

Engenharia Informática	2º Ano	2º Semestre	2018-19	Avaliação Periódica
<b>Projeto</b>	Prazo para divulgação resultados: 24 Junho 2019			
Date: 8 Abril 2019	<b>Data de Entrega: 1 Junho 2019</b>			

## Projeto – FlightClub

### OBJETIVO

O objetivo deste projeto é implementar uma aplicação Web baseada no servidor, utilizando a Framework Laravel, para ajudar à gestão do Aeroclube "FlightClub".

### CENÁRIO

A aplicação Web "FlightClub" será para utilização exclusiva dos sócios do Aeroclube "FlightClub" e tem como objetivo auxiliar a gestão do mesmo. O Aeroclube "FlightClub" tem uma escola de aviação e dispõe de um conjunto de aeronaves que aluga aos seus sócios pilotos ou que utiliza na sua escola.

A aplicação Web "FlightClub" deverá fazer a gestão dos sócios e utilizadores, gestão das aeronaves do aeroclube, gestão das licenças e certificados dos pilotos, registo e análise dos movimentos (voos).

### SÓCIOS E UTILIZADORES

Os sócios do aeroclube poderão ser de três categorias: "pilotos", "não-pilotos" e "aeromodelistas". Cada categoria de sócio tem uma quota diferenciada e um conjunto de regalias distintas. Todos os sócios com quotas em dia podem aceder à aplicação Web "FlightClub", embora com diferentes privilégios (existem 3 perfis que serão explicados mais à frente: Normal, Piloto e Direção).

No contexto da aplicação os sócios são considerados utilizadores - todos os sócios são utilizadores e vice-versa (só os sócios é que podem utilizar a aplicação). Os sócios são criados por elementos da direção e são desde logo considerados utilizadores (embora desativados). Depois do registo de sócio ser criado por um elemento da direção, é enviado um e-mail para o sócio com uma hiperligação para confirmar a conta. Depois de confirmada a conta o utilizador poderá aceder à aplicação com as suas credenciais (email/senha). A senha inicial de todos os utilizadores é gerada automaticamente durante o processo de criação do sócio, e é igual à sua data de nascimento – com o formato ano-mês-dia (exemplo: "1990-07-23"). Enquanto o utilizador não alterar a senha inicial, ao entrar na aplicação deverá ser redirecionado para uma página de alteração de senha.

Os sócios que não pagam as quotas podem deixar de poder aceder à aplicação, caso a direção assim o decida.

A aplicação deverá permitir à direção do Aeroclube (sócios que pertencem à direção) criar, alterar e apagar registos de sócios (se necessário, apagar através de *soft deletes*). Cada registo de sócio deverá incluir a seguinte informação: "nº de sócio", "nome informal" e "nome completo" ("nome informal" corresponde ao nome pelo qual os sócios são conhecidos – geralmente o primeiro e último nome), "sexo", "data de nascimento", "e-mail" (a usar como credencial de acesso), "foto", "NIF", "telefone", "endereço", "tipo de sócio" (P- piloto, NP – não piloto, A - aeromodelista), "quotas em dia" (indicação se as quotas estão ou não pagas), "sócio ativo" (indicação se o sócio está ou não ativo), "password inicial" (indicação se o sócio já alterou a password inicial) e "direção" (indicação se o sócio pertence ou não à direção). Para além desta informação base, os sócios pilotos terão também informação específica (por exemplo, o nº da licença de piloto) que será detalhada posteriormente (na secção relativa à gestão dos pilotos).

Todos os utilizadores (sócios) podem também alterar uma parte da sua informação pessoal ("nome informal", "nome completo", "e-mail", "NIF", "foto" e "telefone" e "endereço"), bem como consultar toda a informação sobre os outros sócios (exceto se tem ou não as quotas em dia e se é ou não um utilizador ativo).

## **AERONAVES**

A aplicação deverá fazer a gestão das aeronaves do aeroclube, cuja lista e informação detalhada deve ser disponibilizada a todos os sócios. Os elementos da direção podem criar, alterar e apagar registos de aeronaves (se necessário, apagar através de *soft deletes*). Cada aeronave deverá incluir uma "matrícula", "marca", "modelo", "nº de lugares", "conta-horas" (total de horas que a aeronave esteve em uso) e "preço hora".

### **CONTA-HORAS**

Note que as aeronaves incluem um dispositivo, designado de conta-horas, que contabiliza o total de horas que a aeronave esteve a ser utilizada (com o motor ligado). Pode-se considerar que o conta-horas de uma aeronave é similar ao conta-quilómetros de um automóvel, mas em vez de se contabilizar os kms percorridos, contabiliza-se o tempo de utilização. A informação sobre as "horas" da aeronave advém do valor atual do conta-horas.

A unidade nos conta-horas das aeronaves é a décima de hora, que corresponde a 6 minutos. Se por exemplo o conta-horas tiver o valor 2583 significa que a aeronave foi utilizada durante 258 horas e 3 décimos de hora, ou seja, 258 horas e 18 minutos.

Para facilitar os cálculos do tempo e do preço de utilização das aeronaves, a aplicação irá manter uma tabela que mapeia os valores do conta-horas (1 a 10) com os tempos e preços de utilização.

## PILOTOS

Os sócios pilotos são sócios que têm uma licença de piloto - incluindo os pilotos alunos que têm uma licença especial de aluno-piloto. Para que os sócios pilotos possam voar nas aeronaves do Aeroclube deverão ter uma licença de piloto válida e um certificado médico válido.

Apesar de cada piloto poder ter várias licenças e a cada licença poderem ser averbadas várias qualificações (ex: "SEP – *Single Engine Piston*", "MEP – *Multiple Engine Piston*" ou "FI – *Flight Instructor*"), e apesar dos dados e estrutura das licenças de piloto e certificados médicos ser bastante complexa, no contexto da aplicação, a informação sobre as licenças e certificados médicos será reduzido ao mínimo necessário para o funcionamento do Aeroclube.

Assim, para cada piloto só será guardada uma licença (a licença válida que é utilizada para voar as aeronaves do Aeroclube) e um certificado médico (o certificado médico válido). A licença deverá incluir o "nº da licença", "tipo de licença", "instrutor" (se permite ou não dar instrução), "validade", "confirmado" (indica se a licença já foi confirmada pela direção) e "imagem" (cópia digital da licença em formato PDF). O certificado médico deverá incluir o "nº certificado", "classe", "validade", "confirmado" (indica se o certificado já foi confirmado pela direção) e "imagem" (cópia digital do certificado em formato PDF).

Para além das licenças e certificados válidos, um piloto só poderá voar uma aeronave em concreto depois de fazer uma adaptação à mesma por um instrutor (piloto instrutor) qualificado, e depois deste o declarar como autorizado a pilotar a aeronave. Por último, só poderá voar na aeronave se mantiver a proficiência de voo, ou seja, se nos 90 dias anteriores tiver efetuado 3 descolagens/aterragens numa aeronave do mesmo tipo/classe.

É responsabilidade da direção do Aeroclube, depois de analisar os processos dos pilotos, manter atualizada para cada aeronave uma lista de pilotos autorizados a voar essa aeronave. A aplicação deverá permitir que qualquer membro da direção atualize estas listas de acordo com as validades da licença e certificado médico, adaptações às aeronaves, manutenção da proficiência de voo e cumprimento de regras internas de utilização das aeronaves. A aplicação deverá auxiliar a direção neste processo informando os utilizadores (membros da direção e os próprios) da aproximação ou ultrapassagem dos limites de validade das licenças e certificados. A aplicação dos restantes critérios (adaptação, proficiência e regras internas) é da exclusiva responsabilidade da direção sem intervenção ou ajuda por parte da aplicação.

## MOVIMENTOS

As aeronaves e os pilotos têm que manter um registo de todos os movimentos (voos) realizados numa caderneta de voo (*flight log*). Isto significa que quando um voo é realizado, o piloto tem que registar os dados desse voo em duas cadernetas (ambas em papel), a do piloto (*Pilot Logbook*) e a da aeronave (Diário de Navegação). Os dados dos registos de ambas as cadernetas estão sujeitas à fiscalização da entidade reguladora da aviação civil - em Portugal, Autoridade Nacional da Aviação

Civil (ANAC) -, pelo que é fundamental que os dados dos dois registos do mesmo voo (nas duas cadernetas) sejam consistentes.

Para além do registo em papel, quando um piloto voa numa das aeronaves do Aeroclube está obrigado a fazer o registo digital do movimento (voo) na aplicação Web "FlightClub" (faz parte das regras internas de utilização das aeronaves).

Os dados de registo dos movimentos (voo) incluem um identificador único ("id") e os seguintes dados: "data do voo", "hora descolagem" (ex. 12:35), "hora aterragem", "aeronave" (matrícula, marca, modelo), "nº diário" (nº da caderneta/diário de navegação da aeronave), "nº serviço" (nº do serviço/movimento dentro do diário), "piloto", "nº licença piloto", "tipo licença piloto", "validade licença piloto", "nº certificado médico piloto", "classe certificado médico piloto", "validade certificado médico piloto", "natureza do voo" (T – Treino; I – Instrução; E – Especial), "aeródromo de partida" e "aeródromo de chegada" (código e nome dos aeródromos), "nº de aterragens", "nº de descolagens", "nº pessoas a bordo", "conta-horas inicial", "conta-horas final", "tempo voo" (calcular o tempo total de voo a partir dos valores do conta-horas), "preço voo" (calcular a partir do tempo de voo e preço hora da aeronave), "modo de pagamento" (N – numerário; M – Multibanco; T – Transferência; P – Pacote de horas), "nº de recibo" (nº do recibo em papel onde o voo foi pago), "confirmado" (indica se o movimento já foi confirmado pela direção) e "observações". Se for um voo de instrução deverá também incluir o "tipo de instrução" (D – Duplo Comando; S – Solo), "instrutor", "nº licença instrutor", "tipo licença instrutor", "validade licença instrutor", "nº certificado médico instrutor", "classe certificado médico instrutor", "validade certificado médico instrutor".

Note que a informação sobre o piloto ou instrutor num dado movimento, inclui sempre o piloto, que corresponde ao ID do sócio piloto, e informação sobre a licença e certificado médico (números, tipos e validades). Isto deve-se ao facto de um piloto poder ter várias licenças e certificados médicos ao longo da sua vida. Desta forma, o registo do movimento mantém a informação de qual o piloto (pessoa) que fez o voo e qual a sua licença e certificado médico que estavam ativos e válidos no momento em que o voo foi realizado (informação fundamental para efeitos legais).

## CONFLITOS

Quando um movimento é registado, os valores do "conta-horas" ("conta-horas inicial" e "conta-horas final") devem ser validados de forma a que não se sobreponha a um movimento registado anteriormente e de forma a garantir que não existem "buracos" temporais por preencher. Por exemplo, se o maior valor de "conta-horas final" registado num movimento da aeronave for igual a 2453, então, no novo movimento o "conta-horas inicial" deve ser também 2453. Se o valor for superior, considera-se que existe um buraco temporal, se o valor for inferior, então existe uma sobreposição. Em ambos os casos, a aplicação deve avisar (com uma mensagem) o utilizador e pedir que este confirme os valores do "conta-horas". Caso o piloto opte por confirmar os valores, mesmo que aparentemente estejam incorretos, a aplicação deverá aceitar o registo – a responsabilidade dos dados é sempre do piloto e o erro poderá estar num registo anterior e não no

novo registo. Neste caso, como os valores estão inconsistentes, o registo do movimento deverá incluir o campo "conflito" com um de dois valores: "S - Sobreposição" e "B - buraco" e o campo "justificação conflito" com texto em que o piloto pode explicar o porquê de manter o conflito em aberto. Note que os movimentos podem não ser registados pela ordem em que foram efetivamente realizados, pelo que um aparente conflito pode ser apenas devido a algum movimento em falta.

## **ALTERAÇÕES E CONFIRMAÇÃO DOS MOVIMENTOS**

Os registos digitais dos movimentos são para consulta interna do Aeroclube, mas também estão sujeitos à fiscalização pela ANAC. Por essa razão, é fundamental que os registos dos movimentos não sejam alterados. No entanto, enquanto a direção não confirmar os dados do movimento, o piloto que o registou pode alterá-lo ou removê-lo. Depois de confirmado pela direção, o movimento já não pode ser alterado ou removido.

Durante o processo de confirmação dos movimentos pela direção, é verificada a não existência de conflitos (no conta-horas), e é feito o cruzamento da informação dos registos de movimentos digitais com os registos em papel do diário de navegação da aeronave e dos recibos em papel.

## **UTILIZADORES**

A aplicação é apenas acessível aos sócios do Aeroclube. Os utilizadores anónimos só podem aceder ao login ou a páginas para ativar a conta ou para recuperar a senha de entrada. Existem 3 tipos de sócios (utilizadores): os sócios normais, os sócios pilotos e os sócios da direção. Os diferentes tipos de sócios (utilizadores) têm acesso a:

- "Normal" – ver e alterar o seu perfil de utilizador, consultar a lista de sócios e consultar informação completa sobre as aeronaves (incluindo os movimentos);
- "Piloto" – têm acesso a todas as funcionalidades dos sócios "normais". Podem também atualizar a informação da sua licença de piloto e certificado médico, registar movimentos (voos) e alterar e apagar os seus movimentos (enquanto não estiverem confirmados);
- "Direção" – têm acesso a todas as funcionalidades dos sócios "normais" e sócios "pilotos". Para além disso podem alterar ou remover movimentos ainda não confirmados (de qualquer piloto) e confirmar movimentos. São responsáveis pela gestão das aeronaves e respetivos preços por hora, pela gestão dos sócios (acrescentar, alterar ou remover sócios) e utilizadores da aplicação (sócios), gestão da informação sobre as licenças e certificados médicos dos pilotos (atualizar ou confirmar a informação) e gerir a lista de pilotos autorizados a voar as aeronaves do aeroclube.

## **REQUISITOS FUNCIONAIS**

Os requisitos funcionais do projeto serão apresentados na forma de *User Stories*, organizados por tipo de utilizador. É também incluída uma *User Story* de "desafio" cuja implementação requer

alguma análise e investigação independente por parte dos estudantes (não será dada nenhuma solução nem sugestão de solução para esta *User Story*).

## **UTILIZADOR ANÓNIMO**

- US 1. Como utilizador anónimo quero autenticar-me na aplicação com credenciais válidas (email e senha). Só os sócios ativos é que poderão autenticar-se e entrar na aplicação.
- US 2. Como utilizador anónimo quero ativar a minha conta na aplicação, recebendo no meu email uma mensagem com uma hiperligação de ativação. Quando o utilizador acede ao endereço da aplicação que está associado à hiperligação de ativação, a aplicação ativa a conta.  
Note que o email de ativação é enviado automaticamente pela aplicação, sempre que um elemento da direção cria um novo sócio.
- US 3. Como utilizador anónimo quero recuperar a senha de entrada na aplicação, passando para a aplicação o meu endereço de email, e se este estiver associado a uma conta válida, recebendo nesse email uma mensagem com um link para reiniciar a senha de entrada.

## **SÓCIO "NORMAL"**

- US 4. Como utilizador/sócio normal quero sair da aplicação (fechar a sessão / *logout*).
- US 5. Como utilizador/sócio normal quero alterar a minha senha de entrada na aplicação.
- US 6. Como utilizador/sócio normal quero visualizar o meu perfil, nomeadamente os campos, "nº de sócio", "tipo de sócio" (P- piloto, NP – não piloto, A - aeromodelista), "nome informal" e "nome completo" ("nome informal" corresponde ao nome pelo qual os sócios são conhecidos – normalmente o primeiro e último nome), "sexo", "data de nascimento", "e-mail" (a usar como credencial de acesso), "foto", "NIF", "telefone", "endereço", "quotas em dia" (indicação se as quotas estão ou não pagas), " sócio ativo" (indicação se o utilizador associado ao sócio está ou não ativo) e "direção" (indicação se o sócio pertence ou não à direção).
- US 7. Como utilizador/sócio normal quero alterar o meu perfil, nomeadamente os campos, "nome informal", "nome completo", "e-mail", "foto" (upload de imagem), "data de nascimento", "NIF", "telefone" e "endereço".
- US 8. Como utilizador/sócio normal quero consultar lista de todos os sócios ativos (valor de "sócio ativo" = 1). Para cada sócio na lista quero visualizar a "foto", "nº de sócio", "nome informal", "e-mail", "telefone", "tipo de sócio", "nº de licença" (se for piloto) e se pertence à "direção".
- US 9. Como utilizador/sócio normal quero filtrar/pesquisar a lista de todos os sócios ativos (US8) pelo "nº de sócio", "nome informal", "e-mail", "tipo de sócio" e "direção".

- US 10. Como utilizador/sócio normal quero consultar a lista de aeronaves do Aeroclube e para cada aeronave obter a "matrícula", "marca", "modelo", "nº de lugares", "total de horas" e "preço hora".
- US 11. Como utilizador/sócio normal quero consultar a lista de todos movimentos (voos) das aeronaves do Aeroclube. Para cada movimento quero visualizar o "id do movimento", "matrícula da aeronave", "data do voo", "hora decolagem", "hora aterragem", "tempo voo", "natureza do voo", "piloto" (nome informal), "código do aeródromo de partida", "código do aeródromo de chegada", "nº de aterragens", "nº de decolagens", "nº diário", "nº serviço", "conta-horas inicial", "conta-horas final", "nº pessoas a bordo", "tipo de instrução", "instructor" (nome informal), "confirmado" e "observações".
- US 12. Como utilizador/sócio normal quero filtrar/pesquisar a lista de movimentos (US11) pelo "id do movimento", "aeronave", "piloto", "instructor", "data do voo" (intervalos de tempo – por exemplo, movimentos entre o dia X e o dia Y, movimentos do dia X ou anteriores, movimentos do dia X ou posteriores ou movimento do dia X), "natureza do voo" e "confirmado".
- US 13. Como utilizador/sócio normal quero obter informação estatística sobre os movimentos (voos) das aeronaves do Aeroclube. Essa informação deve ser apresentada de forma textual e, de preferência, também através de gráficos. A informação pretendida deverá cruzar o total de horas por mês/aeronave; ano/aeronave; mês/piloto e ano/piloto.
- O cruzamento de dados permite ao utilizador obter informação como por exemplo: (mês/aeronave) - no mês 02 do ano 2017, a aeronave X voou 12,8 horas; (ano/piloto) – no ano de 2016, o piloto X voou 34,3 horas. Se a informação for textual, as linhas representam uma dimensão (por exemplo, os meses) e as colunas outra dimensão (por exemplo, as aeronaves), ou vice-versa. Se a informação for gráfica, uma das dimensões corresponde ao eixo dos Xs e outra dimensão ao eixo dos Ys.

## **SÓCIO "PILOTO"**

- US 14. Como sócio piloto, na visualização do perfil de utilizador (US6), quero ver informação sobre a minha licença de piloto e sobre o meu certificado médico e a lista de aeronaves do Aeroclube que estou autorizado a voar. A licença de piloto deve incluir o "nº da licença", "tipo de licença", "instructor" (se permite ou não dar instrução), "validade", "confirmado" (indica se a licença já foi confirmada pela direção), "cópia digital da licença". O certificado médico deverá incluir o "nº certificado", "classe", "validade", "confirmado" (indica se o certificado já foi confirmado pela direção) e "cópia digital do certificado". Em ambos os casos, a cópia digital deverá ter o formato PDF e no perfil de utilizador basta apresentar uma hiperligação para ver/ ou descarregar (*download*) o PDF.
- US 15. Como sócio piloto, no perfil de utilizador quero alterar a informação sobre a minha licença de piloto ou alterar a informação sobre o meu certificado médico. Depois de alterar a



informação sobre a licença ou certificado médico, o campo "confirmado" das mesmas deverá ficar a falso – indicando que a informação não está confirmada pela direção. Os campos para alterar a licença e certificado são os mesmos que são referidos na US14. Para acrescentar/alterar a cópia digital da licença ou do certificado, é necessário fazer o upload de um documento PDF.

US 16. Como sócio piloto quero acrescentar movimentos relativos aos meus voos, ou seja, voos em que eu seja o piloto ou o instrutor. Os campos relativos ao instrutor são opcionais e só deverão ser preenchidos se o movimento for relativo a um voo de instrução.

Quando um voo envolve dois pilotos (piloto + instrutor), apenas um dos pilotos preenche o movimento – pode ser o piloto ou o instrutor a preencher o registo. Independentemente de quem preenche, a aplicação deverá garantir que todos os pilotos envolvidos no movimento estão autorizados a voar a aeronave em questão.

O registo do movimento inclui os seguintes dados: "data do voo" (dia), "hora decolagem" (ex. 12:35), "hora aterragem", "aeronave" (matrícula da aeronave), "nº diário" (nº da caderneta/diário de navegação da aeronave), "nº serviço" (nº do serviço/movimento dentro do diário), "piloto" (por omissão, pré-selecionar o piloto que está a registar o movimento), "natureza do voo" (T – Treino; I – Instrução; E – Especial), "aeródromo de partida" e "aeródromo de chegada" (códigos e nomes dos aeródromos), "nº de aterragens", "nº de decolagens", "nº pessoas a bordo", "conta-horas inicial", "conta-horas final", "tempo voo" (calcular o tempo total de voo a partir dos valores do conta-horas), "preço voo" (calculado automaticamente – ver US 18), "modo de pagamento" (N – numerário; M – Multibanco; T – Transferência; P – Pacote de horas), "nº de recibo" (nº do recibo em papel onde o voo foi pago) e "observações".

Caso o movimento seja relativo a um voo de instrução, deverá incluir também o "tipo de instrução" (D – Duplo Comando; S – Solo) e "instrutor".

O registo do movimento deverá também incluir dados sobre as licenças e certificados médicos atuais do piloto e do instrutor (caso seja um voo de instrução). Estes dados deverão ser preenchidos automaticamente a partir da informação associada ao piloto (licença e certificado médico atual). Os dados relativos às licenças e certificados médicos são: "nº licença piloto", "tipo licença piloto", "validade licença piloto", "nº certificado médico piloto", "classe certificado médico piloto", "validade certificado médico piloto", "nº licença instrutor", "tipo licença instrutor", "validade licença instrutor", "nº certificado médico instrutor", "classe certificado médico instrutor" e "validade certificado médico instrutor".

Depois de criado um movimento, o campo "confirmado" tem que ter o valor falso - indica que o movimento ainda não foi confirmado pela direção (está "pendente").

US 17. Como sócio piloto quero alterar ou remover movimentos relativos aos meus voos (em que sou piloto ou instrutor), que ainda não estejam confirmados pela direção. Os campos a



alterar são os mesmo que os campos para criar um movimento (US16). Se o movimento já estiver confirmado pela direção, não pode ser alterado ou removido.

- US 18. Como sócio piloto quero que ao acrescentar ou alterar um movimento, o tempo e o custo de voo seja calculado automaticamente, tendo em conta os valores arredondados da tabela que mapeia para cada aeronave, os valores do conta-horas com os tempos e preços de aluguer da aeronave (ver também US22)
- US 19. Como sócio piloto quero que ao acrescentar ou alterar um movimento (imediatamente antes de gravar), seja avisado da existência de possíveis conflitos nos valores do conta-horas inicial e final. O aviso deverá ser severo (por exemplo, usando uma cor vermelha) se houver uma sobreposição ou "normal" (por exemplo, usando a cor amarela) se corresponder a um buraco. Se houver um aviso de conflito, o sócio piloto poderá verificar, e se necessário alterar os dados do movimento, ou simplesmente guardar os dados com o conflito por resolver – note que o problema que originou o conflito poderá não ser da responsabilidade do movimento que está a ser inserido ou alterado. Caso grave os dados com o conflito por resolver, o registo do movimento deverá incluir o campo "conflito" com um de dois valores: "S - Sobreposição" e "B - buraco" e o campo "justificação conflito" com texto em que o piloto pode explicar o porquê de manter o conflito em aberto.
- US 20. Como sócio piloto, em complemento à US12, quero que ao filtrar/pesquisar a lista de movimentos, para além dos filtros "normais" tenha acesso a uma opção para filtrar apenas os meus movimentos (movimentos nos quais eu seja o piloto ou instrutor).

## **SÓCIO DA DIREÇÃO**

- US 21. Como sócio da direção quero acrescentar, alterar ou remover aeronaves do Aeroclube. Cada aeronave deverá incluir uma "matrícula", "marca", "modelo", "nº de lugares", " horas" (total de horas que a aeronave esteve em uso) e "preço hora".
- As aeronaves são removidas com *soft deletes* se estiverem associados a algum movimento. Caso contrário, os registos deverão ser apagados da base de dados.
- US 22. Como sócio da direção, em complemento à US21, quero manter a tabela que mapeia para cada aeronave, os valores do conta-horas (unidades de 1 a 10 – 10 corresponde a 1 hora) com os tempos e preços de aluguer. Esta tabela será usada para calcular tempos e preços dos movimentos, com os valores já arredondados.
- Por exemplo, se um voo tiver uma duração de 6 unidades no conta-horas, corresponde a 6/10 de hora, ou seja, 36 minutos. Se a aeronave custar 107€ por hora, o valor calculado seria 64,2€. Na tabela pode-se definir que 6 unidades de tempo nessa aeronave correspondem a 35 minutos e 65€, permitindo desta forma a utilização de valores arredondados e mais fáceis de gerir.
- US 23. Como sócio da direção quero consultar e filtrar/pesquisar a lista de sócios do aeroclube. O funcionamento desta listagem e filtro deverá ser similar à consulta e filtro de sócios pelos

sócios "normais" (US8 e US9), com uma diferença fundamental: deverão aparecer todos os sócios, incluindo os sócios que não estão ativos.

A informação a apresentar sobre cada sócio deverá incluir todos os dados referidos na US8, acrescidos dos dados "quotas em dia" (se tem as quotas em dia) e "sócio ativo". Também os campos a filtrar deverão incluir os campos referidos na US9, acrescidos das "quotas em dia" e "sócio ativo".

A lista de sócios apresentada à direção deverá incluir botões ou hiperligações para abrir a ficha (perfil) de sócio, ativar/desativar o sócio ou declarar quotas como pagas ou por pagar.

**US 24.** Como sócio da direção quero gerir os sócios do aeroclube, ou seja, criar novos sócios, alterar a informação de cada sócio ou remover sócios (*soft delete* se necessário). Os dados de cada sócio que podem ser alterados pela direção são o "nº de sócio", "nome informal" e "nome completo", "sexo", "data de nascimento", "e-mail" (a usar como credencial de acesso), "foto" (*upload* de imagem), "NIF", "telefone", "tipo de sócio" (P- piloto, NP – não piloto, A - aeromodelista), "quotas em dia" (indicação se as quotas estão ou não pagas) e "sócio ativo" (indicação se o utilizador associado ao sócio está ou não ativo) e "direção" (indicação se o sócio pertence ou não à direção).

Quando um sócio (utilizador) é criado, a senha de entrada deverá ser igual à data de nascimento do sócio no formato ano-mês-dia (exemplo: "1990-07-23").

Os sócios são removidos com *soft deletes* se estiverem associados a algum movimento (como piloto ou como instrutor). Caso contrário, os registos deverão ser apagados da base de dados.

**US 25.** Como sócio da direção quero que, depois de criar um novo sócio, a aplicação envie automaticamente para esse sócio uma mensagem de email para ativar a conta. Essa mensagem deverá incluir uma hiperligação de ativação da conta (US2).

Em complemento à US24, quero também que na página para alterar a informação dos sócios, esteja disponível um botão que permita reenviar para o sócio um email de ativação da conta.

**US 26.** Como sócio da direção quero fazer a gestão das quotas sem ter que abrir e alterar a ficha de cada sócio individualmente. Para esse efeito na lista de sócios (US23), e para cada sócio, deverei ter acesso a uma opção (botão / hiperligação) para declarar que se esse sócio tem ou não as quotas em dia. Deverei também ter acesso a uma opção para fazer *reset* à informação das quotas dos sócios, ou seja, declarar como não pagas todas as quotas de todos os sócios.

Note que, como a aplicação não faz a gestão de pagamentos nem do histórico das quotas pagas ou em atraso, considera-se que o campo "quotas em dia" refere-se ao facto do sócio ter ou não pago as quotas do ano atual. Assim, no início do ano é feito um "*reset*" e consideram-se todas as quotas como não pagas. Durante um determinado período as quotas ficam a pagamento e sempre que um sócio pagar as quotas, um elemento da direção deverá

declarar as quotas desse sócio como pagas. No fim do período de pagamento, os sócios que não pagaram as quotas poderão (se a direção assim desejar) ser declarados como sócios inativos.

- US 27. Como sócio da direção quero fazer a gestão dos sócios ativos ou não ativos, sem ter que abrir e alterar a ficha de cada sócio individualmente. Para esse efeito na lista de sócios (US23), e para cada sócio, deverei ter acesso a uma opção (botão / hiperligação) para ativar ou desativar esse sócio. Deverei também ter acesso a uma opção para desativar todos os sócios que não têm as quotas em dia.

Note que os sócios poderão ser desativados por não pagarem as quotas, por falta de pagamento dos alugueres das aeronaves ou por incumprimento de alguma regra interna do Aeroclube.

- US 28. Como sócio da direção quero alterar ou remover qualquer movimento de voo (de qualquer piloto) enquanto o movimento não está confirmado. Depois de confirmado um movimento, ele passa a ser imutável – não pode ser alterado ou removido. A alteração do movimento funciona de forma similar e usa os mesmos campos que a alteração de um movimento por parte dos pilotos (US17). A diferença é que os elementos da direção podem escolher qualquer piloto ou instrutor e têm também acesso à funcionalidade de confirmar o registo.

- US 29. Como sócio da direção quero confirmar qualquer movimento (voo) que ainda não esteja confirmado – depois de confirmado passa a ser imutável. Qualquer operação de confirmação de movimento deverá ela própria ser confirmada pelo utilizador antes de ser gravada na Base de Dados. A operação de confirmação de movimento deverá estar disponível aos elementos da direção no próprio registo do movimento (US28) - através de um botão ou hiperligação - ou selecionando vários movimentos na consulta à lista de movimento (US11) e aplicando a operação de confirmação a todos os movimentos selecionados (neste caso, o utilizador só confirma a operação uma vez, embora a operação possa afetar vários movimentos em simultâneo).

- US 30. Como sócio da direção quero consultar uma lista de assuntos pendentes, que apresente os movimentos com conflitos, e os movimentos, licenças de piloto e certificados médicos por validar. A lista deverá incluir a informação mínima que identifique univocamente cada elemento e uma hiperligação que possibilite ao utilizador saltar para o registo do movimento ou de sócio associado.

- US 31. Como sócio da direção quero consultar e gerir a informação específica dos sócios pilotos do aeroclube, ou seja, quando consultar, acrescentar ou alterar o registo de um sócio (US24) que é piloto, quero ter acesso à informação da licença de piloto e do certificado médico, bem como alterar essa informação, tal como acontece na US15. Quero também ter a opção de confirmar essa informação (da licença e do certificado em separado).

- US 32. Como sócio da direção quero gerir a lista de pilotos autorizado a voar cada aeronave do aeroclube. Essa gestão deverá ser feita por aeronave, numa página que poderá ser acessível

a partir de uma hiperligação na ficha da aeronave (US21) ou acrescentando na lista de aeronaves (US10) uma hiperligação para cada aeronave. A gestão deverá ser feita mantendo duas listas: "Pilotos Autorizados" e "Pilotos Não Autorizados", e acrescentando ou removendo pilotos destas listas – sempre que um piloto for acrescentado à lista de pilotos autorizados é removida da lista de não autorizados, e vice-versa.

## **DESAFIO**

**US 33.** Como utilizador/sócio normal quero obter informação sobre os movimentos (voos) das aeronaves do Aeroclube na forma de linhas temporais. Para cada aeronave deverão estar disponíveis 2 linhas temporais, uma com base nas datas (dias e/ou semanas e/ou meses) e outra com base no conta-horas da aeronave.

As linhas temporais deverão ser apresentadas na vertical da página, com os eventos (datas ou conta-horas) mais recentes no topo da página. Em ambos os casos, o espaçamento vertical deve ser uniforme – a unidade temporal a utilizar (por exemplo, o dia ou a semana ou o conta-horas) deverá ocupar sempre o mesmo espaço na vertical.

No caso de uma linha temporal com base em datas, os movimentos podem ser marcados sobre a linha temporal e ao lado dessa linha e na posição exata da data ou intervalo de datas (dia, semana, mês, etc.), deverá ser apresentada informação sobre o movimento ou movimentos que ocorreram nessa data ou intervalo de datas. A informação sobre o/s movimento/s poderá ser gráfica e/ou resumida e poderá incluir hiperligações para os detalhes do movimento.

No caso de uma linha temporal com base no conta-horas, cada movimento será representado por uma secção da própria linha temporal – as secções podem ser separadas por linhas, ou serem representadas por retângulos de cores diferenciadas. Ao lado da linha temporal deverá ser apresentada informação sobre o movimento associado, na forma gráfica e/ou resumida, mas incluindo sempre a data do voo. A representação do movimento poderá incluir uma hiperligação para o detalhe do movimento. Note que nesta linha temporal (com base em conta-horas) os conflitos de movimentos devem ser facilmente visualizados – os buracos aparecem como secções vazias da linha temporal e as sobreposições são representadas com 2 movimentos na mesma zona da linha temporal.

A apresentação das linhas temporais poderá ser feita uma aeronave de cada vez, ou várias em simultâneo e com um tipo de linha temporal ou ambos os tipos – a implementação mais simples envolve a apresentação de apenas uma aeronave e um tipo de linha, e implementações mais complexas podem envolver várias aeronaves em simultâneo e/ou vários tipos de linhas temporais. Independentemente da forma como é apresentada a informação, a decidir pelos estudantes, deverá ser possível filtrar por data de forma a limitar o tamanho da linha temporal – por exemplo, ver apenas os movimentos entre a data X e a data Y.

# REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais do projeto são os seguintes:

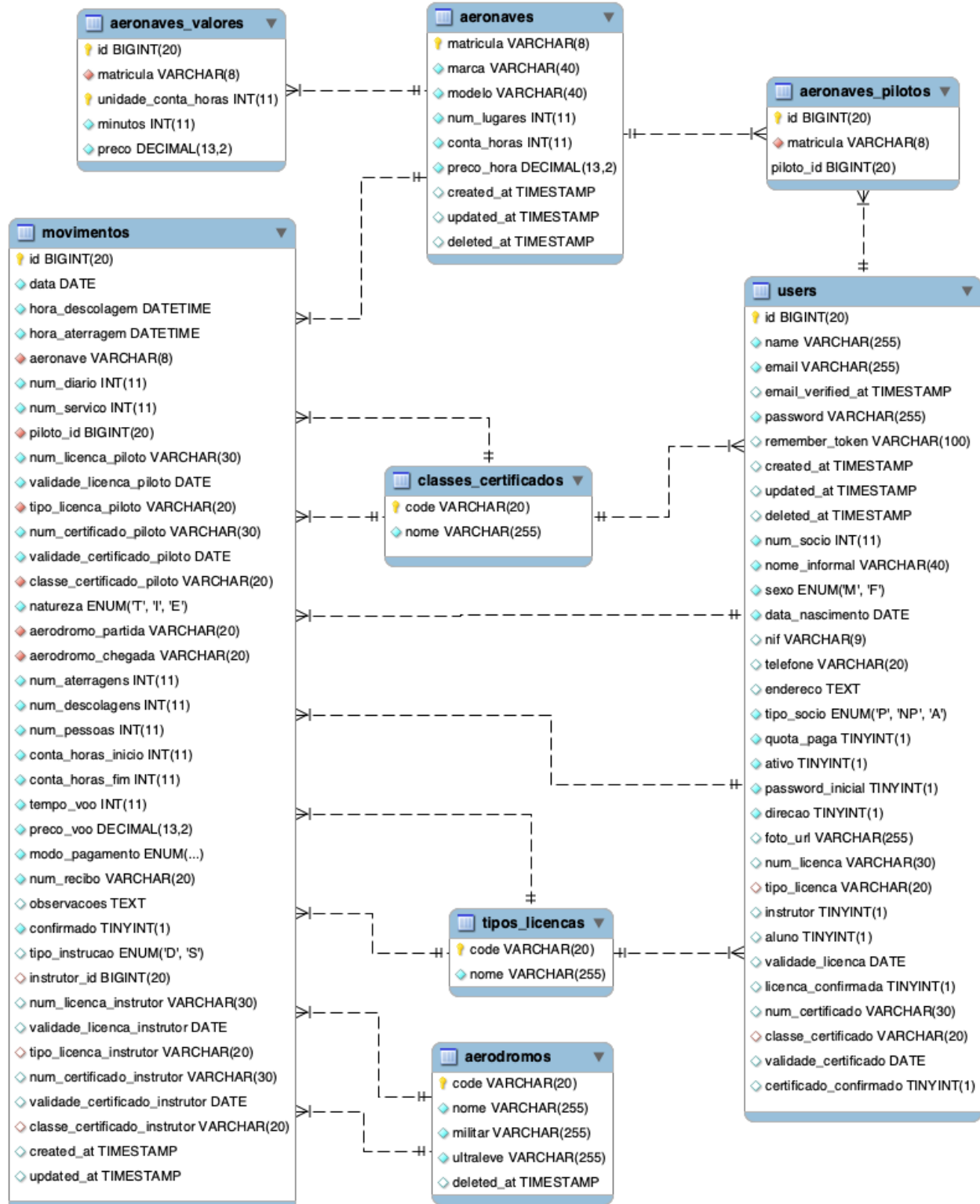
- RNF 1. Aplicação tem que usar a Framework Laravel 5.8;
- RNF 2. Aplicação tem que usar uma base de dados MySQL com a estrutura de dados fornecida. A estrutura não pode ser alterada (exceto correções publicadas no moodle);
- RNF 3. A estrutura, arquitetura e código da aplicação deve seguir as recomendações, convenções e boas práticas do Laravel, bem como as diretrizes PSR-2;
- RNF 4. Quando são apresentados dados na forma de listas ou tabelas, o nº potencial de registos (linhas) deverá ser tido em conta. Isto implica que quando o conjunto de dados tem alguma dimensão (por exemplo, os sócios ou movimentos) as listas e tabelas deverão ser paginadas e deverão permitir a ordenação por colunas;
- RNF 5. Os dados dos formulários deverão ser sempre validados no servidor, de acordo com as regras extrapoladas do cenário, requisitos funcionais (User Stories), estrutura da base de dados e da execução dos testes;
- RNF 6. A aparência e disposição dos elementos (layout) visuais deverá ser consistente ao longo de toda a aplicação e adaptada aos objetivos e ao contexto de utilização da mesma;
- RNF 7. A usabilidade da aplicação deverá ser uma das principais preocupações durante a implementação da mesma – a aplicação deve ser o mais "amigável" e simples possível;
- RNF 8. Aplicação deve ser o mais fiável possível;
- RNF 9. Aplicação deve ter o melhor desempenho possível;
- RNF 10. Aplicação deve ser o mais segura possível;
- RNF 11. A aplicação deve garantir a proteção dos dados e privacidade dos utilizadores, garantido que nenhum utilizador não autorizado pode ver ou alterar os dados ou documentos para os quais não deveria ter acesso.

## ESTRUTURA E DADOS DA BASE DE DADOS

A estrutura (ver diagrama ER na próxima página) da base de dados é fornecida pelos docentes (através de migrações) e inclui as tabelas:

- **users** – tabela com informação dos utilizadores ou sócios (neste projeto um utilizador é um sócio e vice-versa), incluindo os sócios pilotos e sócios que pertencem à direção. Inclui também colunas usadas pelo sistema de autenticação do Laravel;
- **tipos\_licencas, classes\_certificados, aerodromos** – tabelas com listas de valores possíveis para os tipos de licenças de piloto, classes de certificado médico e aeródromos. A utilizar exclusivamente para obter (ler) valores – os valores estão pré-preenchidos e no âmbito deste projeto não é necessário inserir, atualizar ou apagar registos destas tabelas;
- **aeronaves** – tabelas com as aeronaves do Aeroclube;

- **aeronaves\_pilotos** – tabela com a lista de pilotos autorizados a voar as aeronaves;
- **aeronaves\_valores** – tabela com o mapeamento entre as unidades do conta-horas e os valores de tempo (minutos) e de preço;
- **movimentos** – tabela com todos os movimentos (voos).





Para além da estrutura dos dados, são fornecidos dados de exemplo através da funcionalidade de *Database Seeding* do Laravel.

Para criar a estrutura da base de dados e preencher os dados, substituir a pasta "database" do Laravel com o conteúdo do ficheiro "database.zip" disponibilizado no Moodle. De seguida executar os comandos na raiz do projeto (a partir da *shell* acessível através do comando "homestead ssh"):

```
> cd <root/folder/of/Laravel/project>
> php artisan migrate
> composer dump-autoload
> php artisan db:seed
```

## TESTES E ROTAS

Serão fornecidos um conjunto de testes que vão validar o cumprimento da maioria das *User Stories*. Para tal, será necessário substituir a pasta "tests" do Laravel com o conteúdo do ficheiro "tests.zip" que será disponibilizado no Moodle. O conjunto inicial de testes será atualizado no decurso do desenvolvimento do projeto, até 2 semanas antes (**18 de Maio de 2019**) do prazo de entrega do projeto, pelo que é fundamental acompanhar (através do Moodle) as alterações aos testes.

Para executar os testes é obrigatório seguir as orientações referentes às rotas e formulários da tabela abaixo. Para além das rotas aqui especificadas, **os estudantes podem adicionar outras rotas** que considerem necessárias à implementação do projeto.

### **A nota do projeto depende da correta execução dos testes.**

#### *Considerações sobre as rotas:*

- A coluna "U.S." refere-se às User Stories que a rota poderá afetar (direta ou indiretamente). No entanto, este mapeamento depende da forma como o projeto é implementado e da forma como o projeto é analisado - não precisa de ser seguido à risca.
- A coluna "notas adicionais" poderá ter uma descrição mais ou menos completa de uma proposta de solução que não é obrigatória seguir à risca. No entanto, alguma da informação que contém (por exemplo, nomes dos campos) afeta os testes.

U.S.	Método / Verbo	Rota (route)	Notas adicionais
-	GET	/	Página inicial (não especificada em nenhuma <i>user story</i> ). Poderá incluir algum conteúdo específico ou redirecionar para outro conteúdo qualquer (à descrição do estudante).
1	GET	/login	Mostra o formulário para o utilizador introduzir as credenciais (email e senha)
1	POST	/login	Entra na aplicação com as credenciais (email e senha). Campos mínimos: <b>email e password</b> .
4	POST	/logout	Sai da aplicação.
3	GET	/password/reset	Mostra o formulário para o utilizador pedir para recuperar a password. Nesse formulário tem que introduzir o e-mail
3	POST	/password/email	Pedir para enviar para o e-mail um link para fazer o <i>reset</i> à senha de entrada. Campo mínimo: <b>email</b> .



3	GET	/password/reset/{token}	Mostra o formulário onde o utilizador irá recuperar a sua senha de entrada, passando o seu e-mail, a nova senha e a confirmação da nova senha.
3	POST	/password/reset	Altera a senha de entrada na sequência de um pedido de recuperação de senha de entrada. Campos mínimos: <b>email, password, password_confirmation</b> .
2	GET	/email/verify/{id}	Pedido HTTP invocado pela hiperligação do e-mail de ativação da conta (gerado pelo Laravel).
5	GET	/password	Mostra o formulário onde o utilizador pode alterar a sua senha
5	PATCH	/password	Altera a senha do utilizador atual. Campos mínimos: <b>old_password, password, password_confirmation</b> .
10, 21	GET	/aeronaves	Mostra a lista de aeronaves do Aeroclube
21, 22	GET	/aeronaves/{aeronave}/edit	Mostra os dados de uma aeronave do Aeroclube num formulário pré-preenchido. Inclui também o mapa que cruza as unidades do conta-horas com o tempo e preço
21, 22	GET	/aeronaves/create	Mostra o formulário para criar uma nova aeronave, incluindo também o mapa que cruza as unidades do conta-horas com o tempo e preço
21, 22	POST	/aeronaves	Cria uma nova aeronave. Inclui também o mapa que cruza as unidades do conta-horas com o tempo e preço. Campos mínimos: <b>matricula, marca, modelo, num_lugares, conta_horas, preco_hora, tempos[], precos[ ]</b>
21, 22	PUT	/aeronaves/{aeronave}	Altera uma aeronave. Inclui também o mapa que cruza as unidades do conta-horas com o tempo e preço. Os campos mínimos são os mesmos que os campos mínimos da rota anterior (POST /aeronaves)
21, 22	DELETE	/aeronaves/{aeronave}	Apaga uma aeronave. Apaga também o mapa que cruza as unidades do conta-horas com o tempo e preço. Se não for possível apagar a aeronave, faz um "softdelete" sem apagar o mapa.
32	GET	/aeronaves/{aeronave}/pilotos	Mostra a lista de pilotos autorizados a voar uma aeronave.
32	POST	/aeronaves/{aeronave}/pilotos/{piloto}	Acrescenta um piloto à lista de pilotos que estão autorizados a voar a aeronave.
32	DELETE	/aeronaves/{aeronave}/pilotos/{piloto}	Remove um piloto da lista de pilotos que estão autorizados a voar a aeronave.
18	GET	/aeronaves/{aeronave}/precos_tempos	Para uma determinada aeronave, devolve num objeto JSON a tabela que mapeia as unidades do conta-horas em tempo e preço (cada elemento da tabela é um objeto com 3 atributos: unidade, minutos, preço).
8, 9, 23, 24, 26, 27	GET	/socios	Mostra e filtra a lista de sócios (utilizadores). Se acedida por elementos da direção, deve incluir links e/ou botões para gerir as quotas, ativar/desativar sócios, criar novos sócios e abrir as fichas de sócios. Os filtros devem ser especificados na <i>query string</i> (exemplo: /socios?tipo=P&nome=Silva). Os parâmetros possíveis são: <b>num_socio, nome_informal, email, tipo, direcao, quotas_pagas, ativo</b> . Nota: os parâmetros nome_informal e email são relativos a filtros de sub-strings (texto inclui uma sub-string)
6, 7, 14, 15, 24, 31	GET	/socios/{socio}/edit	Mostra os dados de um sócio (utilizador) num formulário pré-preenchido – a utilizar nos perfis privados de sócios "normais" ou sócios pilotos (incluindo licenças e certificado médico). Utilizado também para a direção ver e alterar informação detalhada sobre um sócio, incluindo as licenças e certificados médicos dos pilotos, bem como para apagar o sócio ou reenviar um e-mail de ativação para esse sócio. A informação a mostrar e os campos que podem ser alterados dependem do utilizador atual
24, 31	GET	/socios/create	Mostra o formulário para criar um novo sócio (para qualquer tipo de sócio, incluindo piloto)
24, 25, 31	POST	/socios	Cria um novo sócio (qualquer tipo de sócio, incluindo piloto). Campos mínimos: <b>num_socio, name, nome_informal, sexo, email, nif, data_nascimento, telefone, endereco, tipo_socio, quota_paga, ativo, direcao, aluno, instrutor, num_licenca, tipo_licenca, validade_licenca, licenca_confirmada, num_certificado, classe_certificado, validade_certificado, certificado_confirmado</b> . De notar que os campos relacionados com os pilotos serão ignorados no caso dos sócios não serem pilotos. Para além dos campos atrás referidos (que também existem na tabela "users") o formulário deverá permitir o upload da foto do sócio e das cópias digitais da licença e certificado através dos campos (<input type="file" ...>) <b>file_foto, file_licenca e file_certificado</b> .

7, 15, 24, 31	PUT	/socios/{socio}	Altera a informação relativa a um sócio, incluindo a licença e certificado médico no caso dos pilotos. Os campos são os mesmos que os campos da rota anterior (POST /socios)
24	DELETE	/socios/{socio}	Apaga um sócio e toda a informação relacionada com o mesmo
26	PATCH	/socios/{socio}/quota	Declara uma quota de um sócio como paga ou "por pagar". Campo mínimo: <b>quota_paga</b> (true/false - paga ou não paga)
26	PATCH	/socios/reset_quotas	Declara as quotas de todos os sócios como "por pagar"
27	PATCH	/socios/{socio}/ativo	Ativa ou desativa um sócio. Campo mínimo: <b>ativo</b> (true/false – sócio ativo ou desativado)
27	PATCH	/socios/desativar_sem_quotas	Desativa todos os sócios com quotas "por pagar"
25	POST	/socios/{socio}/send_reactivate_mail	Envia novamente um e-mail de ativação da conta do sócio
14, 15, 31	GET	/pilotos/{piloto}/certificado	Imagem (PDF) do certificado médico do piloto. Apenas acessível ao próprio ou a um membro da direção
14, 15, 31	GET	/pilotos/{piloto}/licenca	Imagem (PDF) da licença de piloto. Apenas acessível ao próprio ou a um membro da direção
11, 12, 17, 20, 28, 29	GET	/movimentos	Mostra e filtra a lista de movimentos (voos). Caso o utilizador atual possa alterar ou apagar um movimento, este deve incluir uma hiperligação para o abrir. Se acedida por elementos da direção, deve permitir a seleção de vários movimentos e a confirmação simultânea dos mesmos. Os filtros devem ser especificados na <i>query string</i> (exemplo: /movimentos?aeronave=D-EAYV&data_inf=2018-06-23). Os parâmetros possíveis são: <b>id, aeronave, data_inf, data_sup, natureza, confirmado, piloto, instrutor, meus_movimentos</b> .
17, 18, 19, 28, 29	GET	/movimentos/{movimento}/edit	Mostra os dados de um movimento (voo) num formulário pré-preenchido – a informação a mostrar e os campos que podem ser alterados dependem do utilizador atual e do movimento estar ou não confirmado. Pode incluir opções (botões) para apagar o movimento ou para o confirmar.
16, 18, 19	GET	/movimentos/create	Mostra o formulário para criar um novo movimento (voo)
16, 18, 19	POST	/movimentos	Cria um novo movimento (voo). Campos mínimos: <b>data, hora_descolagem, hora_aterragem, aeronave, num_diario, num_servico, piloto_id, natureza, aerodromo_partida, aerodromo_chegada, num_aterragens, num_descolagens, num_pessoas, conta_horas_inicio, conta_horas_fim, tempo_voo, preco_voo, modo_pagamento, num_recibo, observacoes, tipo_instrucao, instrutor_id</b> Os campos relacionados com a instrução (tipo_instrucao e instrutor_id) deverão estar vazios (string vazia) no caso de a natureza do movimento ser diferente de Instrução. De notar que ao guardar os dados na base de dados, a aplicação deverá preencher automaticamente os campos num_licenca_piloto, tipo_licenca_piloto, validade_licenca_piloto, num_certificado_piloto, validade_certificado_piloto, num_licenca_instrutor, tipo_licenca_instrutor, validade_licenca_instrutor, num_certificado_instrutor, validade_certificado_instrutor.
17, 18, 19, 28, 29	PUT	/movimentos/{movimento}	Altera um movimento (desde que não esteja já confirmado). Os campos mínimos são os mesmos que os campos mínimos da rota anterior (POST /movimentos)
17, 28	DELETE	/movimentos/{movimento}	Apaga um movimento (desde que não esteja já confirmado).
13	GET	/movimentos/estatisticas	Mostra informação estatística dos movimentos. Pode incluir parâmetros na <i>query string</i> para definir o tipo ou tipos de dados ou gráficos serão apresentados.
30	GET	/pendentes	Lista os assuntos pendentes para tratamento pela direção, nomeadamente, os movimentos com conflitos e os movimentos, licenças e certificados por confirmar. Para cada assunto inclui uma hiperligação para abrir a ficha onde poderá tratar desse assunto.
33	GET	/aeronaves/{aeronave}/linha_temporal	Mostra as linhas temporais das aeronaves. Pode incluir parâmetros na <i>query string</i> para filtrar as aeronaves ou os limites temporais ou para definir a forma como a informação é apresentada.

## PASSOS PARA EXECUTAR OS TESTES:

1. Criar uma base de dados MySQL exclusivamente para testes, com o nome:  
`ainet_prj_testes`
2. Aplicar as "Migrations" do projeto à base de dados de testes:
  - a. No ficheiro ".env" do projeto, alterar o parâmetro com o nome da base de dados:  
`DB_DATABASE=ainet_prj_testes`
  - b. Executar o comando: `php artisan config:clear`
  - c. Aplicar as migrações com o comando: `php artisan migrate:fresh`
  - d. No ficheiro ".env" do projeto, voltar a alterar o parâmetro com o nome da base de dados, para a base de dados usada originalmente pelo projeto (`ainet_prj`).  
`DB_DATABASE= ainet_prj`
  - e. Voltar a executar o comando: `php artisan config:clear`
3. Substituir a pasta "tests" com o conteúdo fornecido pelos professores;
4. Mover o ficheiro phpunit.xml (fornecido pelos professores) para a pasta raiz do projeto;
5. Abrir o ficheiro ".env" e garantir que a opção "APP\_URL" tem o domínio do projeto;
6. Para correr todos os testes, na raiz do projeto (a partir da shell acessível através do comando "homestead ssh") executar o comando:  
`phpunit`
  - i. Se o comando não funcionar, experimente:  
`vendor/bin/phpunit`
7. Para correr apenas um teste executar o comando:  
`phpunit --filter TestName`  
por exemplo:  
`phpunit --filter US01`
8. Para correr os testes no formato "Dox" adicionar a flag `--testdox`. Exemplos:  
`phpunit --testdox`  
`phpunit --testdox --filter US04`

## NOTAS PARA ADEQUAR O PROJETO AOS TESTES

Para que os testes executem corretamente poderá ser necessário fazer alguns ajustes ao desenho (HTML) das páginas. Para minimizar esses ajustes sigam as seguintes regras:

- O projeto deverá ser desenvolvido em Português (nos testes são verificadas algumas páginas a partir de valores em português – por exemplo, se o utilizador não estiver ativo deverá incluir a palavra "desativado" no conteúdo HTML)
- No contexto dos testes, a ordem dos atributos HTML é relevante (apesar de não o ser no HTML). Por exemplo, os atributos "value", "checked" ou "selected" dos campos dos formulários devem ser os últimos atributos dos elementos "input" ou "option". Exemplos de HTML cuja ordem dos atributos é relevante (seguir a sintaxe aqui exemplificada):

```
<input type="text" name=". . ." value=". . .">
<form method="POST" action=". . .">
```

- Todos os valores dos atributos devem ser definidos com aspas (em vez de plicas). O seguinte HTML `<a href='someurl'>` poderá não passar nos testes porque usa plicas em vez de aspas. Em alternativa, o HTML `<a href="someurl">` já iria passar nos mesmos testes.
- No caso de implementarem paginações, o tamanho mínimo de cada página é de 15 elementos.

## CONFIGURAÇÃO

A configuração do projeto deverá ser feita através do ficheiro `.env` e deverá incluir (no mínimo) as seguintes configurações:

- O projeto deverá estar configurado para utilizar o endereço URL:  
`http://ainet.prj.test/`
- O projeto deverá utilizar uma base de dados MySQL com o nome: `ainet_prj`
- Os testes deverão usar a base de dados MySQL com o nome:  
`ainet_prj_testes` (criar uma BD com esse nome – ver secção sobre testes)
- O projeto deverá enviar e-mails (por exemplo, quando é criado um novo sócio) através de uma conta do serviço: `mailtrap.io`.
  - Criar uma conta no serviço (<https://mailtrap.io>)
  - Configurar o ficheiro `.env` para usar essa conta

## ENTREGA

A entrega do projeto deverá incluir 2 ficheiros (ambos obrigatórios):

- Ficheiro Excel com o **relatório** – o relatório inclui a identificação dos elementos dos grupos e informação sobre a implementação do projeto. O modelo do relatório irá ser publicado posteriormente pelos professores;
- Ficheiro zip com todo o **código** do projeto – este ficheiro inclui uma cópia de todos as pastas do Laravel, exceto a pasta "vendor".

# AVALIAÇÃO

Os critérios de avaliação do projeto são os seguintes:

Número	Peso	Critério
1	<b>85%</b>	Requisitos Funcionais
2	<b>15%</b>	Requisitos Não Funcionais

- Os requisitos funcionais serão avaliados considerando a implementação de todas as *User Stories* especificadas. A nota de cada *User Story* dependerá da qualidade, usabilidade, confiabilidade e desempenho da implementação. Para avaliar as *User Stories* serão efetuados testes automáticos (publicados ou não) e manuais, testes de usabilidade informais e análise ao código.

**A nota dos requisitos funcionais depende em parte da correta execução dos testes fornecidos.**

- O peso de cada uma das *User Stories* é o seguinte (total = 85%):

Nº U.S.	Peso (%)	Nº U.S.	Peso (%)	Nº U.S.	Peso (%)
1	1	12	4	23	1
2	1	13	6	24	3
3	1	14	2	25	1
4	1	15	3	26	2
5	1	16	4	27	2
6	2	17	2	28	2
7	3	18	3	29	4
8	2	19	4	30	3
9	3	20	1	31	1
10	1	21	3	32	4
11	3	22	3	33	8

- Os requisitos não funcionais serão avaliados globalmente para todo o projeto. Alguns serão analisados visualizando a estrutura e o código do projeto; outros, analisando a operação e o comportamento da aplicação e outros por meio de testes aleatórios (automáticos ou manuais), por exemplo, para verificar se a privacidade é mantida ou se aplicação é segura.