 Padrão Arquitetural

O Padrão arquitetural escolhido para o projeto foi o MVC, sendo que a arquitetura foi modularizada utilizando TDD, em alguns Domínios não foi necessária a criação da camada de View, e a camada Model foi mantida com a mesma nomenclatura. Por fim, foi-se utilizado Adapter para auxiliar as Views.

## 6.2 C4 model - Diagrama de Contexto

Tem-se abaixo, o Diagrama de Contexto da Aplicação, onde é apresentado a macro arquitetura desta.

Os usuários, podendo ser tanto um Empregado como um Cliente, que acessa o aplicativo Mobile através de seus dispositivos Android, se comunica com uma API que realiza a autenticação do usuário e, ambos por sua vez, se comunicam com a Bases de Dados, onde são persistidas todas a informações referentes a Aplicação.



## 7. Frameworks de Trabalho

No projeto, foi-se utilizada uma combinação de frameworks e tecnologias para desenvolver uma solução simples e completa.

Para o desenvolvimento do Front-End, foi utilizado o Android Studio como a IDE principal. Essa IDE proporciona um ambiente de desenvolvimento integrado completo para criar interfaces de atraentes e responsivas para dispositivos Android. Foi utilizado somente o XML para definir a estrutura e o layout da interface do usuário.

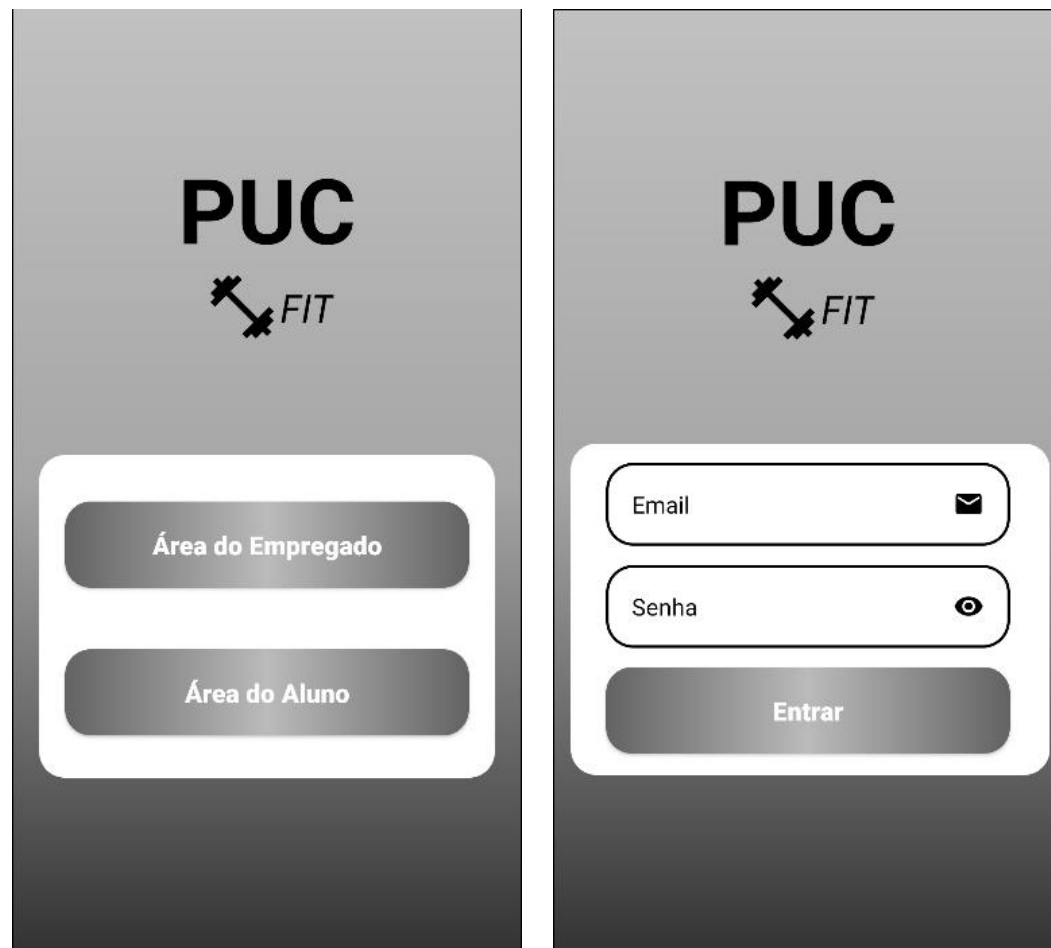
Já para o Back-End, foi utilizada a linguagem Java, que é uma das linguagens mais populares para o desenvolvimento de aplicativos empresariais.

Para a persistência de dados, utilizou-se o Firestore Database, que é um banco de dados NoSQL escalável e de tempo real fornecido pelo Firebase. Ele permitiu armazenar e gerenciar dados de forma eficiente, além de fornecer recursos de sincronização em tempo real para manter os dados atualizados em todos os dispositivos.

Além dos frameworks e tecnologias mencionados acima, foi utilizada também outra ferramenta do Firebase, o Firebase Authentication para gerenciar a autenticação de usuários do sistema.

Em resumo, utilizou-se a IDE Android Studio tanto para o Back-End (em Java), quanto para o Front-End (XML) e, o Firestore Database para persistência de dados. Essas ferramentas permitiram desenvolver uma solução completa oferecendo uma interface atraente, porém simples e responsiva, com uma lógica de negócios eficiente e armazenamento de dados seguro e escalável.

## **8. Estrutura Base do Front End**





Bem vindo(a)!

 professor1@pucfit.com

CADASTRAR ALUNOS

LISTA DE AULAS

Deslogar

Cadastre um novo(a) aluno(a) à sua lista.

Nome: Adriano Sobrenome: Alves

ADICIONAR ALUNO(A)

Clique em uma pessoa para deletá-la.

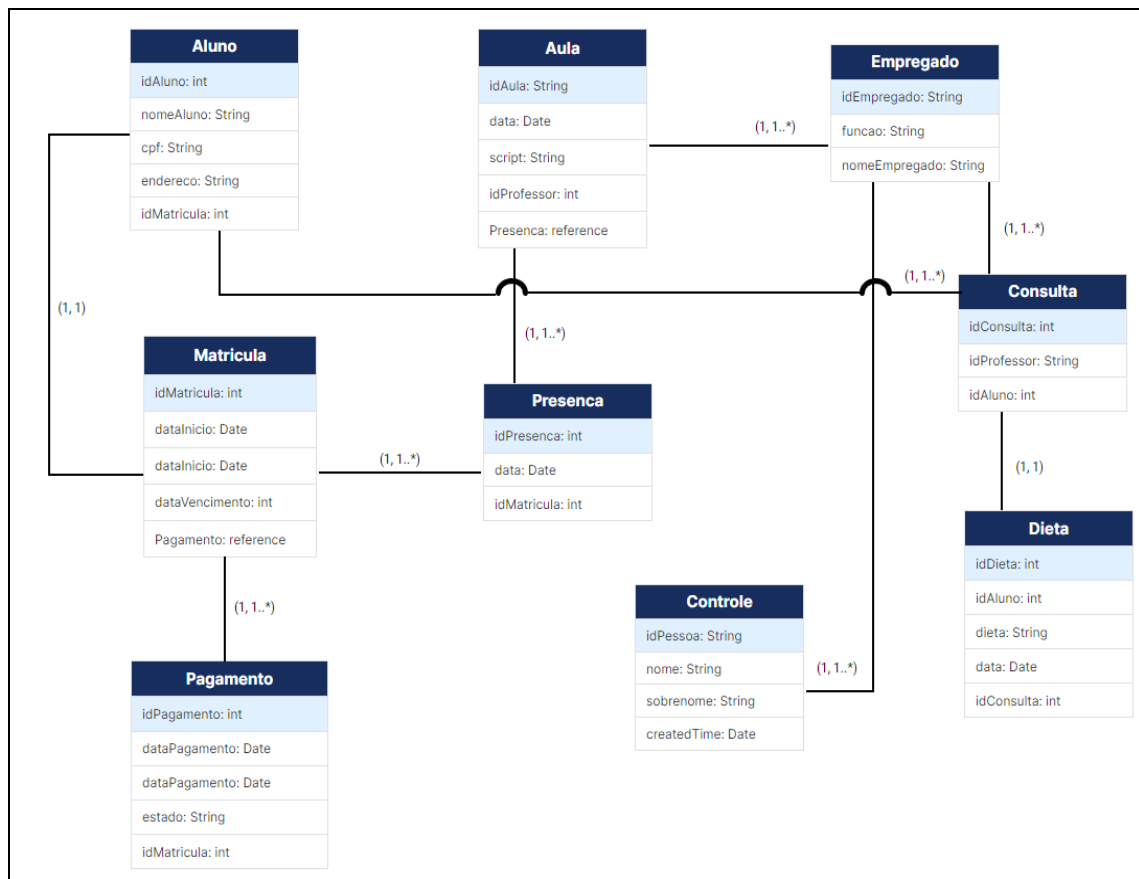
ID: aalves Nome: Adriano Sobrenome: Alves Criado em 17-05-23	ID: asantos Nome: Adriana Sobrenome: Santos Criado em 17-05-23
ID: cfreitas Nome: Cristiane Sobrenome: Freitas Criado em 17-05-23	ID: fsantana Nome: Francisco Sobrenome: Santana Criado em 17-05-23
ID: clopes Nome: Claudio Sobrenome: Lopes Criado em 17-05-23	ID: bandrade Nome: Bruno Sobrenome: Andrade Criado em 17-05-23
ID: érodrigues Nome: Édson Sobrenome: Rodrigues Criado em 17-05-23	ID: afreitas Nome: Aline Sobrenome: Freitas Criado em 18-05-23

Aula

Script da aula

NOVA AULA ATUALIZAR

## 9. Modelo Relacional ou Projeto de Banco de Dados NoSQL



## 10. Plano de Testes

Número	Caso de uso	Objetivo do caso de teste	Entradas	Resultados esperados
1	Cadastrar aluno	Validar se o sistema permite cadastrar um novo(a) aluno(a) no controle pessoal do Professor/Instrutor, quando os dados são preenchidos corretamente.	- Clicar em cima dos campos “Nome” e “Sobrenome” e selecionar os nomes desejados; - Clicar em “Adicionar aluno(a)”.	O sistema deve apresentar uma mensagem de sucesso e representar um retângulo com o “ID”, “Nome”, “Sobrenome” e a data em que o aluno foi cadastrado.
2	Cadastrar aluno	Validar se o sistema não permite que o usuário cadastre um aluno(a) com algum campo em branco.	- Clicar em cima dos campos “Nome” e “Sobrenome” e tentar deixar algum dos campos (ou ambos) em branco.	O sistema não deve permitir que o usuário deixe o(s) campo(s) em branco.
3	Excluir aluno	Validar se o sistema permite efetuar a exclusão de um aluno(a) no controle pessoal do Professor/Instrutor.	- Clicar em cima do retângulo com os dados do aluno(a).	O sistema deve apresentar uma mensagem de sucesso e não deve mais representar as informações do aluno(a) na tela.
4	Excluir aluno	Validar se o sistema não permite efetuar a exclusão de um aluno(a) no controle pessoal do Professor/Instrutor, quando o usuário não realiza a ação corretamente.	- Clicar fora do retângulo com os dados ou pressionar o dedo em cima e arrastar para alguma direção (não efetuando um "clique").	O sistema não deve exibir nenhuma mensagem de erro e deve continuar a representar as informações do aluno(a) na tela.
5	Cadastrar aula	Validar se o sistema permite cadastrar uma nova aula.	- Clicar em cima do botão "Nova aula".	O sistema deve apresentar uma mensagem de sucesso.
6	Cadastrar aula	Validar se o sistema não permite cadastrar uma nova aula, quando não houver conexão com a Internet.	- Preencher os campos "Aula" e "Script da aula" corretamente; - Forçar a perda da conexão com a Internet; - Clicar em "Nova aula".	O sistema não deve realizar nenhuma ação até que a conexão com a Internet seja reestabelecida.

## 11. Apropriação de Horas no Projeto

Histórico de apropriação de horas		
Data do registro	Atividade	Quantidade de horas
05/04/2023	Revisão de todo o material disponibilizado na disciplina do Projeto Integrado (textos e vídeos).	17
07/04/2023	Mapeamento da documentação de escopo para identificação de atores, requisitos e outras informações necessárias para o início do Projeto.	10
08/04/2023	Desenvolvimento dos Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Software.	6
09/04/2023	Identificação de ferramenta e desenvolvimento de diagrama de Caso de Uso.	14
13/04/2023	Criação do Protótipo de Interface navegável e seus itens relacionados.	22
25/04/2023	Identificação de ferramenta e desenvolvimento de diagrama de Classes de Domínio.	12
26/04/2023	Definição do Padrão Arquitetural do Projeto.	8
28/04/2023	Identificação de ferramenta e desenvolvimento de diagrama de Contexto.	10
30/04/2023	Descrever Frameworks e Estrutura Base do Front End.	12
04/05/2023	Identificação de ferramenta e desenvolvimento do Projeto de Banco de Dados.	20
05/05/2023	Definição do Plano de Testes	4
15/05/2023	Implementação / codificação	56
18/05/2023	Armazenamento da codificação em repositório de código (GitHub).	1
18/05/2023	Finalização do Projeto com as descrições Finais.	3

## 12. Código da Aplicação

Repositório com o código da aplicação (GitHub):

<https://github.com/lribeirocardoso/PUCfit.git>

Link do vídeo e do APK (Google Drive):

<https://drive.google.com/drive/folders/1fkIeDnFJ85K1aEAIPBTvza20MW3nABE?usp=sharing>



Obs.: existem dois vídeos dentro da pasta "App\_versãofinal". Ambos os vídeos são similares, sendo um realizando testes em meu aparelho pessoal e o outro no emulador da própria IDE Android Studio.

Recomendo ver o vídeo do emulador e, caso o vídeo não abra por algum motivo, assistir ao gravado pelo meu dispositivo.

**Link da aplicação hospedada no Appetize.io:**

<https://appetize.io/app/5ufs46t7m6pj6lrbgqfwmz3jsu?device=pixel4&osVersion=11.0&scale=75>

### **13. Avaliação Retrospectiva**

Durante todo o período do curso, em principal na realização deste projeto, obtive muitas lições aprendidas e desafios vencidos. Além de tudo ser muito novo para mim, pelo fato de estar em processo de transição de carreira, um dos maiores foi em relação ao cronograma. Manter uma rotina para adequar ao cronograma foi extremamente complicado, devido às outras atividades.

O resultado obtido foi satisfatório, mesmo porque o pequeno projeto foi feito do zero, sendo modelado conforme alguns conhecimentos adquiridos no curso.

#### **13.1 Objetivos Estimados**

Os objetivos estimados, desde o início do projeto, sempre foram focados no cronograma (em como distribuir o tempo) e na real possibilidade de usabilidade da aplicação no cotidiano de uma academia.

#### **13.2 Objetivos Alcançados**

Os objetivos propostos em relação ao cronograma foram alcançados, frequentemente com dificuldades, mas sempre alcançados de forma que, aos poucos o projeto foi entregue com algumas funcionalidades que atendiam os critérios.

Em relação ao objetivo proposto da usabilidade da aplicação no cotidiano, um protótipo foi entregue a um Personal Trainer (que ministra aulas de CrossFit em uma academia) e ele relatou que utilizou a "Lista de aulas" tanto com os scripts diários de aulas, quanto como um banco de dados pessoal para a criação de novos scripts futuros, podendo surgir assim, uma nova "feature" para a aplicação.

### ***13.2 Lições aprendidas***

	<b>Retrospectiva (Lições Aprendidas)</b>	
	<b>Descrição da Lição</b>	<b>Classificação</b>
1	Conhecimento adquirido no decorrer de desenvolvimento do Projeto.	Positiva
2	Utilização de conceitos teóricos ao Projeto.	Positiva
3	Ideias para serem aplicadas em projetos reais.	Positiva
4	Desenvolvimento de um cronograma adequado e conciso.	Positiva
5	Planejamento detalhado para evitar retrabalho.	Negativa

## **14. Referências**

FROZZA, Angelo A.; SCHREINER, Geomar A.; MELLO, Ronaldo dos S. **Projeto de Bancos de Dados NoSQL**. Short courses of the 37th Brazilian Symposium on Data Bases. Búzios – RJ, 2022.