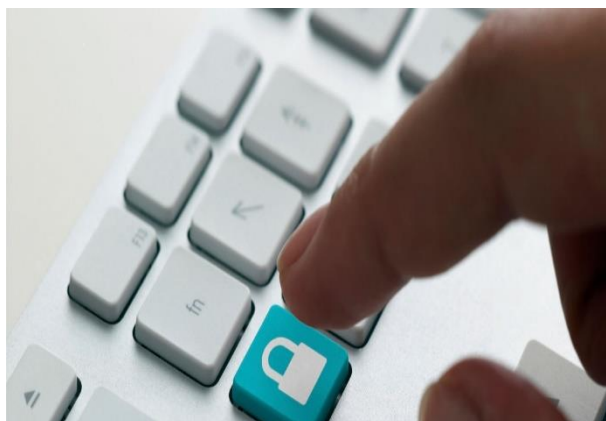


# Trabalho Prático

## Segurança em Aplicações Web

CTeSP – Desenvolvimento para a WEB e Dispositivos Móveis



Janeiro 2019

Hugo Silva-8180378

Marco Carneiro-8180382

## Conteúdo

Introdução .....	3
Desenvolvimento .....	4
Registo.....	4
Login .....	7
Recuperação de Password.....	9
Gestão de erros e logs.....	10

## Introdução

Este relatório foi elaborado para a disciplina de SAW e tem como objetivo descrever a elaboração da nossa aplicação em PHP que contém o seguinte conjuntos de funcionalidades;

- Registo de utilizadores;
- Recuperação da conta (“Forget me”);
- “Remember me”;
- Autenticação;
- Três áreas:

Uma área pública visível para todos os utilizadores (registados e não registados onde devem ser listados todos os filmes existentes no clube de vídeo, mas sem informação do seu estado;

Uma área para utilizadores registados (clientes)

- ♣ Contendo um perfil do utilizador;
- ♣ Um utilizador pode associar uma imagem ao seu perfil;
- ♣ Alterar os seus dados;
- ♣ Não pode ter acesso ao perfil dos outros clientes
- ♣ Contendo uma área para consulta de filmes com informação da sua disponibilização.
- ♣ Possibilitar requisitar um filme disponível, quando é feita uma requisição de um filme, o filme deve passar a indisponível para aluguer e passar a fazer parte da tabela de filmes alugados.
- ♣ Deve ser possível ter acesso a uma listagem de todos os filmes já alugados por si.

Uma área para Administração

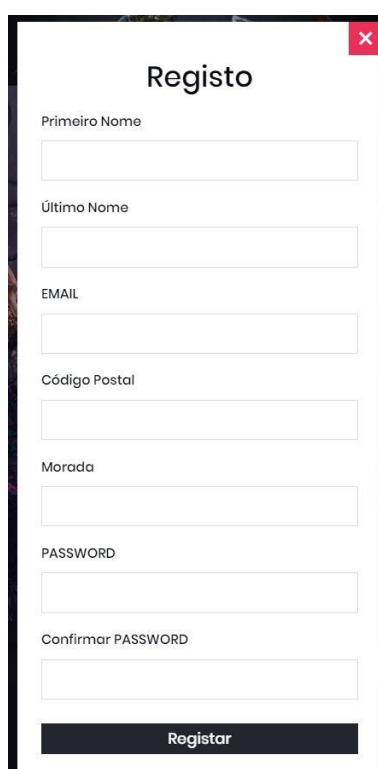
- ♣ Contendo um perfil do Administrador

- ♣ Contendo uma listagem dos clientes existentes na plataforma com uma opção de visualização dos filmes alugados por um cliente
- ♣ Uma área que permita fazer a manutenção dos filmes, inserir, alterar, eliminar com indicação do seu estado (disponível, indisponível, brevemente)
- ♣ Uma área para dar entrada de filmes alugados, sempre que é dada entrada de um filme, esse filme tem de sair da tabela de filmes alugados.

Ao logo deste relatório irão ser descritas as medidas utilizadas para garantir a segurança da nossa plataforma.

## Desenvolvimento

### Registo



The image shows a web registration form titled "Registo" with a close button (X) in the top right corner. The form contains the following fields:

- Primeiro Nome
- Último Nome
- EMAIL
- Código Postal
- Morada
- PASSWORD
- Confirmar PASSWORD

At the bottom of the form is a dark button labeled "Registar".

A medida que tomamos no formulário de registo foi usar o “require” no HTML pois assim o utilizador apenas pode clicar em registar se tiver inserido todos os campos obrigatórios.

De seguida ao Registar é enviado um mail com um link para o utilizador confirmar o registo, assim que o utilizador confirma o registo é possível fazer o login.

## Execução do Registo:

```
//Valida pass
function validapass($pass,$passveri){
    if($pass == $passveri){
        $int = 0;
        $por = 0;
        $arri = str_split($pass);

        //Verifica se tem Maiusculas
        foreach ($arri as $testcase) {
            if (ctype_upper($testcase)) {
                $int = $int + 1;
            }
        }
        //Verifica se tem inteiros
        if (ctype_digit($testcase)) {
            $por = $por + 1;
        }
    }

    if($por != 0 && $int != 0){
        return true;
    } else{
        echo "<script language='javascript' type='text/javascript'>alert('Password Invalida, a password deve ter maiusculas, minusculas e números!');window.location.href='movies.php'</script>";
        return false;
    }
    else{
        echo "<script language='javascript' type='text/javascript'>alert('Password não coincidem!');window.location.href='movies.php'</script>";
        return false;
    }
}
```

A primeira verificação a ser feita quando o utilizador clicar em registar é se a password cumpre os requisitos mínimos (Ter pelo menos uma letra maiúscula, uma minúscula e pelo menos um número inteiro).

De seguida foi elaborada uma função que verifica se o email com que o utilizador se pretende registar já se encontra na base de dados.

```
function verificarUtilizadorExistente($email, $conn)
{
    //Verifica se o mail é valido
    if (filter_var($email, FILTER_VALIDATE_EMAIL) === false) {
        echo "<script language='javascript' type='text/javascript'>alert('E-MAIL INVALIDO!');window.location.href='movies.php'</script>";
        return true;
    } else{
        $stmt = $conn->prepare("SELECT * FROM users WHERE email = (?)");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $result = $stmt->get_result();
        $stmt->close();

        if ($result!=null && $result->num_rows > 0) {
            echo "<script language='javascript' type='text/javascript'>alert('ERRO! Utilizador já existente!');window.location.href='movies.php'</script>";
            return true;
        }

        return false;
    }
}
```

Depois foi criada a função de registo que chama as funções que verificam se a password tem os requisitos obrigatórios e se o email não está ainda registado. Caso estas duas funções apresentem o resultado esperado (password válida e email válido) então os dados (que foram sanitizados de acordo com o padrão a que pertencem) são inseridos na base de dados.

A inserção é feita através de queries parametrizadas. Assim com os dados sanitizados e a inserção feita através de queries parametrizadas estamos a prevenir o SQL Injection (tipo de ataque que consiste na inserção de comandos SQL não autorizados em instruções SQL, para descobrir informação armazenada nas bases de dados).

Assim que a inserção é feita é enviado um email de confirmação para o utilizador confirmar o seu registo.

```
function RegistrarUtilizador($conn,$nome,$pnome, $pass, $passveri,$email,$morada,$codigo){
    $pnome = filter_var($pnome, FILTER_SANITIZE_STRING);
    $nome = filter_var($nome, FILTER_SANITIZE_STRING);
    $morada = filter_var($morada, FILTER_SANITIZE_STRING);
    $codigo = filter_var($codigo, FILTER_SANITIZE_STRING);
    $email = filter_var($email, FILTER_SANITIZE_EMAIL);
    //Verifica estado da pass
    $estadopass = validapass($pass,$passveri);
    //Se estiver dentro dos parametros estabelecidos é incipetada
    if($estadopass == true){
        $hashed_password = password_hash($pass, PASSWORD_DEFAULT);

        ///Verifica se email é valido e não está registado
        $verificautilizador = verificarUtilizadorExistente($email,$conn);
        if($verificautilizador == false){
            $token = bin2hex(random_bytes(50)); //Cria o token
            $stmt=$conn->prepare("INSERT INTO `users` (`nome`,`apelido`,`email`,`codigo postal`,`morada`,`password`,`token`) VALUES (?, ?, ?, ?, ?, ?, ?)");
            $stmt->bind_param('sssssss',$pnome, $nome,$email,$morada,$codigo,$hashed_password,$token);
            $stmt->execute();
            $stmt->close();
            mandamail($email,$token,$pnome);
        }
    }
}
```



**Segurança das Aplicações Web** <hugsaf2132@gmail.com>  
para mim ▾

quarta, 11/12, 17:08 (há 6 dias) ☆ ↩ ⋮

Confirmar Registo -Movie Points  
Clique no link abaixo para confirmar o seu registo:

[tp.com/tp/phpmovies/confirmaremail.php?token=32fa8cbf0bc7343e447dd49d7c19acc045ab62a04788b85394361a4e32569ab55970ec77b543e9b958d2b85ba584d2b97ca4&email=hugsaf2132@gmail.com](http://tp.com/tp/phpmovies/confirmaremail.php?token=32fa8cbf0bc7343e447dd49d7c19acc045ab62a04788b85394361a4e32569ab55970ec77b543e9b958d2b85ba584d2b97ca4&email=hugsaf2132@gmail.com)

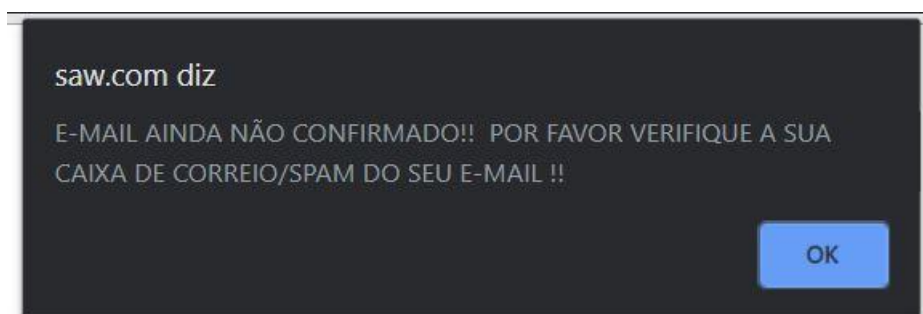
↩ Responder

➡ Encaminhar

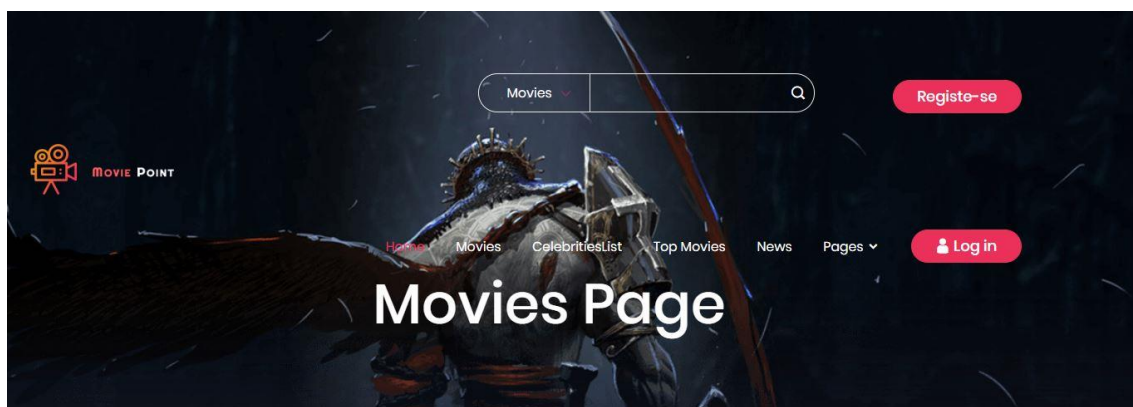
Assim que o utilizador clicar neste link estará pronto para fazer login no nosso site.

## Login

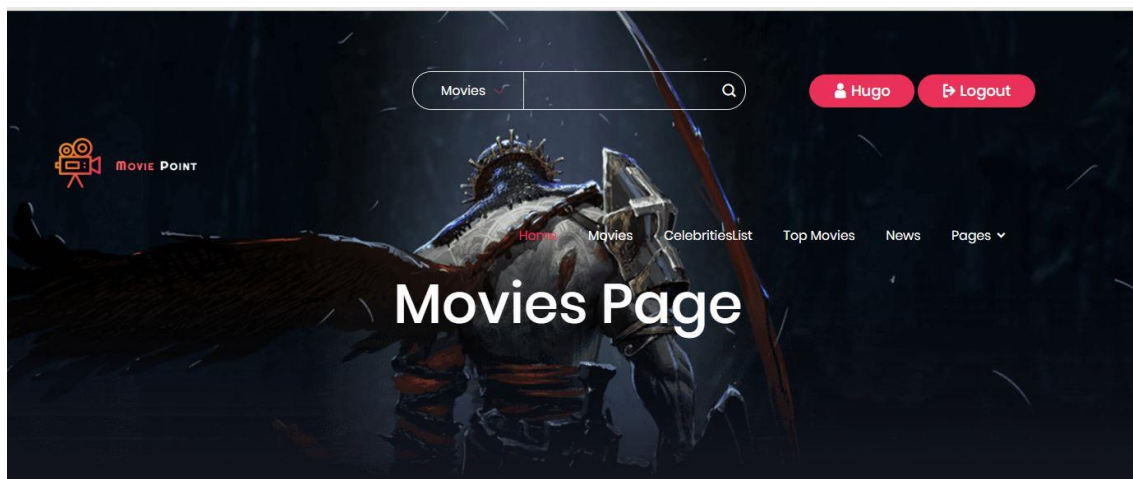
O login só é permitido se o utilizador tiver confirmado o email, caso contrário é informado de que o email ainda não foi confirmado.



Caso o login seja efetuado corretamente o cliente passa a ter acesso ao estado dos filmes.



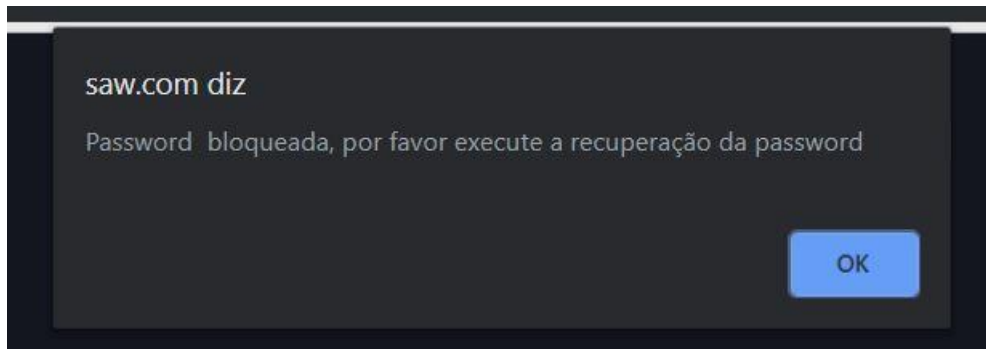
Sem login efetuado.



Com login efetuado.

Outra medida que tomamos para garantir a segurança é que o utilizador apenas tem cinco tentativas para inserir a palavra-passe, caso erre as cinco fica bloqueado e é obrigado a redefinir a pass. Isto previne um número

ilimitado de tentativas de inserção de passwords por pessoas que estejam a tentar “advinha-la”.



```
function tent( $email,$conn,$int1){

    $myEmail = filter_var($email, FILTER_SANITIZE_EMAIL);

    $stmt = $conn->prepare("SELECT tent FROM users WHERE email = ?");

    $stmt->bind_param("s", $myEmail);
    $stmt->execute();
    $result = $stmt->get_result();
    $row = $result->fetch_assoc();
    $stmt->close();
    if( $int1 == 1){ /// Verifica se pass esta bloqueada
        echo "<script language='javascript' type='text/javascript'>alert('Password bloqueada, por favor execute a recuperação da password');window.location.href='movies.php'</script>";
    }else{

        if( $row["tent"] == 1){
            ///Passa o estado da pass a bloqueado
            $int= 1;
            $stmt = $conn->prepare("UPDATE users SET passblock = ? WHERE email = ?;");
            $stmt->bind_param("is", $int,$myEmail );
            $stmt->execute();
            $stmt->close();
            ///Update ao numero de tentativas
            $int= 5;
            $stmt = $conn->prepare("UPDATE users SET tent = ? WHERE email = ?;");
            $stmt->bind_param("is", $int,$myEmail );
            $stmt->execute();
            $stmt->close();
            echo "<script language='javascript' type='text/javascript'>alert('Password bloqueada for favor processada à recuperação da password!!');window.location.href='movies.php'</script>";

        } else{

            $row["tent"] = $row["tent"] + 1;
            $stmt = $conn->prepare("UPDATE users SET tent = ? WHERE email = ?;");
            $stmt->bind_param("is", $row["tent"],$myEmail);
            $stmt->execute();
            $stmt->close();

            $aux= $row["tent"];
            echo "<script language='javascript' type='text/javascript'>alert('Password errada restam $aux tentativas!!');window.location.href='movies.php'</script>";
        }
    }

}
```

Código que demonstra a verificação do número de tentativas que o utilizador tem e quando estas chegam a zero bloqueia o utilizador sendo que este tem de proceder à recuperação de password para voltar a poder efetuar login.




### Recuperação de Password

A recuperação da password é feita através de um link enviado através de email.

Quando o utilizador insere a password é feita a verificação se esta cumpre os requisitos mínimos (Ter pelo menos uma letra maiúscula, uma minúscula e pelo menos um número inteiro). Caso a password não cumpra os requisitos mínimos o utilizador é informado, se obedecer aos requisitos mínimos é sanitizada e encriptada. De seguida inserida na base de dados (sempre através de queries parametrizadas).

## Update Perfil

**Editar Utilizador**



Nome

Apelido

Email

Morada

Código de Postal

Imagem

Nenhum ficheiro selecionado

O utilizador pode atualizar os seus dados aqui, uma vez inseridos os dados serão sanitizados antes de ser colocados na base de dados. As inserções são feitas através de queries parametrizadas.

## Gestão de erros e logs

Para garantir uma boa análise da nossa plataforma criamos ficheiros de erros e de logs para depois poderem ser analisados pelo administrador para detetar possíveis ataques ou falhas de segurança.

log\_29-Dec-2019.log - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

Erro-Login, Dupla Entrada SQL, 29-Dec-2019, 19:15:27, 127.0.0.1

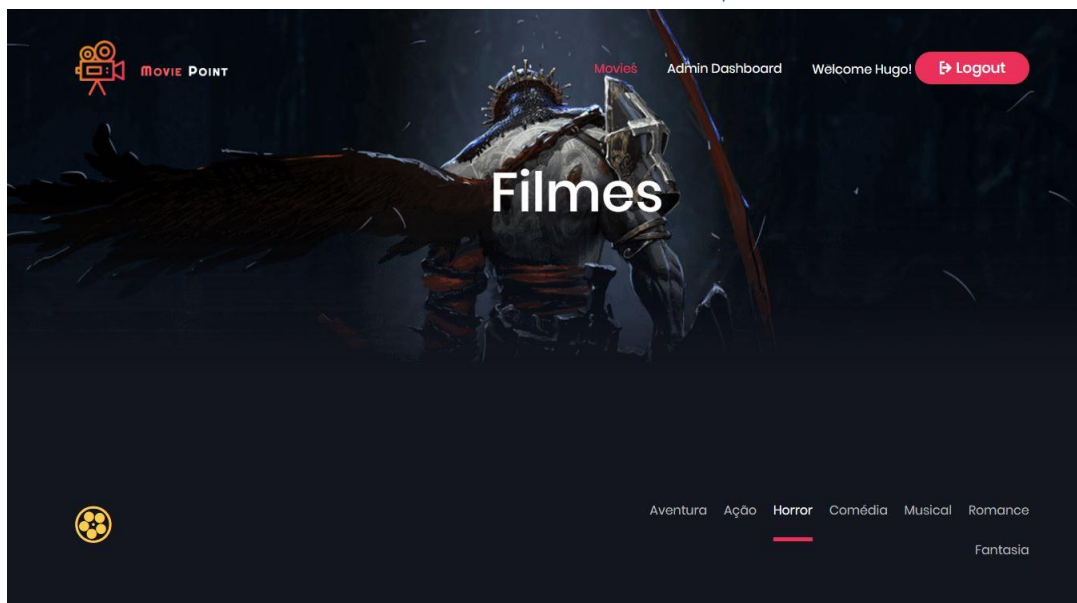
Para este ficheiro é passado a hora, uma descrição do erro, a hora em que decorreu bem como o IP da máquina em que ocorreu.

```
// Check connection
if ($conn->connect_error){
    wh_log("Erro-Conexão à base de dados");
    die("Connection failed: " . $conn->connect_error);
}
```

Neste exemplo, se a conexão à base de dados não for feita corretamente é chamada a função que escreve no ficheiro o erro.

## Gestão de Admin

Quando o login é realizado é verificado o tipo de utilizador, caso seja admin ou funcionário é disponibilizado no menu um link para a parte do BackOffice. Assim prevenimos que utilizadores não autorizados acedem ao BackOffice.



### Aluguer de Filmes

O cliente depois ter feito o login pode reservar filmes caso estes estejam disponíveis para aluguer.

Feito o aluguer é possível terminá-lo. Caso seja terminado o filme vai para a tabela de filmes entregues



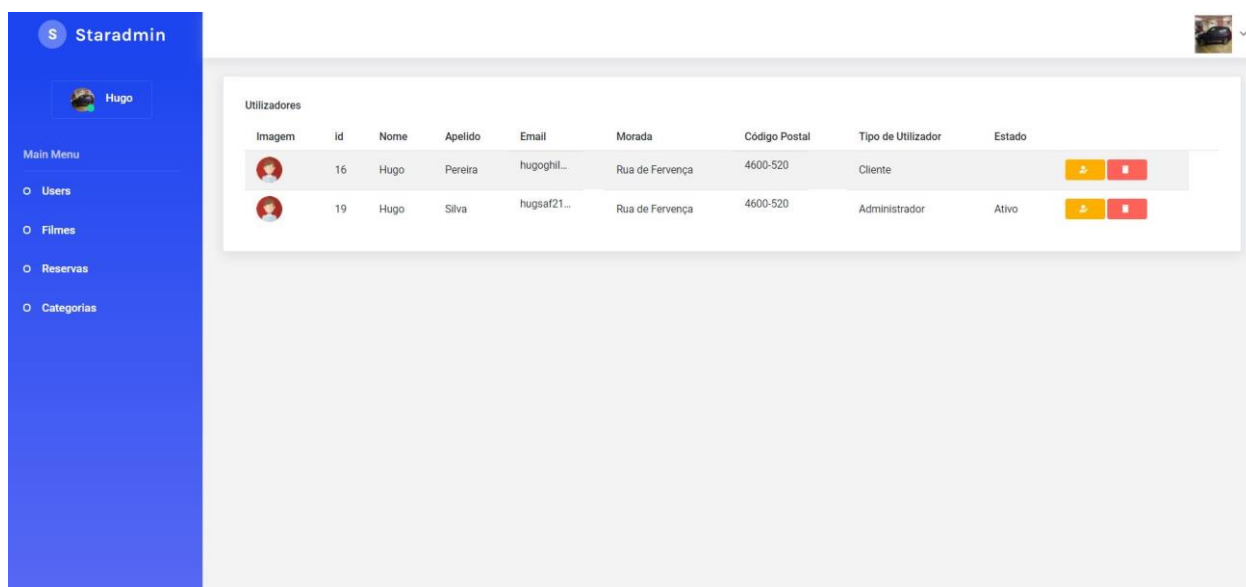
Nome filme	Data de entrega	Estado
Velocidade Furiosa	2020-01-19	Por Entregar

O cliente tem acesso ao seu histórico de filmes alugados.



Nome filme	Data de entrega	Terminar aluguer
Velocidade Furiosa	2020-01-19	Finalizar Aluguer

E tem acesso aos filmes que tem alugados, com a opção de terminar o registo.



No BackOffice o admin pode editar filmes, utilizador, categorias e ver todas as reservas feitas. O layout desta área foi feito com base na simplicidade para que o admin facilmente encontre as funcionalidades desejadas. Todas as inserções feitas pelo administrador são através de queries parametrizadas.

## Conclusão

Em suma, o Trabalho Prático serviu para consolidar todos os conteúdos lecionados nas aulas da disciplina Segurança em Aplicações WEB. Ao longo da realização do projeto ficamos a conhecer melhor as medidas que devemos tomar enquanto programadores para garantir a segurança das nossas plataformas.