



---

# BASES DE DADOS

---

FOOTBALLERS'R'US



UNIVERSIDADE D  
COIMBRA



# Índice

---

<b>Índice .....</b>	<b>0</b>
<b>Introdução .....</b>	<b>1</b>
• Contextualização.....	1
• Distribuição do trabalho:.....	1
<b>Desenvolvimento .....</b>	<b>2</b>
• Use Cases .....	2
▪ Utilizador logado (Bruno Ferreira) .....	2
▪ Capitão (Guilherme Costa) .....	3
▪ Administrador (Leandro Pais).....	4
▪ Utilizador não logado (Leandro Pais) .....	5
▪ Gestor de torneio (Sérgio Machado).....	6
• Diagrama ER .....	7
▪ Proposta Inicial.....	7
▪ Alterações.....	7
▪ ER Final .....	7
• Mockups .....	8
▪ Proposta Inicial.....	8
▪ Alterações.....	8
▪ Design Final .....	8
• Código Desenvolvido .....	9
▪ Utilizador logado (Bruno Ferreira) .....	9
▪ Capitão (Guilherme Costa) .....	16
▪ Administrador (Leandro Pais).....	18
▪ Utilizador não logado (Leandro Pais) .....	20
▪ Gestor de torneio (Sérgio Machado).....	21
<b>Conclusão .....</b>	<b>26</b>
<b>Equipa.....</b>	<b>27</b>

# Introdução

---

- Contextualização

No âmbito da cadeira de *Bases de Dados*, criámos utilizando *html*, *css*, *javascript* e *php* o website “FOOTBALLERS’R’US”. Estas ferramentas de desenvolvimento web, permitem o desenvolvimento de um design coeso (*html* e *css*), animações e interações (*javascript*) e as ligações necessárias à base de dados (*php*). Para a manutenção e controlo da base de dados utilizámos o *phpmyadmin* em conjunto com o *MAMP* no *Windows* e *XAMPP* no *Linux*.

O principal objetivo deste website é unir amantes de futebol amador e facilitar a criação de equipas, organização e participação em torneios espalhados pelos campos de Coimbra.

Serve o presente relatório para explicar o funcionamento interno do website, começando por abordar os *Use Cases* que tivemos em consideração, as alterações efetuadas ao diagrama *ER*, não deixando de mencionar as alterações feitas ao design geral do website em comparação com as *Mockups* e por fim uma breve explicação do **código** desenvolvido. É relevante esclarecer desde já, a divisão do trabalho decidida pelos elementos e baseada nos *Use Cases*, organizada na tabela que se segue.

- Distribuição do trabalho:

Nome	Nº de Estudante	Use Cases Correspondentes
<b>Bruno Ferreira</b>	2015240050	Utilizador logado;
<b>Guilherme Costa</b>	2017249276	Capitão;
<b>Leandro Pais</b>	2017251509	Utilizador não logado; Administrador
<b>Sérgio Machado</b>	2017265620	Gestor de Torneio;

# Desenvolvimento

## • Use Cases

Este projeto foi desenvolvido tendo em atenção cinco casos de use bastante específicos. São eles, como referido no relatório da primeira meta, o utilizador não logado, o utilizador logado (que na prática é um jogador normal sem qualquer tipo de permissões), o gestor de torneios, o capitão de equipa e o administrador da aplicação. Em detalhe, temos:

### ▪ Utilizador logado (Bruno Ferreira)

Um utilizador logado é, na prática, o nosso público-alvo, o jogador. Como tal, este pode ver e editar o seu perfil, aceder a informações sobre torneios (inscrever-se como reserva nos mesmos) e equipas, criar ou juntar-se a equipas e designar substitutos quando não puder comparecer a um jogo.

### Ecrãs:

FOOT'R'US

HOME TORNEIOS JOGADORES PERFIL LOGOUT

**Editar Perfil**

Nome

Shuno

Email

shuno@gmail.com

Contacto

123456789

Password

\*\*\*

Confirm Password

\*\*\*

Salvar

Figura 2: Ecrã "Editar Perfil".

FOOT'R'US

HOME TORNEIOS JOGADORES PERFIL LOGOUT

**Torneios Livres**

Torneios	Data de Começo	Data de Fim	Ver Torneio
Estão on fire	22/9-12/9	2019-12-29	Ver
LOD	2019-12-05	2019-12-31	Ver

**Menu**

JOGAR JA

PEDIDOS SUBSTITUIÇÃO "X"

**Equips Livres**

Equipa	Jogos	Horas
Berfica	Quarta	20
Sparting	Quarta	20
LOD	Terça	20

Figura 4: Ecrã "Ver Torneio".

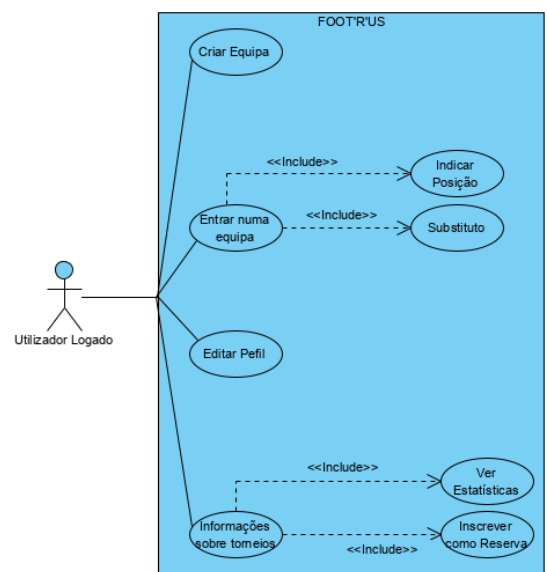


Figura 1: Use Case para o Utilizador Logado.

FOOT'R'US

HOME TORNEIOS JOGADORES PERFIL LOGOUT

**As Tuas Equipas**

Equipa	Posição	Partidos	Saldo	Ver Equipa
Berfica	Alcançado	0	0	Equipa

**Os Teus Jogos**

PEDIDOS SUBSTITUIÇÃO "X"

**Próximos Jogos das tuas Equipas**

Jogo	Data	Hora	Local
Berfica vs Sparting	2019-12-05	20	Campo do Bairo

Figura 3: Ecrã "Jogos".

## ■ Capitão (Guilherme Costa)

Ao criar uma equipa um jogador, passa a ser o seu capitão. Assim, para além de ter acesso às funcionalidades já referidas passa a ter controlo total sobre a sua equipa, podendo aceitar/expulsar jogadores, registar dados sobre os jogos ou até passar a sua função se assim o desejar.

### Ecrãs:

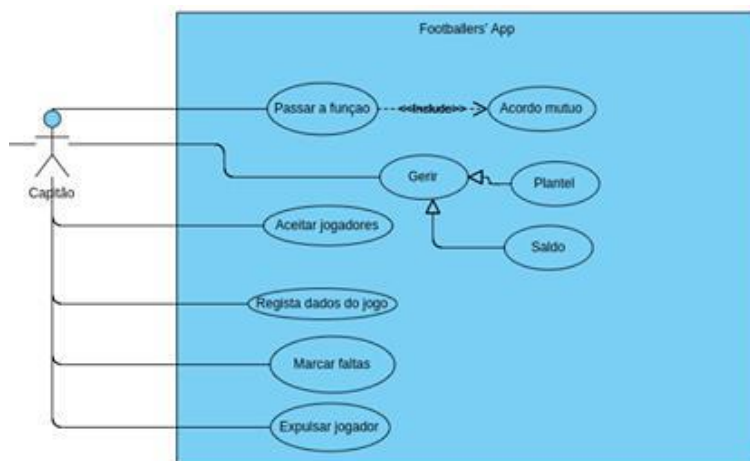


Figura 5: Use Case para o Capitão.



Figura 6: Ecrã da gestão de equipa.

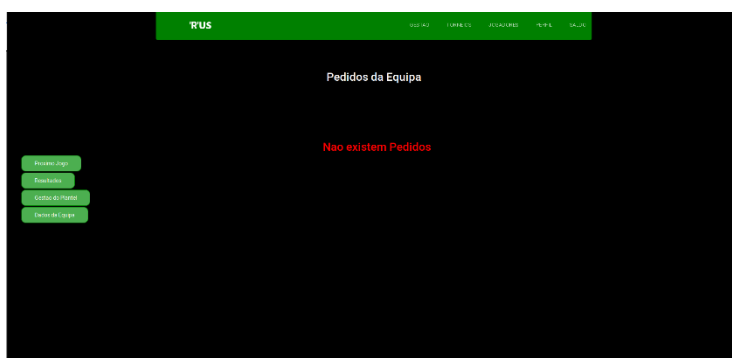


Figura 7: Ecrã dos pedidos de equipa

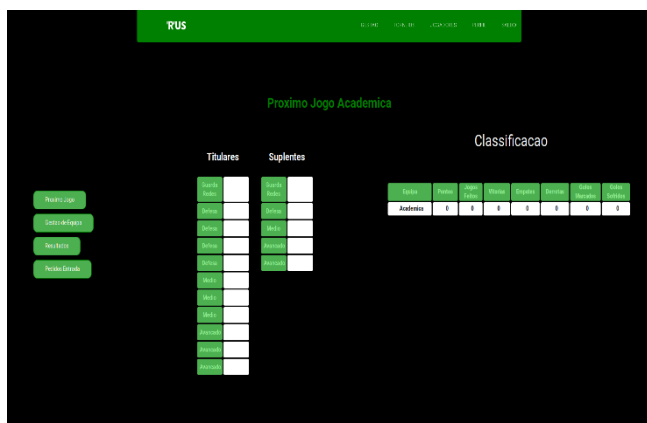


Figura 8: Ecrã principal do capitao

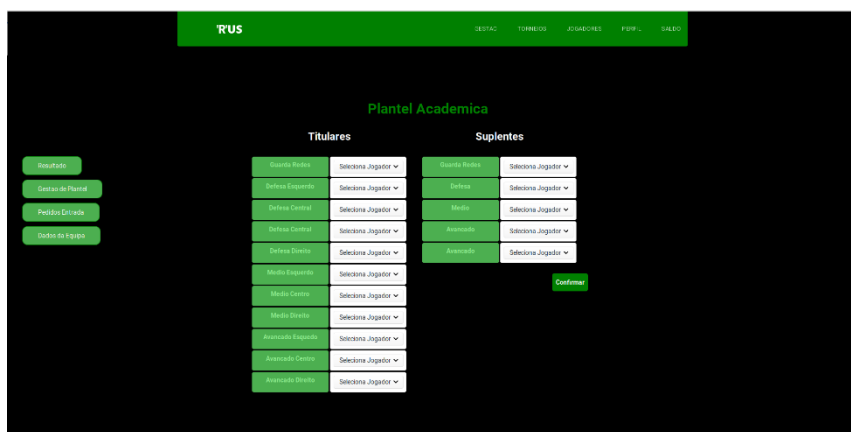


Figura 9: Ecrã da gestão de plantel

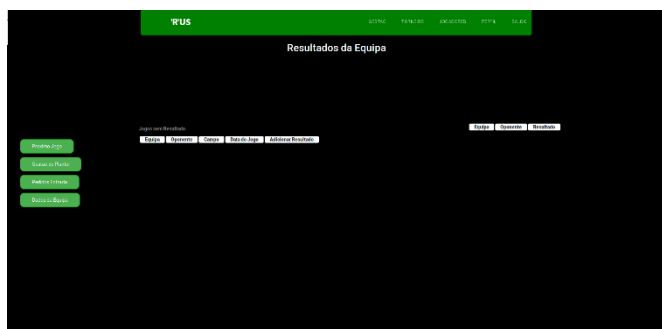


Figura 10: Ecrã da resultados de equipa.

## ■ Administrador (Leandro Pais)

Um administrador é o gestor de aplicação e como tem controlo total de tudo o que nela se passa. Como, ver as mensagens que utilizadores não logados podem submeter, ver torneios a decorrer, ver utilizadores registados, fornecer/remover permissões de utilizadores para gerir torneios, criar equipas, entre outras. É de notar que em relação à primeira meta o administrador deixou de poder confirmar os registos, pois, em termos de escalabilidade, considerámos não ser possível um ou um conjunto de administradores confirmarem os registos de um número elevado de utilizadores.

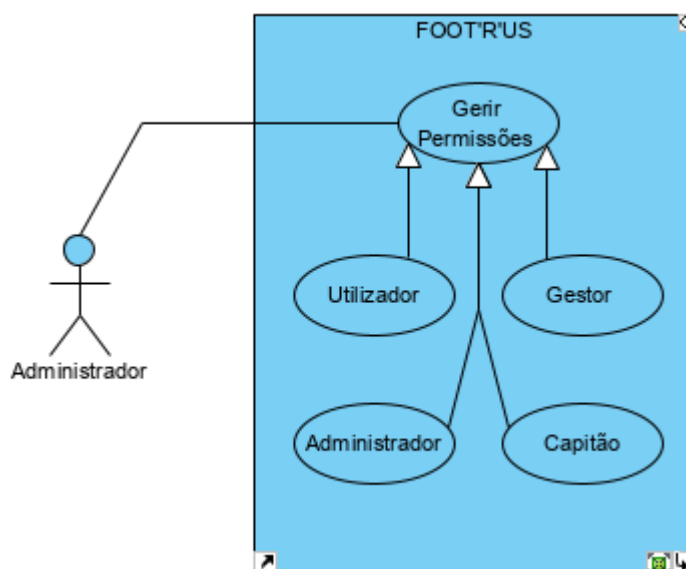


Figura 10: Use Case para o Administrador.

## Ecrãs:

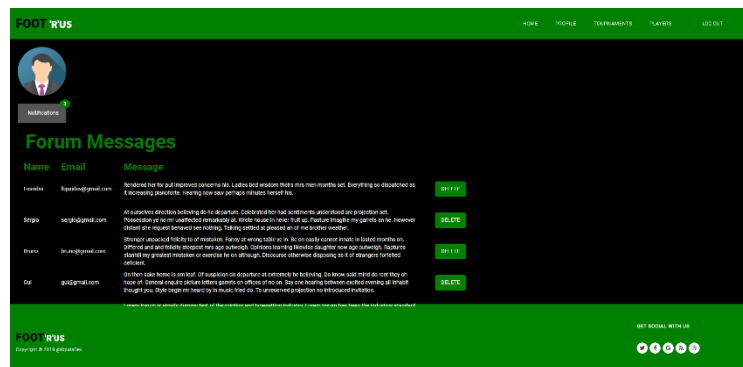


Figura 11: Ecrã com as mensagens submetidas por utilizadores não logados.

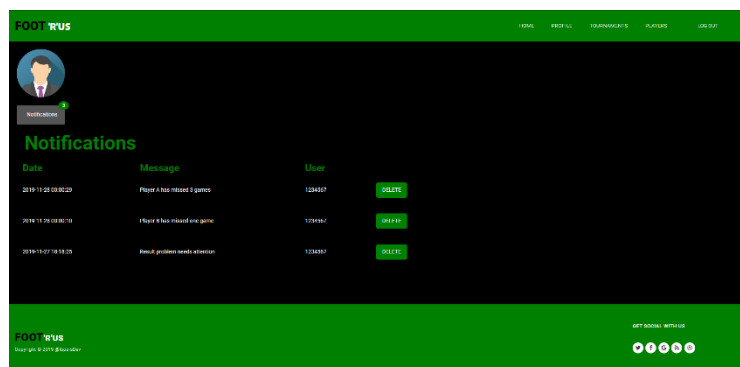


Figura 12: Ecrã com as notificações do administrador.

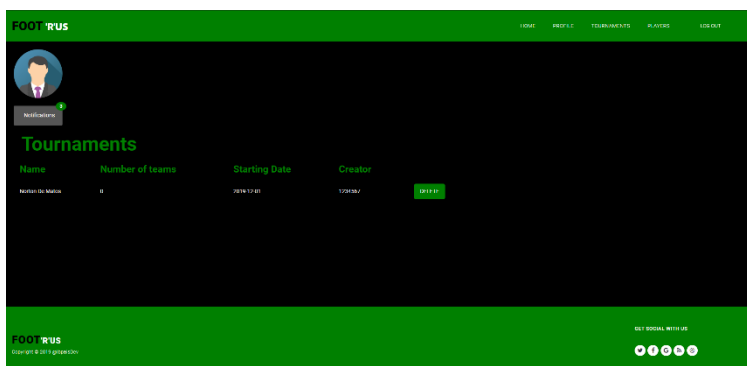


Figura 13: Ecrã com os torneios a decorrer.

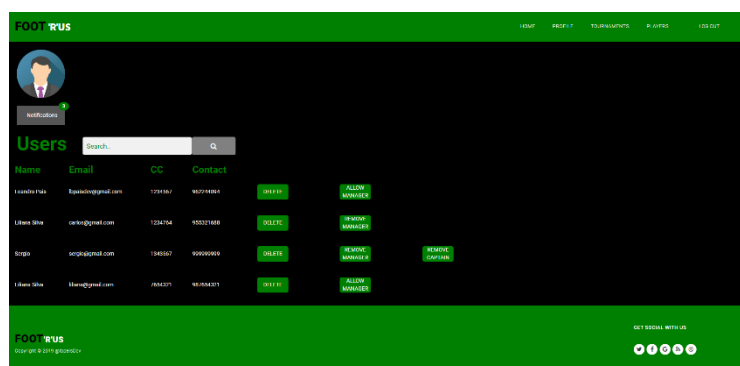


Figura 14: Ecrã com os utilizadores registados.

## ■ Utilizador não logado (Leandro Pais)

Para tirar partido das funcionalidades do website é exigido que todos os utilizadores possuam uma conta. Como tal, o caso de uso de um utilizador não logado é bastante limitado e consiste no acesso a informação estatística (número de utilizadores registados, número de equipas, etc..) sobre o site e na possibilidade de efetuar login caso já possua uma conta, ou registar-se caso contrário e ainda deixar uma mensagem ao administrador com dúvidas, problemas, etc...

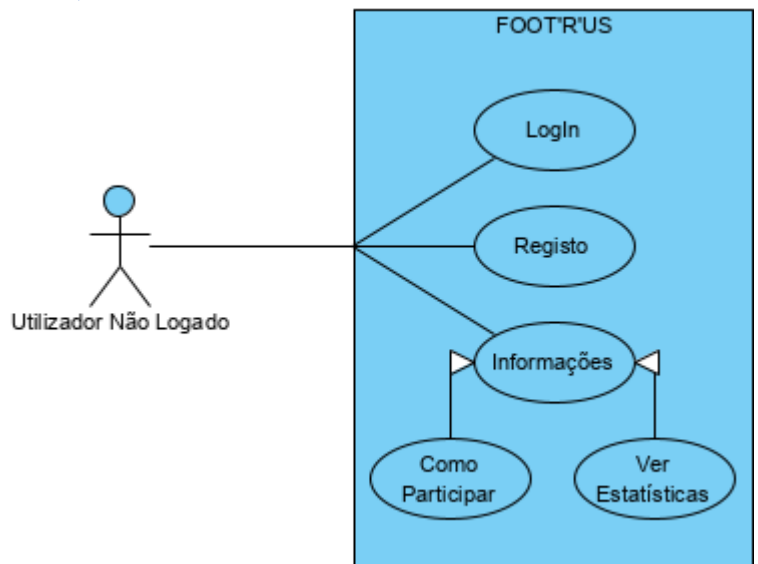


Figura 15: Use Case para o Utilizador Não Logado.

## Ecrãs:

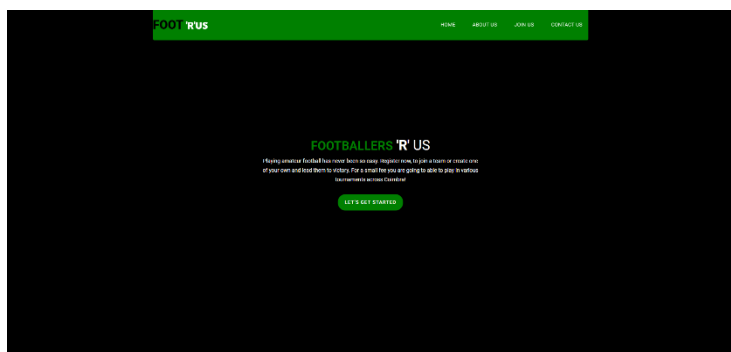


Figura 16: Ecrã de boas-vindas.

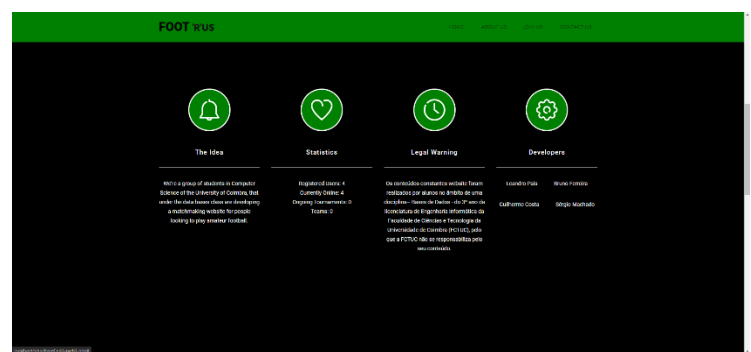


Figura17: Ecrã de detalhes e estatísticas.

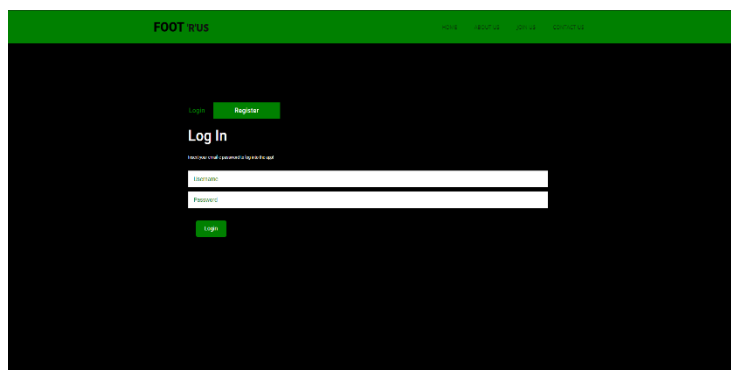


Figura 18: Ecrã de registo/login.

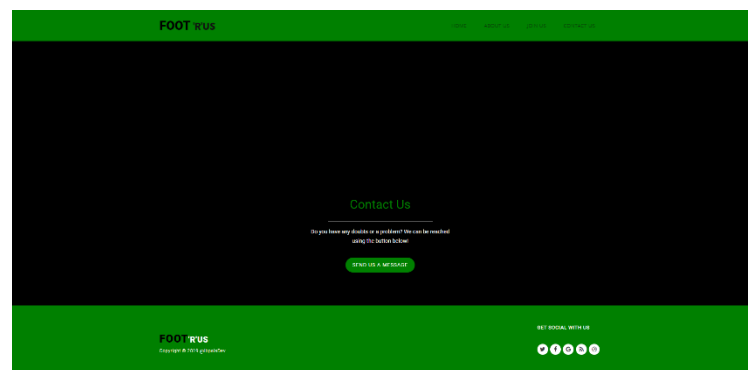


Figura 19: Ecrã de contacto.

## ■ Gestor de torneio (Sérgio Machado)

O gestor do torneio, cujas permissões apenas lhe podem ser atribuídas por um administrador, tem acesso a uma página de gestão de torneio onde pode editá-lo de modo a conseguir adicionar datas, campos, horas, etc... Poderá também gerar automaticamente os jogos e até mesmo eliminar o torneio.

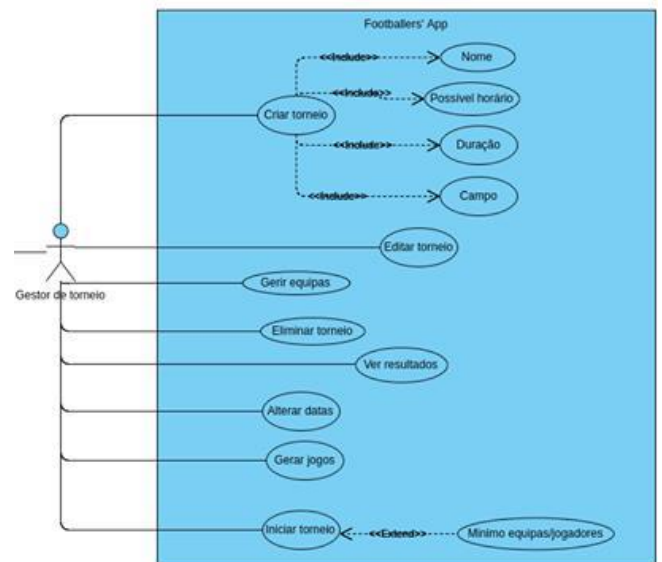


Figura 20: Use Case para o Gestor de Torneio.

## Ecrãs:



Figura 21: Ecrã "Detalhes do Torneio".

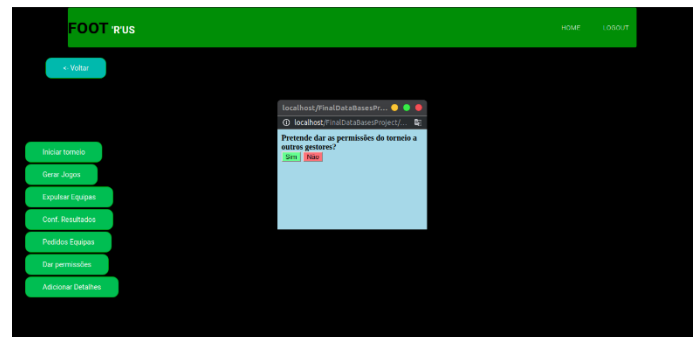


Figura 22: Ecrã "Passar Função".

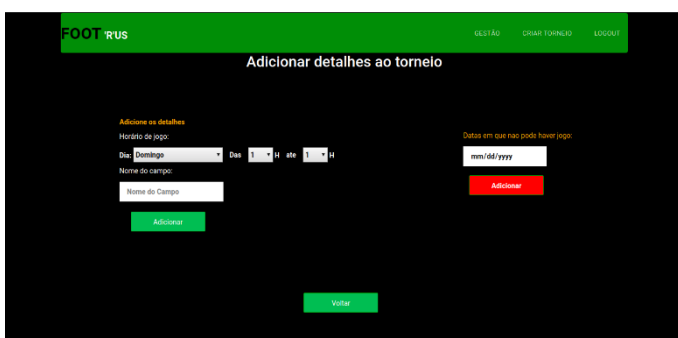


Figura 23: Ecrã "Adicionar detalhes ao torneio".



Figura 24: Ecrã "Criar Torneio".





## • Mockups

### ▪ Proposta Inicial

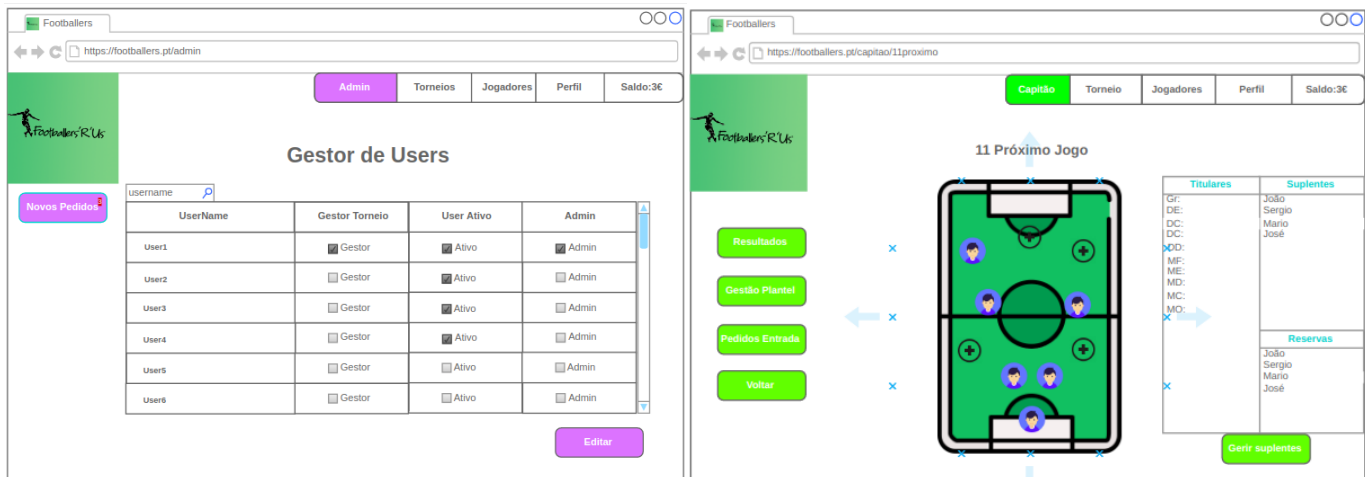


Figura 27: Exemplos da proposta inicial do design.

Na primeira meta o design não foi uma prioridade e, como se pode observar na figura 12, ficou algo confuso. Não existindo uma consistência na paleta de cores utilizadas, nenhum tipo de simetria nem organização dos elementos no ecrã. Com isto em mente, decidimos que para entregar um produto completo iríamos necessitar de melhorar o design.

### ▪ Alterações

Assim, escolhemos uma paleta de cores - branco, verde, cinzento e preto - representativas do futebol. Optámos também por um design minimalista em que temos uma barra de navegação superior, uma zona intermédia onde é mostrada toda a informação relevante e um footer com informações.

### ▪ Design Final

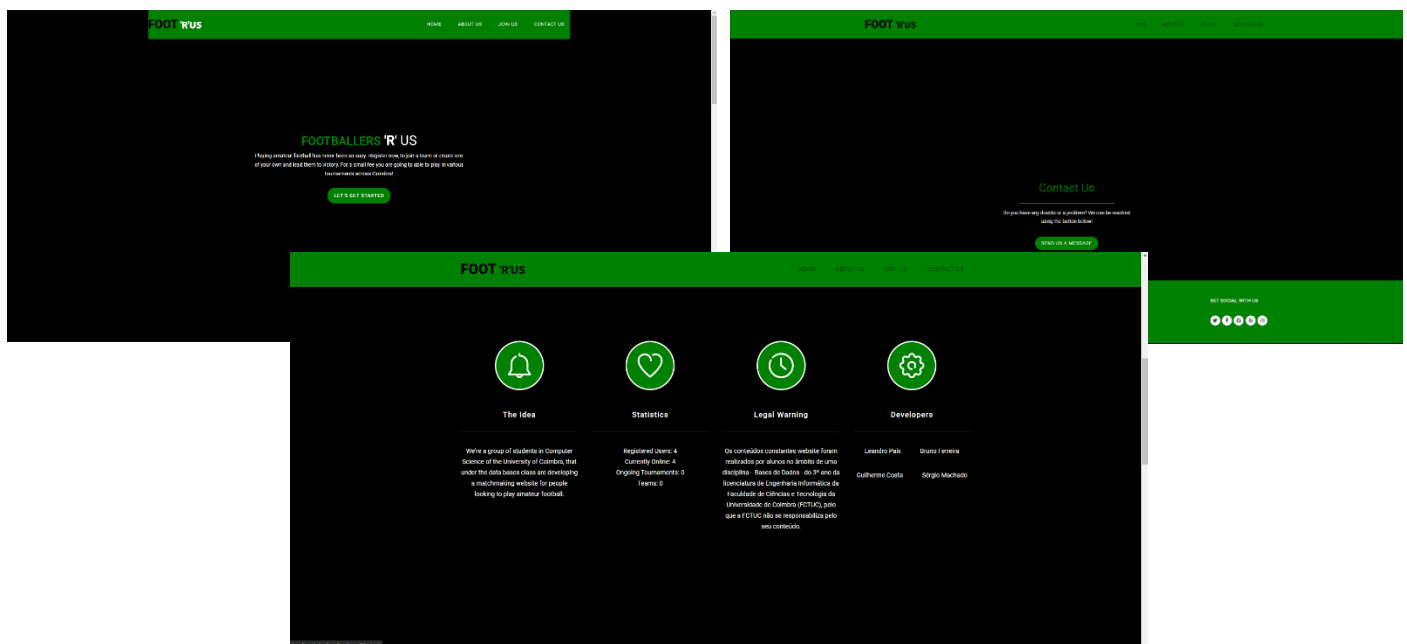


Figura 28: Exemplo do design final.

## • Código Desenvolvido

### ▪ Utilizador logado (Bruno Ferreira)

#### ➤ Resumo:

Quando o utilizador inicia sessão no website Footballers'R'Us, é demonstrado um menu inicial em que o mesmo pode consultar Torneios e Equipas Livres, que de forma rápida, pode entrar num torneio como reserva ou pode pedir para entrar numa equipa.

#### ➤ Home (checkTeam.php, home.php, homeWTeam.php):

Existe a possibilidade de um utilizador ter ou não equipa. Para apresentar uma homepage dependente dessa condição criamos um script (checkTeam.php) que é sempre corrido assim que o utilizador clica na Tab “Home” e esse script redireciona o utilizador para a correta homepage dependendo de ter ou não equipa.

```
// verificar se o user tem equipa ou nao
$sql="SELECT user.username,user.name, player.position,player.teamname
FROM user,player where user.cc=player.user_cc and user_cc=$cc";
$result1 = $conn->query($sql);
$row1=mysqli_fetch_assoc($result1);
// se tiver equipa
if(mysqli_num_rows($result1) > 0){
    header('Location: ../frontEnd/homeWTeam.php');
}
// se nao tiver equipa
else{
    header('Location: ../frontEnd/home.php');
}
```

Figura 29: Verificação inicial da equipa do utilizador.

```
<div class="col-lg-4">
<div class="text-content">
<h3> As Tuas Equipas </h3>
<br>
<table class="table">
<tr>
<th>Equipas</th>
<th>Posição</th>
<th>Faltas Dadas</th>
<th>Saldo</th>
<th>Ver Equipa</th>
</tr>
</table>
<?php
$sql="SELECT user.username,user.name, player.position,player.gamesmissed,player.teamname,player.
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
echo "<tr>";
$url="./equipaPlay.php?t_name=".$row['teamname']."&torneio_id=".$row['tournament_tournam
$ssaldo=$row['balance'];
if($ssaldo=NULL){
$ssaldo=0;
}
echo "<td>".$row['teamname']."</td>.<td>".$row['position']."</td>.<td>".$row['gamesm
echo "</tr>";
}
}
?>
</table>
</div>
</div>
```

Figura 30: Criar uma tabela presente numa das 3 colunas.

A homepage, independentemente de ser de um utilizador com equipa ou sem equipa, é uma página dividida em 3 colunas em que em cada se representa a informação descrita no Ponto de vista de Utilizador. Cada tabela é demonstrada a partir de resultados obtidos através de comandos SQL sobre a base de dados.

É uma página constituída por uma única coluna centrada na página em que mostra ao utilizador uma simples tabela de jogos a que o utilizador foi escolhido como substituto de outro utilizador. Simples Select para obter todas as informações relativas ao jogo em que o cc do substituto está presente na tabela “substitute”.

#### ➤ As tuas Substituições (paginaSubstitutos.php):

```
<div class="parallax-content baner-content">
<div class="container">
<div>
<div class="text-content" style="position:relative; bottom:100px;important">
<h2>
<em>Pedidos de Substituição</em>
</h2>
<div>
<div class="col-lg-6" style="position:relative; left:20%;important">
<div class="text-content">
<br>
<table class="table">
<tr>
<th>Jogo</th>
<th>Data</th>
<th>Hora</th>
<th>Local</th>
</tr>
</table>
<?php
$sql="SELECT game.teamone,game.teamtwo,game.gamedate,game.starttime,game.gamefield from game,sub
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
echo "<tr>";
echo "<td>".$row['teamone']." Vs ".$row['teamtwo']."</td>.<td>".$row['gamedate']."</td>
echo "</tr>";
}
}
?>
</table>
<button class="button" style="position:relative; left:85%; onlick="voltar();" ><b>Voltar</b></button>
</div>
</div>
```

Figura 31: Substituições.

## ➤ As suas Equipas (equipaPlay.php):

Novamente uma página dividida em 3 colunas em que na coluna “Titulares” é apresentada uma tabela como nome de jogador e a respetiva posição do mesmo no plantel. A coluna “Suplentes” é também uma tabela em que são apresentados os suplentes da equipa. A tabela “Gerir disponibilidade”

apresenta ao utilizador todos os

jogos da equipa, informações do jogo e respetiva disponibilidade quanto à ida ao jogo. A tabela tem como última coluna um botão que redireciona para a página “disponibilidade.php” e envia como parâmetro o nome da equipa (PK das equipas), o id do jogo e o id correspondente à disponibilidade do player para aquele jogo, tabela “playergameavailable”.

É necessária a verificação se existe na tabela “playergameavailable” o player e o respetivo jogo pois só assim é possível demonstrar na tabela “Gerir Disponibilidade ” a disponibilidade do player para cada jogo. Se não existir na tabela “playergameavailable” o player e o respetivo jogo, criar-se anteriormente ao display da tabela “Gerir disponibilidade”.

```
<tr>
<th>Jogo</th>
<th>Data</th>
<th>Hora</th>
<th>Disponibilidade</th>
<th>Gerir</th>
</tr>

</tr>
<?php
// verificar se cada jogo tem uma tabela disponibilidade de cada player
// SELECT game.gameid from game, playergameavailable WHERE game.tournamentid=1 and playergameavailable.game_gameid=game.gameid and playergameavailable.user_id=1
$query="SELECT game.gameid from game where game.gameid not in (SELECT playergameavailable.game_gameid from playergameavailable WHERE playergameavailable.user_id=1)";
$result = mysqli_query($conn, $query);
if(mysqli_num_rows($result) > 0){
// adicionar a tabela playergameavailable para cada jogo do jogador
while($row = mysqli_fetch_assoc($result)){
    $gameId=$row['gameid'];
    $query1="INSERT INTO `playergameavailable` (`availableid`,`isavailable`,`user_cc`,`game_gameid`) VALUES (NULL, NULL, '$cc', '$gameId')";
    if ($conn->query($query1)) {
        $_SESSION['registration_success'] = true;
        $_SESSION['loggedin'] = true;
    } else {
        throw new Exception($conn->error);
    }
}
}

// display dos jogos da equipa
$query="SELECT game.gameid,game.gamedate,game.teamone,game.teamtwo, game.starttime,playergameavailable.isavailable,player.position,playergameavailable.availableid from game,playergameavailable WHERE game.tournamentid=1 and playergameavailable.game_gameid=game.gameid";
$result = mysqli_query($conn, $query);
if (mysqli_num_rows($result) > 0) {
// output data of each row
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    $url="./disponibilidade.php?t_name=".$teamname."&game_id=".$row["gameid"]."&available_id=".$row["availableid"];
    // echo $url;
    if($row['isavailable']==1){
        $disponibilidade="Vou";
    }
    else if($row['isavailable']==NULL){
        $disponibilidade="Sem Resposta";
    }
    else{
        $disponibilidade="Não Vou";
    }
    echo "<td>".$row["teamone"]." vs ".$row["teamtwo"]."</td>.<td>".$row["gamedate"]."</td>.<td>".$row["starttime"]."</td>.<td>".$disponibilidade."</td>";
    // echo "<td>".$row["tournamentname"]."</td>.<td>".$row["freespots"]."</td>.<td>".$row["currentteams"]."</td>.<td>".$row["gameweekday"]."</td>";
    echo "</tr>";
}
}
```

Figura 32: Código.

➤ Disponibilidade (disponibilidade.php, substituto.php, substitutoOut.php):

Sequência de páginas que têm de garantir que a disponibilidade que o user seleciona é adicionada na tabela “playergameavailable”, o número de faltas que o utilizador dá aumenta ou diminui conforme ele diz que não vai ou vai (supondo que primeiramente ele selecionou a opção “Não vou” ).

São apresentadas 2 tabelas com checkboxes em que o utilizador pode escolher uma reserva do torneio ou um utilizador do website. É necessário garantir que seleciona só uma checkbox de só uma tabela e que esse jogador selecionado não seja jogador de outra equipa que joga no torneio.

Foi adicionada a possibilidade de o utilizador adicionar uma pessoa de fora do website para fazer a substituição. Basta o utilizador completar o devido form.

É necessário confirmar se o utilizador já fez outro pedido de substituição (de um utilizador do website ou não), pois não faria sentido o jogador ter 2 pedidos de substituição para o mesmo jogo, assim, é eliminado o primeiro pedido de substituição criado.

```
// verificar se clicou num jogador das tabelas
else if(isset($_POST['validar2'])){
    //verificacoes nome
    $flagEverythingOk = true;

    // verificar se as duas tabelas estão selecionadas
    if(isset($_POST['escolha1']) && isset($_POST['escolha2'])){
        $flagEverythingOk=false;

        // echo $_POST['escolha'];
        $escolha=$_POST['escolha1[0]'];
        // $colunas=mysqli_num_rows($escolha);
        echo "<script> var r=confirm('$escolha') </script>";
    }

    // verificar se não selecionou mais que uma checkbox
    if(isset($_POST['escolha1'])){
        $count1=count($_POST['escolha1']);
        if($count1>1){
            $flagEverythingOk=false;
        }
    }

    // verificar se não selecionou mais que uma checkbox
    if(isset($_POST['escolha2'])){
        $count2=count($_POST['escolha2']);
        if($count2>1){
            $flagEverythingOk=false;
        }
    }

    // se já só estiver uma checkbox selecionada
    // verificar se o user escolhido não está inscrito em nenhuma equipa do torneio
    if(isset($_POST['escolha1'])){
        foreach($_POST['escolha1'] as $escolha){
            $sql=" SELECT player.user_cc, player.teamname
            from player, captain_team
            where player.user_cc=$escolha
            and player.teamname in (SELECT captain_team.team_teamname from captain_team WHERE
            $result = mysqli_query($conn, $sql);
            if (mysqli_num_rows($result) > 0) {
                $flagEverythingOk=false;
                echo "<script> alert('Erro!O jogador que selecionou é jogador por outra equi";
            }
        }
    }
}
```

Figura 33: Código.

## ➤ Informações jogadores e Torneios (jogadores.php e torneios.php):

Duas páginas que são acedidas através da Tab e que têm somente uma coluna centrada na página. Apresentam, respetivamente, (i) todos os utilizadores das aplicações com os seus devidos dados (username, nome, contacto e email) e (ii) todos os torneios com as suas devidas informações (equipas inscritas, entre que datas ocorrem, dias da semana a que são os jogos e as respetivas horas).

Na tabela de torneios é também adicionado no fim de cada linha um botão que redireciona para a página específica do torneio (torneioPage.php) em que passa como variável o id do torneio.

```
// se já se estiver uma checkbox selecionada utilizar o seu value
// verificar se o user escolhido não está inscrito em nenhuma equipa do torneio
if(isset($_POST['escolha2'])) {
    foreach($_POST['escolha2'] as $escolha) {
        $sql = "SELECT player.user_cc, player.teamname
        from player, captain_team
        where player.user_cc=$escolha
        and player.teamname in (SELECT captain_team.team_teamname from captain_team WHERE captain_team.team_teamname=$escolha)";
        $result = mysqli_query($conn, $sql);
        if (mysqli_num_rows($result) > 0) {
            $flagEverythingOk=false;
            echo "<script> alert('Erro! O jogador que selecionou é jogador por outra equipa neste torneio!')";
        }
    }
}

// submit para DataBase
if($flagEverythingOk=true){

    // se ele já tiver um substituto, apagar esse e inserir o novo
    $sql="SELECT * FROM 'substitute' WHERE user_cc=$cc and game_gameid=$gameid";
    $result = mysqli_query($conn, $sql);
    if (mysqli_num_rows($result) > 0) {
        // delete da substituição relativa ao jogo
        // DELETE FROM 'substitute' WHERE 'user_cc'='123456789 and game_gameid=1
        $query1="DELETE FROM 'substitute' WHERE 'user_cc'=$cc and game_gameid=$gameid";
        if ($conn->query($query1)) {
            $SESSION['registration_success'] = true;
            $SESSION['logged_in'] = true;
        } else {
            throw new Exception($conn->error);
        }
    }
    // tirar a falta para no algoritmo seguinte adicionar
    $query10="UPDATE player set player.gamesmissed=player.gamesmissed+1 where player.user_cc=$cc and game_gameid=$gameid";
    if ($conn->query($query10)) {
        $SESSION['registration_success'] = true;
        $SESSION['logged_in'] = true;
    } else {
        throw new Exception($conn->error);
    }
}
```

Figura 34: Código.

```
<div classe="text-content">
    <h2>
        <em>Torneios Livres</em>
    </h2>
    <table class="table">
        <tr>
            <th>Torneios</th>
            <th>Vagas</th>
            <th>Equipas Inscritas</th>
            <th>Dias de Jogo</th>
            <th>Horas de Jogo</th>
            <th>Data de Inicio</th>
            <th>Data de Fim</th>
            <th>Ver Torneio</th>
        </tr>
        <tr>
            <td colspan="8">
                // $sql = "SELECT username, email, contact_name FROM 'user' ORDER BY 1";
                // $sql="SELECT 'tournamentid', 'tournamentname', 'freespots', 'currentteams', 'startdate', 'enddate' FROM 'tournament' Order";
                $sql="SELECT tournament.tournamentid, tournament.tournamentname, tournament.freespots, tournament.currentteams, tournament.startdate, tournament.enddate FROM 'tournament'";
                $result = mysqli_query($conn, $sql);
                if (mysqli_num_rows($result) > 0) {
                    // define data of each row
                    while($row = mysqli_fetch_assoc($result)) {
                        echo "<tr>";
                        $url="torneioPage.php?torneio_id=".$row['tournamentid'];
                        // echo $url;
                        echo "<td>".$row['tournamentname']. "</td>". "<td>".$row['freespots']. "</td>". "<td>".$row['currentteams']. "</td>". "<td>".$row['startdate']. "</td>". "<td>".$row['enddate']. "</td>". "<td>".$row['url']. "</td>";
                        echo "</tr>";
                    }
                }
            </td>
        </tr>
    </table>
</div>
```

Figura 35: Código.



- Informações, Equipas e Inscrições (torneioPage.php, entrarEquipa.php, criarEquipa.php, resultadosTorneio.php, marcadoresTorneio.php, plantel.php, inscreverReserva.php):

Esta página é um dos blocos centrais do website, é a página que permite ao utilizador entrar e criar equipas, ver resultados, melhores marcadores e inscrever-se como reserva dos torneios.

A página é dividida em 3 colunas, (i) classificações, (ii) informações torneio e (iii) reservas torneio e do lado esquerdo da página estão 5 botões, “Entrar Equipa”, “Criar Equipa”, “Resultados Torneio”, “Melhores Marcadores” e “Reservas Torneio”.

A tabela classificações mostra por ordem decrescente as equipas do torneio tendo em conta os pontos que têm. É também mostrado quantos jogos cada equipa já realizou, o número de vitórias, empates e derrotas. É também colocado um botão “Plantel” que redireciona para a página “plantel.php” enviando como parâmetros o nome da equipa e o id do torneio.

A página “plantel.php” são duas simples tabelas já mencionadas em que uma mostra as posições dos titulares e a segunda tabela os suplentes.

A página “entrarEquipa.php” é uma janela popup que aparece quando o utilizador pressiona o botão “Então Equipa” na página “torneioPage.php”. A página é constituída por um form em que os inputs são options. A página tem 2 inputs, a equipa que pretende juntar-se, as equipas que são mostradas pertencem ao torneio e já foram aprovadas pelo gestor, e a posição que pretender ocupar na equipa, Gr, defesa, médio e avançado.

```
// verificar se o user já é player numa equipa do torneio
$sql = "SELECT player.user_cc, player.teamname
from player, captain_team
where player.user_cc=$cc
and player.teamname in (SELECT captain_team.team_teamname from captain_team WHERE captain_team.tournament_tournamentid=$torId)";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    $flagEverythingOk=false;
    echo "<script> alert('Erro!Já é jogador por outra equipa!'); </script>";
}

// verificar se o user já é reserva do torneio
$sql10 = "SELECT reservastorneio.user_cc from reservastorneio
WHERE reservastorneio.tournament_tournamentid=$torId
and reservastorneio.user_cc=$cc";
$result10 = mysqli_query($conn, $sql10);
if (mysqli_num_rows($result10) > 0) {
    $flagEverythingOk=false;
    echo "<script> alert('Erro!Já é reserva do torneio!'); </script>";
}

$posicaoEquipa=$_POST['posicao'];

// submit para DataBase
if($flagEverythingOk=true){
    $sql = "SELECT username FROM 'user' where cc=$cc";
    $result = mysqli_query($conn, $sql);
    $user=mysqli_fetch_assoc($result);
    $username=$user['username'];

    $query = "INSERT INTO 'request_team' ('reqid', 'team_name', 'user_name', 'accepted', 'pos', 'user_cc') VALUES (NULL, '$nomeEquipa', '$username', 'accepted', '$pos', '$cc')";
    if ($conn->query($query)) {
        $_SESSION['registration_success'] = true;
        $_SESSION['logged_in'] = true;
        // $url="/torneioPage.php?torneio_id=".$torId;
        // header('Location:'.$url);
        echo "<script>window.close();</script>";
    }
}
```

Figura 37: Código.

```
<div class="col-lg-6" style="position:relative; right:10%">
<div class="text-content">
<h3> Classificações </h3>
<br>
<table class="table">
<tr>
<th>Equipas</th>
<th>Jogos</th>
<th>Pontos</th>
<th>Vitórias</th>
<th>Empates</th>
<th>Derrotas</th>
<th>Ver Plantel</th>
</tr>
</table>
</div>
```

Figura 36: Código.

É necessário verificar se o user já não está numa outra equipa do torneio ou se não é reserva de um torneio.

A página “criarEquipa.php” é igualmente uma janela popup em que existe um form com só um input, o nome da equipa. É necessário verificar que o utilizador não faz parte de uma outra equipa do torneio ou reserva do torneio, e que não existe nenhuma equipa com o mesmo nome dentro do torneio.

Assim que é criada a equipa é também criado um request com valor de “accepted” a NULL. Este “accepted” será alterado para 1 ou 0 consoante a permissão do gestor do torneio que equipa participe ou não no torneio.

```
<div class="text-content" style="position:relative; bottom:150px; !important">
  <h2>
    <em>Criar Equipa</em>
  </h2>
</div>
<div>
  <p>Criar Equipa para o Torneio</p>
  <p><font size = "5"> <b>
    <?php
      echo $torneio["tournamentname"];
    ?>
  </font></b></p>
  <p>Atenção!!<br>Ao Criar um equipa está-se a tornar capitão da mesma!</p>
  <form action="" method="post">
    <input name="equipaName" type="text" value="Nome Equipa">
    <br>
    <p><b>Escolhe uma Posição</b></p>
    <select class="form-control" style="position:relative; left:26%;max-width:50%;" name="posicao">
      <option class="btn" value="Avançado">Avançado</option>
      <option class="btn" value="Medio">Médio</option>
      <option class="btn" value="Defesa">Defesa</option>
      <option class="btn" value="Gr">Guarda-Redes</option>
    </select>
    <br>
    <button name="validar1" class="button" type="submit" style="position:relative; background:red; !i
    <button name="validar2" class="button" type="submit" style="position:relative; display:inline; le
```

Figura 38: Código.

A última página dependente da página “torneioPage.php” é a página “inscreverReserva.php”.

Novamente uma página com uma só coluna só com 3 opções de botões, (i) “Cancelar”, (ii) “Inscrever” e (iii) “Desinscrever”.

São 2 simples opções, mas que é necessário verificar se o utilizador já é reserva do torneio, se já uma equipa ao torneio e total de atingido necessário a mais player).

As páginas “resultadosTorneio.php” e “marcadoresTorneio.php” são duas simples páginas com uma só coluna em que mostram, respetivamente, os resultados do torneio (equipa1 R1-R2 equipa2) e os melhores marcadores dos torneios, ordem decrescente tendo em conta o número de golos.

```
<div class="col-lg-6" style="position:relative; left:26%; !important">
  <div class="text-content">
    <h3><b>Melhores Marcadores</b></h3>
    <br>
    <table class="table">
      <tr>
        <th>Username</th>
        <th>Nome</th>
        <th>Equipa</th>
        <th>Golos Marcados</th>
      </tr>
      <tr>
        <td colspan="4">
          <?php
            $sql="SELECT player.goalsscored,player.teamname,user.username,user.name from player, user,captain_team where pl
            $result = mysqli_query($conn, $sql);
            if (mysqli_num_rows($result) > 0) {
              // output data of each row
              while($row = mysqli_fetch_assoc($result)) {
                echo "<tr>";
                echo "<td>".$row["username"]."</td>".$row["name"]."</td>".$row["teamname"]."</td>".$row["goalsscored"]."</td>";
                echo "</tr>";
              }
            }
          ?>
        </td>
      </tr>
    </table>
    <button class="button" style="position:relative; left:85%;" onclick="voltar();" ><b>Voltar</b></button>
  </div>
</div>
```

Figura 39: Código.

```
// verificar se o user já é player numa equipa do torneio
$sql=" SELECT player.user_cc, player.teamname
from player, captain_team
where player.user_cc=$cc
and player.teamname in (SELECT captain_team.team_teamname from captain_team WHERE captain_team.tournament_tournamentid=$torId);";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
  $flagEverythingOk=false;
  // echo "<script> alert('Erro!Já é jogador por uma equipa!'); </script>";
  echo "<script> var r=confirm('Erro!Já é jogador por uma equipa!');";
  echo "if(r==true){";
  echo "window.location.replace('./torneioPage.php?torneio_id=$torId');";
  echo "};";
  echo "else{";
  echo "window.location.replace('./torneioPage.php?torneio_id=$torId');";
  echo "};";
  echo "</script>";
}

// ver se o user já é reserva do torneio
$sql1="SELECT COUNT(user_cc) as count1 from reservastorneio WHERE reservastorneio.user_cc=$cc and reservastorneio.tournament_tournamentid=$torId";
$result0 = mysqli_query($conn, $sql1);
$aux0=mysqli_fetch_assoc($result0);
$aux1=$aux0['count1'];
if($aux1>0){
  $flagEverythingOk=false;
  // echo "<script> alert('Já é Reserva do Torneio!'); </script>";
  echo "<script> var r=confirm('Já é Reserva do Torneio!');";
  echo "if(r==true){";
  echo "window.location.replace('./torneioPage.php?torneio_id=$torId');";
  echo "};";
  echo "else{";
  echo "window.location.replace('./torneioPage.php?torneio_id=$torId');";
  echo "};";
  echo "</script>";
  // echo "<script>window.close();</script>";
  // header('Location: ./torneioPage.php?torneio_id='.$torId);
}

// se o numero de reservas de um torneio já atingiu o limite
if($numRes>4){
  $flagEverythingOk=false;
  echo "<script> var r=confirm('0 Número total de Reservas foi atingido!');";
  echo "if(r==true){";
  echo "window.location.replace('./torneioPage.php?torneio_id=$torId');";
  echo "};";
  echo "else{";
  echo "window.location.replace('./torneioPage.php?torneio_id=$torId');";
  echo "};";
  echo "</script>";
}
```

Figura 40: Código.

faz parte de que pertença se o número reservas foi (não sendo inscrição de nenhum



## ➤ Editar Perfil (editPerfil.php)

A página “editPerfil.php” é uma página com só uma coluna, centrada ao centro do ecrã, constituída por um form. O form tem como valores o nome, contacto, email, password e verificação de password do utilizador. O form mostra ao utilizador as suas informações, sendo só necessário que o utilizador as altere, se assim entender.

O form é sucedido de verificações, tais como, o email tem de ter a forma de um email (FILTER\_SANITIZE\_EMAIL), as passwords inseridas são iguais, etc.

```
//verificacoes contacto
if ((strlen($contacto) != 9)) {
    $flagEverythingOk = false;
    $_SESSION['e_phonenumber'] = "Phone number must be nine digits!";
    echo "<script> alert('Phone number must be nine digits!'); </script>";
}

if (is_numeric($contacto) == false) {
    $flagEverythingOk = false;
    $_SESSION['e_phonenumber'] = "Phone number has to consist of only numbers!";
    echo "<script> alert('Phone number has to consist of only numbers!'); </script>";
}

//verificacoes email
$emailB = filter_var($email, FILTER_SANITIZE_EMAIL);
if ((filter_var($emailB, FILTER_VALIDATE_EMAIL) == false) || ($emailB != $email)) {
    $flagEverythingOk = false;
    $_SESSION['e_email'] = "Incorrect email address!";
    echo "<script> alert('Incorrect email address!'); </script>";
}

$query2 = "SELECT email FROM user WHERE email LIKE '$email'";
$result = $conn->query($query2);
$count_emails = $result->num_rows;
if ($count_emails > 1) {
    $flagEverythingOk = false;
    $_SESSION['e_email'] = "Account with this email already exists!";
    echo "<script> alert('Account with this email already exists!'); </script>";
}

//verificacoes password
if ((strlen($pass) < 4) || (strlen($pass) > 20)) {
    $flagEverythingOk = false;
    $_SESSION['e_password'] = "Password under 4 or over 20 characters!";
    echo "<script> alert('Password under 4 or over 20 characters!'); </script>";
}

if ($pass != $repass) {
    $flagEverythingOk = false;
    $_SESSION['e_pass'] = "Passwords are different!";
    echo "<script> alert('Passwords are different!'); </script>";
}
```

Figura 41: Código.



➤ Gestão de duplicados na gestão do plantel

➤ Addresult:

Ecrã com resultado por definir com a possibilidade de adicionar as faltas e golos da sua equipa onde também aparecem as faltas e golos já adicionados.

➤ deleteplayer.php:

Página não visível que elimina um determinado jogador da equipa.

➤ faltas.php:

Ecrã que adiciona faltas a um determinado jogo onde só são apresentados os jogadores da sua equipa

➤ golos.php:

Ecrã que adiciona golos a um determinado jogo onde só são apresentados os jogadores da sua equipa

➤ newcapitan.php:

Página não visível que define o novo capitão e atualiza os dados

```
$sql = "SELECT teamname, count(name) as t from user WHERE goal ON user.cc = goals.usercc and goals.game_id = $id GROUP BY name ORDER BY COUNT(name) DESC";
$result = $db->query($sql);
if($result){
    $sum_of_results = mysql_num_rows($result);
    echo "<table id='goals' class='mytable' style='width: 40%;' <!-- style='width:70%;' -->";
    <tr>
        <th style='text-align: center;'>Nome/Tho
        <th style='text-align: center;'>Equipa/Tho
        <th style='text-align: center;'>faltas/Tho
    </tr>

    for ($i = 1; $i <= $sum_of_results; $i++){
        $row = mysql_fetch_assoc($result);
        $player = $row['name'];
        $team = $row['teamname'];
        $goals = $row['t'];
        echo "<tr>
            <td> $player </td>
            <td> $team </td>
            <td> $goals </td>
        </tr>";
    }
    echo "</table>";
}

$sql = "SELECT name, teamname, count(name) as t from user WHERE goal ON user.cc = faults.usercc and faults.game_id = $id GROUP BY name ORDER BY COUNT(name) DESC";
$result = $db->query($sql);
if($result){
    $sum_of_results = mysql_num_rows($result);
    echo "<table id='faults' class='mytable' style='width: 40%;' <!-- style='width:70%;' -->";
    <tr>
        <th style='text-align: center;'>Nome/Tho
        <th style='text-align: center;'>Equipa/Tho
        <th style='text-align: center;'>faltas/Tho
    </tr>

    for ($i = 1; $i <= $sum_of_results; $i++){
        $row = mysql_fetch_assoc($result);
        $player = $row['name'];
        $team = $row['teamname'];
        $faults = $row['t'];
        echo "<tr>
            <td> $player </td>
            <td> $team </td>
            <td> $faults </td>
        </tr>";
    }
    echo "</table>";
}
```

Figura 44: Faltas e golos no result.php.

## ■ Administrador (Leandro Pais)

### ➤ Notificações:

O administrador tem uma funcionalidade onde recebe todo o tipo de notificações que os utilizadores considerem relevantes.

```
<form method="post" action="userAdminNotification.php">
</php
$result = $conn->query("SELECT * FROM notifications WHERE isvisible=1");

$resultsCount = mysqli_num_rows($result);
if ($resultsCount > 0) {
    echo "<h2><b>Notificações</b></h2>";
    <table id="msgTable">
    <tr>
    <th style="display:none;">id</th>
    <th>Date</th>
    <th>Message</th>
    <th>User</th>
    </tr>";
    for ($i = 1; $i <= $resultsCount; $i++) {
        $row = mysqli_fetch_assoc($result);

        $id = $row["id"];
        $date = $row["notdate"];
        $message = $row["message"];
        $user = $row["user_cc"];

        echo "<tr>";
        <td style="display:none;">$id . "</td>";
        <td> $date . "</td>";
        <td> $message . "</td>";
        <td> $user . "</td>";
        <td><form method="post" action="userAdminNotification.php"><button type="submit" name="delete" value=" . $id . " class="button">DELETE</button></form>
        </td>";
        </tr>";
    }
    echo "</table>";
} else {
    echo "<div id=response>No notifications found...</div>";
}
?>
```

Figura 45: Mostrar notificações.

Por exemplo, caso haja um problema com aplicação, o administrador pode resolvê-lo e descartar a notificação. Esta funcionalidade trata-se de uma busca por todas as linhas da tabela “notifications” que têm o atributo “isvisible” a 1 e imprimindo-as numa tabela html. O descartar nada mais é do que um “UPDATE” que coloca o “isvisible” a 0.

```
<?php
$conn = OpenCon();

if (isset($_POST['delete'])) {
    $id = (int) $_POST['delete'];
    $conn->query("UPDATE tournaments SET isvisible=0 WHERE id=$id");
}
```

Figura 46: “Eliminar” da base de dados.

### ➤ Torneios:

O administrador tem acesso a uma página onde pode ver e/ou apagar os torneios a decorrer. De forma semelhante às notificações, esta é constituída por uma tabela html onde são impressos os dados obtidos pela busca SQL. O descartar passa por manipular o atributo “isvisible”.

```
<form method="post" action="userAdminTournamentsInfo.php">
</php
$result = $conn->query("SELECT * FROM tournament WHERE isvisible=1");

$resultsCount = mysqli_num_rows($result);
if ($resultsCount > 0) {
    echo "<h2><b>Torneios</b></h2>";
    <table id="msgTable">
    <tr>
    <th style="display:none;">id</th>
    <th>Name</th>
    <th>Number of teams</th>
    <th>Starting Date</th>
    <th>Creator</th>
    </tr>";
    for ($i = 1; $i <= $resultsCount; $i++) {
        $row = mysqli_fetch_assoc($result);

        $id = $row["tournamentid"];
        $name = $row["tournamentname"];
        $nteam = $row["currentteams"];
        $startdate = $row["startdate"];
        $user_cc = $row["user_cc"];

        echo "<tr>";
        <td style="display:none;">$id . "</td>";
        <td> $name . "</td>";
        <td> $nteam . "</td>";
        <td> $startdate . "</td>";
        <td> $user_cc . "</td>";
        <td><form method="post" action="userAdminTournamentsInfo.php"><button type="submit" name="delete" value=" . $id . " class="button">DELETE</button></form>
        </td>";
        </tr>";
    }
    echo "</table>";
} else {
    echo "<div id=response>No tournaments found...</div>";
}
?>
```

Figura 47: Mostrar torneios.

## ➤ Utilizadores:

O administrador na página dos utilizadores pode, caso assim escolha, alterar as permissões dos mesmo. Novamente, este processo passa por ir buscar todos os utilizadores registados que

“isvisible” a 1, imprimi-los numa tabela html com os respetivos botões. Estes botões atualizam atributos na tabela do utilizador como, “iscaptain”, “isadmin” e “ismanager” para 0 ou 1 consoante o desejado. Para facilitar o trabalho do administrador esta página vem ainda equipada com um botão de pesquisa que permite ao administrador encontrar qualquer utilizador rapidamente utilizando o número do CC (proveniente de

```
if (isset($_POST['search'])) {
    $search = $_POST['search'];
    if ($search != "") {
        $result = $conn->query("SELECT * FROM user WHERE isvisible=1 AND ( cc LIKE '%$search%' OR name LIKE '%$search%' OR email LIKE '%$search%' );");
        $resultsCount = mysqli_num_rows($result);
        if ($resultsCount > 0) {
            echo
            <table id="msgTable">
            <tr>
            <th>Name</th>
            <th>Email</th>
            <th>CC</th>
            <th>Contact</th>
            </tr>';
            for ($i = 1; $i <= $resultsCount; $i++) {
                $row = mysqli_fetch_assoc($result);
                $name = $row['name'];
                $email = $row['email'];
                $cc = $row['cc'];
                $contact = $row['contact'];
                $ismanager = $row['ismanager'];
                $iscaptain = $row['iscaptain'];
                echo '
                <tr>
                <td> . $name . '</td>
                <td> . $email . '</td>
                <td> . $cc . '</td>
                <td> . $contact
                </td>
                <td><form method="post" action="userPlayersInfo.php"><button type="submit" name="delete" value=" . $cc . ' class="button">DELETE</button></form>
                </td>';
                if ($ismanager == 1) {
                    echo ' <td><form method="post" action="userPlayersInfo.php"><button type="submit" name="removeMan" value="
                    . $cc . ' class="button">REMOVE MANAGER</button></form>';
                } else {
                    echo ' <td><form method="post" action="userPlayersInfo.php"><button type="submit" name="allowMan" value="
                    . $cc . ' class="button">ALLOW MANAGER</button></form>';
                }
                if ($iscaptain == 1) {
                    echo ' <td><form method="post" action="userPlayersInfo.php"><button type="submit" name="removeCap" value="
                    . $cc . ' class="button">REMOVE CAPTAIN</button></form>';
                }
                echo ' </td>';
            }
            echo ' </table>';
        }
    }
}
```

Figura 48: Parte do código para a efetuar pesquisa.

têm o

```
$result = $conn->query("SELECT * FROM user WHERE isvisible=1 AND ( cc LIKE '%$search%' OR name LIKE '%$search%' OR email LIKE '%$search%' );");
```

Figura 49: SQL.

uma notificação), o nome ou ainda o email. Como se pode ver na figura X este comando SQL utiliza o “LIKE” para comparar a variável “search” (cujo valor é inserido pelo utilizado) com cada um dos campos já referidos e devolve o correspondente utilizador, caso exista.

## ➤ Log Out:

Com este botão é feito o log out do utilizador, a sessão php é destruída e o atributo “isLoggedIn” é atualizada para 0.

```
<?php
if (array_key_exists('logOut', $_POST)) {
    logOut();
}
?>
<form id="logout" method="post">
    <input type="submit" name="logOut" class="button" value="LOG OUT" />
</form>
```

Figura 50: Botão de log out.

```
//User logout
function logOut()
{
    //Continue sessions
    session_start();
    //Issue destroy command
    session_destroy();
    //Redirect to user not logged in
    header("Location: ../frontEnd/userNL.php");
}
```

Figura 51: Função que efetua o log out.

- Utilizador não logado (Leandro Pais)

➤ **Estatísticas**

O utilizador não logado como não possui qualquer tipo de funcionalidade, o código é concentrado no html e css havendo apenas algumas

qurys para obter dados estatísticos. Estes são obtidos, geralmente, contando o número de linhas devolvidos pela busca geral de uma tabela. No caso particular dos utilizadores logados, a tabela dos users é percorrida e devolve o número de utilizadores onde o atributo “isLoggedIn” se encontra a 1.

➤ Login

```
if (isset($_POST['loginUser'])) {
    $login = $_POST['login'];
    $pass = $_POST['pass'];
    $query = "SELECT * FROM user WHERE username LIKE '$login' AND password LIKE '$pass'";
    $result = $conn->query($query);
    $row = mysqli_fetch_assoc($result);
    if (mysqli_num_rows($result) > 0) {
        $_SESSION['loggedin'] = true;
        $_SESSION['cc'] = $row['cc'];
        header('Location: ../frontEnd/mainNav.php');
    }
}
```

*Figura 53: Verificar o login.*

Figura 53: Verificar o login.

É nesta página também que é efetuado o login, se trata de uma busca pela linha que contém o username e password correspondente.

➤ **Registro**

```

<form method="post" action="/userlib.php">
<div>
<input type="text" placeholder="Username" name="username">
</input>
if (isset($_SESSION['e_login'])) {
    echo <div class="error">$_SESSION['e_login'] . '</div>;
unset($_SESSION['e_login']);
}
</if>
</div>
<div>
<input type="text" placeholder="Email" name="email">
</input>
if (isset($_SESSION['e_email'])) {
    echo <div class="error">$_SESSION['e_email'] . '</div>;
unset($_SESSION['e_email']);
}
</if>
</div>
<div>
<input type="text" placeholder="CC" name="cc">
</input>
if (isset($_SESSION['e_cc'])) {
    echo <div class="error">$_SESSION['e_cc'] . '</div>;
unset($_SESSION['e_cc']);
}
</if>
</div>
<div>
<input type="password" placeholder="Password" name="password">
</input>
if (isset($_SESSION['e_password'])) {
    echo <div class="error">$_SESSION['e_password'] . '</div>;
unset($_SESSION['e_password']);
}
</if>
</div>
<div>
<input type="password" placeholder="Repeat Password" name="passwordRepeat">
</input>
if (isset($_SESSION['e_pass'])) {
    echo <div class="error">$_SESSION['e_pass'] . '</div>;
unset($_SESSION['e_pass']);
}
</if>
</div>
<div>
<input type="text" placeholder="Name" name="name">
</input>
if (isset($_SESSION['e_name'])) {
    echo <div class="error">$_SESSION['e_name'] . '</div>;
unset($_SESSION['e_name']);
}
</if>
</div>

```

Figura 54: Mostrar erros cliente-side.

O registo trata-se de um form que recebe os dados necessário de um utilizador, verifica se estão de acordos com as normas estabelecidas para o site e, caso não existe nenhum utilizador com informações iguais, faz o devido registo inserindo os dados na tabela correspondente.

[illegible]

Figura 55: Gerar erros server-side.



- Gestor de torneio (Sérgio Machado)

➤ Página Inicial

Na página inicial são apresentados todos os torneios. Para isso é feita uma query (esta busca todos os torneios criados por aquele gestor ou torneios partilhados por outros gestores) à database e posteriormente os dados são apresentados no ecrã

```
$result = $connection->query( "SELECT tournamentid, user_cc,creationdate, tournamentname, creationdate,freespots,
                             startdate, enddate, alreadystarted from tournament where user_cc=$quick_cast or shared = '1' ORDER BY startdate DESC");
$num_of_results = mysqli_num_rows($result);
```

Figura 56: Query para buscar os torneios do gestor.

Figura 58: Print dos torneios no ecrã

estiver correto.

```
<script>
    function deleteSubjects(subjectId){
        window.location.href = 'gerirTorneio.php?edit_id='+ subjectId + '';
        console.log(subjectId);
    }

    function tournamentDetails(subjectId){
        window.location.href = 'seeDetails.php?edit_id='+ subjectId + '';
        console.log(subjectId);
    }
</script>
```

Figura 57: Javascript com a ligação dos botões detalhes e gerir

➤ Criar Torneio

O código desta página é constituído pelo html que apresenta os inputs e botões para o gestor inserir os dados e pelo php que espera que o botão adicionar sejam clicados, verifica e adiciona a database se tudo

[illegible]

Figura 59: Exemplo de verificação dos dados inseridos

```
$query = "INSERT INTO tournament (tournamentname, classificationtable, creationdate, startdate, enddate, maxteams,
    currentteams, freeslots, user_cc) VALUES ('$nome_torneio', null, CURRENT_TIMESTAMP, '$start_date', '$end_date',
    $number_max, 0, $number_max, $quick_cast)";
mysqli_query($db, $query);
```

Figura 60: SQL para criar torneio

```
if (count($errors) == 0) {
    $quick_cast = ($SESSION['cc']);
    $quick_cast2 = $SESSION['last_created_tournament'];
    $query = "INSERT INTO tournamentdetails (gameweekday, gamestarttime, gamefield, gameendtime,
    tournament_tournamentid) VALUES ('$dia_semana', $primeira_hora, '$nome_campo', $segunda_hora, $quick_cast2)";
    $conn = mysqli_query($db, $query);
}
```

Figured to run this SQL and provide advanced code assistance. Disable this inspection via problem menu (Alt+Enter).

+Shift+Enter More actions Alt+Enter

Figura 61: SQ para criar horário de jogo

```
console_log($quick_cast2);
$query = "INSERT INTO dates (nocontestdate, tournament_tournamentid) VALUES ('$noPlay',$quick_cast2)";
$res = mysqli_query($db, $query);
```

Figura 62: SQL para inserir data em que não pode haver jogo

## ➤ Detalhes

Busca à base de dados os possíveis horários dos jogos, as datas e dá ao gestor a possibilidade de apagar os mesmos.

```
$result = $connection->query( query: "SELECT detailsid, gameweekday, gamestarttime, gamefield, gameendtime  
FROM tournamentdetails WHERE tournament_tournamentid = $torId;");  
$num_of_results = mysqli_num_rows($result);
```

Figura 63: Comando SQL para buscar os possíveis horários daquele torneio

```
$result = $connection->query( query: "SELECT nocontestdate from dates WHERE tournament_tournamentid = $torId;");  
$num_of_results = mysqli_num_rows($result);
```

Figura 64: Comando SQL para buscar as datas em que não podes haver jogos

```
for ($i = 1; $i <= $num_of_results; $i++)  
{  
  
    $row = mysqli_fetch_assoc($result);  
  
    $detailsid = $row['detailsid'];  
    $gameweekday = $row['gameweekday'];  
    $gamestarttime = $row['gamestarttime'];  
    $gameendtime = $row['gameendtime'];  
    $gamefield = $row['gamefield'];  
  
    echo  
        '<tr>  
        <td>'. $gameweekday. '</td>  
        <td>'. $gamestarttime. '</td>  
        <td>'. $gameendtime. '</td>  
        <td>'. $gamefield. '</td>  
        <form method="post">  
        <td><input name="ajax" id="apagar1" class="bbt"  
        </form>  
        </tr>';  
}
```

Figura 65: Print dos horários do torneio

```
for ($i = 1; $i <= $num_of_results; $i++)  
{  
  
    $row = mysqli_fetch_assoc($result);  
  
    $nocontestdate = $row['nocontestdate'];  
  
    echo  
        '<tr>  
        <td>'. $nocontestdate. '</td>  
        <form method="post">  
        <td><input name="ajax2" id="apagar1"  
        </form>  
        </tr>';  
}  
  
echo '</table>';  
e{
```

Figura 66: Print das datas sem jogos

```
if(isset($_POST['ajax']) and isset($detailsid)) {  
    $db = OpenCon();  
    $query = "DELETE FROM tournamentdetails where detailsid = $detailsid";  
    $result = mysqli_query($db, $query);  
}
```

Figura 67: Apagar horários

```
if(isset($_POST['ajax2']) and isset($nocontestdate)) {  
    $db = OpenCon();  
    $qiko = $_SESSION['editTourId'];  
    $query = "DELETE FROM dates where nocontestdate = $nocontestdate and tournament_tournamentid=$qiko";  
    $result = mysqli_query($db, $query);  
}
```

Figura 68: Apagar datas em que não pode haver jogos



➤ Gestão de torneio:

**Adicionar detalhes:**

-O código é semelhante ao presente na página de criar torneio.

**Dar permissões:**

-Verifica se foi aquele gestor a criar o torneio e se foi este pode partilhar as permissões de gestão de torneio com outros gestores ou retirar as mesmas.

```
$db = OpenCon();  
$query = "UPDATE tournament SET shared = '1' where tournamentid=$quick";  
$resp = mysqli_query($db, $query);  
console_log($resp);  
  
echo '<p style="color:green;">Permissões a outros gestores dada com sucesso!</p>';
```

Figura 69: Dar permissões de gerir o torneio a outros gestores

```
$resp = mysqli_query($db, $query);  
console_log($resp);  
echo '<p style="color:green;">Permissões a outros gestores retirada com sucesso!</p>';
```

Figura 70: Retirar permissões de gerir o torneio a outros gestores

**Confirmar resultados:**

-Para cada jogo torneio busca os jogos que já tem o resultado inserido por parte dos capitães das duas equipas. Se os resultados foram diferentes cabe ao gestor averiguar qual o verdadeiro resultado e inseri-lo.

```
$result = $connection->query('query: "SELECT gameid, teamone, teamtwo, goalsteam1cap1, goalsteam1cap2, goalsteam2cap1,  
goalsteam2cap2, goalsteam1manager, goalsteam2manager from game WHERE goalsteam1cap1 is not null and goalsteam1cap2  
is not null and goalsteam2cap1 is not null and goalsteam2cap2 is not null and goalsteam1cap1!=goalsteam2cap1 and  
goalsteam1cap1=goalsteam2cap1 and goalsteam1manager is null and goalsteam2manager is null");  
$num of results = mysqli_num_rows($result);
```

Página 16: Busca os jogos do torneio em que os resultados não coincidem

```
(isset($_POST['lale']) and isset($_POST['re1']) and isset($_POST['re2'])) {  
    if( $_POST['re1']>=0 and $_POST['re2']>=0) {  
        $_SESSION['tutu1'] = $_POST['re1'];  
        $_SESSION['tutu2'] = $_POST['re2'];  
        console_log($_SESSION['tutu1']);  
        console_log($_SESSION['tutu2']);  
        $query = "UPDATE game SET check_game='0' WHERE gameid=$gameid";  
        mysqli_query($connection, $query);  
        header( string: 'Location: updateResult.php?upt_id='.$gameid.'');  
    }  
}
```

Página 17: Inserção dos resultados corretos por parte do gestor

## Gerar os jogos:

-Provavelmente será a funcionalidade mais complexa do gestor. Começa-se por buscar o número de equipas no torneio e os seus nomes. Se as equipas forem um número ímpar é adicionada uma equipa fantasma para que o algoritmo de criação de jogos seja mais eficaz. Além disso também se busca os dias e horas em que pode haver jogos bem como as datas em que não pode haver.

```
$basicQ = $db->query( query: "SELECT * FROM game WHERE game.tournament tournamentid=$quicko");
$basicResults = mysqli_num_rows($basicQ);

$result0 = $db->query( query: "SELECT teamname FROM request WHERE accepted='1' and tournament tournamentid=$quicko");
$num_of_results0 = mysqli_num_rows($result0);

$result = $db->query( query: "select currentteams, enddate, startdate from tournament where tournamentid=$quicko");
$num_of_results = mysqli_num_rows($result);
```

Página

Figura 71: Busca o número das equipas e os seus nomes

```
$result2 = $db->query( query: "SELECT tournamentdetails.gameweekday from tournamentdetails WHERE
tournamentdetails.tournament tournamentid=$quicko");
$numberOfDays = mysqli_num_rows($result2);

$resultStart = $db->query( query: "SELECT tournamentdetails.gamestarttime from tournamentdetails WHERE
tournamentdetails.tournament tournamentid=$quicko");
$numberOfStart = mysqli_num_rows($resultStart);
```

Figura 72: Busca os dias em que há jogos e a data de início e fim no torneio

-Depois disto, verifica se existe pelo menos duas equipas inscritas no torneio e se com os dias de jogos existentes da para realizar pelo menos uma “1ª mão”. Se isto não se verificar então o gestor não poderá gerar jogos e terá de adicionar mais horários para haver jogos.

```
if($teamsNumber < 2){
    echo '<p style="color: red">É necessário ter pelo menos duas equipas inscritas no torneio!</p>';
}else{
    if((((($daysDiff/7)*$numberOfDays)<(((($teamsNumber-1)*$teamsNumber)/2))){
        echo '<p style="color: red">Tem de adicionar mais campos/horas durante a semana uma vez que as
    }elseif ($basicResults>2) {
        echo '<p style="color: red">Já gerou jogos para este torneio! Pode agora inciar o torneio.</p>';
        echo '<p style="color: black">Pretende apagar os jogos para pode-los gerar outra vez?</p>';
        echo '<form method=post>
            <button id="boto2" type="submit" name="deletetudo" class="leButton">Apagar</button>
        </form>';
    }
```

Figura 73: Verifica se é possível gerar jogos

-Se todas as condições foram reunidas, então serão calculadas o número de primeiras e segundas mãos que o torneio poderá ter. Em seguidas as equipas serão divididas em dois. Equipas que começam por jogar em “casa” e aquelas que começam por jogar “fora”.

```
for ($i = 0; $i < $teamsNumber / 2; $i++) {
    $Home[$i] = $Teams[$i];
    $Away[$i] = $Teams[$teamsNumber - 1 - $i];
}
```

Página 21: Divisão de equipas casa/fora

-Posteriormente são geradas todas as combinações possíveis entre equipes para os jogos.

```
for ($i = 0; $i < $gamesCount; $i++) {
    if (($i % 2) == 0) {
        for ($j = 0; $j < $teamsNumber / 2; $j++) {
            $calendar[$i][] = [$Away[$j], $Home[$j]];
        }
    } else {
        for ($j = 0; $j < $teamsNumber / 2; $j++) {
            $calendar[$i][] = [$Home[$j], $Away[$j]];
        }
    }

    $pivot = $Home[0];
    array_unshift( &array: $Away, $Home[1]);
    $carryover = array_pop( &array: $Away);
    array_shift( &array: $Home);
    array_push( &array: $Home, $carryover);
    $Home[0] = $pivot;
} //endfor
```

Página 22: Algoritmo round-Robin, gera todas as combinações possíveis entre equipes.

-Por fim é selecionada a data do primeiro jogo. A partir daí vão sendo gerados jogos a partir dos dias da semana em que se pode realizar o jogo. Este processo é feito calculando a diferença de dias dos mesmos escolhidos para o horário em que acontece o torneio. A partir daí é usada a data para o primeiro jogo e usa-se a função do date do php para acrescentar a diferença de dias para cada jogo.

```
foreach ($FinalCalendar as $ke) {
    $index = $j + $startIndex < $numberOfDays ? $j + $startIndex : $j + $startIndex - $numberOfDays;

    if (in_array($dateToInsert, $NoContesto)){
        $trial = $dateToInsert;
        $dateToInsert = date( 'format: Y-m-d', strtotime( time: $trial . ' + ' . $data3[$index] . ' day' ));
        $lastInsertedDate = $dateToInsert;
        continue;
    }

    if ($a%2==1) {
        $query = "INSERT INTO game(gamefield, gamedate, starttime, endtime, tournament_tournamentid, teamone, teamtwo) VALUES
        ( '$GameField[$index]', '$dateToInsert', '$StartTime[$index]', '$EndTime[$index]', $quicko, '$ke[0]', '$ke[1]' )";
        $resp = mysqli_query($db, $query);
    } elseif ($a%2==0){
        $query = "INSERT INTO game(gamefield, gamedate, starttime, endtime, tournament_tournamentid, teamone, teamtwo) VALUES
        ( '$GameField[$index]', '$dateToInsert', '$StartTime[$index]', '$EndTime[$index]', $quicko, '$ke[1]', '$ke[0]' )";
        $resp = mysqli_query($db, $query);
    }
    $j = ($j + 1) % $numberOfDays;
    $cycles++;
}
```

Página 23: Gera e insere os dados na base de dados.

### Iniciar o torneio:

-Nesta parte a variável “alreadystarted” presente na database é updated para true e então os jogos já poderão ser vistos pelos jogadores.

# Conclusão

---

O desenvolvimento deste processo não nos permitiu apenas aprimorar conhecimentos pré-existentes de tecnologias de desenvolvimento web como o *html*, o *css* e o *javascript*, mas também adquirir novos conhecimentos referentes a bases de dados (por exemplo, modulação de dados), com o *SQL*, e à sua utilização nas páginas web utilizando o *php*. No entanto, gostaríamos de apontar aquilo que consideramos se um aspeto a melhorar, o enunciado. Este, por vezes, não foi claro o suficiente e sofreu diversas alterações ao longo do tempo que também não facilitaram a tarefa, no futuro, seria interessante possuir uma checklist de funcionalidades desde o início para que os alunos possam ter mais sensibilidade sobre as especificações necessárias para que o projeto vá mais de encontro ao esperado pelo professor. Apesar dos vários problemas que enfrentámos ao longo do percurso, o balanço que fazemos é positivo pois para além dos conhecimentos adquiridos (já referidos anteriormente) conseguimos ainda melhorar a qualidade do nosso trabalho através de boas práticas de programação que ajudaram a tornar os milhares de linhas de código percetíveis.

# Equipa

A equipa responsável pelo desenvolvimento da aplicação, desde as decisões de design até à elaboração do código é constituída por:

Nome	Nº de Estudante	Email
<b>Bruno Ferreira</b>	2015240050	brunocar.f@hotmail.com
<b>Guilherme Costa</b>	2017249276	uc2017249276@student.uc.pt
<b>Leandro Pais</b>	2017251509	lbpais@student.dei.uc.pt
<b>Sérgio Machado</b>	2017265620	sergiomanuelmachado@hotmail.com