

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Desenvolvimento de Aplicações Web
Teste Final, Época Normal, Semestre de Verão, 20/21
Duração: 2 horas

Divida as respostas em pelo menos duas folhas fisicamente independentes, de forma a permitir a correcção em paralelo dos dois seguintes grupos de questões:

- Questões 1, 3, e 5 (questões ímpares).
- Questões 2, 4, e 6 (questões pares).

Na questão 1, a resposta deve incluir o texto completo da opção e não apenas o seu número.

1. (6) Para cada uma das seguintes questões, seleccione a opção de resposta correcta. A escolha de uma opção errada contribui negativamente para o resultado final com um terço da cotação da questão.
 - 1.1. No protocolo HTTP, o *header* **Content-Location** tem semântica definida:
 - i. Apenas nas mensagens de pedido.
 - ii. Apenas nas mensagens de resposta.
 - iii. Nas mensagens de pedido e de resposta.
 - iv. Nas mensagens de pedido e de resposta com um método idempotente.
 - 1.2. A realização de um pedido de método **GET** para `https://example.com/projects/123/delete`, deve ser interpretado por um intermediário como sendo:
 - i. Equivalente ao pedido de método **DELETE** para `https://example.com/projects/123`.
 - ii. Um pedido idempotente mas não *safe*.
 - iii. Um pedido idempotente e *safe*.
 - iv. Nenhuma das anteriores.
 - 1.3. Na plataforma Spring MVC, por omissão, o construtor de uma classe anotada com **@RestController** é chamado:
 - i. Uma vez por cada pedido HTTP, independentemente do *handler* que processa o pedido.
 - ii. Uma vez por cada pedido HTTP processado por um *handler* presente nessa classe.
 - iii. Uma vez por cada utilizador distinto.
 - iv. Uma vez por cada instância da aplicação.
 - 1.4. Assumindo o processo de construção de aplicações para execução na plataforma **browser** usando na unidade curricular, qual o resultado da avaliação da seguinte expressão num módulo da aplicação:

```
const m = require('utils')
```

 - i. Erro de execução, porque a função **require** não está definida.
 - ii. Pedido HTTP de método **GET** com caminho `/utils`.
 - iii. Pedido HTTP de método **GET** com caminho `/utils.js`.
 - iv. Referência para um objecto, se o módulo **utils** estiver presente em **node_modules**.
 - 1.5. No contexto de uma *single page application*, a avaliação da seguinte expressão resulta em:

```
window.location.pathname = '/projects'
```

 - i. Na realização de um pedido HTTP de método **GET** para o caminho `/index.html`.
 - ii. Na realização de um pedido HTTP de método **HEAD** para o caminho `/index.html`.
 - iii. Na realização de um pedido HTTP de método **GET** para o caminho `/projects`.
 - iv. Na realização de um pedido HTTP de método **GET** para o caminho `/projects.html`.
 - 1.6. O resultado da avaliação da expressão **JSX** `<Div />` é:
 - i. Um elemento HTML, do mesmo tipo do obtido na avaliação da expressão `document.createElement('div')`.
 - ii. Um elemento HTML, do mesmo tipo do obtido na avaliação da expressão `document.createElement('Div')`.
 - iii. O resultado da avaliação da expressão `Div({})`.
 - iv. Nenhuma das anteriores.

2. (2) No formato Siren, quais as diferenças entre *links* e acções?
3. (2) No contexto do protocolo HTTP, descreva qual a relação entre negociação de conteúdos e *caching*. Nomeadamente, indique as consequências na organização do sistema de *cache* decorrentes da existência de negociação de conteúdos, bem como a informação extra que é necessário incluir nas mensagens HTTP.
4. (2) Tendo em consideração o modelo de construção de aplicações para a plataforma *browser* usado na unidade curricular, indique qual o propósito e forma da utilização do sistema NPM.
5. (3) Realize um ou mais componentes para a plataforma Spring MVC de forma a expôr um recurso no caminho */handlers*. Um pedido de método **GET** a este recurso deve retornar um objecto JSON. Cada campo deste objecto representa um *handler* usado no processamento de pelo menos um pedido, sendo o valor do campo um objecto com:
 - O número de vezes que o *handler* foi utilizado.
 - O tempo médio de execução dos pedidos a esse *handler*.

Assuma que todos os *handlers* são do tipo **HandlerMethod**. Use o método **getShortLogMessage** para obter uma representação textual do *handler*. Valorizam-se soluções em que o cálculo do tempo de processamento inclua não só o tempo de execução do *handler* mas também o da maioria dos intermediários envolvidos (e.g. filtros e interceptores).

6. (5) Realize um componente React que recebe um URI e que apresenta uma caixa de texto (*textarea*) e um botão. A caixa de texto deve apresentar o conteúdo do *body* presente na resposta a um pedido HTTP de método GET ao URI recebido, independentemente do *status code* da resposta. Caso o pedido resulte numa excepção, a caixa de texto deve apresentar o texto associado a essa excepção.

Um clique no botão deve desencadear um novo pedido HTTP ao URI definido e consequente apresentação da resposta. Enquanto um pedido estiver em curso, o botão deve permanecer inactivo (*disabled*).

O componente deve ser sensível a mudanças no URI definido. Caso o novo URI seja diferente do URI usado no último pedido (terminado ou em curso), deve ser desencadeado um pedido para o novo valor do URI e cancelado eventuais pedidos em curso.