

Sistemas de Informação e Bases de Dados Aplicados à Saúde

Licenciatura em Engenharia Biomédica

2023 – 2024

Relatório Final - módulo P3

Nome: Catarina Macedo N^o:1211278

Turma: 2E

Entrega: 19/06/2024

Índice

1.	Introdução ao tema	4
2.	Objetivos	4
3.	Desenvolvimento	4
3.1.	Base de Dados	4
3.1.1	Modelo conceitual	4
3.1.2.	Modelo relacional	6
3.2.	Front-End	7
3.2.1.	Identidade da marca	7
3.2.2.	Organização e estrutura das páginas	7
3.2.3.	Diagrama de fluxo	11
3.2.4.	Ferramentas	11
3.3.	Ligação Front-End com a Base de dados	12
3.3.1.	Organização do projeto	12
4.	Conclusão	17
5.	Bibliografia	18

Figura 1:	Logótipo.....	7
Figura 2:	Botão de login e sign in.	7
Figura 3:	Opções da Navbar do módulo.....	8
Figura 4:	Container – Clientes.....	8
Figura 5:	Tabela plano de treino.	8
Figura 6:	Formulário Plano de Treino.....	9
Figura 7:	Sugestões de exercícios.	9
Figura 8:	Botão "Criar Novo Cliente".	9
Figura 9:	Tabelas de exercícios.....	10
Figura 10:	Estatística.	10
Figura 11:	Alert para a saída do módulo.	11
Figura 12:	Diagrama de fluxo.	11
Figura 13:	Validação no login.	13
Figura 14:	Mensagem de erro.....	13
Figura 15:	Procura dos clientes do fisiologista.	13
Figura 16:	Homepage do fisiologista.	13
Figura 17:	Parte inicial da página cliente_info.php.	14
Figura 18:	Homepage.php	14
Figura 19:	Client_info.php.....	14
Figura 20:	query da “Tabela Plano de Treino”.	15
Figura 21:	Botão de eliminar da Tabela Plano de Treino.	15
Figura 22:	Botão de criar nova linha.	15
Figura 23:	Valor do input "nome" do perfil.....	16
Figura 24:	Contagem de clientes por género.....	16
Figura 25:	Gráficos gerados.	17

1. Introdução ao tema

Este projeto, desenvolvido no âmbito da unidade curricular de Sistemas de Informação e Bases de Dados Aplicados à Saúde, foca-se na criação de um módulo de apoio para um estúdio de fisiologistas.

Este módulo visa facilitar a gestão da elaboração e atribuição de planos de treino personalizados, permitindo o registo de uma avaliação física e funcional do cliente e a elaboração do plano de treino. Sendo assim, após a realização da avaliação física, o fisiologista é responsável por conceber um plano de treino adaptado às necessidades específicas desse cliente.

Para ser possível realizar os planos de treino é necessário criar uma lista de exercícios, com informações como o nome do exercício, equipamento utilizado, número de séries, duração de cada série, ritmo de execução, carga, tempo de descanso, entre outros. Assim, após a tipologia dos exercícios estar implementada no sistema, é necessário que os planos sigam as normas implementadas pelo estúdio, que consistem em incluir no máximo 8 exercícios de treino de força, num total de 20 minutos, e obrigatoriamente 10 minutos de treino aeróbico.

2. Objetivos

Os objetivos deste projeto são: desenvolver uma WebApp para um módulo de apoio à gestão dos planos de treino de um estúdio, integrar um sistema de base de dados eficiente para armazenar e recuperar informações dos planos de treino, garantir a segurança e privacidade dos dados dos utilizadores, criar uma interface de utilizador intuitiva para a gestão dos planos de treino e otimizar a aplicação para diferentes dispositivos, garantindo a responsividade do design.

3. Desenvolvimento

Nesta secção, serão apresentadas as diversas etapas do desenvolvimento da WebApp, incluindo a Base de Dados (secção 3.1), o Front-End (secção 3.2) e a Integração do Front-End com a Base de Dados (secção 3.3).

3.1. Base de Dados

A base de dados inicialmente desenvolvida sofreu ligeiras alterações de forma a atender a todos os requisitos do enunciado. Assim, o exercício foi separado em duas entidades distintas: "Exercício" e "Execução".

3.1.1 Modelo conceitual

Um modelo conceitual é uma representação abstrata e de alto nível de um sistema ou domínio de negócios. Descreve os conceitos e relações entre as entidades, com uma representação abstrata de uma estrutura para a base de dados.

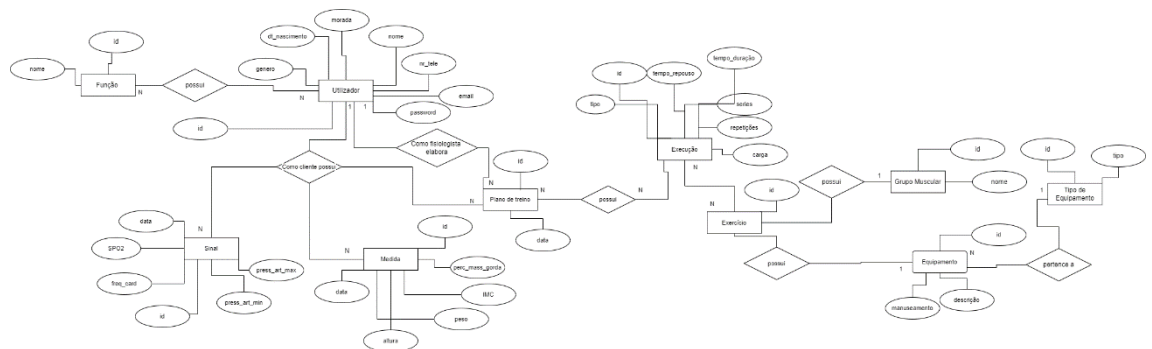
As principais entidades do modelo são: Utilizador, Função (papel desempenhado pelo utilizador), Sinal, Medida, Plano de Treino, Exercício, Execução, Equipamento, Tipo de Equipamento e Grupo Muscular.

- Utilizador:

- Esta entidade é responsável por armazenar as informações sobre os utilizadores do programa.
- Os atributos email e password são utilizados pelo utilizador para que este seja capaz de realizar o login.
- Função:
 - Esta entidade diz respeito às funções que podem ser aplicadas ao utilizador, como cliente e fisiologista. Nesta entidade existe dois atributos onde é guardado o nome da função e o seu respetivo id, que será utilizado para referenciar a função de cada utilizador.
- Sinal:
 - Armazena os dados recolhidos na avaliação física, contendo vários parâmetros essenciais para o fisiologista elaborar o plano de treino.
- Medida:
 - Tem uma utilidade semelhante à entidade sinal.
- Plano de treino:
 - Esta entidade armazena e agrupa o id do cliente, que usufrui do plano de treino, e do fisiologista, que cria o plano de treino, e dos exercícios pertencentes ao plano de treino, contendo uma data.
- Tipo de equipamento:
 - Esta entidade armazena os tipos de equipamentos presentes no estúdio, ou seja, se são para exercícios de força ou aeróbicos. Desta forma o equipamento terá um id associado que diz respeito a esta informação.
- Grupo muscular:
 - Contém um objetivo semelhante à entidade anterior, mas neste caso armazena que grupo muscular é ativado durante o exercício.
- Equipamento:
 - Esta entidade tem como finalidade descrever o manuseamento do equipamento e uma pequena descrição sobre este, de forma a facilitar a adaptação do cliente a esta. Nela armazena o id do tipo de equipamento.
- Exercício:
 - Esta entidade contém o nome do exercício, o id do equipamento associado e o id do grupo muscular associado.
- Execução:
 - A entidade execução descreve como será o exercício em termos técnicos, descrevendo a carga, tempo de descanso, duração entre outras. Contém o id do equipamento e do grupo muscular.

Sendo assim, apenas existe uma entidade para a gestão de utilizadores, acompanhada pela tabela Função, de modo a evitar a redundância de informação de clientes e fisiologistas, uma vez que o utilizador pode desempenhar ambos os cargos, ser fisiologista e ao mesmo tempo ser cliente do estúdio. Deste modo, a escalabilidade do sistema é assegurada, uma vez que a entidade Utilizador detém os campos email e password permitindo efetuar o login. Ao mesmo tempo a entidade Função assegura que existe um mapeamento entre o utilizador e as permissões que este tem no sistema.

Ambas as entidades sinal e medida estão associadas ao cliente e possuem uma data. Deste modo, é também possível obter o histórico de evolução do cliente ao longo do tempo. Sendo assim possível, no futuro, implementar essa funcionalidade se assim for decidido.



3.1.2. Modelo relacional

O modelo relacional é fundamental para a organização, manipulação e garantia da integridade dos dados numa base de dados.

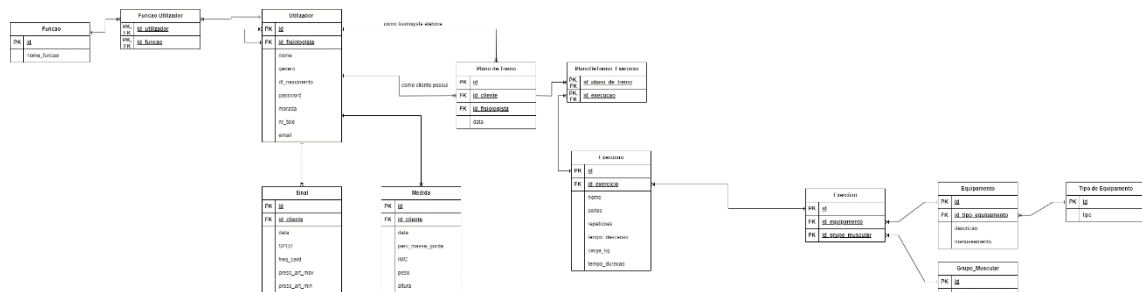


Figura 2 - modelo relacional.

Para a elaboração do modelo relacional tomou-se por base o modelo conceitual, sem esquecer as 3FN. A terceira forma normal (3FN) é importante para uma base de dados, dado que ajuda a evitar redundância de dados e a manter a consistência das informações. Esta permite reduzir o espaço de armazenamento e facilita a manutenção e consulta dos dados. E ainda, promove uma estrutura mais eficiente e organizada para a base de dados [1].

Podemos verificar que a primeira forma normal foi cumprida, tendo em conta, que em todas as entidades da base de dados, os atributos apresentam um caráter atômico, ou seja, não exibem uma estrutura/conteúdo complexa. A segunda forma normal também foi cumprida, visto que, os atributos não-chave dependem da chave primária [1]. Por fim, verificamos também a 3 forma normal, dado que nenhum atributo não-chave não depende de outros atributos não-chave [1].

3.2. Front-End

O Front-End refere-se à interface do utilizador e serve para facilitar a interação entre o utilizador e a aplicação. É a camada visual da aplicação, onde todos os elementos gráficos e funcionais são apresentados, como botões, menus, formulários e conteúdos multimédia. O objetivo principal do Front-End é proporcionar uma experiência de utilizador agradável, intuitiva e eficiente.

3.2.1. Identidade da marca

Para organizar o design do Front-End do módulo, foi realizada uma pesquisa sobre identidade visual de uma marca. Considerando que o módulo é destinado a um ginásio de fisioterapeutas, era pretendido transmitir uma sensação de calma, tranquilidade e felicidade aos utilizadores, através do logótipo e cores escolhidas [2], [3].

Assim, foi adicionado o verde como cor complementar às cores base, preto e branco, de modo a tornar o módulo mais atrativo para os utilizadores [3].



Figura 1: Logótipo.

3.2.2. Organização e estrutura das páginas

O estúdio criado contém uma página que apresenta a empresa, referindo os seus valores, os seus serviços, os contactos da empresa e um mapa com a localização do estúdio.

Na *navbar* da página de apresentação foram colocados dois botões: botão de *login* e de *sign in*, como é demonstrado na figura seguinte.



Figura 2: Botão de login e sign in.

O botão de *login* redireciona o visitante para a página de *login*, onde o utilizador poderá colocar as suas credenciais. Já o botão de *sign in* direciona o utilizador para uma página onde poderá criar as suas credenciais.

Ao realizar o *login*, o fisiologista vai para a Homepage do módulo. Na *navbar* presente no módulo do fisiologista, há a possibilidade de aceder ao perfil do fisiologista, à biblioteca de exercícios, estatísticas, voltar à Homepage ou sair do módulo, como é possível visualizar na seguinte figura.

Homepage
Perfil
Estatística
Biblioteca de exercícios
Sair

Figura 3: Opções da Navbar do módulo.

Ao entrar na Homepage o utilizador vai visualizar os clientes que lhe estão atribuídos.

Nesta página há a possibilidade de aceder às informações de cada cliente através de um botão presente na caixa deste, como demonstrado na seguinte figura.

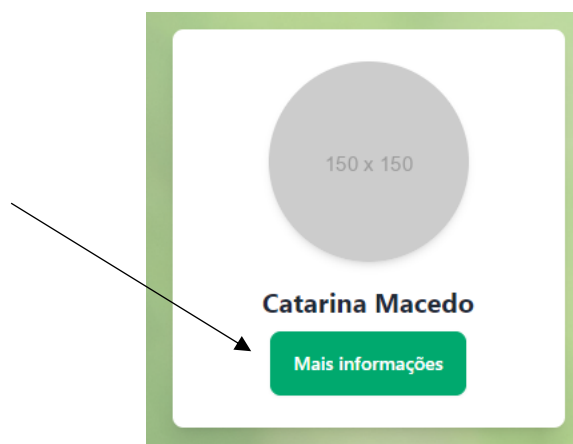


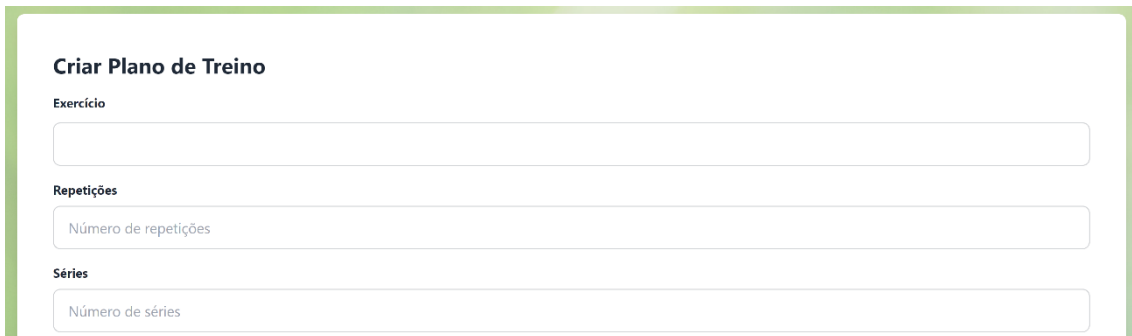
Figura 4: Container – Clientes.

Ao entrar na página do cliente é possível visualizar as informações do cliente e as tabelas do plano de treino, medidas e sinais com as suas respetivas datas. Nestas tabelas podemos criar novas linhas ou eliminar.

Tabela Plano de Treino									Criar Nova Linha
Exercício	Equipamento	Tipo	Repetições	Séries	Carga Kg	Descanso min	Duração min	Grupo Muscular	Ações
Agachamento Smith	Smith	musculacao	1	1	200	1	1	pernas	Eliminar

Figura 5: Tabela plano de treino.

Em todas as tabelas existe o botão “Criar Nova Linha” que redireciona o utilizador para um formulário, que permite inserir novas informações em cada tabela. Este formulário foi elaborado de forma que o utilizador estivesse o mais condicionado possível nas suas respostas. Assim, simplifica-se a gestão de dados, o seu processamento e introdução na base de dados,



Criar Plano de Treino

Exercício

Repetições


Número de repetições

Séries

Número de séries

Figura 6: Formulário Plano de Treino.

Nas tabelas “Criar Plano de treino” e “Criar Novo Exercício”, a escrita no campo exercício é livre, mas haverá a recomendação dos exercícios existentes na base de dados, como apresentado na seguinte figura.



Criar Plano de Treino

Exercício

Repetições

Número de

Séries

Número de

Descanso (min)

Tempo de descanso em minutos

Supino

Agachamento

Leg Press

Lounges

Corrida

Bicicleta

Figura 7: Sugestões de exercícios.

Na Homepage há também a possibilidade de adicionar novos clientes, através de um botão “Criar Novo Cliente”. Da mesma forma, redireciona o utilizador para um formulário.

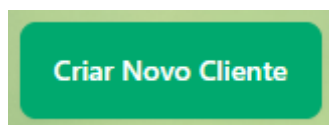


Figura 8: Botão "Criar Novo Cliente".

No perfil do fisiologista o utilizador consegue visualizar os seus dados e editá-los.

Coop Schutte

Nome:	Coop Schutte
Morada:	90 Larry Parkway
Email:	cschutter0@yellowbook.com
Numero de telemovel:	+351919300500
<button>Editar</button>	

Figura 9: Local de edição no perfil.

Na biblioteca de exercícios o fisiologista consegue visualizar os exercícios já existentes e criar novos exercícios. Na figura seguinte, encontra-se representada a tabela da página “Biblioteca de exercícios”.

Biblioteca de Exercícios				
Exercícios				
Exercício	Grupo Muscular	Equipamento	Tipo de Exercício	Ações
Levantamento Terra	costas	Halteres	musculacao	<button>Eliminar</button>
Desenvolvimento de Ombros	ombro	Halteres	musculacao	<button>Eliminar</button>
Rosca Direta	biceps	Halteres	musculacao	<button>Eliminar</button>

Figura 10: Tabelas de exercícios.

Com o botão “Criar Novo Exercício”, o utilizador é redirecionado para um formulário que permite a criação de um novo exercício.

Nas estatísticas, presentes na *navbar*, estão apresentados dois gráficos-exemplo de informação que poderia ser relevante para o fisiologista.

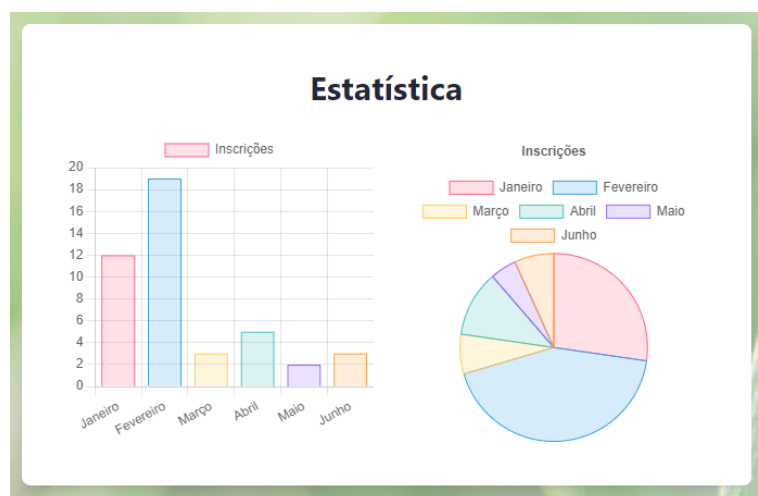


Figura 11: Estatística.

127.0.0.1:5500 diz

Tem a certeza?

OK

Cancelar

Se o utilizador seleccionar a opção “OK” é redireccionado para o *login*. Para voltar à página de apresentação é apenas necessário seleccionar o logótipo existente na página de *login*.

Um diagrama de fluxo é uma representação gráfica de um processo ou sistema, que mostra as etapas ou atividades e a sequência em que ocorrem. Neste caso, o diagrama de fluxo serve para demonstrar o fluxo do Front-End do módulo P3.



Através deste diagrama de fluxo é possível perceber o percurso pelas várias páginas do Front-End.

Foram utilizadas várias ferramentas para facilitar a criação do frontend da aplicação, entre as quais:

- DaisyUI;
- Chart.js;

- `Animate.css`.

Para aplicar as funcionalidades essenciais ao módulo, foi utilizado o DaisyUI, uma biblioteca baseada no Tailwindcss [4]. O DaisyUI é uma extensão que simplifica o desenvolvimento de interfaces responsivas e estilizadas, fornecendo componentes prontos a usar, como botões, cartões e barras de navegação [4]. Tailwindcss, por sua vez, é um framework CSS que oferece classes de baixo nível para criar designs personalizados de forma rápida e eficiente [5].

O Chart.js foi utilizado para criar gráficos na seção de estatísticas [6]. O Chart.js é uma biblioteca JavaScript de código aberto que permite a criação de gráficos interativos e dinâmicos, suportando vários tipos de gráficos [6]. Este é altamente personalizável e facilita a apresentação visual dos dados [6].

Para colocar animações na página de apresentação e na homepage, foi utilizado o Animate.css [7]. Esta biblioteca de animações CSS oferece uma vasta gama de animações pré-definidas, como fades, bounces e flips, que podem ser facilmente aplicadas a elementos HTML. O Animate.css melhora a interatividade e a experiência do utilizador ao incorporar animações dinâmicas de forma simples e eficiente [7].

Estas ferramentas, em conjunto, permitiram desenvolver um Front-End atrativo, responsivo e funcional, melhorando a experiência do utilizador e a visualização de dados.

3.3. Ligação Front-End com a Base de dados

O PHP é uma linguagem de programação focada principalmente em scripts do lado do servidor, permitindo a criação de páginas dinâmicas, a recolha de dados de formulários, gestão páginas com conteúdo dinâmico ou enviar e receber cookies [8].

O PHP é utilizado na WebApp para gerar html e para permitir a ligação entre a interface do utilizador e a base de dados. Isto inclui, incluir novos dados na base de dados, atualizar os dados, eliminar dados e ainda demonstrar os dados guardados na base de dados [8].

Tendo em conta o contexto da aplicação, isto transpõem-se na possibilidade de através do Front-end criar novas linhas nas tabelas criadas, atualizar a informação do fisiologista, criar novos clientes e visualizar toda a sua informação.

A introdução do PHP no projeto resultou em pequenas alterações no Front-End, que serão detalhadas ao longo desta secção.

3.3.1. Organização do projeto

O projeto está dividido em várias pastas entre as quais, a pasta “assets” que contém as imagens utilizadas na WebApp e as folhas de estilo, a pasta “configs”, que possui um ficheiro com algumas informações sobre a base de dados, a pasta “layouts” que carrega o “header” e o “footer” utilizados em todas as páginas e a pasta “helper” que inclui funções integradas em todo o ficheiro.

3.3.2. Aplicação do PHP

Na página de login, o fisiologista pode aceder à WebApp inserindo as suas credenciais. Este formulário inclui validações que garantem que o utilizador preencha todos os campos de forma correta.

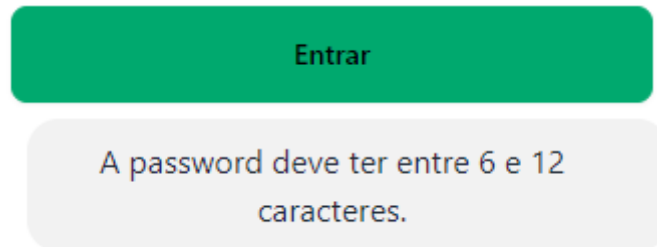


Figura 14: Validação no login.

Este módulo foi exclusivamente desenvolvido para o fisiologista, limitando o acesso à WebApp apenas a este utilizador. Portanto, quando um utilizador que não seja fisiologista tenta entrar no módulo, uma mensagem de erro é exibida.



Figura 15: Mensagem de erro.

Ao entrar na sessão é possível aceder aos clientes que lhe correspondem, isto é realizado através de uma *query* que procura o id do utilizador de acesso, no campo `id_fisiologista` dos utilizadores.

```
<?php
$ligacao = new PDO("mysql:host=" . MYSQL_HOST . ";dbname=" . MYSQL_DATABASE . ";charset=utf8", MYSQL_USERNAME, MYSQL_PASSWORD);
$queryClientes = $ligacao->prepare("SELECT * FROM utilizador WHERE id_fisiologista = :id");
$queryClientes->execute([':id' => $_SESSION['user']]);
$clientes = $queryClientes->fetchAll(PDO::FETCH_OBJ);
?>
```

Figura 16: Procura dos clientes do fisiologista.

Assim, ao entrar na “*Homepage.php*” serão dispostos todos os clientes do fisiologista, como é possível observar.

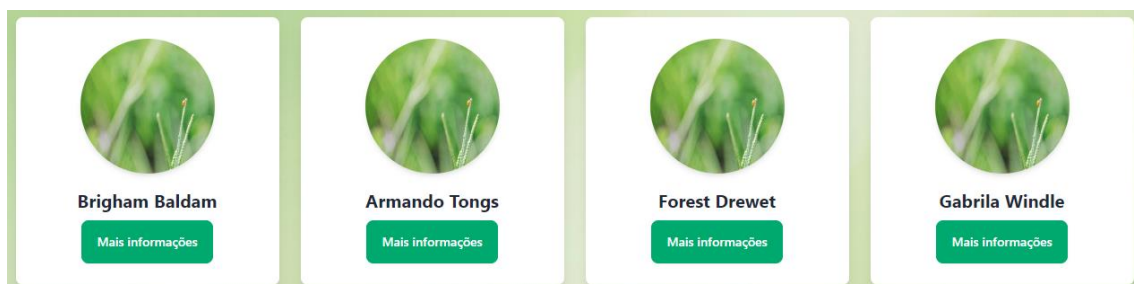


Figura 17: Homepage do fisiologista.

Ao entrar no botão mais informação, entramos na página “*cliente_info.php*” e o nome do cliente correspondente aparece escrito na página.



Brigham Baldam

Figura 18: Parte inicial da página *cliente_info.php*.

Isto é realizado mediante o envio do id do cliente no acesso à página “*cliente_info.php*”.

```
<h2 class="text-center text-xl font-bold mb-2"><?= $cliente->nome ?></h2>
<div class="text-center">
  <a class="btn btn-success text-white" href="client_info.php?id_cliente=<?= $cliente->id ?>">Mais
    informações</a>
</div>
```

Figura 19: *Homepage.php*

```
$parametros = [
  ':id' => $_GET['id_cliente'],
];

$ligacao = new PDO("mysql:host=" . MYSQL_HOST . ";dbname=" . MYSQL_DATABASE . ";charset=utf8", MYSQL_USERNAME, MYSQL_PASSWORD);
$comando = $ligacao->prepare("SELECT * FROM utilizador WHERE id = :id");
$comando->execute($parametros);
$resultadoCliente = $comando->fetchAll(PDO::FETCH_OBJ);
```

Figura 20: *Client_info.php*

Assim, nas tabelas da página “*cliente_info.php*” encontram-se disponibilizadas as informações guardadas na base de dados relativas ao cliente. Estas são as mesmas apresentadas no capítulo Front-End, ou seja, “Tabela Plano de Treino”, “Tabela Medidas” e “Tabela Sinais”.

Tanto a “Tabela Medidas” e “Tabela Sinais” apenas demonstram os dados diretos da entidade “medida” e “sinal”, contudo para a “Tabela Plano de Treino”, foi necessário a procura do id do plano de treino associado ao cliente, execuções existentes, o nome do exercício cujo id estivesse presente na execução e ainda o tipo de equipamento que é guardado dentro do exercício. Isto resultou numa maior complexidade em termos de *query* e de exposição de resultados.

```

$comando = $ligacao->prepare("
SELECT
    ex.id AS exec_id,
    exe.nome AS exercicio,
    eq.descricao AS equipamento,
    te.tipo AS tipo,
    ex.repeticoes,
    ex.series,
    ex.carga_kg,
    ex.tempo_descanso,
    ex.tempo_duracao,
    gm.nome AS grupo_muscular
FROM plano_treino pt
JOIN plano_treino_execucao pte
ON pte.id_plano_treino = pt.id
JOIN execucao ex
ON ex.id = pte.id_execucao
JOIN exercicio exe
ON exe.id = ex.id_exercicio
JOIN equipamento eq
ON eq.id = exe.id_equipamento
JOIN tipo_equipamento te
ON eq.id_tipo = te.id
JOIN grupo_muscular gm
ON gm.id = exe.id_grupo_muscular
WHERE pt.id_cliente = :id;
");
$comando->execute($parametros);
$exercicios = $comando->fetchAll(PDO::FETCH_OBJ);

```

Figura 21: query da “Tabela Plano de Treino”.

Em todas as tabelas disponibilizadas é possível apagar dados. A funcionalidade de edição não foi implementada, uma vez que os registos são simples de criar e sua permanência na base de dados não é obrigatória ou necessária, como no caso dos utilizadores.

Para todas as tabelas foi criado um ficheiro que permitia eliminar a linha em específico da tabela, como “eliminar_execucao.php”, “eliminar_medida.php” e “eliminar_sinal.php”

```

<td><a href="eliminar_execucao.php?id_cliente=?= $_GET['id_cliente']">&id_exec=?= $exercicio->exec_id ?>"
class="btn btn-error">Eliminar</a></td>

```

Figura 22: Botão de eliminar da Tabela Plano de Treino.

Os ficheiros PHP utilizados para eliminar as linhas das tabelas contêm uma estrutura semelhante, a obtenção do id do cliente e do id da entidade a ser eliminada, quer seja execução, sinal ou medida e o pedido à base de dados para fazer o delete desse registo.

Da mesma forma em todas as tabelas há a possibilidade de acrescentar mais linhas, com a criação de um novo registo. Para isso, os botões criados contêm o id do cliente no href de forma a poder ser utilizado na nova página para a inserção de um novo registo.

```

<button class="btn btn-success text-white" onclick="location.href='create_PT.php?id_cliente=?= $_GET['id_cliente'] ?>'">
Criar Nova Linha</button>

```

Figura 23: Botão de criar nova linha.

A inserção de novas medidas e novos sinais é feita através de um simples POST com os dados introduzidos nos campos dos formulários. No entanto, a inserção de novas linhas no plano de treino é uma tarefa mais complexa. Primeiramente, é criada uma nova execução. Em seguida, verifica-se se o cliente já possui um plano de treino associado. Caso não exista, é criado um novo plano, associando o id do cliente e do fisiologista. Com o plano de treino disponível, o id da nova

execução e do plano de treino são registados na tabela "plano_treino_execucao", que associa ambas as entidades.

Em todos formulários de criação de novos registos à restrição máxima possível de preenchimento livre por parte do utilizador. Utilizando inputs onde só seja possível seleccionar valores que estejam já na base de dados.

```
<div class="mb-4">
  <label for="equipment" class="block text-sm font-bold mb-2">Equipamento</label>
  <select id="equipment" name="equipment" class="select select-bordered w-full">
    <?php foreach ($equipamentos as $equipamento) : ?>
      <option value="<?= $equipamento->id ?>"><?= $equipamento->descricao ?></option>
    <?php endforeach; ?>
  </select>
</div>
```

Figura 24: Input do equipamento no formulário de criação de um novo exercício.

Na Biblioteca de Exercícios é utilizada uma lógica semelhante às entidades “medidas” e “sinais” para criar, apagar e demonstrar os registos.

A única entidade que é atualizada é o utilizador autenticado, isto é realizado através do perfil. Ao entrar no perfil, os valores dentro dos inputs contêm a informação do utilizador.

```
value="<?= ($resultadoCliente->nome) ?>"
```

Figura 25: Valor do input "nome" do perfil.

A informação pode ser facilmente editada reescrevendo os dados no formulário. Ao clicar no botão "Editar", a informação é atualizada através do ficheiro “update_profile.php”. Neste ficheiro, os valores inseridos nos campos do formulário são utilizados para atualizar os dados do utilizador na base de dados.

Relativamente às estatísticas, realizou-se um estudo relativo ao género mais predominante dentro dos clientes do fisiologista.

```
foreach ($clientes as $cliente) {
    if ($cliente->genero === 'Male') {
        $maleCount = $cliente->total;
    } elseif ($cliente->genero === 'Female') {
        $femaleCount = $cliente->total;
    }
}
```

Figura 26: Contagem de clientes por género.

No fim foram gerados dois gráficos, um de barras e um “pie” para visualizar os resultados.

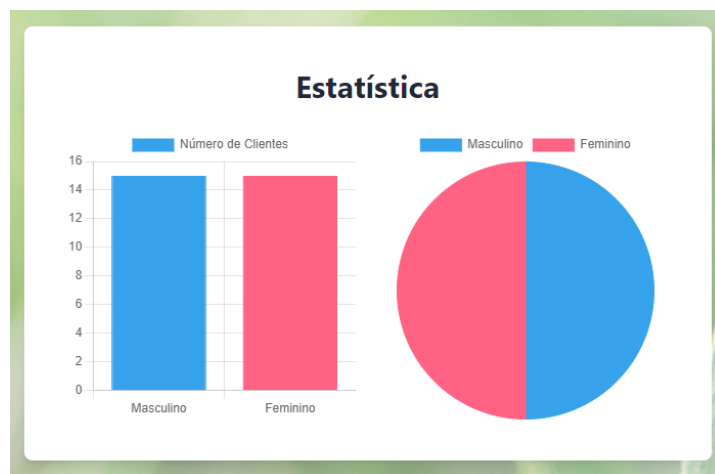


Figura 27: Gráficos gerados.

4. Conclusão

Na concepção da Base de Dados, foi essencial criar uma estrutura que evitasse redundâncias e facilitasse a demonstração dos dados necessários para o módulo, o que se revelou um obstáculo. A Base de Dados idealizada inicialmente foi alterada, pois não continha a estrutura ideal para que fosse possível realizar os pedidos à base de dados necessários ao bom funcionamento do Front-End.

Relativamente ao Front-End, a criação de uma interface intuitiva e eficaz foi uma tarefa que exigente, uma vez que foi necessário a elaboração de um fluxo claro e de fácil entendimento. Para além disso, a construção da base em html e posterior estilização envolveu a pesquisa de várias formas e ferramentas de construção de html, folhas de css e formas de dar responsividade ao template.

Dominar o PHP apresentou-se como uma tarefa extremamente desafiadora, uma vez que não possuía experiência prévia com esta linguagem. Adquirir competências em PHP no curto período disponível foi particularmente exigente, especialmente considerando a sua importância para o funcionamento da aplicação.

Devido à limitação de conhecimento em PHP, algumas partes do projeto necessitaram de alterações na fase final para assegurar a implementação de todas as funcionalidades exigidas. Além disso, a estrutura da base de dados teve de ser ajustada para facilitar tanto a inserção de dados quanto a sua exibição no Front-End.

Este módulo apresenta algumas limitações, tais como a impossibilidade de modificar os dados dos clientes e a restrição a um único plano de treino por paciente. Adicionalmente, verifica-se a ausência de algumas validações essenciais, como a obrigatoriedade de haver 8 exercícios de treino de força, num total de 20 minutos, e obrigatoriamente 10 minutos de treino aeróbico.

Em suma, apesar das suas falhas, a WebApp cumpre com maior parte dos requisitos preestabelecidos, provando-se ser bastante útil para a gestão de planos de treino.

5. Bibliografia

- [1] H. Proença, J. Muranho, and P. Prata, “Base de Dados I.” Accessed: Jun. 18, 2024. [Online]. Available: https://www.di.ubi.pt/~pprata/bd/BD_05_06_T7.pdf
- [2] C&I, “The Psychology of Color in Marketing: Choosing the Right Palette for Your Brand.” Accessed: May 28, 2024. [Online]. Available: <https://c-istudios.com/the-psychology-of-color-in-marketing-choosing-the-right-palette-for-your-brand/>
- [3] Josh Gallant, “Color palette trends to inspire your marketing strategy.” Accessed: May 28, 2024. [Online]. Available: <https://unbounce.com/landing-page-design/color-palettes-2022/>
- [4] daisyUI, *All daisyUI components*. 2024. Accessed: May 28, 2024. [Online]. Available: <https://daisyui.com/components/>
- [5] Get started, “tailwindcss,” 2024. Accessed: May 28, 2024. [Online]. Available: <https://tailwindcss.com/>
- [6] Chart.js, “Get Started,” 2024. Accessed: May 28, 2024. [Online]. Available: <https://www.chartjs.org/docs/latest/>
- [7] Animate.css, “Animate.css,” 2024, Accessed: May 28, 2024. [Online]. Available: <https://animate.style/>
- [8] PHP, “What can PHP do?” Accessed: Jun. 18, 2024. [Online]. Available: <https://www.php.net/manual/en/intro-whatcando.php>