

## Exame Prático de avaliação final, Ep. Normal

### Introdução

Pretende-se avaliar o aluno quanto às competências adquiridas durante a disciplina. Assim, com este trabalho, o aluno deve demonstrar estudo, análise crítica, desenho e desenvolvimento de aplicações e serviços Web.

O trabalho pode ser desenvolvido individualmente ou em grupo com até 3 elementos. É possível outras configurações se propostas e devidamente justificadas ao docente, estando sujeitas a aprovação. Quando o trabalho for desenvolvido em grupo cada elemento deve identificar corretamente durante a apresentação o seu contributo e responsabilidade.

O trabalho está dividido em três partes, todas de entrega e apresentação obrigatória.

### Parte 1 [ 4 valores ]

1. Escolha uma API Web para estudar (Pode consultar uma lista de APIs disponíveis em <https://github.com/public-apis/public-apis>) e sugira ao docente a escolha efetuada. Tenha em atenção os requisitos listados abaixo.
  - Obrigatórios:
    - Arquitetura de serviços do tipo REST;
    - Utilização de mais do que um Verbo (método) do protocolo HTTP;
  - Preferenciais:
    - Com disponibilização de mais do que um formato de mensagem (XML e JSON);
    - Com Autenticação (requisito obrigatório se em grupo).

Elabore uma apresentação [ **2 Valores** ] e um conjunto de pedidos em Ficheiro *Postman Collection* [ **1 valor** ] descrevendo API Web selecionada. Deve conter pelo menos:

- Descrição dos objetivos da API Web
- Descrição do modo de funcionamento da API Web, incluindo exemplos funcionais;
- Descrição de um mínimo de 5 funções da API Web;
- Avaliação e análise crítica da API Web.

Durante a apresentação presencial [ **1 Valor** ] deve demonstrar pelo menos um exemplo do funcionamento da API, dando realce a método de autenticação (se existir) e identificação de relações do tipo “1 para muitos” (one-to-many ou many-to-one) entre dois dos recursos disponibilizados.

**Resultados a entregar:** Apresentação (PDF) e Ficheiro *Postman Collection*.

### Parte 2 [ 8 valores ]

Defina o tema do seu trabalho e sugira ao docente a sua escolha. Prepare a base de dados de suporte aos seus serviços e preencha-a com dados suficientes para a apresentação do seu trabalho. Considera-se que 30 registos por tabela é um volume de dados suficiente.

2. Utilize uma estratégia *Design-first* conforme lecionada para o desenvolvimento de uma camada de serviços REST [ **6 valores** ] para disponibilizar os recursos para uma aplicação Web. Tenha em atenção os seguintes requisitos mínimos obrigatórios:
  - Arquitetura de serviços do tipo REST;
  - Utilização de pelo menos 4 verbos (métodos) do protocolo HTTP - CRUD (do inglês: Criar, Ler, Atualizar e Apagar);
  - Disponibilizar pelo menos 4 recursos diferentes;
  - Utilização de pelo menos uma relação de cardinalidade 1:n entre dois dos recursos;
  - Disponibilização das representações de estado dos recursos em JSON;
  - Documentar a API com recurso ao formato OpenAPI 3.0;

## Exame Prático de avaliação final, Ep. Normal

- Disponibilizar uma ‘Collection’ para o Postman para consulta dos recursos;
- Utilização do MySQL como SGBD;
- Utilização de Node.js como servidor aplicacional para implementação da camada de serviços.
- Disponibilização de configuração para a aplicação *multi-container* (pelo menos duas imagens) MySQL + NodeJs.

Valoriza-se o desenvolvimento/utilização de:

- Filtros através de parâmetros HTTP;
- Utilização de relação de cardinalidade m:n entre dois dos recursos;
- Relatório do trabalho [ 2 valores ].

**Resultados a entregar:** API REST, e Relatório.

## Parte 3 [8 valores]

Pode usar o mesmo tema e dados da Parte 2, ou definir um novo tema para esta parte da prova. Sugira ao docente a sua escolha, e prepare antecipadamente os dados de suporte aos seus serviços para a apresentação do seu trabalho. Considera-se que 30 registos por tabela é um volume de dados suficiente.

3. Utilize uma estratégia *Code-first* para o desenvolvimento de uma camada de serviços REST com base na utilização da framework Loopback4 [4 valores] para disponibilizar os recursos para uma aplicação Web de Backoffice [2 valores] construída com recurso à biblioteca React-Admin. Tenha em atenção os seguintes requisitos mínimos obrigatórios:
  - Arquitetura de serviços do tipo REST;
  - Utilização de pelo menos 4 verbos (métodos) do protocolo HTTP - CRUD (do inglês: Criar, Ler, Atualizar e Apagar);
  - Disponibilizar pelo menos 4 recursos diferentes;
  - Utilização de pelo menos uma relação de cardinalidade 1:n entre dois dos recursos;
  - Disponibilização das representações de estado dos recursos em JSON;
  - Documentar a API com recurso ao formato OpenAPI 3.0;
  - Disponibilizar uma ‘Collection’ para o Postman para consulta dos recursos;
  - Utilização do MySQL como SGBD;
  - Desenvolvimento de uma Aplicação Web (Cliente de backoffice);
  - Disponibilização de configuração para a aplicação *multi-container* (pelo menos duas imagens).

Valoriza-se o desenvolvimento/utilização de:

- Filtros através de parâmetros HTTP;
- Utilização de relação de cardinalidade m:n entre dois dos recursos;
- Inclusão no relatório de uma avaliação da validação do HTML5 e CSS (W3C Validator ou outro).
- Disponibilizar a utilização de filtros nas listagens;
- Relatório do trabalho [ 2 valores ].

**Resultados a entregar:** API REST, Aplicação Web e Relatório.

## Relatório e Ambiente de trabalho colaborativo

Todos os elementos do grupo devem criar uma conta no Github (<https://github.com>) e juntarem-se numa organização (gratuita) do Github. Definam o nome da **organização** como **infYYdw1gXX** – substituindo YY pelo ano lectivo e XX pelo número do grupo (**copiar o nome do grupo partilhado no moodle**). É em repositórios dentro da organização que devem partilhar os trabalhos e o relatório.

## Exame Prático de avaliação final, Ep. Normal

O docente partilhou em <https://github.com/exemploTrabalho/reportDW> um exemplo genérico de como podem organizar os repositórios, incluindo os elementos necessários para os relatórios.

Os relatórios devem ser escritos usando Markdown (<https://www.markdownguide.org/>). Podem gerar documentos em pdf através da extensão [Markdown PDF do VS Code](#).

O grupo deve também criar uma conta gratuita no DockerHub como **infYYdw1gXX** – substituindo YY pelo ano lectivo e XX pelo numero do grupo (**copiar o nome do grupo partilhado no moodle**), e deve criar nesta os repositórios públicos para partilhar as imagens dos *containers*. Defina ligações a estes repositórios e às imagens do DockerHub no relatório.

## Apresentação e Discussão de resultados

Para a apresentação cada aluno deverá demonstrar que o conteúdo do CD/DVD/Pen entregue para avaliação está coerente com os elementos apresentados.

A apresentação consiste na instalação de todos os elementos num computador disponibilizado na sala de exame, demonstração do cumprimento dos requisitos, identificação de características diferenciadoras e passíveis de valorização, discussão das principais dificuldades e resultados atingidos.

Durante a apresentação o aluno necessita demonstrar o domínio no controlo dos itens em avaliação e discutir os resultados atingidos.

Durante ou após a apresentação, o docente poderá elaborar questões para aferição das competências, do conhecimento e domínio individual das tecnologias utilizadas.

A avaliação da prestação do aluno durante a apresentação e discussão dos resultados será refletida na classificação final do trabalho e poderá originar diferenças na classificação dos diferentes elementos caso o trabalho seja desenvolvido em grupo.

## Conteúdos a entregar

Não obstante a utilização da plataforma GitHub, a formalização da **entrega** deve ser realizada **individualmente** através da entrega de um ficheiro zip contendo uma cópia individual dos repositórios à data da entrega em CD/DVD/Pen, entregue ao docente na data de realização do exame.

O aluno deve entregar individualmente no dia do exame um CD/DVD/Pen contendo as versões digitais dos seguintes elementos:

1. Numa pasta de nome ‘parte1’ a Apresentação e Ficheiro *Postman Collection*.
2. Numa pasta de nome ‘parte2’ a API WEB e o Relatório.
3. Numa pasta de nome ‘parte3’ a API WEB, a Aplicação Web e o Relatório.

## Utilização de ferramentas de Inteligência Artificial

Os estudantes podem recorrer a serviços de inteligência artificial (por exemplo, assistentes de texto, geração de código ou outras ferramentas semelhantes) como apoio no desenvolvimento do trabalho. No entanto, é obrigatório documentar de forma clara e precisa o processo de utilização, incluindo as *prompts* principais ou instruções fornecidas à ferramenta e a forma como estas contribuíram para o resultado obtido. A avaliação terá em consideração não apenas a qualidade do resultado final, mas também a transparência e a reflexão crítica sobre o processo seguido.

## **Exame Prático de avaliação final, Ep. Normal**

### **Datas importantes**

Data da prova de exame de avaliação final em época normal :

- Entrega de todos os conteúdos em formato digital.
- Apresentação do trabalho.

NOTA: Todos os documentos entregues para avaliação devem estar em formato PDF (exceto se código fonte).

### **Entrega das Sugestões/Propostas de Temas**

A proposta de temas deverá ser entregue no portal da disciplina **e enviada por e-mail para o docente** até, pelo menos, 5 dias antes da data do exame.

Utilize para o efeito o endereço de e-mail do docente [mao@umaia.pt](mailto:mao@umaia.pt) .

NOTA: Os alunos cuja proposta de trabalho tenha sido apresentada e aceite durante a avaliação contínua não necessitam apresentar uma nova proposta. Neste caso devem manter a proposta apresentada e anteriormente aceite pelo docente.