

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Desenvolvimento de Aplicações Web
Teste Final de Época Especial, Semestre de Verão, 21/22
Duração: 2 horas

1. (6) Para cada uma das seguintes questões, indique qual a resposta correta. Cada resposta incorreta subtrai 1/3 pontos à classificação total do conjunto de questões deste grupo.
 - 1.1. No protocolo HTTP, um pedido de método GET para o recurso `https://example.com/projects/123/delete`:
 - i. Solicita a remoção do recurso com URI `https://example.com/projects/123`.
 - ii. Solicita uma representação do recurso com URI `https://example.com/projects/123`.
 - iii. Solicita uma representação do recurso com URI `https://example.com/projects/123/delete`.
 - iv. Deve resultar sempre numa resposta com *status code* 400 (*bad request*).
 - 1.2. No protocolo HTTP, uma mensagem de resposta de *status code* 500:
 - i. Não pode ter *payload*.
 - ii. Pode ter um *payload* usando um qualquer *media type*.
 - iii. Tem de ter um *payload* usando o *media type* `application/problem+json`.
 - iv. Tem de ter um *payload* usando o *media type* `application/json`.
 - 1.3. Na norma *Web Linking* (RFC 8288), o atribute `rel` de um *link*:
 - i. Contém o URI do recurso destino do *link*.
 - ii. Contém o URI do recurso contexto do *link*.
 - iii. Contém o *media-type* usado na representação do recurso destino do *link*.
 - iv. Contém o tipo da relação entre dois recursos, representado eventualmente na forma de um URI.
 - 1.4. Na biblioteca *Spring MVC* e na configuração por omissão:
 - i. É sempre criada uma nova instância dum *interceptor* por cada pedido.
 - ii. Não são criadas instâncias de *interceptors* porque todos os métodos têm de ser estáticos.
 - iii. Apenas são criadas novas instâncias de *interceptors* quando todas as instâncias existentes estiverem a ser usadas.
 - iv. Nenhuma das anteriores.
 - 1.5. A utilização de módulos NPM usando o sistema *CommonJS* em ficheiros *Javascript* destinados à execução em *browsers* requer:
 - i. A instalação de extensões no *browser*.
 - ii. A utilização da ferramenta *webpack* ou similar.
 - iii. A utilização da biblioteca React ou similar.
 - iv. A utilização da sintaxe JSX.
 - 1.6. O primeiro parâmetro da função `createElement` da biblioteca React é:
 - i. Sempre uma *string*.
 - ii. Sempre um componente.
 - iii. Sempre uma instância de um componente.
 - iv. Nenhuma das anteriores.

2. (2) No desenho de APIs HTTP, quais as vantagens da utilização de métodos idempotentes?
3. (2) No contexto do desenvolvimento de *Single Page Applications*, qual o propósito do uso do método `preventDefault`, presente em objectos evento?
4. (2) No contexto da biblioteca React, o recurso identificado por <https://reactjs.org/docs/hooks-rules.html> (acedido a 2022-09-09) tem presente a seguinte afirmação na sua representação

Don't call Hooks inside loops, conditions, or nested functions. Instead, always use Hooks at the top level of your React function, before any early returns.

Indique qual a razão para esta limitação.

5. (4) Realize um ou mais componentes para a plataforma Spring MVC, de forma a que um recurso seja exposto no caminho `/failures`. Um pedido de método `GET` para esse recurso deve retornar uma mensagem com uma representação contendo um objeto JSON. Esse objecto deve representar os últimos 10 pedidos que resultaram numa resposta com *status code* maior ou igual a 500. A representação de cada pedido deve incluir: o método do pedido, o URI do pedido, o *status code* da resposta, e o nome do controlador e do método responsável pelo processamento desse pedido (caso o processamento tenha sido realizado por um controlador).
6. (4) Realize um componente para uso com a biblioteca React que receba uma lista de URLs através das suas propriedades e apresente uma lista de items. Cada item está associado a um URL da lista e deve conter:
 - O URL e o texto "...", enquanto o pedido ao URL estiver a ser realizado.
 - O URL e o código de estado da resposta, ou o erro ocorrido, depois do pedido ter sido concluído, com ou sem sucesso.

A ordem dos items deve respeitar a ordem dos URLs. A sua solução pode criar e usar componentes auxiliares. O componente realizado deve reagir correctamente a mudanças nas suas propriedades.