

Instituto Superior de Engenharia de Lisboa  
Licenciatura em Engenharia Informática e de Computadores  
**Desenvolvimento de Aplicações Web**  
Teste Final de Época de Recurso, Semestre de Verão, 21/22  
**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, um pedido de método PUT para o recurso `https://example.com/projects`:
    - i. Solicita a criação ou atualização do recurso com URI `https://example.com/projects`.
    - ii. Solicita a atualização do recurso com URI `https://example.com/projects`, devendo resultar em erro caso o recurso não exista.
    - iii. Solicita a criação dum recurso com URI `https://example.com/projects/{id}`, para um valor de `id` controlado pelo servidor.
    - iv. Solicita a representação do recurso com URI `https://example.com/projects`.
  - 1.2. No protocolo HTTP, uma mensagem de resposta com *status code* 401:
    - i. Deve originar o reenvio da mensagem de pedido associada, sem alterações, após decorrido um intervalo de tempo.
    - ii. Deve conter o header de resposta `WWW-Authenticate`.
    - iii. Deve conter o header de resposta `Authorization`.
    - iv. Deve conter um *payload* com o *media type* `application/json`.
  - 1.3. Uma mensagem de pedido com método GET para o recurso `https://example.com/projects/123?accept=application/json`:
    - i. Tem de resultar numa mensagem de resposta com `Content-Type` igual a `application/json`.
    - ii. Tem de resultar numa mensagem de resposta com `Content-Type` igual a `text/json`.
    - iii. Tem de resultar numa mensagem de resposta com `Content-Type` igual a `application/json` ou a `application/problem+json`.
    - iv. Pode resultar numa mensagem de resposta com qualquer valor de `Content-Type`.
  - 1.4. No contexto da plataforma Spring MVC
    - i. Vai existir uma instância implementando a interface `HttpServletRequest` partilhada por todos os pedidos.
    - ii. Vai existir uma instância implementando a interface `HttpServletRequest` partilhada por todos os pedidos para o mesmo controlador.
    - iii. Vai existir uma instância implementando a interface `HttpServletRequest` partilhada por todos os pedidos para o mesmo *handler*.
    - iv. Nenhuma das anteriores.
  - 1.5. No contexto de uma *single page application* usando a biblioteca React Router, a ativação pelo utilizador do elemento DOM produzido em resultado de `<Link to="/path"/>`, realiza:
    - i. Um pedido de método GET ao URI com caminho `/path`.
    - ii. Um pedido de método GET ao URI com caminho `/index.html`.
    - iii. Um pedido de método GET ao URI com caminho `/bundle.js`.
    - iv. Nenhuma das anteriores.
  - 1.6. A avaliação da seguinte expressão JSX `<Example render={()=><Inside />} />`
    - i. É equivalente à avaliação da expressão `React.createElement(Example, null, Inside)`.
    - ii. É equivalente à avaliação da expressão `React.createElement(Example, {render: () => createElement(Inside)})`.
    - iii. É equivalente à avaliação da expressão `React.createElement(Example(Inside()))`.
    - iv. Nenhuma das anteriores.

2. (2) No contexto do protocolo HTTP, indique as vantagens do conceito de interface uniforme, fornecido por este protocolo.
3. (2) Qual o propósito e estrutura do header **Link**, definido na especificação RFC 8288 - *Web Linking*.
4. (2) No contexto da biblioteca React, o recurso identificado por <https://reactjs.org/docs/hooks-rules.html> 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 conjunto de recursos seja exposto no caminho `/status/{method}`. Um pedido de método **GET** para um desses recursos deve retornar uma mensagem com uma representação contendo um objeto JSON. Esse objeto deve representar o tempo máximo e mínimo do processamento de pedidos com método **method**. Caso o servidor ainda não tenha recebido pedidos com o método **method**, a resposta ao pedido deve ter **status code** 404.
6. (4) Realize um componente React com as seguintes propriedades: **uri** do tipo **string**, e **period** do tipo **number**. O componente deve realizar um pedido HTTP de método **GET** ao URI definido por **uri**, com a periodicidade de **period** milissegundos. Após a conclusão de cada pedido, o componente deve apresentar o *status code* da resposta e o tempo que a resposta demorou a ser recebida. Em caso de erro, deve apresentar a descrição do erro. Caso o pedido não esteja concluído em **period** milissegundos, deve ser apresentada a mensagem "timeout" e o resultado do pedido deve ser ignorado. O componente realizado deve reagir correctamente a mudanças nas suas propriedades.