

ADC Projeto: Edição 2024/25

Exercício de Avaliação Individual

1. Âmbito e objetivo da avaliação individual

Âmbito

A avaliação individual visa a demonstração de competências individuais adquiridas pelos estudantes no decurso das sessões de formação, com particular enfoque nas seguintes competências:

- i. Desenvolvimento com a plataforma Google App. Engine (projetos Java / Maven), programação de operações REST (com POST) e gestão da persistência de dados com a utilização da tecnologia Google Datastore, bem com demonstração de competências para *deployment* local e *deployment* remoto (na Google Cloud Platform)
- ii. Desenvolvimento de serviços (*endpoints*) REST associados a um conjunto de operações, cuja correção e execução conforme os requisitos indicados, pode ser demonstrado com a ferramenta Postman. As operações requeridas são indicadas na secção 5 deste enunciado.

A avaliação pode envolver **opcionalmente como fatores adicionais valorativos**:

- iii. Desenvolvimento de aplicações-cliente Web utilizando Javascript e/ou Flutter (como aplicações de exemplo de demonstração das operações REST indicadas). Neste caso, os estudantes podem demonstrar uma ou mais operações REST com interação a partir de um Browser ou a partir de uma App Móvel, mesmo que rudimentar do ponto de vista da interface de utilização.
- iv. Desenvolvimento de operações REST que envolvem gestão de dados persistentes pode ainda adotar a solução Google Storage, como opção dos estudantes.

Objetivo.

O objetivo do trabalho é o desenvolvimento, teste e demonstração de um conjunto de operações REST a implementar, cobrindo os requisitos apresentados no presente enunciado e utilizando as tecnologias indicadas. Este objetivo visa aferir as competências individuais adquiridas pelos estudantes antes de iniciarem o desenvolvimento do projeto ADC em equipa. A aprovação na avaliação individual em sessão de avaliação individual é um requisito OBRIGATÓRIO para prosseguimento dos estudantes na unidade curricular na fase seguinte d desenvolvimento do projeto em equipa.

2. Materiais a desenvolver para o trabalho

Para realização e entrega do trabalho, os estudantes deverão preparar os seguintes materiais:

- **Realizarem o exercício de avaliação individual como projeto individual, com código fonte** em repositório e conta individual GitHub de cada estudante. Todos os elementos no repositório GitHub deverão ter data de atualização anterior à data e hora da sessão de avaliação individual

- **Preenchimento individual de um formulário (Google Form)** com respostas a questões em que cada estudante caracterizará os objetivos que alcançou, comparativamente aos requisitos do trabalho individual.

Nota: a ausência dos anteriores materiais conforme indicado implica em FALTA e NÃO APROVAÇÃO na avaliação individual

Estrutura do projeto (GitHub) e identificação do projeto de avaliação individual

Cada aluno deverá desenvolver, apresentar e ter disponível para demonstração na sessão de avaliação, o repositório GitHub acima indicado. O repositório do projeto deve ser identificado com o título **APDC-2025-Individual-NNNN**, em que NNNN é o número de aluno. O projeto deverá ter uma estrutura de um projeto Maven, podendo conter uma diretoria suplementar na raiz do projeto, designada “INFO”, onde

os estudantes podem incluir quaisquer ficheiros com informação ou indicações que julguem importantes. Os repositórios deverão estar prontos para serem descarregados para um IDE (ou para clonagem na linha de comando), logo no início da sessão de demonstração e avaliação presencial.

3. Processo de avaliação nas sessões de avaliação individual

- Todos os estudantes serão escalonados para sessões individuais de avaliação que ocorrerão na presença de pelo menos um dos docentes. Na avaliação individual seguir-se-á um guião de avaliação em que será assinalada e verificada pelo docente cada um dos passos da avaliação. Estes passos permitirão demonstrar todas as operações suportadas e a sua conformidade com os requisitos.
- A avaliação do trabalho decorre da avaliação dos materiais indicados anteriormente
 - Resposta ao formulário.
 - Cumprimento dos objetivos e requisitos obrigatórios para implementação.
 - Valorização adicional de objetivos opcionais no caso de serem realizados.

A avaliação ainda terá em conta:

- Demonstração presencial e discussão sobre os objetivos realizados ou respostas a questões que forem colocadas durante a sessão de avaliação.

A avaliação envolve o preenchimento do Guião de Demonstração, com a sequência de passos a demonstrar, em que se registará o cumprimento ou não dos objetivos e a execução e correção das operações indicadas nos requisitos de implementação.

4. Datas importantes

Exercício de Avaliação Individual (ADC, 2025)

O período de desenvolvimento do trabalho individual decore entre 4/Abril/2025 e 14/Abril/2025, devendo os estudantes organizarem a sua agenda tendo em conta st período e as datas importantes a seguir indicadas:

Desenvolvimento do trabalho (código no repositório GitHub):	DEADLINE: Até 2ª Feira 14/Abril/2024, 12h30 (Data máxima de todos os elementos no repositório GitHub de cada estudante.
Indicação do escalonamento dos alunos para sessões individuais de avaliação e salas onde realizarão as avaliações individuais	Será anunciada (através de mensagem enviada pelo sistema CLIP) a partir de 2ª Feira 7/Abril
URL e formulário para preenchimento dos alunos sobre referenciamento dos objetivos realizados e requisitos cumpridos no trabalho realizado, conforme os requisitos indicados a seguir	URL será anunciado (através de mensagem enviada pelo sistema CLIP) partir de 2ª Feira 7/Abril, sendo o formulário aberto para respostas a partir dessa data. O formulário deverá ser preenchido até ao início das sessões de avaliação individual, sendo o fechado na 3ª Feira 13/Abril.
Sessões de avaliação individual	Serão escalonadas, com indicação dos alunos em cada sessão. As sessões serão distribuídas entre 3ª Feira 15/Abril e 4ª Feira 16/Abril. informação do escalonamento será comunicada por mensagem (através do sistema CLIP) a partir de 4ª Feira 9/Abril

A aplicação e operações a desenvolver implicam demonstração a partir de um componente CLIENTE (lado cliente) e um componente SERVIDOR (lado SERVIÇO CLOUD)

Lado CLIENTE:

- **No mínimo, de forma obrigatória**, as operações a executar do lado do CLIENTE podem ser demonstradas com a ferramenta Postman para demonstrar as operações necessárias). Neste caso é relevante que os estudantes preparem as configurações de modo a facilitar a demonstração com base em invocações RST (POST) já pré-configuradas com argumentos de entrada (dados JSON), para obviarem a demonstração das diversas operações implementadas durante a sessão de demonstração.
- **Opcionalmente** a demonstração pode ser suportada com desenvolvimento de aplicação em ambiente Web-Browser ou App móvel desenvolvida com Flutter). Neste caso podem mostrar-se uma ou mais operações, podendo a demonstração envolver a combinação do uso da aplicação (para as operações suportadas) e as restantes não suportadas na aplicação) serem mostradas com o Postman.

Lado SERVIDOR (ou SERVIÇO COUD):

- Este componente é obrigatório devendo suportar a funcionalidade REST e estar suportado no Cloud App Engine, dados persistentes (Google Datastore) ou opcionalmente Google Datastore e Google Cloud Storage. O demonstrador deverá ser desenvolvido de forma a poder ser compilado, executar com *deploy* local e demonstrado em *deploy* remoto na Google Cloud Platform, no tempo estipulado, logo no início da sessão de avaliação presencial.
- Será conveniente que para efeitos da demonstração das operações a suportar que a aplicação do lado servidor esteja apta a usar já uma base de dados pré-criada e vazia, associada ao projeto par deploy remoto.

No início da sessão de demonstração, cada aluno partirá de um projeto “vazio” (deve descarregar o projeto do GitHub para o IDE que pode ser Eclipse, IntelliJ ou qualquer outro à escolha do estudante) e estar pronto para usar o ambiente de linha de comando (Shell) para demonstrar o deploy local e remoto

Durante a avaliação individual, cada estudante passará pelos seguintes passos nos seguintes tempos de referência, pelo que durante a preparação os estudantes devem treinar a sua demonstração com base nos tempos indicados na seguinte tabela:

1. Demonstração de criação do projeto com clonagem do projeto no IDE (ou ambiente de linha de comando SHELL), recorrendo aos <i>sources</i> no repositório individual GitHub no qual se verificarão as datas do conteúdo, com o respetivo descarregamento a partir da plataforma GitHub e repositório individual	~3 minutos
2. Compilar localmente, demonstrando não existirem erros de compilação/build e demonstração de conseguirem realizar o <i>deploy</i> local para acessibilidade HTTP local (ainda que não seja este o <i>deploy</i> depois usado para a restante demonstração)	Entre 1 a 2 minutos
3. <i>Deployment</i> remoto e demonstração da acessibilidade HTTPS (URL) na Google Cloud Platform: com demonstração de prontidão para execução na nuvem (GAE+Storage)	De ~3 a 5 minutos
4. Demonstração da sequência de operações REST com verificação do cumprimento dos respetivos requisitos enunciados obrigatórios na secção 5 seguinte. Esta parte implica a demonstração sucessiva de 10 passos de um guião de teste a seguir na prova de avaliação presencial	~15-20 min
5. Apresentação e demonstração de operações/requisitos adicionais opcionais para verificação e avaliação valorativa (apenas no caso dos estudantes que tenham realizado essas operações opcionais valorativas)	Entre 1 e 2 minutos por operação

Na sessão de avaliação e demonstração cada estudante deve estar pronto para esclarecer ou dar resposta a qualquer questão sobre a sua implementação ou respetivo código desenvolvido.

Está previsto que as sessões individuais de demonstração possam decorrer até 30 minutos, para se completar todos os passos de verificação do Guião de Avaliação Individual. Em cada sessão de avaliação, cerca de 5 a 6 estudantes realizarão em paralelo a sua avaliação individual. De acordo com esta previsão, os estudantes serão escalonados para *slots* de 1 hora, como máximo tempo de referência.

5. Operações e requisitos a desenvolver para serem demonstradas

Requisitos do componente Cliente

Funcionalidade do lado cliente: as operações REST indicadas à frente podem ser demonstradas com a ferramenta Postman (como requisito mínimo obrigatório). Para efeitos de demonstração de operações REST (POST) que devem ser desenvolvidas no exercício de avaliação individual será importante que os estudantes tenham as operações POST configuradas no Postman com os dados (JSON), para ser mais fácil e mais rápido

Exercício de Avaliação Individual (ADC, 2025)

fazerem a demonstração da sua execução, com dados de entrada (JSON) pré-configurados para exemplificação (independentemente desses dados poderem ter que ser modificados pontualmente durante as demonstrações).

Opcionalmente, no caso de demonstração do cliente com aplicação JAVASCRIPT FLUTTER, os estudantes devem escolher o browser da sua preferência e que tenham usado e testado durante os desenvolvimentos (ex, Google *Chrome*, Mozilla *Firefox*, Apple *Safari*, MS Edge, Opera, Brave, etc.)

Requisitos do componente servidor (solução CLOUD)

Deverá executar no ambiente Cloud (*Google App Engine + Persistência com Datastore*) ou opcionalmente podendo envolver o Google Cloud Storage, sendo o deploy remoto demonstrado em conta Google Cloud Platform do estudante.

Sobre o uso de *tokens* para controlo de acesso nas operações REST

Notar que todas as operações a implementar devem ser feitas com pedidos POST, usando-se sempre que necessário tokens de autenticação, (obtidos em operação de LOGIN). Tal significa que as operações são solicitadas pelo cliente, em sessões autenticadas. Assim, as operações devem enviar ao servidor o Token de autenticação obtido na operação de LOGIN (em que o utilizador forneceu um username e password verificados), devolvendo a operação de LOGIN o token de autenticação no caso do LOGIN ter sido realizado com sucesso.

Como referência inicial (que pode ser estendida pelos estudantes), deve usar-se a seguinte estrutura para os Tokens de autenticação quando usados nas operações que os devem usar em sessões após obtenção do Token em operação prévia de LOGIN:

<Output> = operation(<Input>, <Token>) em que:

operation: <opid> // Código ou identificador de operação

// po

Input: input JSON data // Dados JSON – argumentos da operação REST

Output: output JSON do resultado da operação // Retorno (dados JSON) da operação REST

Se o <Token> enviado numa operação REST após autenticação estiver correto e válido (correspondendo assim a um token efetivamente emitido pelo servidor numa operação válida de LOGIN com sucesso e assim correspondente a uma sessão autenticada válida do utilizador). Cada operação que utilize o token válido (como operação realizada numa sessão autenticada) será atendida pelo servidor (desde que o utilizador seja suposto ter as permissões para a operação, dado o seu Role (papel). Noutro caso, a operação deve resultar em erro

(seja por não validade da sessão, por *token* inválido, por *token* de sessão expirada, ou por Role não adequado). O retorno da operação em causa deve originar um resultado que refletirá o erro em cada caso.

Para implementação dos tokens de autenticação, deve usar-se (como base de inspiração) a utilização de tokens nas exemplificações nas sessões de formação. Os tokens podem conter os elementos que considerados necessários para garantir a segurança da autenticação dos utilizadores (em sessões autenticadas) e o controlo de acesso às operações (de acordo com o role do utilizador na sua conta de LOGIN). Os estudantes devem criar a estrutura de dados JSON adequada com as informações necessárias para esse controlo, com todos os elementos do token considerados relevantes para o efeito.

OP1: Criação e registo de contas de utilizadores **Operação REST, POST**

Deve suportar-se o registo e criação de até 4 contas de utilizadores na aplicação (devendo ser mostrada a possibilidade de registo de alguns utilizadores). O registo deve permitir que na criação de contas os utilizadores registem atributos (à escolha dos alunos), mas que devem incluir pelo menos os seguintes dados (independentemente de outros que os alunos considerem como valorização):

Atributos obrigatórios para criação e registo das diferentes contas a prever

- *Email* do utilizador: terá que ter formato <string>@<string>. <dom>
 - <dom> representa um *top-level domain* DNS
 - Exemplo: petermurphy3456@campus.fct.unl.pt
- *Username* (ou Nickname), exemplo: petermurphy3456: <String>
- Nome completo: <String> <String> <String> etc
 - Exemplo: Manuel Francisco da Silva Marques
- *Telefone*: número de telefone (exemplo +3512895629)
- *Password*: PWD
 - Deve conter uma combinação de caracteres alfabéticos (maiúsculas e minúsculas) , numéricos e sinais de pontuação (combinação do tamanho ou regra que pode ser definida por cada estudante)
 - Ex: 2025adcAVALind!!!
 - A inserção da PWD durante o registo deve sempre ser confirmada duas vezes
- Perfil da conta: pode ser “público” ou “privado”

Atributos adicionais (sem preenchimento obrigatório na criação e registo das diferentes contas a prever (isto é que não são de preenchimento obrigatório na operação de registo de contas, mas que podem ser atualizados depois pelo utilizador)

- Número de cartão do cidadão
- ROLE
 - Pode ser ENDUSER, BACKOFFICE, ADMIN, PARTNER
- NIF do utilizador
- Entidade empregadora
 - Exemplo: Smart Forest S.A.
- Função
 - Exemplo: Professor, Estudante, Gerente, Técnico de Floresta, etc.
- Morada
 - Exemplo: “Rua dos alunos de APDC2324, No 100, Piso 2, Porta 116)
- NIF da entidade empregadora
 - Exemplo: 177888999
- Estado da conta

Exercício de Avaliação Individual (ADC, 2025)

- Deve ter um dos seguintes valores: ATIVADA, SUSPENSA ou DESATIVADA

Atributos adicionais opcionais valorativos na criação e registo das diferentes contas a prever (isto é que não são de preenchimento obrigatório na operação de registo de contas, mas que podem ser atualizados depois pelo utilizador)

- Foto (jpeg)

As contas, após serem criadas, terão sempre estes dois atributos com os seguintes valores:

ROLE: enduser

ESTADO DA CONTA: DESATIVADA

O ROLE inicialmente criado com o valor enduser pode depois ser modificado para um dos outros ROLES, mas esta operação só pode ser feita por um utilizador com ROLE backoffice ou admin.

Notas complementares:

- Não vai ser preciso no exercício confirmarem a conta com confirmação do Email como é habitual em muitas aplicações, para efeitos de ativação da conta. Também não será preciso verificar se no Email fornecido é válido, nem testar que é possível mandar EMail ao utilizador da conta registada ou enviar SMS para o telefone fornecido. Todos estes aspetos ou demais fatores de confirmação da abertura de contas, a serem suportados, serão sempre considerados elementos opcionais se os estudantes os quiserem considerar.
- Como estratégia, em tempo de *deployment* da aplicação, a aplicação deve logo criar automaticamente uma conta de um utilizador com *username* "root", e role ADMIN *password* pré-inicializada bem como os restantes campos obrigatório como acima, devendo a respetiva conta ficar logo registada como conta em estado ATIVADA na base de dados e de modo que esse utilizador possa fazer operações como as que se referem a seguir.

OP2: Login Operação REST, POST

Operação de "Login" de sessão

O utilizador fornece o IDENTIFICADOR (ex., *Username* ou *Email*) e a *password*. Se as credenciais são corretas (autenticação pelo servidor): recebe um *token* para a sessão que entre quaisquer outros atributos terá

- USER: o identificador (ou *username*, *UserID* ou *Email*)
- ROLE: o role (ou código de role)
- VALIDITY: validade do token (correspondente à validade da sessão), com as propriedades:

- VALID_FROM: data da emissão (tempo do servidor)
- VALID_TO: data de expiração (tempo do servidor)
- VERIFICADOR: VERIF VERIFICADOR: ex., um “*magic (random) number*” emitido pelo servidor que traduz uma prova de autenticidade da emissão pelo servidor dos dados do *token*, para realização de operações na sessão de LOGIN. A forma de criar este VERIFICADOR é opcional, devendo ser definida por cada aluno.

Quando o utilizador conclui com sucesso o LOGIN tem acesso a uma “página de boas-vindas”, que indica de forma visível qual o PAPEL em que pode atuar e em que pode realizar as restantes operações suportadas para o seu papel (ver abaixo)

OP3: ChangeRole (Mudança de role de utilizador)

Operação REST, POST

Esta operação permite mudar o role de utilizadores de contas criadas, mas só pode ser executada de acordo com as seguintes permissões:

- BACKOFFICE: pode passar conta de role ENDUSER para PARTNER (ou vice-versa)
- ADMIN pode passar qualquer conta de qualquer role para qualquer role
- ENDUSER: não pode mudar role de nenhuma conta

OP4: ChangeAccountState (Mudança de estado de conta)

Operação REST, POST

Esta operação permite mudar o estado de contas de ACTIVADA para DESATIVADA (ou vice-versa) mas só pode ser executada com as seguintes permissões a partir do role do utilizador que a executa

- ADMIN pode mudar o estado de qualquer conta, de qualquer estado para qualquer estado.
- BACKOFFICE: pode passar o estado de contas de DESATIVADA para ATIVADA ou vice-versa.

OP5: ReoveUserAccount (Remoção de contas de utilizadores)

Operação REST, POST

Esta operação serve para remover contas da aplicação, devendo obedecer aos seguintes requisitos de autorização:

- ADMIN: pode remover qualquer conta dado o userID ou Email da conta
- BACKOFFICE: pode remover qualquer conta dado o userID ou Email da conta, desde que a conta tenha Roles ENDUSER ou PARTNER

OP6: ListUsers (Listar utilizadores com contas registadas)

Operação REST, POST

Esta operação serve para listar os utilizadores (contas) registadas, mostrando os seguintes atributos:

- Utilizadores com role ENDUSER: podem listar os utilizadores registados e que têm perfil ENDUSER, mostrando apenas os atributos: *username*, *email* e nome (mas apenas os que têm role ENDUSER) e só se as respetivas contas estiverem com perfil público e as contas em estado ATIVO.
- Utilizadores com role BACKOFFICE: podem listar todos os atributos de utilizadores registados (mas apenas os que têm role ENDUSER) e independentemente de estarem com perfil público ou privado ou do estado da conta.
- Utilizadores com role ADMIN: podem listar todos os atributos de utilizadores registados, independentemente de estarem com perfil público ou privado, estado da conta ou role.

Quando haja atributos não preenchidos, a listagem deve indicar para esses atributos “NOT DEFINED”

OP7: ChangeAccountAttributes (Modificação de atributos de contas de utilizadores)
Operação REST, POST

Operação executável por utilizadores com qualquer role, que permite modificar os respectivos atributos da conta, do seguinte modo:

- Utilizadores com role ENDUSER: podem modificar todos os atributos da própria conta ou adicionar informação de atributos opcionais não definidos, exceto os atributos *Username* (ou UserID), *Email* ou Nome (inicialmente registados na criação da conta desde que ATIVADA, mas não podem mudar os atributos de controlo tal como ROLE e ESTADO)
- Utilizadores com role BACKOFFICE de conta ativada: podem modificar todos os atributos ou acrescentar informação de atributos não definidos de contas de utilizadores ENDUSER ou PARTNER, exceto os atributos *Username* (ou UserID) ou EMAIL
- Utilizadores com role ADMIN: podem modificar todos os atributos de qualquer conta com qualquer ROLE ou adicionar informação de atributos opcionais não definidos em contas de qualquer ROLE, podendo mudar os atributos de controlo tal como ROLE e ESTADO

OP8: ChangePassword (Modificação de password do utilizador)
Operação REST, POST

Operação de mudança de *password*. Exige a *password* atual e a inserção da nova *password*, com dupla verificação (inserção da password duas vezes com comparação entre elas). Esta operação só pode ser suportada por cada utilizador na sua própria conta (independentemente do seu ROLE)

OP9: SessionLogout (LOGOUT de uma sessão)
Operação REST, POST

Operação de “Logout” da sessão

Cliente executa a operação e como resultado o servidor revoga o TOKEN (mesmo que este ainda esteja válido) e a sessão é redirecionada no cliente para uma página que notifica LOGOUT (desaparecendo a informação que se “via” na sessão aberta). Após o *Logout*, não é possível realizar operações em nome do utilizador (mesmo que se reutilize um token que ainda estava válido), a não ser que o utilizador faça de novo LOGIN com sucesso em nova sessão e obtenha assim um novo TOKEN válido. Assim, após realização de uma operação de LOGOUT, um utilizador só pode realizar novas operações após nova operação de LOGIN

OP10: CreateWorkSheet (Registrar ou modificar folhas de obra)
Operação REST, POST

Esta operação visa permitir registar folhas de obra associadas a processos de reordenamento do território para que possam ser alocadas a entidades que as vão executar (contas PARTNER).

Uma folha de obra deve ter os seguintes atributos:

Atributos de registo de obra

Atributos obrigatórios na criação de uma folha de obra

Referência da obra: uma identificação, ex: O234/CM/2024
Descrição: uma frase com sumário de descrição da obra, ex: “Obra para reflorestação da propriedade P34567 com registo de parcela de terreno rústico RP-PTR-2459
Tipo de alvo da obra: Propriedade Pública ou Propriedade Privada
Estado de adjudicação: ADJUDICADO ou NÃO ADJUDICADO

Atributos adicionais que serão preenchidos quando a obra for atribuída após adjudicação

Se a obra foi adjudicada, os seguintes campos devem ser preenchidos. Notar que se uma folha de obra criada não foi adjudicada os campos não devem ter valores preenchidos.

Data de Adjudicação: Data

Data prevista de início da obra: Data

Data prevista de conclusão da obra: Data

Conta de entidade: identificador de conta de utilização com role PARTNER

Entidade de adjudicação: Nome da Empresa, ex., Reflorestação Inteligente S.A.

NIF da empresa; ex., 511876234

Estado da Obra: “NÃO INICIADO”, “EM CURSO” ou “CONCLUÍDO”

Observações: Descrição sumária informativa sobre a obra complementando a informação do estado da obra

O registo de obras ou atualização de tributos de adjudicação só pode ser feita por utilizadores com role BACKOFFICE

Modificação dos atributos de Estado da Obra pode ser feita por utilizadores com role PARTNER a quem tenha sido adjudicada a obra.

Operações opcionais (não obrigatórias) que os estudantes podem desenvolver adicionalmente e que serão usadas para efeitos valorativos

Para além das operações anteriores que são obrigatórias, os estudantes podem fazer até duas operações opcionais adicionais se assim o pretenderem (funcionalidade à escolha de cada aluno). As operações devem ser demonstradas e executadas numa sessão LOGIN de um utilizador, com o necessário controlo de permissão que for estabelecido pelo ROLE necessário.

As seguintes operações são exemplos de operações opcionais indicativas. Os alunos podem escolher e implementar qualquer uma das indicadas ou qualquer outra à sua escolha.

Uma das operações interessantes será permitir a um utilizador com role BACKOFFICE listar todas as folhas de obra ou listar folhas de obra dado o estado da folha de obra, exemplo: Obras concluídas, obras adjudicadas, obras registadas por adjudicar, Obras previstas para conclusão na data X.

Outra das operações interessantes seria permitir a um utilizador com role BACKOFFICE listar todas as operações num histórico de modificação de estado de folhas de uma dada obra, como notificações ou comentários por parte da entidade que está a realizar a obra e durante a realização da obra. Exemplo de notificações:

Notificação 1: 23/Abri/2025, 17h00: A reflorestação da área C foi suspensa na data XXX devido à chuva intensa e estado não adequado do terreno”

Exercício de Avaliação Individual (ADC, 2025)

Notificação 2: 7/Maio/2025. 14h2: A intervenção da máquina de arrasto na área D da obra foi interrompida devido a avaria da máquina que estava a ser utilizada, estando previsto que a situação seja ultrapassada em 48 horas.

Notificação 3: 9/Maio/2025, 10h15: A reflorestação da área C foi reiniciada, após verificação de adequabilidade do estado no terreno

6. Critérios de referência para a avaliação

Como indicado (*ver quadro inicial de Datas Importantes*) será depois disponibilizado um guião associado a este enunciado. Este guião estará associado a passos de testes para demonstradores que cobrirá as funcionalidades indicadas (tendo como referência base os requisitos obrigatórios). Assim, os requisitos serão demonstrados “passo a passo” obedecendo a tempos definidos. O sucesso de realização de cada passo da demonstração e o cumprimento do tempo definido para cada passo são fatores que determinam a avaliação individual. Cada passo realizado e demonstrado no guião de avaliação terá uma pontuação que será como referência a seguinte, tendo por base a demonstração nos tempos médios de referência anteriormente indicados

DEMONSTRAÇÕES INICIAIS: total 25 pontos

- De início: ter realizado o formulário do trabalho e ter o código no repositório Github com as datas de desenvolvimento indicadas, estando o repositório descarregável para a sessão de demonstração (10 pontos)
- Conseguir fazer download do projeto para um projeto vazio (IDE) na sessão de avaliação presencial, fazer a compilação e conseguir fazer o deploy local (5 pontos)

- Conseguir fazer *deployment sem erros* para o ambiente *Cloud* (GAE + Persistência) e mostrar que está pronto a operar (10 pontos)

OPERAÇÕES BASE OBRIGATÓRIAS: total 75 pontos

- Demonstrar em sequência o suporte e correção das operações OP1 a OP10: 7,5 pontos por operação suportada e demonstrada na sessão de avaliação

OPERAÇÕES EXTRA: 7,5 pontos por cada operação opcional extra

- Demonstrar em sequência o suporte e correção das operações OP1 a OP10: 7,5 pontos por cada operação extra suportada e demonstrada na sessão de avaliação

AValiação FINAL INDIVIDUAL A ATRIBUIR

- A avaliação final converterá os pontos anteriores em escala de 0 a 20 valores (que podem ser obtidos com um total de 100 pontos). Notar que a soma de todos os pontos (incluindo operações extra) pode totalizar mais de 100 pontos. Neste caso, um total de pontos superior a 100 pontos corresponderá à nota máxima de 20 valores
- A não realização da avaliação individual ou não obtenção de nota positiva (superior a 9,5 valores) implica reprovação à unidade curricular (frequência).
- A avaliação individual conta 20% na nota final de PIPP+ADC (nota única combinada das duas UC).