

# Ficha engweb2024-normal

---

## Avaliação: Engenharia Web

---

16 de Maio de 2023, 14h, Ed.2 - salas 0.20 + 1.03

Licenciatura em Engenharia Informática (3º ano)

### Sinopsis

O objectivo principal deste teste é avaliar os conhecimentos obtidos durante as aulas no desenvolvimento de aplicações Web e outras tarefas afins.

Antes de começares, lê atentamente até ao fim para ficares com uma percepção do todo que se pretende. Vais ver que tomarás decisões mais acertadas depois de uma leitura completa.

Os resultados finais deverão ser enviados ao docente da seguinte forma:

- Enviar por email para: `jcr@di.uminho.pt`;
- Colocar no subject/assunto: `ENGWEB2024::Normal::Axxxxx`, em que `Axxxxx` corresponde ao número do aluno;
- Enviar ao docente um link do github para um repositório novo criado especificamente para o exame com o seguinte conteúdo (esta preparação poderá valer 1 valor do exame):
  - O repositório no GitHub deverá chamar-se `ENGWEB2024-Normal`;
  - Dentro do repositório deverá haver um ficheiro, `PR.md`, contendo uma descrição de como fez a persistência de dados, do setup de bases de dados, respostas textuais pedidas, instruções de como executar as aplicações desenvolvidas, etc...
  - Dentro do repositório deverão existir duas pastas: `ex1`, onde colocarão a aplicação desenvolvida para responder ao primeiro exercício e, `ex2`, onde colocarão a aplicação desenvolvida para responder ao segundo exercício.

Os exercícios que envolvam criação de rotas serão testados com as rotas no enunciado, qualquer rota que seja diferente da pedida será avaliada com 0.

Ambas as aplicações serão colocadas em execução correndo os seguintes comandos:

```
$ npm i
$ npm start
```

Alternativamente, se resolveres implementar o último ponto, tudo será colocado em execução com o comando:

```
$ docker-compose up -d
```

Prepara e configura as tuas aplicações para funcionarem desta forma.

## Recursos

Recursos para a realização da prova:

- [Lista de contratos registados no Portal dos Contratos Públicos para o projeto Portal da Transparência \(até 9 de Maio de 2024\)](#) (dataset em CSV obtido de [dados.gov.pt](#)), este ficheiro tem a seguinte estrutura:

```
idcontrato;nAnuncio;tipoprocedimento;objectoContrato;dataPublicacao;dataCelebracaoContrato;precoContratual;prazoExecucao;NIPC_entidade_comunicante;entidade_comunicante;fundamentacao
10424261;;Consulta Prévia;Seguro de Acidentes de Trabalho para os Funcionários/as da Urbe – Consultores Associados, Lda.;01/01/2024;01/01/2024;3918,75;366;505111667;Urbe – Consultores Associados, L.da;Artigo 20.º, n.º 1, alínea c) do Código dos Contratos Públicos
10424331;;Ajuste Direto Regime Geral;Serviços no âmbito administrativo e apoio nas atividades da Freguesia;02/01/2024;02/01/2024;8100;272;507853024;Freguesia de Candoso (São Martinho);Artigo 20.º, n.º 1, alínea d) do Código dos Contratos Públicos
10424469;;Consulta Prévia;"Serviços de limpeza de bermas, valetas, podas de árvores e manutenção de espaços públicos: manter sem vegetação selvagem as valetas de todas as ruas e caminhos da Freguesia de Carvalhosa, no concelho de Paços de Ferreira. Por vegetação selvagem deve-se compreender toda a vegetação rasteira que cresce nos caminhos, ruas e bermas, bem como arbustos até 1,5 metros de altura que sejam indicados pela entidade adjudicante. A entidade adjudicatária está autorizada a utilizar herbicidas não tóxicos autorizados pelas normas comunitárias e ainda os mecanismos de corte que considere necessário. Proceder à remoção dos detritos que resultem do exercício dos serviços que presta. Manter limpas e desimpedidas as valetas e caixas exteriores de águas pluviais. Manter limpas e desimpedidas as sarjetas e condutas de águas pluviais que sejam de fácil acesso e cuja obstrução ocorra após cada limpeza. Proceder à poda de árvores da Freguesia
...
```

## Exercício 1: Contratos (API de dados)

Neste exercício, irás implementar uma API de dados sobre o dataset fornecido. Encontra-se dividido em 3 partes.

### 1.1 Setup [1val.]

Realiza as seguintes tarefas sem alterares os nomes da base de dados e coleção fornecidos:

- Analisa o dataset fornecido;
- Introdz as alterações que achares necessárias no dataset;

- Importa-o numa base de dados em MongoDB com os seguintes parâmetros:
  - database: `-d contratos`
  - collection: `-c contratos`
- Testa se a importação correu bem.

## 1.2 Queries (warm-up) [0.5+0.5+1+1+1 = 4val.]

Especifica queries em MongoDB para responder às seguintes questões:

1. Quantos registos estão na base de dados;
  2. Quantos registos de contratos têm o tipo de procedimento com valor "Ajuste Direto Regime Geral"?
  3. Qual a lista de entidades comunicantes (ordenada alfabeticamente e sem repetições)?
  4. Qual a distribuição de contratos por tipo de procedimento (quantos contratos tem cada tipo de procedimento)?
  5. Qual o montante global por entidade comunicante (somatório dos contratos associados a uma entidade)?
- Regista estas queries num ficheiro de texto que deverás colocar na pasta **ex1** chamado **queries.txt**.

## 1.3 API de dados [0.5+0.5+1+1+1+1+1+1+1 = 8val.]

Desenvolve agora uma API de dados, que responde na **porta 16000** e que responda às seguintes rotas/pedidos:

- **GET /contratos**: devolve uma lista com todos os registos;
- **GET /contratos/:id**: devolve o registo com identificador **id** (corresponde ao **idcontrato**);
- **GET /contratos?entidade=EEEE**: devolve a lista dos contratos correspondentes à entidade **EEEE**;
- **GET /contratos?tipo=AAA**: devolve a lista dos contratos com tipo de procedimento igual a **AAA**;
- **GET /contratos/entidades**: devolve a lista de entidades comunicantes ordenada alfabeticamente e sem repetições;
- **GET /contratos/tipos**: devolve a lista dos tipos de procedimento ordenada alfabeticamente e sem repetições;
- **POST /contratos**: acrescenta um registo novo à BD;
- **DELETE /contratos/:id**: elimina da BD o registo com o identificador **id**;
- **PUT /contratos/:id**: altera o registo com o identificador **id**.

Antes de prosseguires, testa as rotas realizadas com o Postman ou similar.

## Requisitos para o Exercício 2: API de dados ou json-server

Se por algum motivo não conseguiste obter uma API de dados a funcionar que te permita continuar a realizar este teste, podes avançar para o exercício 2 colocando o dataset num **json-server** e usando este como API de dados (terá uma valorização de 20%, ou seja, há uma redução de 80% relativamente à valorização do exercício 1).

Se optares por esta via, coloca o **json-server** a responder na **porta 16000**.

## Exercício 2: Contratos (Interface) [2+2+2 = 6val.]

Tendo a API desenvolvida, desenvolve agora um novo serviço, que responde na **porta 16001** da seguinte forma:

1. Se colocares no browser o endereço `http://localhost:16001` deverás obter a página principal constituída por:
  - Um cabeçalho com metainformação à tua escolha;
  - Uma tabela contendo a lista de registos, um por linha, com os campos: `idcontrato`, `objectoContrato`, `dataCelebracaoContrato`, `precoContratual`, `NIPC_entidade_comunicante`, `entidade_comunicante`;
  - O campo `idcontrato` deverá ser um link para a página do contrato com esse identificador;
  - O campo `NIPC_entidade_comunicante` deverá ser um link para a página dessa entidade.
2. Se colocares no browser o endereço `http://localhost:16001/:id` deverás obter a página do contrato cujo identificador foi passado na rota:
  - Esta página deverá conter todos os campos do contrato e um link para voltar à página principal.
3. Se colocares no browser o endereço `http://localhost:16001/entidades/:nipc` deverás obter a página da entidade cujo `NIPC_entidade_comunicante` corresponde ao parâmetro passado na rota :
  - Na página de cada entidade deverá constar este identificador e o respetivo nome da entidade;
  - Uma tabela com a lista de contratos dessa entidade (tabela com estrutura semelhante à da página principal);
  - O somatório do valor dos contratos;
  - E um link para voltar à página principal.

## Distribuição das aplicações [1val.]

Tendo chegado a este ponto, deverás ter as duas aplicações a funcionar. Para facilitar a sua distribuição e colocação em funcionamento deverás realizar os seguintes passos:

1. Criar uma especificação `Dockerfile` para cada uma;
2. A partir destas especificações criar as imagens locais das aplicações;
3. Criar uma especificação `docker-compose` que orquestre os três serviços: MongoDB, API de dados e Interface.
4. Testar tudo.

Bom trabalho e boa sorte jcr