



**Universidade de Aveiro**  
**Novas Tecnologias da Comunicação**

**Interface de administração**  
**Laboratório Multimédia 4**  
**2020**

aplans.

email: **admin@ua.pt**  
password: **123456**

email: **normal@ua.pt**  
password: **123456**

**[www.aplans.pt](http://www.aplans.pt)**

Miguel Filipe | **95039**  
David Ribeiro | **94801**  
João Mota | **89160**  
Samuel Santos | **89149**

## Introdução

Para o desenvolvimento desta interface de administração o grupo decidiu utilizar AJAX. Isto porque já tínhamos intenções de desenvolver a própria aplicação para a entrega final com esta tecnologia. Assim, acabou por ser uma introdução à lógica de AJAX que, apesar de ser bastante oportuna, fez com que o tempo de desenvolvimento aumentasse consideravelmente enquanto o grupo se familiarizava com os procedimentos.

Temos consciência de que o objetivo desta entrega era o de concluir por completo a interface de administrador e que tal não foi atingido. No entanto, queremos reforçar que adotamos uma tecnologia que era nova para nós e que ia um pouco mais além do que foi leccionado e treinado nas aulas da Unidade Curricular.

Concluimos a 100% a manutenção da tabela de utilizadores a poucas horas da entrega. O que tínhamos idealizado, e que será definitivamente desenvolvido para a entrega final do projeto, é utilizar este código da tabela de utilizadores e torná-lo dinâmico para que a sua lógica e todas as suas funções funcionem com todas as tabelas, sem ser necessário copiar e repetir o mesmo código em vários ficheiros, dedicados a tabelas específicas.

Para mostrar este potencial, e por acharmos que apenas uma tabela nesta fase seria pouco, copiamos de facto os ficheiros desenvolvidos da tabela de users e adaptamos para a tabela de eventos. Queremos salientar novamente que, apenas recorremos a esta opção por falta de tempo:

E é o tempo o último ponto nesta introdução. Queremos salientar que, por sobrecarga de trabalhos das várias unidades curriculares nesta fase final do semestre, a quantidade de opções que implementámos nesta interface, nomeadamente a pouca quantidade de tabelas que se podem consultar e editar, foi abaixo do esperado. No entanto, esperamos que, o esforço de aplicar uma tecnologia diferente do que seria “tradicional” para esta interface, compense este pouco número de opções.

Durante o desenvolvimento desta fase foram feitas algumas alterações necessárias à base de dados que anexaremos no final deste relatório.

## Descrição das Áreas Funcionais

- **Registo & Login de utilizadores:**

A possibilidade de registar utilizadores já se encontra totalmente implementada, com parâmetros obrigatórios de preenchimento: e-mail, password e nome; e alguns opcionais: número de telemóvel, morada e código postal. Este sistema de registo está ainda munido com um mecanismo de mensagens de alerta, que alerta o utilizador para os campos obrigatórios não preenchidos, assim como para formatos incorretos.

Após o registo é permitido o login, no entanto, para o utilizador normal não temos conteúdo a apresentar de momento, sendo este apenas um sistema de gestão de dados para contas de administração.

- **Diferenciação entre utilizador normal & administrador:**

Após o registo existe uma diferenciação entre utilizadores normais e administradores. De momento apenas apresentamos conteúdo para os administradores.

- **Visualização de dados:**

Após o Login com uma conta com permissões de administração é possível visualizar os dados numa tabela de todos os utilizadores registados, com um sistema de pesquisa, de ordenação ascendente e descendente, ordenação pela categoria pretendida, número de itens por página e sistema de paginação. Sublinhar que o sistema de pesquisa funciona para qualquer uma das categorias da tabela. De salientar ainda que este sistema funciona interligando uns filtros com os outros. Ou seja, é possível fazer uma pesquisa que, por exemplo, procure por “@gmail.com”, ordene os resultados obtidos por cargo do utilizador, na ordem ascendente para que apareçam os administradores em primeiro lugar, selecionar que desejamos que apenas existam 3 resultados por página e, por fim, aceder à página 2 desses mesmos resultados.

- **Manipulação de dados:**

Na tabela de administração é possível adicionar um novo utilizador à base de dados através do botão “+ add row”, ou editar um existente através do botão editar, onde é ainda permitida a alteração de permissões e alteração da palavra passe.

Para o caso da palavra passe, se, ao editar um utilizador, for submetido o formulário de edição sem os campos da mesma preenchidos, a palavra passe antiga é mantida. Ou seja, apenas é alterada a palavra passe se os dois campos de palavra passe forem preenchidos e corresponderem um com o outro.

## Implementação das Tecnologias Utilizadas

Entrado um pouco em detalhe sobre a tecnologia e a lógica que foi implementada para conseguir estes resultados:

### Registo:

Começando pelo registo na plataforma, apresentamos ao utilizador a opção de se registar preenchendo apenas o nome, o email e a palavra passe (duas vezes para confirmação). No entanto, clicando no ícone de “+” o sistema permite-nos inserir o número de telemóvel, a morada e o código postal.

No ficheiro scripts/sc\_register.php:

Para a validação do email utilizámos FILTER\_VALIDATE\_EMAIL.

Já para a validação dos campos do número de telemóvel e do código postal, utilizámos REGEX (s).

`"/(9[1236][0-9]) ?([0-9]{3}) ?([0-9]{3})/";`

Esta primeira, para o telemóvel, permite que existam 9 dígitos, sendo que o primeiro tem que começar por 9, o segundo necessita de ser 1, 2, 3 ou 6 e os restantes podem tomar valores de 0 a 9.

`"/^\d{4}-\d{3}?$/"`

Neste caso, para o código postal, apenas nos limita a que o formato seja “xxxx-xxx”

## Tabela Users:

**/admin/pages/users.php:**

Aquando do carregamento inicial da página (linha 319), é chamada a função `tabela()`, passando como parâmetros os valores que estão selecionados para filtros e ordenação (itens por página sendo 3 o default, ordenar pela categoria sendo o 'nome' default, ordenação sendo ascendente o default, o texto presente na barra de pesquisa que neste momento não existe e, por fim, a página que queremos ver, sendo a primeira o default ).

A função `tabela()` executa um pedido de AJAX à base de dados (`/admin/ajax/users_table`), esperando todos os resultados que correspondam aos filtros e definições aplicadas. Inicialmente, irá receber todos os utilizadores. Tendo a informação executa uma série de ações para que seja possível mostrar todos os dados que recolheu sem recarregar a página. Para isso, apaga e reescreve alguns elementos html como:

- O head da tabela (linha 241), que acompanha a ordenação por categoria, mostrando um pequeno ícone junto da categoria escolhida pelo utilizador. Realçar que este ícone também acompanha a ordem, sendo ela ascendente ou descendente.
- O corpo da tabela (linha 251). Para isto, a cada elemento do array de resultados, é gerado o código html da linha da tabela do utilizador e é adicionado ao elemento com id `#colunaTabela`.
- O menu de navegação por páginas (linha 259). É recebido da base de dados (processado no ficheiro `/admin/ajax/users_table`) o número total de página que irão ser necessárias para mostrar os resultados. Assim, igualmente num ciclo for, são escritos os botões das páginas no menu de navegação com o id `#paginas`.
- O modal para editar os dados (linha 273). Recebendo todos os dados, assim que se clique no botão de editar de um dos utilizadores, esta função gera o código html com os inputs pré-preenchidos e escreve-o no elemento com id `#modal_editar`.

Como podemos ver a partir da linha 327 até à linha 358, segue-se uma série de eventos que são acionados quando o utilizador escreve ou clica num dos parâmetros para filtrar ou ordenar os resultados.

Restam apenas algumas verificações e eventos que ocorrem quando o administrador guarda uma edição de um utilizador, adiciona um novo utilizador ou apaga um utilizador (linha 359 até 413).

Toda a lógica inerente aos pedidos ao servidor em AJAX encontram-se nos ficheiros

```
/admin/ajax/users_table.php  
/admin/scripts/sc_adicionar_row.php  
/admin/scripts/sc_delete_row.php  
/admin/scripts/sc_update_row.php
```

## Conclusão

Como já referimos, sentimos que, numa época mais leve em termos de carga de trabalhos, faríamos mais e melhor. No entanto, é comum a todos os membros do grupo a ideia de que esta fase de desenvolvimento da interface de administrador serviu para nos familiarizar-mos com AJAX e isso facilitará o trabalho futuro no desenvolvimento da aplicação final.

Apenas resalvar que, os ficheiros que consideramos estarem bem construídos são os que correspondem a ações da tabela de utilizadores. Apesar de estar desenvolvido para a tabela de eventos, não tivemos tempo de concluir tudo e a ação de adicionar um evento não ficou concluída.