



IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Anabela Pais Tavares

novembro| 2017





Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

PLATAFORMA ONLINE DE APTIDÃO FÍSICA PARA IDOSOS

ANABELA PAIS TAVARES

RELATÓRIO PARA A OBTENÇÃO DO GRAU DE LICENCIADO

EM ENGENHARIA INFORMÁTICA

Novembro de 2017

Ficha de Identificação

Aluno:

Anabela Pais Tavares

Nº 1011109

Licenciatura: Engenharia Informática

Estabelecimento de Ensino:

Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

Orientador de Estágio:

Nome: Doutor Noel Lopes

Grau académico: Doutorado

Agradecimentos

Queria começar por agradecer ao meu orientador, Professor Doutor Noel Lopes por toda a disponibilidade mostrada ao longo deste trabalho e por me ter dado a oportunidade de fazer um projeto com real interesse, de trabalho futuro e aplicabilidade. Agradecer todas as sugestões e críticas que fizeram deste trabalho o melhor possível. O meu sincero obrigado.

Não podia deixar de agradecer também o apoio e dedicação na ajuda da realização do presente relatório à Professora Filipa Gaudêncio. A sua ajuda foi crucial durante as aulas e também fora das mesmas.

Agradeço ainda à Professora Carolina Vila-Chã e Professora Alexandra Fonseca o apoio dado no projeto de forma direta e indireta. Sem a colaboração destas o resultado final teria certamente ficado aquém do desejado. Devo a qualidade deste projeto também ao vosso contributo.

Além disso, gostaria de agradecer ao meu Pai, Mãe, Irmão e restante família por me terem proporcionado a oportunidade de tirar o curso. Sem o apoio e compreensão deles não teria sido possível este percurso.

Por fim, agradeço aos meus amigos mais próximos e namorado, que desde sempre acreditaram e estiveram presentes quando eu precisei, para me dar força, ânimo e por serem essenciais na minha vida. Sem o apoio destes tudo teria sido mais difícil.

Resumo

O presente relatório descreve o projeto realizado no âmbito da unidade curricular Projeto de Informática, integrada na Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda.

O projeto desenvolvido consiste numa plataforma *web* para gerir a aptidão física dos utentes, do programa Guarda +65, do ginásio do IPG, cujo principal objetivo é o de registar e organizar a documentação dos utentes, permitindo assim monitorizar e avaliar a aptidão física dos vários utentes.

Para o desenvolvimento do projeto foi utilizada a metodologia de desenvolvimento ágil, *Rational Unified Process* (RUP). A plataforma *web* foi desenvolvida utilizando ASP.NET e C# com recurso à base de dados SQLserver.

O projeto foi realizado com sucesso e as opiniões deferidas pelos órgãos envolvidos no programa Guarda +65, foram positivas. A implementação da plataforma utilizou tecnologias atuais, dados reais e um problema real o que tornou o projeto cativante e interessante de desenvolver.

Palavras-chave: *Web*, ASP.NET, Aptidão-física, Guarda +65;

Abstract

This report describes the project carried out within the scope of the final project of the Computer Engineering degree at the Higher School of Technology and Management, Polytechnic of Guarda.

The project consists of a *web* platform to manage the physical fitness of IPGym's Guarda +65 program participants, which classes are on the Polytechnic of Guarda gymnasium – IPGym. Their main objective is to register and organize data about the participants, thus allowing to monitor and evaluate their physical fitness.

This project was developed in an agile development methodology, *Rational Unified Process* (RUP). The *web* platform was developed using ASP.NET and C#, combined with a SQL server database.

The project was carried out successfully and the stakeholders involved in the Guarda +65 program approved it and gave positive feedback. The current solution was developed with current technologies, real data and a real problem which made the project captivating and interesting to be on.

Keywords: *Web*, ASP.NET, Physical-fitness, Guarda +65;

Índice Geral

Agradecimentos	III
Resumo	IV
Abstract.....	V
Índice Geral	VI
Índice de Figuras	IX
Índice de Tabelas	XI
Lista Acrónimos	XII
1. Introdução.....	1
1.1. Motivação	1
1.2. Requisitos funcionais da plataforma.....	1
1.3. Estrutura do Documento	2
2. Estado de Arte	5
2.1. Aplicações existentes.....	5
2.1.1. Physical Test 8.0.....	5
2.1.2. GymMaster	7
2.2. Análise crítica das soluções existentes	8
3. Metodologia.....	11
3.1. Metodologia de desenvolvimento ágil: Rational Unified Process (RUP).....	11
3.2. Descrição das tarefas	13
4. Análise de Requisitos	15
4.1. Objetivos previstos	15
4.2. Diagrama de Contexto	16
4.3. Atores e respetivos casos de uso.....	18
4.4. Diagrama de Casos de uso.....	18
4.5. Descrição dos casos de uso e Diagramas de Sequência	20
4.5.1. Registar novos utentes	20

4.5.2.	Registar Testes de Aptidão Física	23
4.5.3.	Registar Parâmetros de Testes de Aptidão Física.....	25
4.5.4.	Registar Assiduidade do utente	27
4.5.5.	Registar Planeamento de Aulas	29
4.6.	Diagrama de Classes.....	31
5.	Implementação da Solução	35
5.1.	Tecnologias Utilizadas	35
5.1.1.	HTML.....	35
5.1.2.	CSS	36
5.1.3.	ASP.NET	36
5.1.4.	JavaScript	37
5.1.5.	C#	37
5.1.6.	Microsoft SQL Server	37
5.1.7.	T-SQL.....	38
5.1.8.	AJAX Control Toolkit.....	38
5.2.	Diagrama de Hierarquia da plataforma	39
5.3.	Interfaces da Plataforma de Aptidão Física.....	42
5.3.1.	Interface principal da plataforma (Default).....	42
5.3.2.	Interface inserir Utentes.....	43
5.3.3.	Interface da inserção e edição de testes de aptidão física.....	45
5.3.4.	Interface consultar assiduidade dos utentes.....	47
5.3.5.	Calendário das sessões de aulas.....	48
6.	Testes.....	51
7.	Conclusão	53
	Bibliografia.....	55
8.	Anexos.....	57
8.1.	Mapa de Gantt	58
8.2.	Páginas da Plataforma Web.....	59
8.3.	Código da Plataforma	61
8.3.1.	Calcular a idade automaticamente.....	61

8.3.2.	Script SQL para adicionar dados em massa	62
8.3.3.	Imprimir dados em PDF	62
8.3.4.	Atualização do calendário depois de inserida uma nova aula	64
8.4.	Documentação entregue pelo cliente	66
8.4.1.	Testes de Aptidão Física.....	66

Índice de Figuras

Figura 1 – Algumas funcionalidades do Physical Test 8	6
Figura 2 – Módulos da Avaliação Física	7
Figura 3 – Plataforma GymMaster Online	8
Figura 4 – Fases do modelo RUP divididas em várias fases do projeto.....	12
Figura 5 – Diagrama de Contexto da Plataforma	17
Figura 6 – Diagrama de casos de uso	19
Figura 7 – Diagrama de Sequência “Registrar novo utentes”.....	22
Figura 8 – Diagrama de Sequência “Registrar Testes de Aptidão Física”	24
Figura 9 – Diagrama de Sequência “Registrar Parâmetros dos testes”	26
Figura 10 – Diagrama de sequencia “Registrar assiduidade do utente”	28
Figura 11 – Diagrama de sequência “Registrar Planeamento de aulas”	30
Figura 12 – Diagrama de classes da plataforma	32
Figura 13 - Modelo ER da plataforma.....	33
Figura 14 – Diagrama de Hierarquia da Plataforma.....	40
Figura 15 – Aspeto da página Login da plataforma.	41
Figura 16 – Página testes de aptidão física da plataforma.....	41
Figura 17 - Página dos parâmetros dos testes de aptidão física da plataforma.	42
Figura 18 – Página inicial da plataforma.....	43
Figura 19 – Preenchimento do campo data de aniversário.....	44
Figura 20 – Dropdownlist das freguesias	44

Índice de Figuras

Figura 21 – Demonstração do funcionamento das caixas de texto “invisíveis”.....	45
Figura 22 – Inserção de um novo teste de aptidão física.....	46
Figura 23 – Exemplo de edição de um teste de aptidão física.....	46
Figura 24 – Lista de utentes e sua presença.....	47
Figura 25 – Interface calendário da plataforma.....	48
Figura 26 – Caso de teste inserir sem todos os dados preenchidos.	51
Figura 27 – Mapa de Gantt.....	58

Índice de Tabelas

Tabela 1 – Casos de Uso	18
Tabela 2 – Descrição do caso de uso “Registar novos utentes”	20
Tabela 3 - Descrição do caso de uso “Registar Testes de Aptidão Física”	23
Tabela 4 – Descrição do caso de uso “Registar Parâmetros dos Testes”	25
Tabela 5 - Descrição do caso de uso “Registar Assiduidade do utente”	27
Tabela 6 – Descrição do caso de uso “Registar Planeamento de Aulas”	29

Lista Acrónimos

AJAX	Asynchronous Javascript And XML
ASP	Active Server Pages
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
ER	Entidade Relacionamento
EER	Enhanced Entity-Relationship
HTML	HyperText Markup Language
IDE	Integrated Development Environment
IMC	Índice de Massa Corporal
IPG	Instituto Politécnico da Guarda
NIF	Número de Identificação Fiscal
PDF	Portable Document Format
PHP	Hypertext Preprocessor
RFID	Radio-Frequency IDentification
RUP	<i>Rational</i> Unified Process
SGBD	Sistema de Gestão de Base de Dados
SQL	Structured Query Language
TIC	Tecnologias da Informação e Comunicação
UML	Unified Modeling Language
XML	eXtensible Markup Language

1. Introdução

O presente documento descreve o projeto desenvolvido pela aluna Anabela Pais Tavares, no âmbito da unidade curricular Projeto de Informática, da licenciatura em Engenharia Informática.

O projeto desenvolvido consistiu numa plataforma *web*, para gerir a informação dos utentes, de modo a avaliar o desempenho e evolução de alguns parâmetros de saúde, tais como, a tensão, o IMC (índice de massa corporal), massa gorda, entre outros.

1.1. Motivação

Atualmente, verifica-se, por parte da população em geral, a adoção de práticas saudáveis. Assente nesta ideia, surgiu o projeto “colocar todos a mexer”, promovido pelo Instituto Politécnico da Guarda e a Câmara Municipal da Guarda.

Neste contexto existiu a necessidade de criar uma plataforma *web* que possibilitasse às entidades participantes o registo da informação dos utentes (idosos), para posteriormente ajudar a monitorizar a evolução em diferentes parâmetros da aptidão física.

A nível pessoal, um dos aspetos que levaram à escolha deste projeto, foi a possibilidade de aplicar os conhecimentos adquiridos ao longo do curso e acima de tudo, poder criar uma plataforma *web* que vai ser uma mais valia para a comunidade.

1.2. Requisitos funcionais da plataforma

Numa fase inicial, após várias reuniões com as docentes envolvidas no projeto, foram definidos os requisitos funcionais para desenvolvimento da plataforma *web*. De seguida apresentam-se os requisitos definidos:

- Plataforma *online*

- Registrar informação do utente
- Registrar assiduidade do utente
- Registrar testes de avaliação
- Planeamento de testes e aulas
- Registrar condição física do utente

Para cumprir os requisitos funcionais foi necessário dividir o trabalho em várias fases. Neste caso:

- No estudo de outras aplicações que vão de encontro aos objetivos previstos;
- Na seleção da metodologia mais adequada para a conceção da plataforma;
- Na análise de requisitos para recolha dos dados necessários para alcançar os objetivos;
- Na escolha das tecnologias que foram utilizadas;
- No desenvolvimento da plataforma *web*, que se realizou em conjunto com a última fase (testes).

1.3. Estrutura do Documento

O relatório é constituído por sete capítulos e está estruturado da seguinte maneira: neste capítulo foi feita uma breve introdução sobre a plataforma *web* e os objetivos previstos para a construção da mesma. No capítulo dois são apresentadas as aplicações existentes no mercado que se enquadram no tema do projeto, assim como, uma comparação das mesmas em relação à plataforma *web* desenvolvida. No capítulo três é definida a metodologia para a análise de requisitos e é feita a descrição das tarefas que

foram realizadas ao longo do desenvolvimento do projeto. No capítulo quatro é descrita toda a análise de requisitos, tais como diagrama de casos de uso, descrição de caso de uso, diagrama de classes, etc. No capítulo cinco são apresentadas as tecnologias e descreve-se a implementação da solução proposta com algumas janelas da plataforma *web* para uma melhor compreensão. No capítulo sete, são expostos os testes que foram feitos ao longo da elaboração da plataforma *web* e por fim, no último ponto encontra-se a conclusão que descreve o trabalho desenvolvido como também alguma consideração mais relevante a ter, e as perspectivas de desenvolvimento que se pretendem efetuar no futuro.

2. Estado de Arte

Uma vez definidos os objetivos propostos para o desenvolvimento do projeto, foi necessário realizar uma pesquisa para o levantamento de soluções existentes.

Após realizada a pesquisa foram encontradas várias aplicações, mas apenas duas vão de encontro aos objetivos definidos inicialmente. Ambas as aplicações foram analisadas para adquirir informações que poderão ser benéficas e não benéficas para o desenvolvimento da plataforma *web* de aptidão para idosos.

As aplicações estudadas, embora realizem os objetivos inicialmente propostos, não se focam unicamente na análise da aptidão física. Uma das aplicações permite analisar um grande leque de parâmetros de saúde, quando na realidade para este projeto não é necessário tanto. A outra aplicação não analisa tantos parâmetros de saúde e destina-se mais para gerir as informações dos utentes no ginásio.

2.1. Aplicações existentes

Das várias aplicações existentes no mercado foram selecionadas duas para análise, a *Physical Test 8.0* [1] ¹ e a *GymMaster* [2] ².

2.1.1. *Physical Test 8.0*

A aplicação *Physical Test 8.0* [1] desenvolvida pela empresa Terrazul Informática Ltda. que é assumidamente líder de mercado há 20 anos consecutivos e, é considerado pelos maiores especialistas do setor como o mais simples, prático e completo *software* na aplicação da avaliação física [1].

A *Physical Test 8.0* é um *software* para avaliação física e prescrição de exercícios que permite um controlo total sobre os objetivos e melhorias de performance dos utentes.

¹ A aplicação *Physical Test 8.0* foi sugerida pela professora Carolina Vila-Chã.

² A aplicação *GymMaster* foi selecionada por se enquadrar nos objetivos propostos e também pelo seu design apelativo.

Foi criado a pensar num ambiente de fácil utilização para que o foco esteja concentrado no bem-estar dos utentes.

Na Figura 1 é apresentado o menu inicial do *Physical Test 8.0*, onde se poder ver os vários módulos que os autores apresentam para avaliação da atividade física dos utentes.

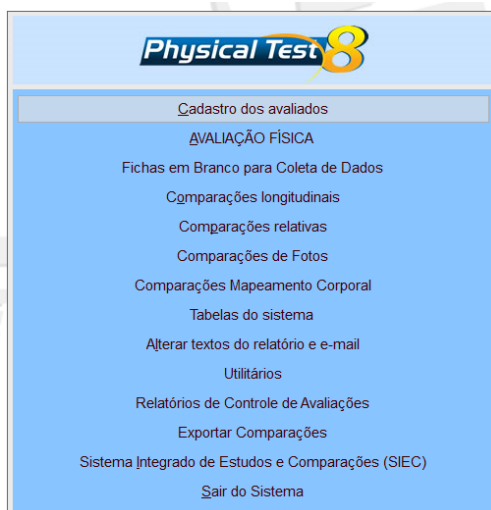


Figura 1 – Algumas funcionalidades do *Physical Test 8*, Fonte [1]

Estes módulos permitem processar vários tipos de informação como por exemplo, o registo de novos utentes onde estarão dados importantes para o sistema realizar os cálculos da avaliação física.

Outro módulo importante é o das “fichas em branco para coleta de dados”. Este permite que caso não haja um computador na sala, se registre os dados dos exercícios realizados pelos utentes e, mais tarde adicioná-los ao *Physical Test 8.0*.

Existem ainda outros módulos que fazem a avaliação física dos utentes, tais como:

- Anamnese, Risco Coronariano, Relação Cintura-Quadril, Perímetros, Composição Corporal, Fotos da Composição Corporal, Ajuda de Vídeos e Fotos, Avaliação Cardiorrespiratória, Testes Neuromotores, Análise Postural Digital, Avaliação Nutricional, Fichas de Ocorrências, Teste de 40 segundos,

Localização de Avaliado, Editar Textos de Relatórios, Impressão de Relatórios, Enviar Relatórios por E-mail, Nome do Avaliador, entre outros.

A Figura 2 apresenta o preenchimento de um formulário onde será feita a avaliação física através de uma atividade a um utente já registado no *Physical Test 8.0*.

Figura 2 – Módulos da Avaliação Física, Fonte [1]

Estes testes vão avaliar o utente e mostrar quais são os pontos críticos deste, dando assim uma perspetiva ao médico ou *personal trainer* de como está a saúde em geral e, assim poderão corrigir alguns pontos para uma melhor qualidade de vida. Além disso para complementar a avaliação do *personal trainer* ou médico, o *Physical Test 8.0* apresenta módulos de prevenção de saúde onde estão disponíveis testes para avaliar se o utente poderá ter problemas coronários, problemas de osteoporose, entre outros.

2.1.2. *GymMaster*

O *GymMaster* é desenvolvido pela *Treshna Enterprises Support* é uma plataforma *web* e, foi criada com o intuito de fornecer apoio a gestão dos ginásios para proporcionar, uma solução de *software* multifacetada que se adapta às necessidades dos ginásios, independentemente da dimensão, tanto a curto prazo como, a longo prazo [2].

A Figura 3 mostra uma visão geral da plataforma em funcionamento. Sendo uma plataforma *web* conseguimos aceder em várias páginas sem problema algum.



Figura 3 – Plataforma GymMaster Online, Fonte [2]

O *GymMaster* demonstra ser um *software* de fácil utilização e bastante abrangente. Permite a gestão de vários recursos, tais como a conta do utente e o seu envolvimento na organização, vários tipos de relatórios sobre os utentes e exercícios a que estes se propõem, reservas de equipamentos ou instalações, controlo de acessos (biometria ou cartões RFID (*Radio-Frequency IDentification*)), alertas de assiduidade do cliente, estatísticas e relatórios diversos, entre outros que permitem uma fácil análise da saúde do utente. Além disso, contempla um aplicativo móvel para o pessoal e os utentes [2].

2.2. Análise crítica das soluções existentes

Ambas as aplicações, com base no que foi apresentado, são boas e reúnem um leque de funcionalidades, que vão de encontro ao objetivo principal do desenvolvimento da plataforma *web* de aptidão para idosos.

A *Physical Test 8.0*, é uma aplicação forte porque reúne funcionalidades apelativas. Trata-se de uma aplicação que possibilita ao utilizador, um leque de módulos bastante completo para análise da saúde do utente como por exemplo, os módulos de prevenção de saúde. Estes permitem aos técnicos verificar a longo prazo se vão existir problemas coronários, problemas de osteoporose, entre outros. O ponto fraco desta aplicação é não ser uma plataforma *web* e, portanto, não haverá acesso fácil a partir de qualquer equipamento com um *browser* e internet.

A *GymMaster*, tem vindo a ganhar força com a facilidade de acesso que permite aos utilizadores, isto é, sendo uma plataforma *web* permite que os técnicos acedam em qualquer situação. Além disso também dispõe de alguns módulos interessantes, como por exemplo o controlo de acessos através de cartões RFID que permite uma verificação mais pormenorizada da assiduidade do utente. O ponto fraco revela-se nas funcionalidades disponíveis, em relação à aplicação anteriormente descrita podemos afirmar que esta aplicação é mais direcionada para a componente geral dos ginásios, deixando o módulo da aptidão física de lado.

A plataforma *web* que se pretende desenvolver vai ao encontro das duas aplicações descritas anteriormente. A facilidade de acesso da *GymMaster* e os módulos de análise da aptidão física da *Physical Test 8.0*, são funcionalidades a ter em conta na implementação da plataforma *web* de aptidão física para idosos.

De seguida, descreve-se a metodologia escolhida para análise da informação dada e a descrição das tarefas que se pretende realizar aquando da implementação da plataforma *web*.

3. Metodologia

Para o desenvolvimento de uma plataforma *web* é necessário seguir certas normas e critérios, agilizando com isto o processo de desenvolvimento. Para facilitar essas tarefas existem diferentes tipos de metodologias de desenvolvimento de *software*, tais como, a metodologia estruturada, metodologia orientada a objetos e metodologias de desenvolvimento ágil.

Optou-se pela escolha da metodologia de desenvolvimento ágil mais precisamente pelo processo RUP (*Rational Unified Process*). O forte envolvimento dos clientes no ciclo de vida do projeto, foi um dos pontos fulcrais na escolha da metodologia e do processo a utilizar. Além disso, a forma incremental (várias versões para demonstração no cliente), colaborativa (reuniões assíduas com o cliente, para ajuste de alguns pontos), direta (fácil de aprender e modificar ao longo do desenvolvimento) e adaptativa (capaz de responder às mudanças até o último instante) destacou-se perante os outros processos de desenvolvimento de *software*.

Nas secções abaixo, é descrito o processo escolhido (RUP) para o desenvolvimento da plataforma *web* e ainda a descrição das tarefas a realizar.

3.1. Metodologia de desenvolvimento ágil: *Rational Unified Process* (RUP)

O RUP é uma estrutura da metodologia de desenvolvimento ágil criada pela *Rational software*. Trata-se de uma estrutura que fornece um conjunto de práticas testadas na indústria de desenvolvimento de *software* e gestão de projetos. Este é documentado recorrendo á notação UML (*Unified Modeling Language*) [3].

O principal objetivo do RUP é atender as necessidades dos utilizadores garantindo assim uma produção de *software* de alta qualidade que cumpra um cronograma e um orçamento previsíveis.

O ciclo de vida do RUP é dividido em quatro fases, onde são tratadas variadas questões, tais como o planeamento, levantamento de requisitos, implementação e testes do *software*. Cada uma destas fases tem um papel fundamental para que o objetivo final seja cumprido.

Sendo assim, as quatro fases estão descritas abaixo: [4]

- Conceção: avaliou-se com as docentes envolvidas no projeto os vários objetivos a ter em conta no desenvolvimento da plataforma *web*;
- Elaboração: analisou-se de forma detalhada o problema e identificou-se os principais casos de uso da plataforma *web*;
- Construção: começou-se a desenvolver a solução com base nas fases anteriormente descritas;
- Transição: realizaram-se testes com dados reais e entregou-se ao cliente a solução final.

Ao longo das fases de desenvolvimento, houve reuniões com as docentes para ajustar funcionalidades da plataforma e apresentar o conteúdo já realizado.

A Figura 4 mostra como as fases e os componentes do processo se articulam. As atividades com maior ênfase são demonstradas através de uma altura maior na linha.

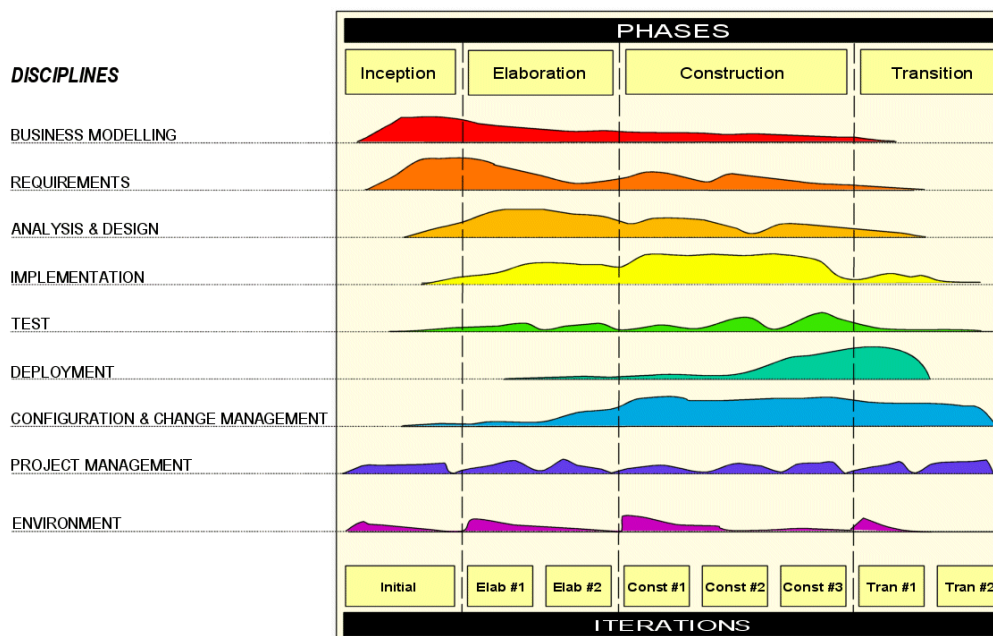


Figura 4 – Fases do modelo RUP divididas em várias fases do projeto, Fonte [1]

3.2. Descrição das tarefas

Com base nas reuniões com as docentes envolvidas e nos objetivos propostos, foram definidos os requisitos funcionais e não funcionais para o desenvolvimento da plataforma *web*. As tarefas iniciais foram as mais complexas uma vez que é nesta fase que são definidos os principais pontos a desenvolver. Depois que esta fase seja concluída, foi possível “tomar rédeas” e implementar a plataforma *web*.

É de salientar que ao longo do processo de desenvolvimento a fase de testes foi intercalada com a fase de codificação, porque estavam a ser corrigidos os erros que se encontravam nos testes.

Neste sentido foram identificadas as seguintes tarefas:

- Tarefa 1 – Recolher todas as funcionalidades que o sistema deverá no final possuir (Análise de Requisitos);
- Tarefa 2 – Desenhar e conceber o Modelo de Base de Dados (Modelo ER (Entidade Relacionamento));
- Tarefa 3 – Construção do *template* para a aplicação;
- Tarefa 4 – Implementação da solução final e testes à aplicação;
- Tarefa 5 – Relatório final.

Por ter várias unidades curriculares em atraso foram necessárias algumas interrupções no projeto, o que levou a que o planeamento inicialmente previsto não fosse cumprido. No Mapa de *Gantt* (Anexo 8.1) mostra-se o tempo real da realização do projeto.

Seguidamente, é descrita a análise dos requisitos para que a plataforma *web* cumpra os objetivos propostos inicialmente.

4. Análise de Requisitos

Os requisitos de um sistema são a informação que descreve o procedimento que é esperado antes de ser desenhado, implementado e testado. Com base nos objetivos propostos pelo cliente é realizada uma análise que define quais os requisitos do sistema a desenvolver.

Neste caso, foram estudados os requisitos do projeto para se obter uma análise pormenorizada do iria ser implementado. Utilizaram-se práticas da metodologia descrita anteriormente (RUP – Capítulo 3.1), como por exemplo, diagramas e símbolos gráficos de modelação descritos na linguagem UML [5].

Nas secções abaixo são descritas as práticas necessárias para implementar a plataforma *web*. Pretende-se que na versão final os requisitos implementados cumpram os objetivos propostos inicialmente.

4.1. Objetivos previstos

Após o estudo dos requisitos funcionais, foram definidos os seguintes objetivos:

- Gerir professores/utilizadores da plataforma web (Só o administrador da plataforma é que terá acesso a esta página)
 - Criar, editar e consultar os dados do professor;
- Gerir utentes (idosos)
 - Criar, editar e consultar os dados dos utentes;
- Gerir Treinos (aulas)
 - Criar e editar novas sessões de aulas;
 - Criar e editar aulas dadas por semana;
 - Frequência por parte do utente (idosos) em aulas.
- Gerir Testes de Aptidão Física
 - Criar, editar e consultar os testes de aptidão física;
 - Criar, editar e consultar os parâmetros dos testes de aptidão física;
 - Criar, editar e consultar os testes de aptidão física realizados pelo utente.

- Gráficos para análise de dados, a longo prazo e a curto prazo.

4.2. Diagrama de Contexto

O diagrama de contexto é composto por fluxos de dados entre o sistema e as entidades externas (utilizadores) permitindo assim, identificar os vários módulos a serem realizados, os relacionamentos e elementos externos à plataforma *web*.

A Figura 5 que se segue, ilustra o diagrama de contexto que traduz a solução para o problema apresentado. Neste caso como entidades externas temos o professor e o administrador do sistema.

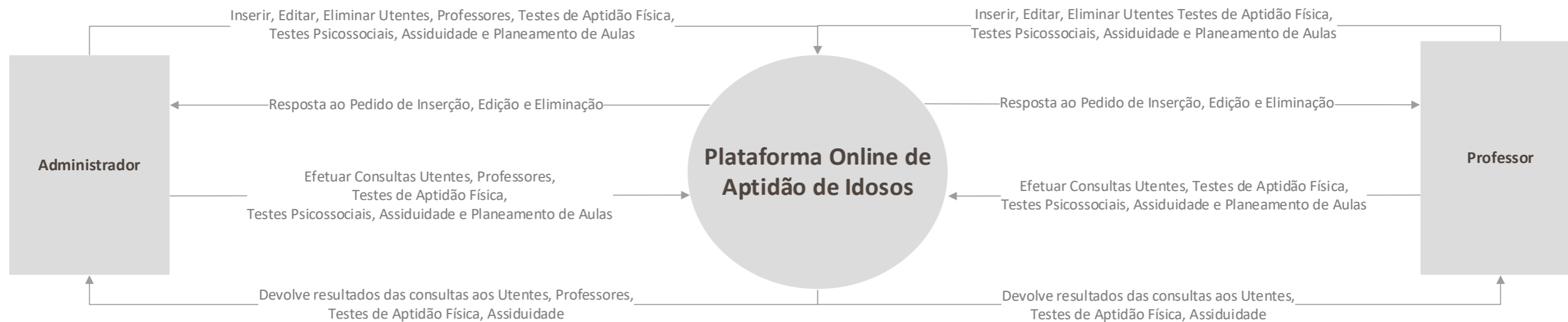


Figura 5 – Diagrama de Contexto da Plataforma

4.3. Atores e respetivos casos de uso

A Tabela 1 tem como objetivo definir os atores e os casos de uso em que participam (de acordo com o diagrama de contexto). Os casos de uso apresentados definiram a maioria dos requisitos da plataforma *web*.

Tabela 1 – Casos de Uso

Atores	Objetivos – Casos de Uso
Administrador	<ul style="list-style-type: none"> • Inserir, editar utentes da aplicação • Inserir, editar professores da aplicação • Inserir, editar testes de aptidão física; • Inserir, editar Assiduidade • Inserir, editar Planeamento de Avaliações
Professor	<ul style="list-style-type: none"> • Inserir, editar utentes da aplicação • Inserir, editar testes de aptidão física; • Inserir, editar Assiduidade • Inserir, editar Planeamento de Avaliações

4.4. Diagrama de Casos de uso

Um diagrama de casos de uso permite ver, de uma forma prática, todas as funcionalidades da plataforma *web*, assim como todas interações que o ator vai ter com a mesma. Além disso, este resulta da junção do diagrama de contexto e da Tabela 1.

A Figura 6 mostra o resultado final do diagrama de contexto, onde é possível ver a fronteira que delimita os casos de uso que vão ser desenvolvidos e as responsabilidades de cada ator do sistema.

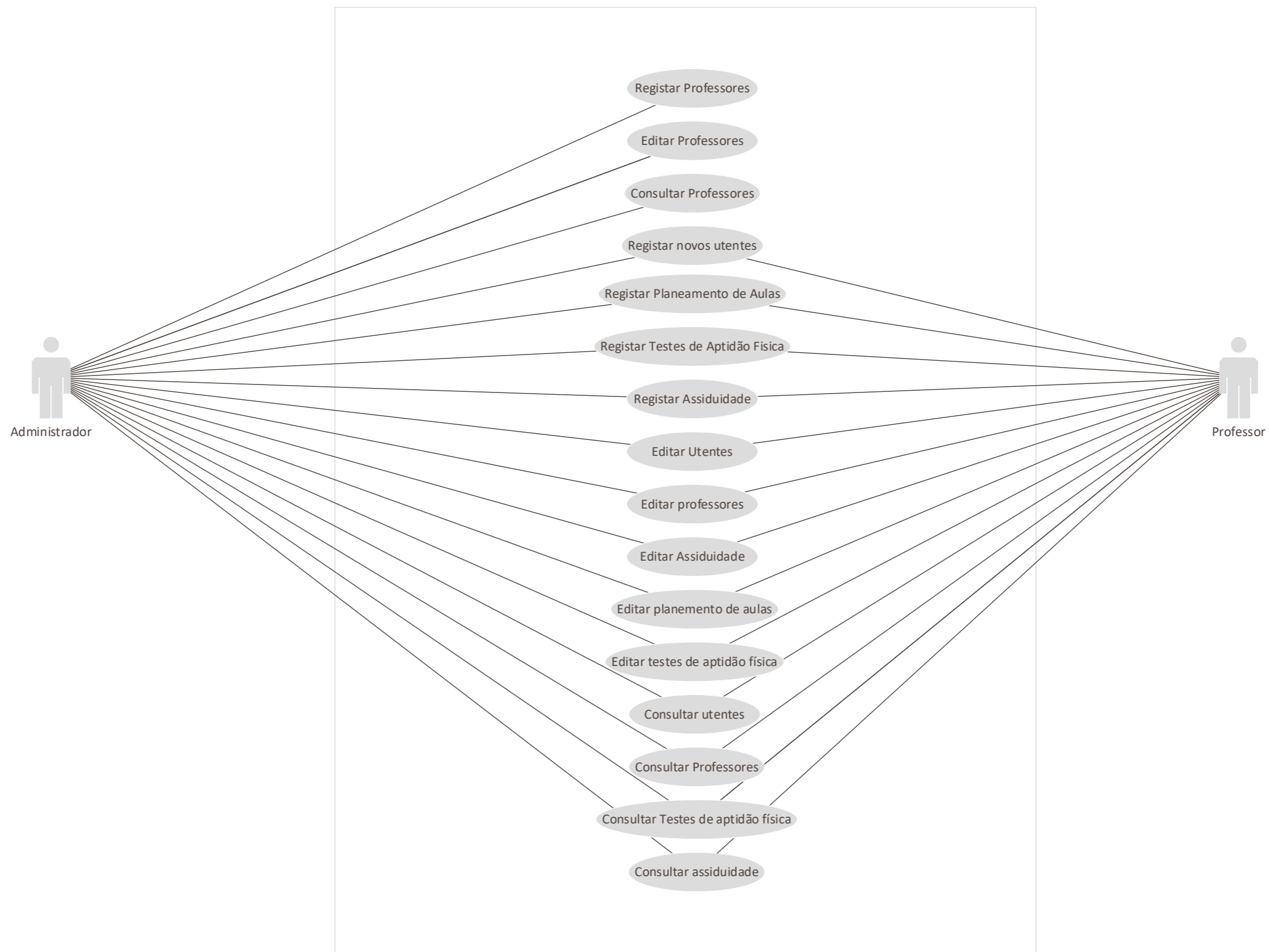


Figura 6 – Diagrama de casos de uso

4.5. Descrição dos casos de uso e Diagramas de Sequência

Na descrição de um caso de uso pressupõe-se que estão reunidas todas as condições para que tudo corra bem (caminho principal). Porém pode ser necessário fazer uma descrição de situações alternativas, ou seja, caminhos alternativos.

Na descrição sequencial (Diagramas de Sequência) são apresentadas as trocas de mensagens entre vários objetos. Estes são utilizados para representar os casos de uso com o objetivo de moldar o fluxo das mensagens, eventos e ações entre objetos e componentes.

Em virtude de definir a sequência de eventos da plataforma *web*, serão apresentadas as descrições dos casos de uso e os diagramas de sequência referentes objetivos propostos inicialmente.

4.5.1. Registrar novos utentes

O título demonstra, claramente, o objetivo deste caso de uso. A primeira das interfaces que vamos ter contato é “Registrar um novo utente” e através desta, podemos ver toda a plataforma *web* a estender-se pelas várias interfaces. Quando o utilizador interage com o sistema vai desencadear uma sequência de eventos que permite desenvolver este caso de uso.

A Tabela 2 permite ver em detalhe como se desenvolveu o caso de uso.

Tabela 2 – Descrição do caso de uso “Registrar novos utentes”.

Nome	Registrar novos utentes
Descrição	Este caso de uso tem como objetivo principal adicionar um utente.
Pré-Condição	<i>Login</i> de um ator válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator escolhe a opção “Adicionar Utente”. 2. O sistema mostra o formulário para adicionar um novo utente. 3. O ator introduz os dados (Nome, Morada, Código-Postal, Contacto, NIF (Número de Identificação Fiscal), E-mail, entre outros) necessários para registo de um novo utente.

	4. O sistema guarda o utente e mostra os dados que foram inseridos.
Caminhos Alternativos	4. a) O utente não foi inserido porque os campos não estão preenchidos.
Suplementos ou Adornos	Fazer um teste onde se verifique: Se insere o cliente sem que os campos obrigatórios (todos) estejam preenchidos;

Para complementar a descrição de caso de uso “Registar novos utentes” (Tabela 2), é apresentado o respetivo diagrama de sequência (Figura 7) que apresenta o cenário principal do caso de uso, onde o utilizador do sistema adiciona um novo utente.

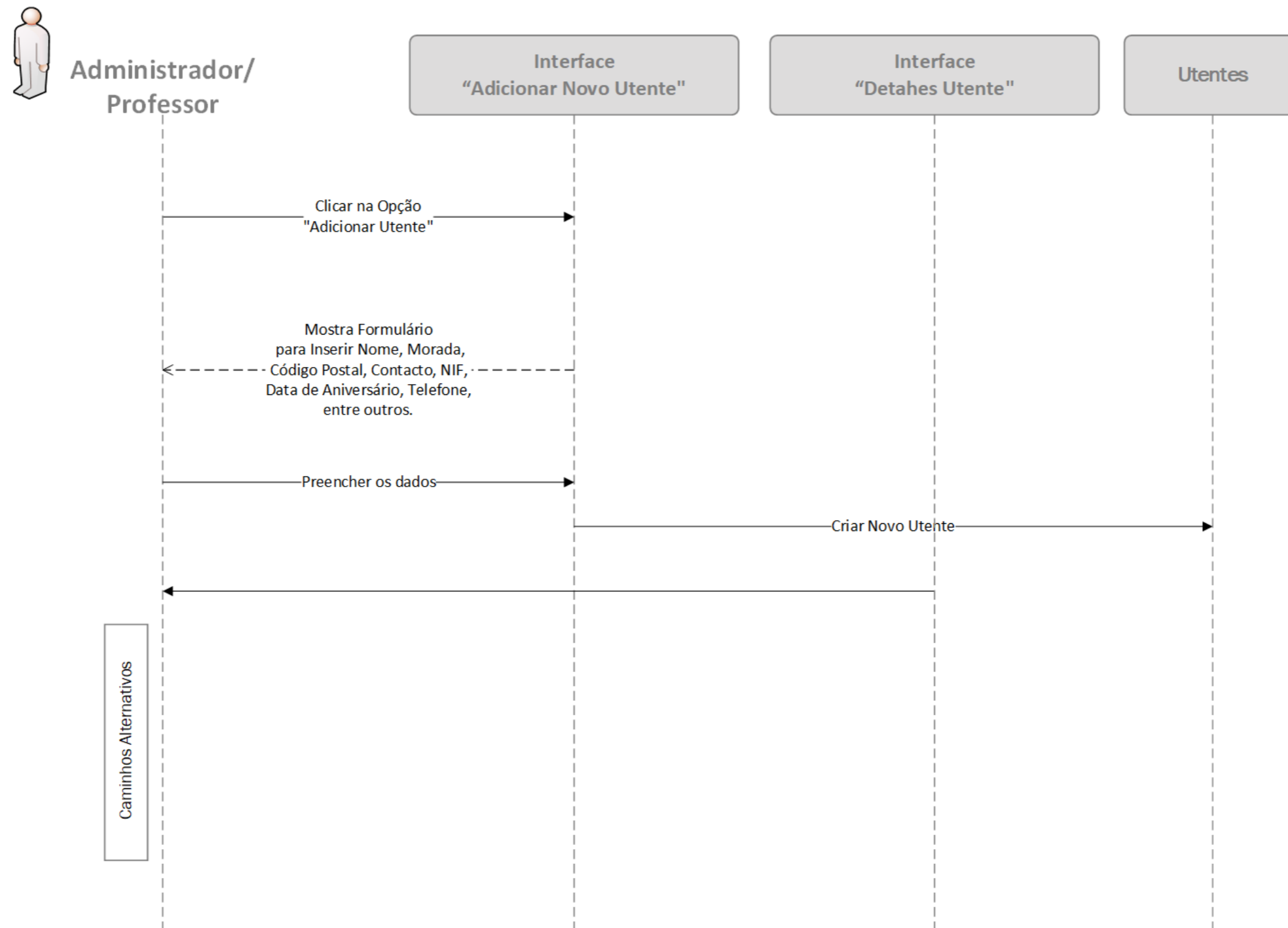


Figura 7 – Diagrama de Sequência “Registrar novo utentes”

4.5.2. Registrar Testes de Aptidão Física

A descrição deste caso de uso vai permitir que a plataforma *web* esteja quase operacional. Trata-se de uma interface que permite ver todos os testes de aptidão física e ao mesmo tempo adicionar mais testes, caso seja necessário.

A interface será composta por uma tabela de testes que já são existentes em base de dados e o caso de uso, é acionado caso o utilizador precise de adicionar mais testes de aptidão física.

A Tabela 3 mostra o processo que o utilizador produz para adicionar um novo teste.

Tabela 3 - Descrição do caso de uso “Registrar Testes de Aptidão Física”.

Nome	Registrar Testes de Aptidão Física
Descrição	Este caso de uso tem como objetivo principal registrar um novo teste de aptidão física.
Pré-Condição	<i>Login</i> de um ator válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator escolhe a opção “Testes Pretendidos”. 2. O sistema mostra uma tabela com os testes já adicionados. 3. O ator clica na opção “Adicionar novo Teste” 4. O ator introduz os dados (Nome do teste, Descrição do teste) para registo de um novo teste de aptidão física. 5. O sistema guarda o novo teste e atualiza a tabela com o novo teste.
Caminhos Alternativos	<ol style="list-style-type: none"> 4. a) O ator esquece-se de inserir o Nome do teste um dos campos.
Suplementos ou Adornos	Fazer um teste onde se verifique: Se o novo teste de aptidão física foi inserido com sucesso.

Na Figura 8 é apresentado o cenário principal do caso de uso, onde o utilizador do sistema adiciona um novo exercício para avaliação da aptidão física do utente.

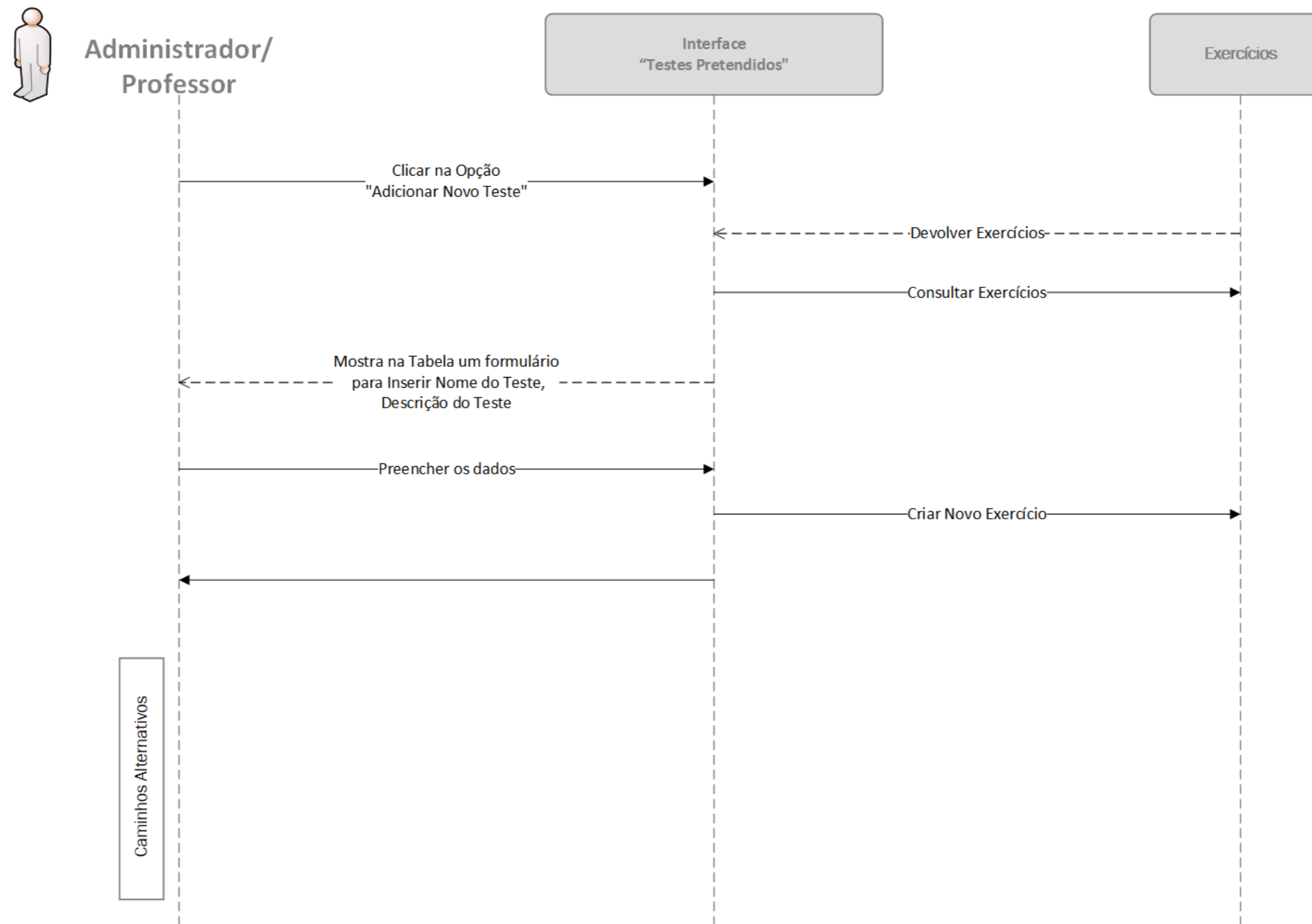


Figura 8 – Diagrama de Sequência “Registrar Testes de Aptidão Física”

4.5.3. Registrar Parâmetros de Testes de Aptidão Física

Depois de adicionados todos os testes para avaliar o utente, houve a necessidade de criar uma interface em que o ator, pudesse adicionar parâmetros de avaliação aos testes de aptidão física. Este caso de uso só deverá ser acionado caso o ator precise de adicionar parâmetros aos testes de aptidão física.

A Tabela 4 mostra o processo que o ator faz para adicionar um novo parâmetro de avaliação a um teste.

Tabela 4 – Descrição do caso de uso “Registrar Parâmetros dos Testes”

Nome	Registrar Parâmetros de Testes de Aptidão Física
Descrição	Este caso de uso tem como objetivo principal registrar um novo parâmetro de um teste de aptidão física.
Pré-Condição	Login de um ator válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator escolhe a opção “Parâmetros do Teste”. 2. O sistema mostra uma tabela com os testes já adicionados. 3. O ator clica na opção “Adicionar novo Parâmetro de um teste” 4. O ator escolhe o Nome do exercício que pretende adicionar e preenche a o nome do parâmetro. 5. O sistema guarda o novo parâmetro do teste e atualiza a tabela.
Caminhos Alternativos	<ol style="list-style-type: none"> 4. a) O ator esquece-se de inserir o nome do parâmetro.
Suplementos ou Adornos	Fazer um teste onde se verifique: Se o nome do parâmetro é adicionado.

Na Figura 9 é apresentado o cenário principal do caso de uso, onde o utilizador do sistema adiciona um novo parâmetro de um teste de aptidão física.

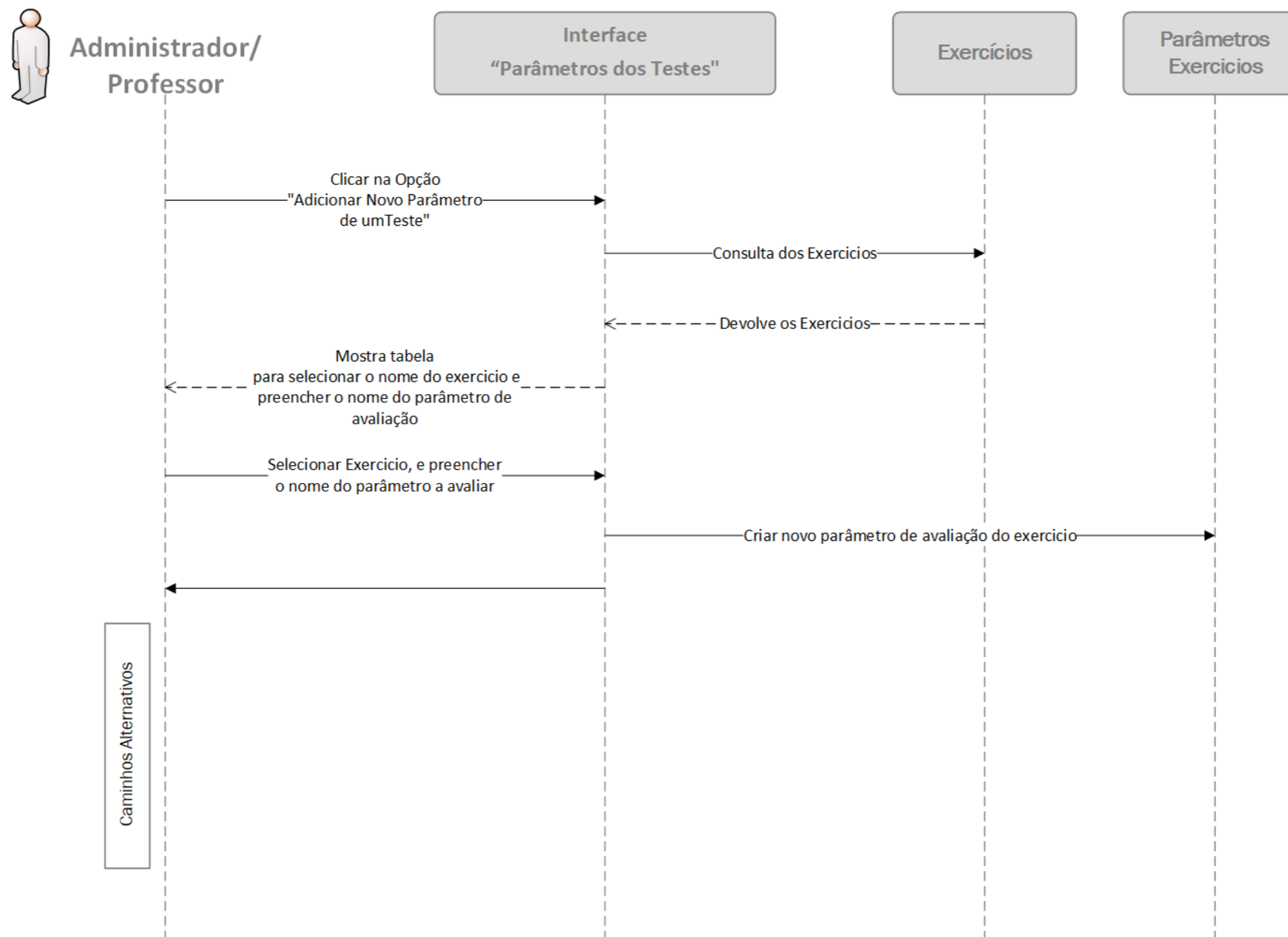


Figura 9 – Diagrama de Sequência "Registrar Parâmetros dos testes"

4.5.4. Registrar Assiduidade do utente

Na descrição deste caso de uso é realizado o preenchimento da presença dos utentes em aula ou não. A interface possuirá uma lista de utentes em que depois é só seleccionar se aquele utente estará presente ou não.

A Tabela 5 mostra como o utilizador processa o registo da assiduidade de um utente.

Tabela 5 - Descrição do caso de uso "Registrar Assiduidade do utente".

Nome	Registrar Assiduidade do utente
Descrição	Este caso de uso tem como objetivo principal registar a assiduidade de um utente em aula.
Pré-Condição	<i>Login</i> de um ator válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator escolhe a opção "Assiduidade". 2. O sistema mostra uma tabela com os utentes que existem. 3. O ator preenche os dados (Data, presença, dados pré-treino e dados pós-treino). 4. O ator clica na opção "Inserir Presença" 5. O sistema guarda a assiduidade do utente naquele dia.
Caminhos Alternativos	<ol style="list-style-type: none"> 3. a) O ator esquece-se de inserir a data.
Suplementos ou Adornos	Fazer um teste onde se verifique: Se foi inserido a assiduidade do utente no dia 28/06/2017.

Na Figura 10 é apresentado o cenário principal do caso de uso, onde o utilizador do sistema adiciona a assiduidade do utente no dia pretendido.

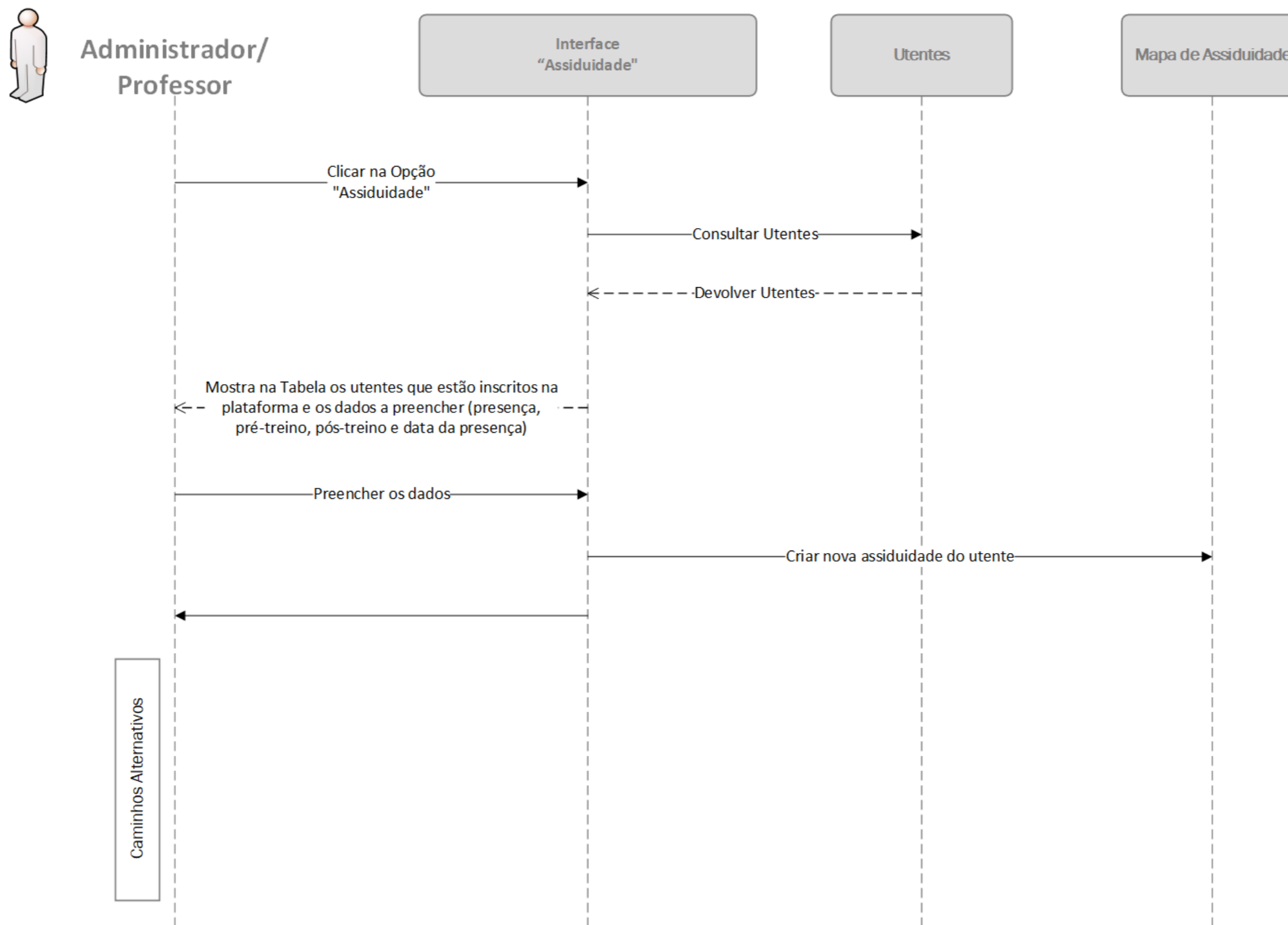


Figura 10 – Diagrama de sequencia “Registrar assiduidade do utente”

4.5.5. Registrar Planeamento de Aulas

Este caso de uso, pretende criar a interface “Planeamento de Aulas”, ou melhor, trata-se de uma interface capaz de planear aulas para uma melhor gestão de horário. Desta forma, os técnicos conseguirão organizar melhor as respetivas aulas a serem dadas.

A Tabela 6 mostra como podemos registar um novo planeamento e depois visualizá-lo.

Tabela 6 – Descrição do caso de uso “Registrar Planeamento de Aulas”.

Nome	Registrar Planeamento de Aulas
Descrição	Este caso de uso tem como objetivo principal registar um planeamento de aulas.
Pré-Condição	Login de um ator válido.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator escolhe a opção “Planear Aulas”. 2. O sistema mostra numa tabela quais os planeamentos de aulas que existem. 3. O ator clica na opção “Adicionar novo Planeamento”. 4. O ator preenche os dados (Exercício, nome do planeamento, dia da semana, objetivos e observações). 5. O ator clica na opção “Inserir”. 6. O sistema guarda o novo planeamento de aula.
Suplementos ou Adornos	Fazer um teste onde se verifique: Se foi inserido a assiduidade do utente no dia 28/06/2017.

Na Figura 11 é apresentado o cenário principal do caso de uso, onde o utilizador do sistema insere um novo planeamento para aulas.

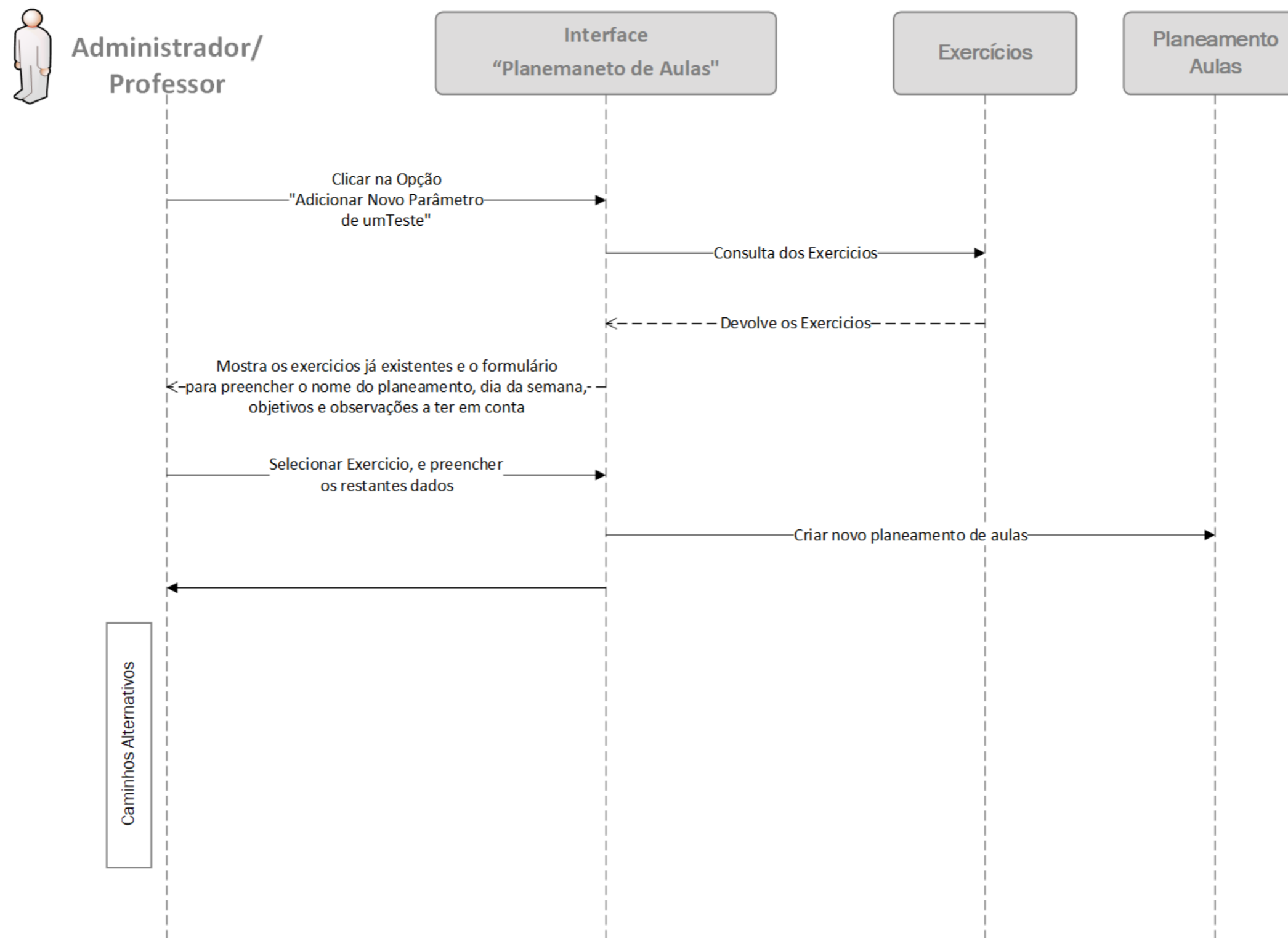


Figura 11 – Diagrama de sequência “Registrar Planeamento de aulas”

4.6. Diagrama de Classes

Nesta secção poderemos ver o diagrama de classes da plataforma, representado pela Figura 12. Este diagrama mostra como as diferentes classes se relacionam entre si, agrupando especialidades e comportamentos, onde cada classe é constituída pelo nome (o que representa a classe no mundo real), atributos (informação que deve ser analisada e/ou armazenada) e por fim as operações que representa o papel dos atores no sistema. Sendo ainda algumas classes existentes na plataforma *web*, serão descritas só as mais pertinentes para o sistema.

Uma das classes mais importante para que a plataforma *web* fosse viável é a classe “Utentes”. Esta junta todos os dados essenciais do utente e a partir dela, conseguiremos obter dados fulcrais para uma pré-análise e pós-análise da saúde do utente. O diagrama na página seguinte (Figura 12) mostra como esta classe faz a ligação a outras classes como por exemplo treinadores, exercícios, assiduidade, questionários de saúde, entre outros.

A classe “Exercicios”, é essencial para o sistema funcionar perante os objetivos propostos. Através desta, é feita a ligação a outras classes, tais como “ParametrosExercicios” e “ExercicioRealizado”. As três classes em conjunto, são fulcrais para a análise da aptidão física do utente, pois guardam os dados necessários para posteriormente ser feito um relatório da saúde do utente. Também ligada a classe “Exercicios”, está a classe “PlaneamentoAulas” onde é realizada a gestão da aula, ou seja, planear para aquele dia da semana, qual o exercício que será feito, quais os objetivos que são pretendidos para a aula e quais as observações a terem em conta quando a aula for dada.

A classe “Assiduidade” servirá de suporte ao utilizador para este quantificar a presença do utente em aula. Esta é de extrema importância para a posterior análise da saúde do utente, isto é, se o utente está presente em todas ou na maioria das aulas que existiram ao longo do tempo. Através deste registo vai ser permitido realizar os testes de aptidão física ou não. O utente que acaba de se registar no programa +65 não faz parte desta classificação e vai fazer os testes de aptidão quando houver disponibilidade.

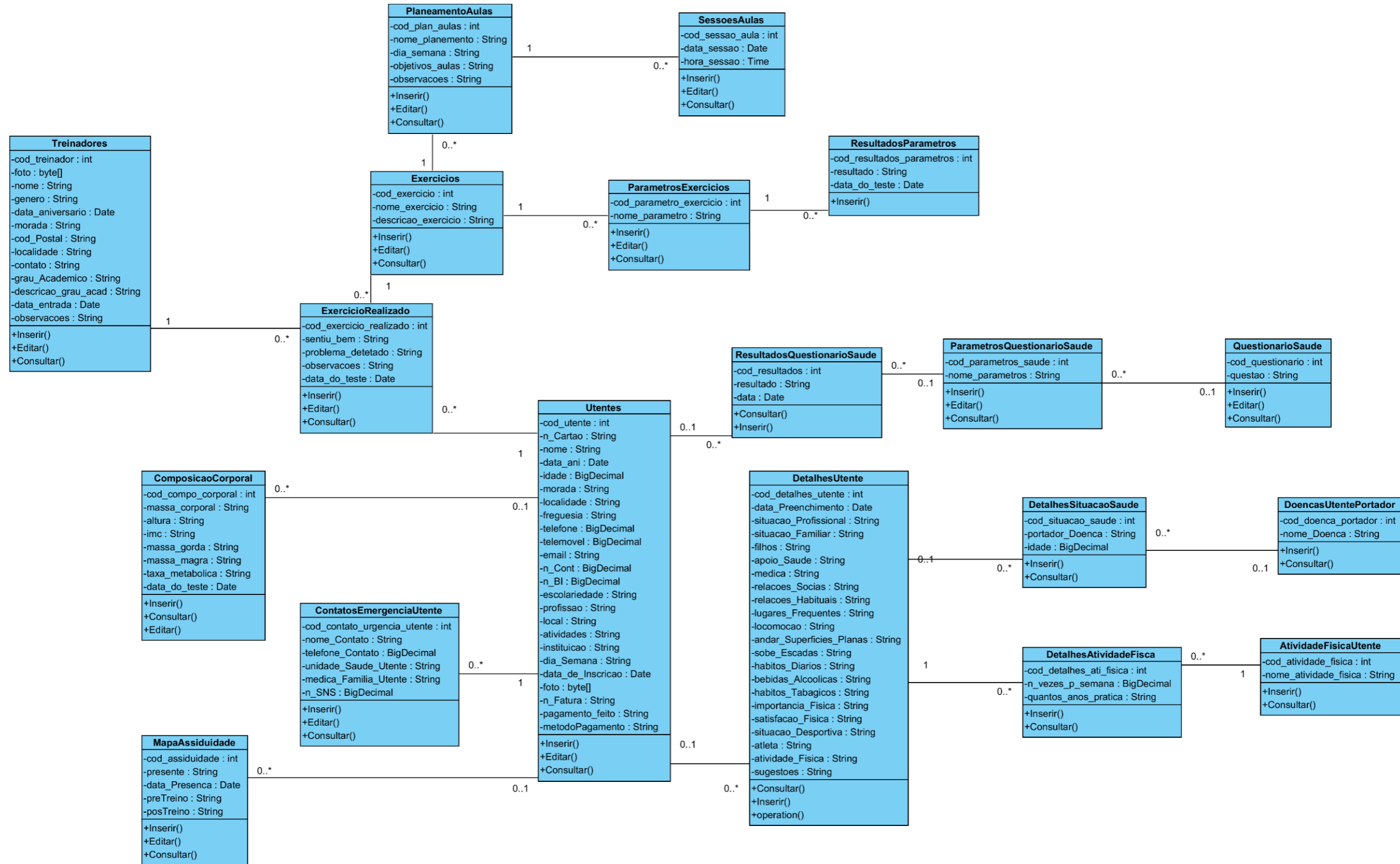


Figura 12 – Diagrama de classes da plataforma

A implementação deste diagrama de classes no modelo relacional originou o EER (*Enhanced Entity-Relationship*) apresentado na Figura 13.

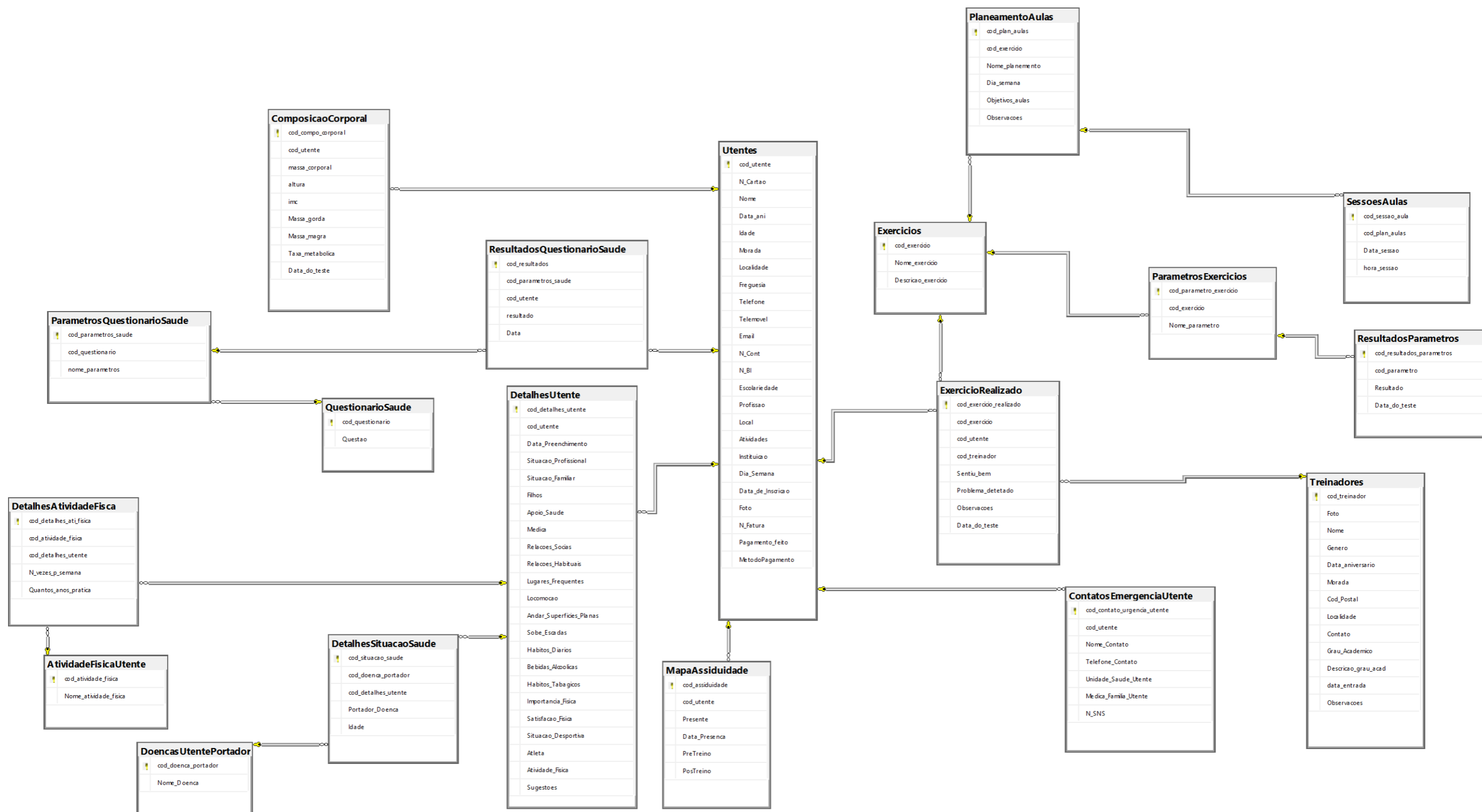


Figura 13 - Modelo ER da plataforma

No capítulo seguinte, são apresentadas as tecnologias que foram utilizadas para desenvolver a plataforma *web*.

5. Implementação da Solução

Com base no estudo elaborado na análise de requisitos (Capítulo 4), procedeu-se ao desenvolvimento da plataforma *web*.

Para realização da plataforma *web* foram seleccionadas as tecnologias que melhor se adequavam ao contexto do projeto. Não foi necessária uma escolha demorada em relação a algumas tecnologias a utilizar, pois já tinham sido utilizadas em unidades curriculares da licenciatura, tais como, ASP.NET, C#, HTML (*HyperText Markup Language*), entre outras. Como principal dúvida na escolha, esteve o *AJAX Control Toolkit*, pois trata-se de uma tecnologia diferente que não tinha conhecimento e a informação é demasiada e na maior parte das vezes confusa.

Durante a implementação da plataforma, foi necessário fazer alguns ajustes aos requisitos definidos, para que a funcionalidades ficassem utilizáveis. No entanto, tentou-se sempre procurar a forma mais eficaz, fácil e rápida de realizar todos os objetivos ajustados com as docentes envolvidas no projeto.

5.1. Tecnologias Utilizadas

Atualmente, é difícil escolher as tecnologias certas para implementação de uma plataforma *web*. Os benefícios que cada tecnologia inclui, são basicamente os mesmos de umas ferramentas para as outras, dificultando assim a escolha que o programador tem que fazer.

Assim, segue-se uma breve explicação sobre cada uma das tecnologias utilizadas e detalhando o porquê de serem necessárias para o desenvolvimento do projeto.

5.1.1. HTML

HTML é a linguagem com que se escrevem as páginas *web*. As páginas *web* podem ser vistas pelo utilizador mediante navegador. Podemos dizer, portanto, que o HTML é a linguagem usada pelos navegadores para mostrar as páginas *web* ao utilizador, sendo hoje em dia a interface mais extensa na rede [6].

Esta linguagem tornou-se no padrão e base para criação de qualquer página *web*. Como se decidiu desenvolver a solução numa página *web*, esta linguagem foi utilizada em todas as páginas *web* desenvolvidas.

5.1.2. CSS

O CSS (*Cascading Style Sheets*) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como HTML ou XML (*eXtensible Markup Language*).

O seu principal benefício é fornecer a separação entre o formato e o conteúdo de um documento. O CSS permite criar e alterar conjuntos de propriedades de estilo. Estas propriedades alteram o *design* e a visualização de qualquer elemento HTML, seguindo um conjunto de regras impostas pelo programador. Assim, com esta linguagem, conseguem-se criar *layouts* e *designs* específicos de modo eficiente. [7]

Com base na pesquisa, decidiu-se utilizar um *framework* chamado *bootstrap* [8]. A *framework* está pré configurado para alterar os elementos básicos de HTML não só para *designs* mais complexos, mas também para tornar as páginas *web* responsivas. Assim, a página consegue ser utilizada e visualizada de modo intuitivo e eficiente quer nos pequenos ecrãs dos telemóveis quer nos maiores ecrãs dos computadores. [9]

A utilização desta *framework* facilitou todo o processo de *design* contruído para a plataforma *web*. Por esta razão, foi devoto mais tempo a outras tecnologias que necessitaram mais dedicação.

5.1.3. ASP.NET

ASP.NET é a plataforma da *Microsoft* para o desenvolvimento de aplicações *Web* e é o sucessor da tecnologia ASP (*Active Server Pages*). Permite através de uma linguagem de programação integrada na *.NET Framework*, criar páginas dinâmicas [10]. Há outras linguagens de programação para a *web*, tais como PHP (*Hypertext Preprocessor*). No entanto, decidiu-se utilizar o ASP.NET pelas seguintes razões:

- Disciplina dedicada à programação na internet e graças a esta conseguiu-se perceber o funcionamento da tecnologia e, aprender um pouco da mesma.
- A enorme quantidade de informação sobre como trabalhar e utilizar a tecnologia.

Esta tecnologia vai ser utilizada em todas as páginas *web* a desenvolver.

5.1.4. JavaScript

JavaScript é uma linguagem de programação utilizada para criar pequenos programas encarregados de realizar ações dentro do âmbito de uma página *web*. Com *JavaScript* podemos criar efeitos especiais nas páginas e definir interatividades com o utilizador. O *browser* do cliente é o encarregado de interpretar as instruções *JavaScript* e executá-las para realizar estes efeitos e interatividades, de modo que o maior recurso, e talvez o único, com que conta esta linguagem é o próprio *browser*. [11]

A utilização do *JavaScript* na plataforma *web* em várias páginas vai permitir que o utilizador interaja mais facilmente com o sistema.

5.1.5. C#

O C# é uma linguagem de programação orientada a objetos criada pela *Microsoft*. Faz parte da plataforma .NET e é originalmente baseada em C++ e JAVA [12]. A linguagem C# foi criada em conjunto com a arquitetura .NET. Embora, como já foi dito, existam outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#), o C# é considerado a linguagem símbolo do .NET porque foi criada praticamente do zero para funcionar na plataforma .NET e a maior parte das classes do .NET *Framework* foram desenvolvidas em C# [13]. Esta tecnologia vai ser utilizada em todas as páginas *web* a desenvolver.

5.1.6. Microsoft SQL Server

O *Microsoft SQL Server* é um SGBD (Sistema de Gestão de Base de Dados) criado pela *Microsoft*. Este é o SGBD usado no projeto principalmente por já vir incluído no *Microsoft Visual Studio* [14].

5.1.7. T-SQL

Transact-SQL é a linguagem primária de acesso a dados utilizada por uma aplicação para manipular dados e objetos em um banco de dados *SQL Server*. Toda aplicação comunica com o *SQL Server*, enviando instruções *Transact-SQL* para o servidor [15].

5.1.8. AJAX Control Toolkit

AJAX é o acrónimo de *Asynchronous Javascript And XML* e podemos definir como um recurso que utiliza algumas tecnologias existentes com o objetivo de promover interatividade e o dinamismo para aplicações *web*. O AJAX não é uma tecnologia, mas um conjunto de tecnologias conhecidas a trabalhar em conjunto, tornando as aplicações finais mais ricas [16].

O *toolkit* vem acrescentar novos controlos neste caso ao *Visual Studio* que foi o IDE (*Integrated Development Environment*) usado na implementação da solução. Os controlos são extensões ao ASP.NET. Quer isto dizer que os controlos usados são os controlos por omissão, mas com a particularidade de usar mais qualquer coisa (extensões AJAX). Entre os mais de 40 novos controlos que o *toolkit* tem, apenas um foi utilizado na solução. O *Calendar Extender*, esta extensão fornece do lado do cliente, um painel *pop-up* com um calendário onde o cliente pode escolher a data.

A documentação disponibilizada pelos criadores [17] contém instruções gráficas de como adicionar os controlos ao *Visual Studio* e também tutoriais para cada um dos controlos do *toolkit*.

Para que se possa usar os controlos AJAX a página *web* terá de ter um *ScriptManager*. O *ScriptManager* está dividido em duas fases. A primeira fase verifica se

a aplicação suporta os conteúdos ASP.NET AJAX e configura-se o ambiente para que se possa usar os mesmos conteúdos. Na segunda fase, o *ScriptManager* constrói uma comunicação assíncrona com o código em execução no cliente para que o *script* possa executar as atualizações necessárias na página.

No caso da solução desenvolvida, o *ScriptManager* foi inserido na *Master Page* mais precisamente no *Form* da página, de modo a que o mesmo tivesse disponível nas outras páginas que descendem desta.

5.2. Diagrama de Hierarquia da plataforma

Após o estudo e análise dos objetivos propostos e delineados todos os requisitos funcionais a desenvolver, é altura de esquematizar a plataforma *web* para compreender o fluxo de dados durante o seu funcionamento.

Deste modo, foi projetado um diagrama do *Website* que tem como principal objetivo mostrar aos utilizadores as funções que cada um pode executar.

A plataforma *web* terá dois tipos de utilizadores e as suas funções são:

- Administrador: que pode fazer todo o tipo de ações na plataforma;
- Professor: que pode fazer todo o processo de análise do utente.

A Figura 14 mostra o mapa da plataforma *web*, onde informa as permissões que cada utilizador tem e quais as respetivas páginas a serem realizadas.

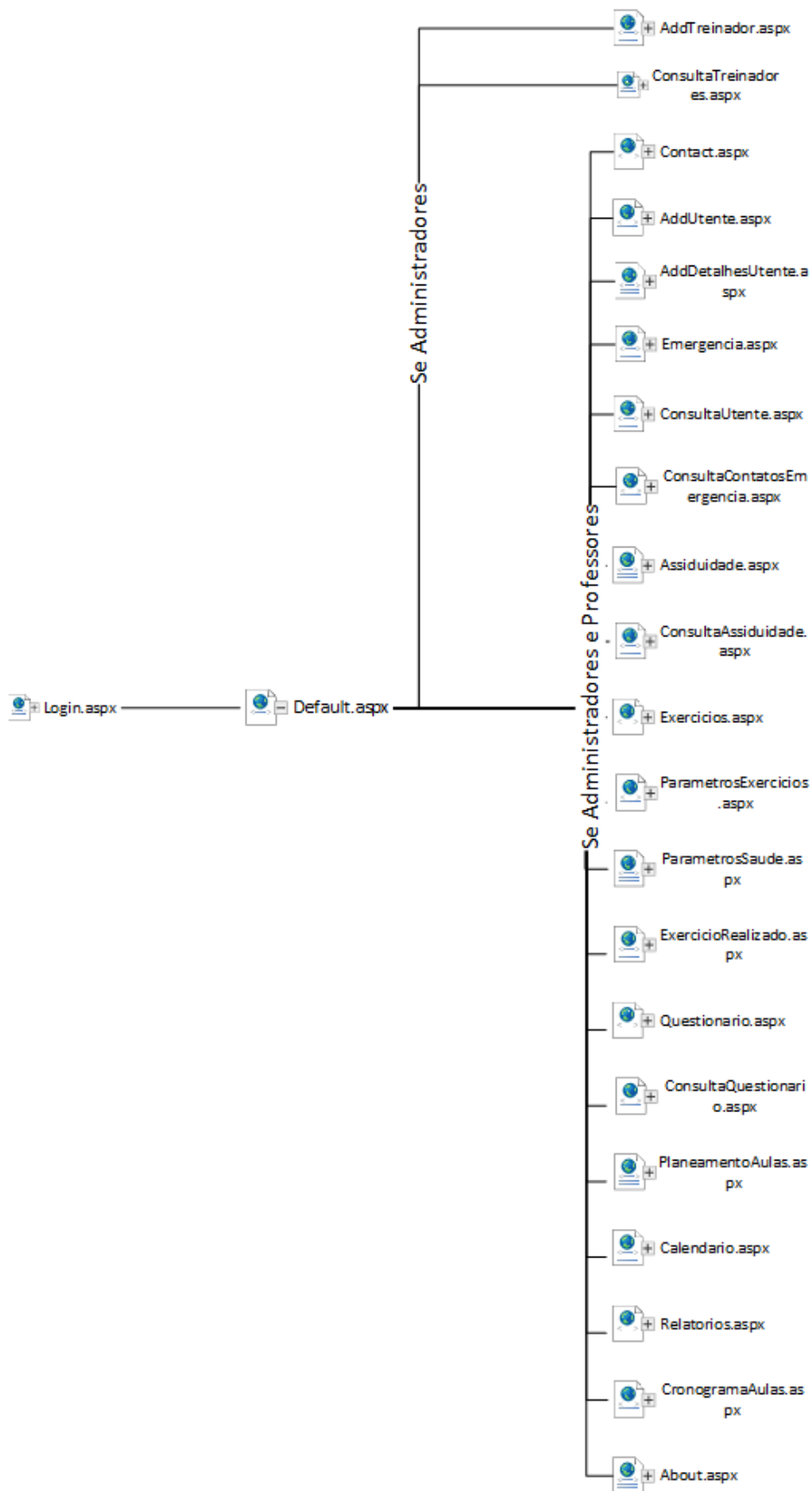


Figura 14 – Diagrama de Hierarquia da Plataforma

Para uma melhor compreensão do fluxo dados existentes na plataforma *web*, é realizada uma pequena descrição das páginas fundamentais.

1. **Login:** página para o utilizador entrar na aplicação. Tem dois campos para o utilizador colocar o *username* e *password* correspondentes. Se o utilizador fizer o *Login* com sucesso, é encaminhado para a página *Default*. A página *Login* existe como medida de segurança para negar o acesso a qualquer utilizador não autorizado.



Figura 15 – Aspeto da página Login da plataforma.

2. **Exercícios:** página que mostra os vários testes de aptidão física. Contém uma tabela onde é possível ver os vários testes de aptidão física e uma pequena descrição dos mesmos. Além disso, se o utilizador pretender adicionar mais testes de aptidão física poderá fazê-lo nesta mesma página.

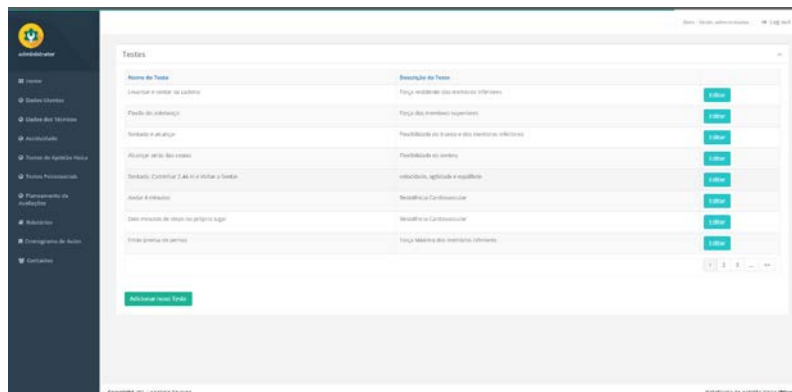


Figura 16 – Página testes de aptidão física da plataforma.

- 3. ParametrosExercicios:** página que mostra os vários parâmetros de avaliação dos testes. Contém os vários testes de aptidão física e os seus parâmetros associados. Poderão ser adicionados mais parâmetros de avaliação aos testes de aptidão física nesta mesma página.

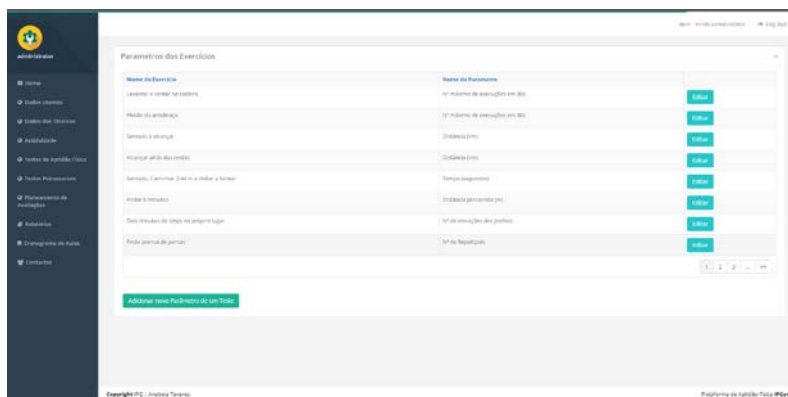


Figura 17 - Página dos parâmetros dos testes de aptidão física da plataforma.

5.3. Interfaces da Plataforma de Aptidão Física

Para a conceção das variadas interfaces, foi necessário debater com as professoras envolvidas no projeto para definir como a informação que iria ser exposta. Depois destes objetivos estarem bem assentes, foi possível prosseguir para a implementação das interfaces.

Durante a implementação, foram superados alguns problemas e desafios que foram aparecendo. De seguida, são apresentados os pontos principais destes desafios e as respetivas soluções. É de salientar que em muitas interfaces o código é semelhante e, portanto, mostra-se um exemplo de cada uma das ações (inserir, editar, pesquisar) que são realizadas nas várias páginas.

5.3.1. Interface principal da plataforma (*Default*)

Desde a fase inicial de análise de requisitos, ficou assente que a página inicial teria que ter uma interface amigável de fácil acesso à informação, como se fosse uma “capa com vários separadores”.

Deste modo, com ajuda da *framework (bootstrap)* conseguiu-se uma interface (pode ser visualizada na Figura 18) limpa, simples, intuitiva e de fácil acesso. É composta por um menu lateral onde vai dar acesso a qualquer ponto de interesse do utilizador. Na parte central da página apenas se deu ênfase ao programa e às instituições que dão vida a este projeto.

Importa salientar, que a ideia principal seria colocar os dados relevantes dos utentes nesta página como por exemplo, um gráfico com os melhores utentes a nível de saúde, outro gráfico com os utentes com a melhor assiduidade, entre outros. No entanto, numa das reuniões com as docentes envolvidas, chegou-se a conclusão que esta página poderia ficar simplesmente com as imagens do programa e das instituições, tornando assim o ambiente da plataforma menos complicado e mais bonito ao primeiro impacto.



Figura 18 – Página inicial da plataforma

5.3.2. Interface inserir Utentes

Sendo os utentes um dos pontos focais deste desafio, deu-se bastante ênfase a este módulo de maneira a que ficasse completamente pronto, ou seja, de rápido acesso, rápida inserção e visualização dos dados.

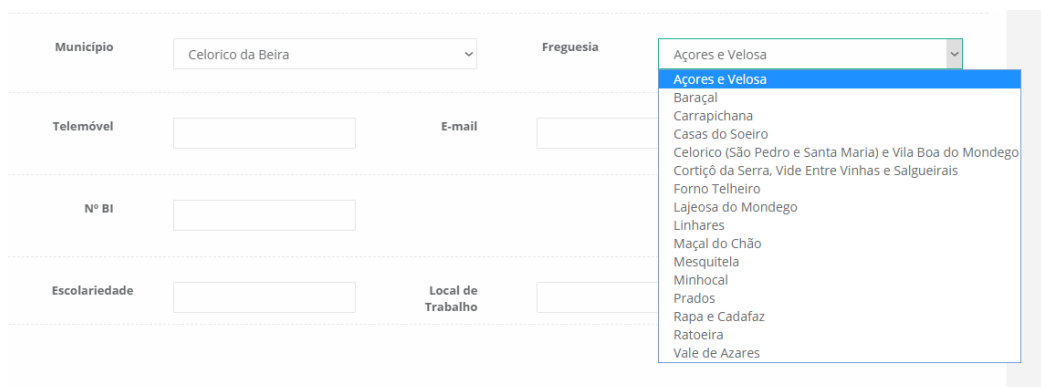
O utilizador da plataforma preenche os vários dados, sendo alguns de preenchimento automático, como por exemplo a idade (Figura 19) que para alguns utentes pode ser um dado bastante sensível de revelar (Anexo 8.3.1). Também de maneira a

simplificar o preenchimento quando o utilizador escolhe o campo município, é dado logo o conjunto de freguesias pertencente a esse município (Figura 20).



The screenshot shows a form titled "Adicionar Utente". It includes a "Foto" field with a "Choose File" button and "No file chosen" text. A "Nº do Cartão" field is empty. The "Nome" field is empty. The "Data de Aniversário" field is filled with "19/11/1945" and has a calendar icon. The "Idade" field is filled with "71".

Figura 19 – Preenchimento do campo data de aniversário.



The screenshot shows the "Município" dropdown set to "Celorico da Beira". The "Freguesia" dropdown is open, showing a list of parishes: Açores e Velosa, Baraçal, Carrapichana, Casas do Soeiro, Celorico (São Pedro e Santa Maria) e Vila Boa do Mondego, Cortiço da Serra, Vide Entre Vinhas e Salgueirais, Forno Telheiro, Lajeosa do Mondego, Linhares, Maçal do Chão, Mesquitela, Minhocal, Prados, Rapa e Cadafaz, Ratoeira, and Vale de Azares. The "Açores e Velosa" option is highlighted.

Figura 20 – Dropdownlist das freguesias

Consequentemente para melhor organização da página, decidiu-se esconder alguns dados que não são pertinentes apresentar. Neste caso, se um utente não for institucionalizado não há necessidade de haver uma caixa de texto para colocar o nome. Sendo assim, só se o utente tiver uma instituição é que aparece a caixa de texto para preencher o nome da instituição (Figura 21).

Institucionalizado Sim Não

Nome da Instituição

Figura 21 – Demonstração do funcionamento das caixas de texto “invisíveis”.

Por fim, quando se procede à inserção dos dados o utilizador poderá ver logo os dados inseridos e se necessitar editá-los. Além disso, se pretender adicionar um contato de emergência poderá fazê-lo.

5.3.3. Interface da inserção e edição de testes de aptidão física

Os testes de aptidão física são a parte principal desta aplicação, é onde se regista todos os testes que são precisos para avaliação da saúde do utente. Sendo assim, é possível dizer que são uma parte fundamental para o bom funcionamento da plataforma *web*.

Durante a conceção desta interface, foi dado pelas docentes envolvidas no projeto, um conjunto de testes de aptidão física que eram importantes ter na plataforma *web*. No entanto, no que se refere à inserção destes testes foi um desafio, eram cerca de quarenta testes de aptidão física para adicionar um a um.

Sendo um trabalho moroso, foi feita uma pesquisa de como adicionar dados em massa através de um ficheiro CSV (*Comma-Separated Values*) que iria ser constituído por um nome do teste e a descrição do teste. Através deste ficheiro CSV e um *script* em SQL (*Structured Query Language*) (Anexo 8.3.2), foi possível adicionar os dados de forma rápida e eficaz. Estes dados ficaram logo disponíveis para consulta e alterações que fossem necessárias.

No entanto, se o utilizador pretender adicionar um novo teste à plataforma terá possibilidade para tal. É dada a hipótese ao utilizador de adicionar um novo teste através do botão “Adicionar novo teste” disponível nesta página. Na Figura 22 é possível ver como é o formulário para a inserção do novo teste.

Figura 22 – Inserção de um novo teste de aptidão física

Contudo, se o utilizador pretender editar um teste de aptidão física já adicionado, poderá o fazer nesta mesma página. A edição destes testes tornou-se então uma das principais funcionalidades do sistema porque poderá ser necessário alterar o nome do teste ou mesmo a descrição do mesmo. Não foi dada importância ao eliminar pois não faria sentido suprimir um teste que já foi realizado por vários utentes e além disso, se for eliminado um teste, já não haverá uma coesão de dados para uma análise da saúde do utente.

Na Figura 23 é possível ver como se procede à edição de dois campos. Estes depois de preenchidos com os novos dados são atualizados e apresentados logo a seguir.

Testes

Nome do Teste	Descrição do Teste	
Levantar e sentar na cadeira	Força resistente dos membros inferiores	Update Cancel
Flexão do antebraço	Força dos membros superiores	Edit
Sentado e alcançar	Flexibilidade do tronco e dos membros inferiores	Edit
Alcançar atrás das costas	Flexibilidade do ombro	Edit
Sentado, Caminhar 2,44 m e Voltar a Sentar	velocidade, agilidade e equilíbrio	Edit

Figura 23 – Exemplo de edição de um teste de aptidão física

Por ser uma parte importantíssima na plataforma optou-se pela simplicidade na interface, mostrando apenas o nome e a descrição do teste numa tabela. Assim sendo, permite ao utilizador uma melhor leitura dos dados disponíveis.

5.3.4. Interface consultar assiduidade dos utentes

Era pertinente saber da assiduidade que os utentes têm em aula. Pretendia-se saber se os utentes iam as aulas frequentemente.

Deste modo, pensou-se numa interface onde seria possível ver através de uma tabela se os utentes estiveram presentes ou não. Para uma melhor leitura destes dados, existe a possibilidade de ordenar pelos vários campos existentes e também pesquisar por um utente em específico. Com as informações da assiduidade dos utentes atualizada, será possível posteriormente analisar se o utente poderá fazer os testes de aptidão física ou não.

Caso o utilizador pretenda imprimir estes dados é possível (Anexo 8.3.3), tanto individualmente, como a tabela com todos os dados. Individualmente basta o utilizador fazer *check* e carregar no botão da opção pretendida, isto é, PDF (*Portable Document Format*), *Microsoft Word*, entre outros. Já se for a lista toda, basta carregar nas várias opções disponíveis para impressão e fica pronto.

Na Figura 24 é possível verificar todas as funcionalidades descritas anteriormente. Contudo, foi necessário adicionar a funcionalidade editar esta lista, assim também será possível editar estes dados caso o utilizador assim o pretenda.

Consultar Assiduidade

Search:

Word Excel PDF Imprimir Imprimir Seleção

Nome	Presente	Data	Pré-Treino	Pós-Treino	Selecionar Imprimir	
Dona Maria	Faltou	01/01/1900			<input type="checkbox"/>	Editar
Dona Maria	Faltou	01/01/1900			<input type="checkbox"/>	Editar
Dona Maria	Faltou	01/01/1900			<input type="checkbox"/>	Editar
Dona Maria	Faltou	07/03/2017			<input type="checkbox"/>	Editar
Dona Maria	Presente	28/06/2017	Pressão sistólica: 46,7	Pressão sistólica: 48,7	<input type="checkbox"/>	Editar
Dona Maria	Presente	26/07/2017	26/07/2017	26/07/2017	<input type="checkbox"/>	Editar
Dona Maria	Presente	26/07/2017	26/07/2017	26/07/2017	<input type="checkbox"/>	Editar
Dona Maria	Faltou	26/07/2017	26/07/2017	26/07/2017	<input type="checkbox"/>	Editar

1 2

Figura 24 – Lista de utentes e sua presença

5.3.5. Calendário das sessões de aulas

Com a criação de variadas aulas nos dias da semana, tornou-se difícil a leitura destas numa simples tabela. Mesmo que os dados fossem filtrados por dias da semana, havia dificuldade de procurar a aula que era necessária.

Entretanto, após uma das reuniões com as docentes, foi possível perceber que a melhor solução para este problema, era um calendário mensal com as várias aulas dispostas pelos dias que tinham sido escolhidos. Assim, o utilizador tinha a leitura perfeita do planeamento que iria ter para aquele mês, tal e qual como aconteceria se fosse um calendário em papel.

Sendo assim, procedeu-se à criação do calendário com recurso as funcionalidades dispostas pelo *Visual Studio*. Por um lado, apresentou-se uma interface mais simples, por outro lado, uma melhor leitura dos dados que pretendíamos. A Figura 25 mostra como ficou a interface final.

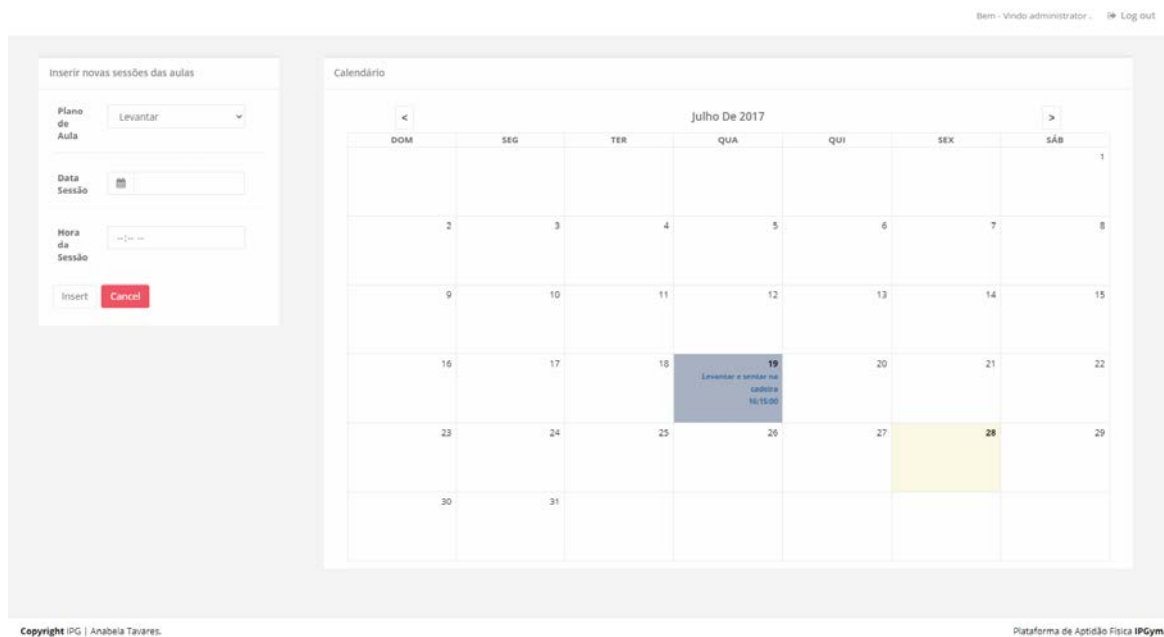


Figura 25 – Interface calendário da plataforma

Em termos de implementação, houve alguns desafios para superar, como por exemplo manter o calendário atualizado. Para que este se mantivesse atualizado e houvesse uma coesão dados na plataforma, houve necessidade de adicionar alguns pontos de referência quando é adicionada uma nova sessão de aula, isto é, sempre que é inserida uma nova sessão é realizado um *refresh* à *form* do calendário para que esta mantenha os dados atualizados.

Sem dúvida que este foi um desafio difícil, pois como se decidiu ter um calendário para melhor visualização dos dados, este teria que estar constantemente atualizado. Em Anexo (Anexo 8.3.4) poderemos ver o processo realizado para manter os dados atualizados no calendário após inserção de uma nova sessão.

No entanto, quando o utilizador acede à página pela primeira vez não há necessidade de fazer *refresh*, pois não houve uma ação na mesma. Sendo assim, só houve necessidade de “forçar” o *refresh* após inserção de dados.

De seguida, é descrita a realização dos testes à plataforma *web*.

6. Testes

Para verificar que a plataforma *web* para garantir a qualidade da aplicação, foi necessário realizar vários tipos de teste para garantir o bom funcionamento da plataforma *web*. O objetivo principal destes testes, é encontrar falhas para que possam ser corrigidas posteriormente.

Para além dos testes finais, durante toda a fase de desenvolvimento da plataforma *web* foram sendo realizados testes graduais com o intuito de validar as funcionalidades desenvolvidas. Para manter a integridade da plataforma *web*, testou-se as várias interfaces da plataforma através da descrição dos casos de uso.

Na Figura 26 é demonstrado um exemplo de um teste realizado à plataforma *web*. Neste caso, foi possível ver se dados não estivessem todos preenchidos, o utilizador não seria capaz de fazer a inserção de o novo utente na plataforma *web*. Sendo a inserção de utentes um dos pontos mais importantes na plataforma.

Nome	Anabela Tavares		
Data de Aniversário	19/11/1945	Idade	71
Morada	Rua Mártires da Pátria		
Distrito	Aveiro	Município	Selecionar Município
		Freguesia	Selecionar Freguesia
Telefone	914881128	Telemóvel	914881128
		E-mail	anabela@tares@gmail.com
NIF		N.º BI	
	Por favor preencha o N.º de Identificação Fiscal		Por favor preencha o N.º de Identificação Civil
Profissão		Escolaridade	
	Por favor preencha a Profissão		Local de Trabalho
Institucionalizado	<input type="checkbox"/> Sim <input type="checkbox"/> Não		
Atividades	<input type="checkbox"/> HIDROGINASTICA <input type="checkbox"/> GINÁSTICA <input type="checkbox"/> FORÇA <input type="checkbox"/> KARATÉ SENIOR	Dias da Semana	<input type="checkbox"/> SEGUNDA-FEIRA <input type="checkbox"/> TERÇA-FEIRA <input type="checkbox"/> QUARTA-FEIRA <input type="checkbox"/> QUINTA-FEIRA <input type="checkbox"/> SEXTA-FEIRA
	Por favor preencha as atividades pretendidas		Por favor preencha os dias

Figura 26 – Caso de teste inserir sem todos os dados preenchidos.

7. Conclusão

Ao longo deste projeto foi desenvolvida uma plataforma *web* para gerir a aptidão física dos idosos do ginásio do IPG (Instituto Politécnico da Guarda). A plataforma desenvolvida visa colmatar a necessidade que os técnicos de aptidão física tinham em organizar e documentar os resultados dos testes de cada utente do programa Guarda +65.

Inicialmente, foram traçados objetivos para a implementação da plataforma *web* com a colaboração dos docentes envolvidos no programa Guarda +65. Posteriormente foi realizada uma pesquisa onde foram encontradas várias aplicações que iam ao encontro dos objetivos anteriormente propostos. No entanto, as aplicações encontradas eram mais complexas do que se desejava, por isso, nenhuma se enquadrava nos objetivos do programa.

De seguida, foi efetuada uma análise de requisitos onde foi possível estudar as funcionalidades que a plataforma *web* deveria ter. Visto que, durante a análise de requisitos surgiram dúvidas, houve necessidade de ajustar as funcionalidades para uma melhor compreensão.

Uma vez definidos os requisitos e ajustadas as funcionalidades, começou-se a desenvolver a solução final. Durante o desenvolvimento, foram surgindo alguns problemas e desafios, onde através de pequenas reuniões com as pessoas envolvidas no projeto foram esclarecidas.

Na reta final do projeto, foram efetuados testes com dados reais e por vezes foram encontrados pequenos erros que foram corrigidos de imediato, isto porque, utilizou-se a abordagem de programar e testar logo. Esta abordagem, foi aplicada durante todo o desenvolvimento para garantir que plataforma *web* satisfizesse os requisitos inicialmente propostos.

No fim do desenvolvimento da solução, foi feita uma apresentação às pessoas envolvidas, de modo a demonstrar todas as funcionalidades da plataforma *web*, para ouvir as opiniões gerais acerca do projeto. Os objetivos principais foram concluídos e as opiniões diferidas foram positivas, tornando assim o projeto uma mais valia para o programa +65.

Como trabalho futuro, prevê-se a melhoria da plataforma *web*. A realização de gráficos dinâmicos que tornaria a plataforma ainda mais apelativa e as validações de campos em outras páginas, que fariam com que a solução ficasse mais completa. Além disso poderiam ser feitas algumas alterações na plataforma *web*, para disponibilizar os testes de aptidão física não só para os utentes do programa +65 bem como, aos frequentadores do ginásio do IPG.

Por fim e concluindo este relatório, tenho de agradecer a todos os envolvidos por toda a ajuda prestada, contribuindo assim para que a plataforma *web* atingisse os objetivos principais propostos. É de salientar que o trabalho desenvolvido pessoalmente é me satisfatório. Certamente que tudo o que foi aprendido neste projeto, será uma mais valia no meu futuro profissional.

Bibliografia

- [1] “*software* de Avaliação Física - Physical Test. | Terrazul Tecnologia.” [Online]. Available: [http://www.terrazul.com.br/site/produtos/16/software -de-Avaliação-Física/Physical-Test](http://www.terrazul.com.br/site/produtos/16/software-de-Avaliação-Física/Physical-Test). [Accessed: 27-Jun-2016].
- [2] “*GymMaster* - Gym *software* and Club Membership Management *software* .” [Online]. Available: [https://www.GymMastersoftware .com/](https://www.GymMastersoftware.com/). [Accessed: 27-Jun-2016].
- [3] J. M. Sampayo, “RUP – Four phases – iTest,” 2009. [Online]. Available: <http://www.itest.pro/rup-four-phases/>. [Accessed: 02-Jul-2016].
- [4] Rational, “Rational Unified Process. Best practices for *software* development teams,” 2001. [Online]. Available: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.PDF. [Accessed: 02-Jul-2016].
- [5] Lucid *software* Inc, “What is Unified Modeling Language | Lucidchart.” [Online]. Available: <https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language>. [Accessed: 28-Jun-2017].
- [6] W3C, “W3C HTML,” 2011. [Online]. Available: <https://www.w3.org/html/>. [Accessed: 27-Jun-2017].
- [7] H. W. Lie and B. Bos, *Cascading style sheets : designing for the Web*. Addison-Wesley, 1999.
- [8] Bootstrap, “About Bootstrap,” 2015. [Online]. Available: <http://tech.queryhome.com/51419/about-bootstrap-framework>. [Accessed: 27-Jun-2017].
- [9] Smashing Editorial, “Responsive *Web* Design: What It Is And How To Use It - SmashingMag,” 2011. [Online]. Available: <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>. [Accessed: 27-Jun-2017].

- [10] Rick Anderson; Samir Patel, “ASP.NET overview | *Microsoft Docs*,” 2010. [Online]. Available: <https://docs.microsoft.com/pt-br/aspnet/overview>. [Accessed: 27-Jun-2017].
- [11] “*JavaScript* | MDN.” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Accessed: 24-Jun-2016].
- [12] N. E. T. Framework, “C# (CSharp) o que é esta linguagem?,” pp. 2007–2009, 2009.
- [13] Suresh Dasari, “Remove Add New Row to GridView on Button Click in ASP.Net using C#, VB.NET Example - ASP.NET, C#.NET, VB.NET, JQuery, JavaScript, Gridview, SQL Server, Ajax, SSRS, XML examples.” [Online]. Available: <http://www.aspdotnet-suresh.com/2015/07/remove-add-new-row-to-gridview-on-button-click-in-aspnet-using-csharp-vbnet-example.html>. [Accessed: 17-Oct-2016].
- [14] TechTarget, “What is *Microsoft SQL Server*? - Definition from WhatIs.com,” 2017. [Online]. Available: <http://searchsqlserver.techtarget.com/definition/SQL-Server>. [Accessed: 27-Jun-2017].
- [15] “Transact-SQL Reference (Database Engine) | *Microsoft Docs*,” 2017. [Online]. Available: <https://docs.microsoft.com/pt-br/sql/t-sql/language-reference>. [Accessed: 27-Jun-2017].
- [16] w3schools.com, “AJAX Introduction,” 2007. [Online]. Available: https://www.w3schools.com/js/js_ajax_intro.asp. [Accessed: 26-Jul-2017].
- [17] “AJAX Control Toolkit | DevExpress.” [Online]. Available: https://www.devexpress.com/Products/AJAX-Control-Toolkit/?utm_source=Web&utm_medium=SiteShortcut&utm_campaign=AjaxControlToolkit&utm_content=AjaxControlToolkit_Web_SiteShortcut. [Accessed: 26-Jul-2017].

8. Anexos

8.1. Mapa de Gantt

ID	Nome da Tarefa	Início	Conclusão	Duração	jul 2016			ago 2016			set 2016			out 2016			nov 2016			dez 2016			jan 2017			fev 2017			mar 2017			abr 2017								
					3/7	10/7	17/7	24/7	31/7	7/8	14/8	21/8	28/8	4/9	11/9	18/9	25/9	2/10	9/10	16/10	23/10	30/10	6/11	13/11	20/11	27/11	4/12	11/12	18/12	25/12	1/1	8/1	15/1	22/1	29/1	5/2	12/2	19/2	26/2	5/3
1	Recolher todas as funcionalidades que o sistema deverá no final possuir	04/07/2016	12/09/2016	10,2s	█																																			
2	Desenhar e conceber o Modelo de Base de Dados (Modelo ER)	13/09/2016	30/09/2016	2,8s											█																									
3	Construção do template para a aplicação	03/10/2016	14/10/2016	2s											█																									
4	Implementação da solução final e testes à aplicação	17/10/2016	14/04/2017	26s											█																									
5	Relatório final	04/07/2016	28/07/2017	56s	█																																			

Figura 27 – Mapa de Gantt.

8.2. Páginas da Plataforma Web

Default: página inicial da aplicação. Contém um menu lateral para encaminhar o utilizador para as outras páginas.

AddUtente: página onde é possível adicionar novos utentes à plataforma. Contém um formulário com vários campos para adicionar um novo utente. Nesta página, a idade é calculada automaticamente através da data de aniversário. Além disso, alguns campos aparecem sempre que há uma alteração na checkbox pretendida, por exemplo, caso o utente seja institucionalizado aparece uma caixa de texto para adicionar o nome da instituição.

ExercicioRealizado: página onde é adicionado qual o exercício realizado pelo utente. Contém uma tabela que mostra os parâmetros que poderão ter sido realizados pelo utente. Além disso, nesta página, é possível adicionar informações adicionais que poderão ser relevantes durante a realização do exercício. Também é possível filtrar os dados pelo exercício.

Relatorios: página que mostra vários parâmetros para gerar relatórios de alguns aspetos da saúde do utente. Contém várias dropdownlists que mostram quais os dados que podem ser seleccionados e a partir de aqui, gerar vários relatórios em vários tipos de ficheiros, tais como, Microsoft Word, Microsoft Excel, entre outros.

AddTreinador: página para adicionar técnicos na plataforma. Contém um formulário que apresenta todos os campos necessários para adicionar um técnico à plataforma. Neste caso, só os Administradores da plataforma é que poderão adicionar outros técnicos.

ConsultarTreinador: página que apresenta todos os técnicos que existem na plataforma. Contém uma lista de técnicos com todos os dados necessários. Só o Administrador é que pode ter acesso a esta página. Não faria sentido os técnicos verem os dados de outros técnicos.

Contact: página que mostra os contatos dos administradores da página. Neste caso, serão só as docentes envolvidas no projeto.

AddDetalhesUtentes: página onde é possível adicionar uma análise da saúde do utente. Contém um formulário com vários campos onde poderão ser adicionados os dados da análise à saúde do utente. Além disso, também poderão ser adicionados outros tipos de dados através de *pop-ups* como por exemplo, novas doenças além das que já existem em base de dados.

Emergencia: página onde é possível adicionar os contatos de emergência do utente. Este inclui um formulário onde há vários campos para preencher os contatos em caso de emergência do utente selecionado.

ConsultaUtente: página que apresenta todos os utentes que existem na plataforma. Contém uma lista de utentes com todos os dados necessários. Estes dados poderão ser impressos em vários tipos de ficheiros, como por exemplo *Microsoft Word* e PDF.

ConsultaContatosEmergencia: página que apresenta os contatos de emergência dos utentes inscritos na plataforma. Contém uma lista com todos os dados necessários. Estes dados poderão ser igualmente impressos como em outras páginas.

Assiduidade: página para adicionar e ver a assiduidade do utente. Contém como uma *grid* onde é possível ver os utentes inscritos e adicionar informações tais como, se estão presentes ou não em “x” dia. Além disso, também é possível adicionar uma pré-análise da saúde do utente.

ConsultarAssiduidade: página que mostra a assiduidade do utente. Contém uma tabela informativa da assiduidade dos utentes. Estes dados, poderão ser editados e impressos caso seja necessário.

ParametrosSaude: página onde é possível adicionar um novo parâmetro de saúde do utente. Inclui um formulário para adicionar parâmetros de saúde, como por exemplo a massa corporal, altura, entre outros. Após a inserção de alguns parâmetros é feito um calculo automático de dados como o IMC.

Questionario: página onde é possível preencher um questionário de saúde para o utente responder conforme o seu estado de saúde. Contém uma tabela onde é possível responder à questão de um a cinco conforme o utente se sentir.

ConsultaQuestionarios: página onde é possível ver os resultados do questionário de saúde previamente preenchido pelo utente.

PlaneamentoAulas: página para adicionar um planeamento de um teste. Mostra um formulário onde é possível adicionar um novo planeamento para um “x” dia da semana.

Calendario: página que mostra um calendário com as sessões de aulas a dar. Contém um calendário onde poderão ser vistas as sessões de aulas programadas para cada dia. Também poderão ser adicionadas novas sessões nesta página.

CronogramaAulas: página que mostra as várias sessões de aulas. Contém as várias sessões de aulas que já foram dadas e poderão ser dadas. Também é possível filtrar por mês e por ano para uma melhor visualização de sessões de aulas. Além disso, é possível imprimir estes dados.

About: página que mostra os dados da plataforma. Por quem foi feita, e quem é o principal administrador da plataforma.

8.3. Código da Plataforma

8.3.1. Calcular a idade automaticamente

```
protected void Data_aniTextBox_TextChanged(object sender, EventArgs e)
{
    TextBox data =
    (TextBox)FormViewAdd_Utente.FindControl("Data_aniTextBox");
    TextBox idades =
    (TextBox)FormViewAdd_Utente.FindControl("IdadeTextBox");
    string idadess = idades.Text;
    RequiredFieldValidator idade =
    (RequiredFieldValidator)FormViewAdd_Utente.FindControl("RequiredFieldValida
torIddade");

    DateTime idad = DateTime.Parse(data.Text);
    TimeSpan ts = DateTime.Today - idad;

    int idadeAtual = DateTime.Now.Year - idad.Year;
    if (DateTime.Now.Month < idad.Month || (DateTime.Now.Month ==
idad.Month && DateTime.Now.Day < idad.Day))
```

```
{
    idadeAtual--;
}

if (idadeAtual <= 0)
{
    idade.ErrorMessage = "A idade não pode ser menor que 0";
}

idades.Text += idadeAtual;
}
```

8.3.2. Script SQL para adicionar dados em massa

```
BULK
INSERT ParametrosExercicios
FROM 'E:\Users\anabe\Desktop\Livro2.csv'
WITH
(
    FIELDTERMINATOR = ';',
    ROWTERMINATOR = '\n'
)
GO
```

8.3.3. Imprimir dados em PDF

```
protected void LinkButtonPDF_Click(object sender, EventArgs e)
{
    GridView1.AllowPaging = false;
    GridView1.DataBind();
    using (StringWriter sw = new StringWriter())
    {
        using (HtmlTextWriter hw = new HtmlTextWriter(sw))
        {
            int colCount = GridView1.Columns.Count - 1;

            PDFPTable table = new PDFPTable(colCount);
            table.HorizontalAlignment = 1;
            table.WidthPercentage = 100;

            int[] colWidths = new int[GridView1.Columns.Count];

            PDFPCell cell;
            string cellText;

            for (int colIndex = 0; colIndex < colCount; colIndex++)
            {
```

```

        colWidths[colIndex] =
(int)GridView1.Columns[colIndex].ItemStyle.Width.Value;

        cellText =
Server.HtmlDecode(GridView1.HeaderRow.Cells[colIndex].Text);

        cell = new PDFPCell(new Phrase(cellText));

        cell.BackgroundColor = new
BaseColor(System.Drawing.ColorTranslator.FromHtml("#e7eaec"));

        table.AddCell(cell);
    }

    //export rows from GridView to table
    for (int rowIndex = 0; rowIndex < GridView1.Rows.Count;
rowIndex++)
    {
        for (int j = 0; j < GridView1.Columns.Count - 1; j++)
        {
            //Esconder coluna do ItemTemplate
            this.GridView1.Columns[5].Visible = false;
            this.GridView1.Columns[6].Visible = false;

            cellText =
Server.HtmlDecode(GridView1.Rows[rowIndex].Cells[j].Text);

            cell = new PDFPCell(new Phrase(cellText));

            if (rowIndex % 2 != 0)
            {
                cell.BackgroundColor = new
BaseColor(System.Drawing.ColorTranslator.FromHtml("#e7eaec"));
            }
            else
            {
                cell.BackgroundColor = new
BaseColor(System.Drawing.ColorTranslator.FromHtml("#FFFFFF"));
            }
            table.AddCell(cell);
        }
        GridView1.RenderControl(hw);
    }

    //Create the PDF Document
    Document PDFDoc = new Document(PageSize.A4, 20f, 20f, 50f,
20f);

    PDFWriter.GetInstance(PDFDoc, Response.OutputStream);
    //open the stream
    PDFDoc.Open();
    //add the table to the document
    PDFDoc.Add(table);
    //close the document stream
    PDFDoc.Close();
    Response.ContentType = "application/PDF";
    Response.AddHeader("content-disposition", "attachment;" +
"filename=ListaTreinadores.PDF");
    Response.Cache.SetCacheability(HttpCacheability.NoCache);

```

```

        Response.Write(PDFDoc);
        Response.End();
    }
}

```

8.3.4. Atualização do calendário depois de inserida uma nova aula

```

protected void InsertButton_Click(object sender, EventArgs e)
{
    SqlConnection connectionInsert = null;

    try
    {
        connectionInsert = new
        SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].Connec
        tionString);
        connectionInsert.Open();

        DateTime datas = DateTime.ParseExact(data.Text, "dd/MM/yyyy",
        CultureInfo.InvariantCulture);
        DateTime horas = DateTime.ParseExact(hora.Text, "HH:mm",
        CultureInfo.InvariantCulture);

        string sqlInsertSessoes = "INSERT INTO [SessoesAulas]
        ([cod_plan_aulas], [Data_sessao], [hora_sessao]) VALUES (@cod_plan_aulas,
        @Data_sessao, @hora_sessao)";
        SqlCommand insertSessoes = new SqlCommand(sqlInsertSessoes,
        connectionInsert);

        insertSessoes.Parameters.AddWithValue("@cod_plan_aulas",
        cod_plan_aula.SelectedValue);
        insertSessoes.Parameters.AddWithValue("@Data_sessao", datas);
        insertSessoes.Parameters.AddWithValue("@hora_sessao", horas);
        insertSessoes.ExecuteNonQuery();

        data.Text = "";
        hora.Text = "";

        // REFRESH PARA ATUALIZAR DADOS
        Response.Redirect(Request.Url.AbsoluteUri);
    }

    catch (Exception ex)
    {
        Response.Write(ex.ToString());
        // send error message
    }
    finally
    {
        if (connectionInsert != null) connectionInsert.Close();
    }
}

```


8.4. Documentação entregue pelo cliente

8.4.1. Testes de Aptidão Física

Parâmetros de Saúde

- **Composição corporal**
 - **Parâmetros a avaliar e registrar**
 - Massa Corporal (Kg):_0
 - Altura (m):_____
 - IMC :_____

- **Dados da balança de bio impedância**
 - Massa magra (kg)
 - Massa gorda (Kg)
 - % de massa Gorda
 - % de Massa Magra
 - % de HO₂
 - Taxa metabólica (Kcal)

- **Circunferências**

Nome do teste	Parâmetros avaliados e a registrar	
Abdominal	(cm)	
Bicipital	Músculo relaxado (cm)	Músculo contraído (cm)
Crural	(cm)	
Geminal	(cm)	

- **Testes de Aptidão Funcional – Bateria de Fullerton (Rikli and Jones, 1999)**

Nome do teste	Parâmetros avaliados	Pontuação
Levantar e sentar na cadeira	Força resistente dos membros inferiores	Nº máximo de execuções em 30s
Flexão do antebraço	Força dos membros superiores	Nº máximo de execuções em 30s
Sentado e alcançar	Flexibilidade do tronco e dos membros inferiores	Distância (cm)
Alcançar atrás das costas	Flexibilidade do ombro	Distância (cm)
Sentado, caminhar 2,44 m e Voltar a Sentar	velocidade, agilidade e equilíbrio	Tempo (segundos)
Andar 6 minutos	Resistência Cardiovascular	Distância percorrida (m)
Dois minutos de steps no próprio lugar	Resistência Cardiovascular	Nº de elevações dos Joelhos

Avaliação da Força e Potência Muscular

- Força Máxima Dinâmica

Nome do teste	Parâmetros avaliados	Parâmetros a introduzir					
		Nº repetições	Carga (Kgs)	Estimativa da Fmax (kg)	Fmax (N)	Velocidade média Propulsiva (m.s ⁻¹)	Potência média Propulsiva (W)
Fmáx prensa de pernas	Força Máxima dos membros inferiores						
Fmáx Supino	Força Máxima da parte anterior do tronco						
Fmáx Remada horizontal	Força Máxima da parte posterior do tronco						

- **Força Máxima Isométrica**

Nome do teste	Parâmetros avaliados	Registo
Fmáx isométrica – quadricípete (leg extension)	Força máxima estática do quadricípete	Força Máxima Produzida (N)
Fmáx isométrica – Mid thigh pull	Força máxima estática dos membros inferiores	Força Máxima Produzida (N)
Fmáx isométrica da mão – Handgrip	Força máxima estática e agarre da mão	Força Máxima Produzida (N)
Fmáx isométrica – parte anterior do tronco	Força máxima estática da parte anterior do tronco	Força Máxima Produzida (N)
Fmáx isométrica – parte posterior do tronco	Força máxima estática da parte posterior do tronco	Força Máxima Produzida (N)

- **Força resistente**

Nome do teste	Parâmetros avaliados	Registo
Flexões de braços	Força resistente trem superior	Nº máximo de flexões corretamente executadas
Curl-up	Força resistente da parede abdominal	Nº máximo de flexões corretamente executadas
Teste de Sorensen	Força resistente dos extensores da coluna	Tempo máximo (s)

• **Potência Muscular**

Nome do teste	Parâmetros a introduzir							
	Carga a usar (20%, 40%, 60% e 80%)	Pmax propulsiva (W)	Pmed propulsiva (W)	Fmax propulsiva (N)	Fmed (N)	Vmax Propulsiva (m.s ⁻¹)	Vmed Propulsiva (m.s ⁻¹)	RFD (N.m.s ⁻¹)
Potência Muscular – exercício prensa de pernas								
Potência Muscular – exercício supino								
Potência Muscular – exercício remada horizontal								

• **Flexibilidade**

Nome do teste	Parâmetros avaliados	Registro
Senta e alcança*	Flexibilidade da região lombar e isquiotibiais	(cm)
Alcançar atrás das costas*	Flexibilidade do ombro	(cm)

Flexão do tronco à frente	Flexibilidade da região lombar	(cm)
Teste de Thomas	Flexibilidade do psoas ilíaco	(cm)
Teste de elevação do MI	Flexibilidade dos membros inferiores	(cm)
Dedos ao chão (ângulo toracolombar)	Flexibilidade toracolombar	(cm)

- **Capacidade Cardiovascular**

Nome do teste	Parâmetros avaliados				
Teste submáximo de Astrand-Ryhming	FCmáxima (bpm)	Carga de trabalho (Kgm/min)	VO ₂ máx (Litros)	VO ₂ máx (Litros) – corrigido para a idade	VO ₂ máx. (ml/Kg/min)
Teste Rockport	Tempo (min)	FC(bpm)	VO ₂ máx Estimado (mL/Kg/min) (introduzir a seguinte formula $\text{VO}_2\text{max} = 132,853 - 0,0769 * (\text{Peso Corporal em lb}) - 0,3877(\text{idade, anos}) + 6,315(\text{sexo; 1- para masc; 0 para few}) - 3,269(\text{tempo, min}) - 0,1565(\text{FC, bpm})$		

- **Testes de equilíbrio**

Nome do teste				
Teste de Romberg	Olhos abertos (s)	Olhos fechados (s)		
Teste de Romberg modificada postura de tadem	Olhos abertos (s)	Olhos fechados (s)		
Postura unipedal	Olhos abertos (s)	Olhos fechados (s)		
Teste de alcance funcional	distância alcançada (tentativa 1)	distância alcançada (tentativa 2)	distância alcançada (tentativa 3)	Média das distâncias alcançadas (cm)

