

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Desenvolvimento de Aplicações Web
Teste Final de Época Normal, Semestre de Inverno, 22/23
Duração: 2 horas

1. (6) Para cada uma das questões seguintes, indique qual a resposta correta. Cada resposta incorrecta subtrai 1/3 pontos à classificação total do conjunto de questões deste grupo.
 - 1.1. No protocolo HTTP, o método DELETE é:
 - i. Não *safe* e não idempotente.
 - ii. *Safe* e não idempotente.
 - iii. Não *Safe* e idempotente.
 - iv. *Safe* e idempotente.
 - 1.2. No protocolo HTTP, o conceito de *interface uniforme* significa que:
 - i. Os URIs tem de ter sempre a mesma estrutura, tal como usar nomes pluralizados (e.g. `/games/123` em vez de `/game/123`) e ter sempre os identificadores da base de dados no caminho e não na *query string* (e.g. `/games/123` em vez de `/games?id=123`).
 - ii. O significado do *status codes* nas mensagens de resposta não depende do recurso acedido.
 - iii. Devem ser usados *media types* baseados em JSON em todas as representações (e.g. `application/json` e `application/problem+json`).
 - iv. Não podem ser usados métodos não definidos no RFC 7231 - *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*.
 - 1.3. No contexto de APIs HTTP, qual a forma que sugere para comunicar informação de erro específica da aplicação, em resposta a pedidos HTTP.
 - i. Definindo e usando um novo *status code* nas mensagens de resposta.
 - ii. Colocando essa informação no *payload* da mensagem de resposta.
 - iii. Definindo e usando um novo *header* nas mensagens de resposta.
 - iv. Retornando uma excepção, criada especificamente para representar esse tipo de informação.
 - 1.4. Para se suportar *deep-linking* no contexto de uma *single page application* é necessário configurar o servidor que serve essa aplicação da seguinte forma.
 - i. Caso o caminho presente num pedido GET não esteja associado a um ficheiro, então é retornada uma resposta de sucesso com o conteúdo de `index.html` em vez de uma resposta com *status code* 404.
 - ii. Caso o caminho presente num pedido GET não esteja associado a um ficheiro, então é retornada uma resposta de redireção para o caminho `/index.html`.
 - iii. O servidor usar a API de história para navegar para o *deep-link*.
 - iv. O servidor usar React Router para navegar para o *deep-link*.
 - 1.5. A avaliação da expressão `JSX <A><p>Hello</p>` é equivalente qual das seguintes expressões:
 - i. `A({children:[document.createElement("p"), "hello"]})`
 - ii. `React.createElement("A", null, React.createElement("p", null, "Hello"))`
 - iii. `React.createElement(A, null, document.createElement("p", null, "Hello"))`
 - iv. `React.createElement(A, null, React.createElement("p", null, "Hello"))`
 - 1.6. No contexto da biblioteca React, uma função *hook*.
 - i. Pode ser chamada dentro de um *callback* associado a um evento.
 - ii. Pode ser chamada dentro da função passada ao `useEffect`.
 - iii. Pode ser chamada dentro de outra função *hook*.
 - iv. Nenhuma das anteriores.

2. (2) No âmbito da biblioteca Spring, indique qual a consequência de anotar classes com a anotação `@Component`.
3. (2) Tendo em conta que os *browsers* modernos já suportam o sistema de módulos ECMAScript Modules (ESM), qual a relevância de se ainda usar uma ferramenta como o webpack?
4. (4) Realize um ou mais componentes para uso com a biblioteca Spring MVC de forma a expor recursos nos caminhos `/handlers` e `/handlers/{handler-id}`. Um pedido de método GET a `/handlers` retorna uma representação JSON contendo uma lista de *links* para caminhos com a forma `/handlers/{handler-id}`, um para cada *handler* que tenha sido executado até ao momento. Um pedido para `/handlers/{handler-id}` retorna uma representação com o número de vezes que o *handler* foi executado, caso `handler-id` seja o identificador de um *handler* executado pelo menos uma vez. Assuma que todos os *handlers* são do tipo `HandlerMethod` e use o método `getShortLogMessage` para obter o identificador dum *handler*.
5. (4) Realize um componente para uso com a biblioteca React, recebendo um URI e um período temporal em milissegundos. O componente deve realizar periodicamente um pedido de método GET para esse URI e apresentar ou atualizar o conteúdo (*payload*) textual da resposta numa caixa de texto (`textarea`), independentemente do *status code* da resposta. Caso a realização do pedido resulte numa exceção, a caixa de texto deve apresentar o texto associado a essa exceção. O componente deve ser sensível a mudanças no URI ou no período temporal.
6. (2) Realize a função

```
useCounter(initial: number): [observed: number, inc: () => void, dec: () => void]
```

para ser usado como *hook* em componentes para a biblioteca React. Este *hook* serve para gerir um contador, retornando a função um *array* com três elementos: o valor atual do contador, uma função para incrementar o contador, e uma função para decrementar o contador.