Respostas Teste Versão – A

1. A imagem é um diagrama cliente-servidor responsável em contexto web por entregar páginas web aos utilizadores.
   No cliente é executado o software (browser) responsável pela apresentação da página web escrita em várias linguagens como, HTML, CSS e JavaScript.
   Quando o cliente quer aceder a uma página, envia um pedido ao servidor, que por sua vez devolve o conteúdo da página.
   O servidor executa o software necessário para suportar as páginas web e mantelas disponíveis na web. Pode também suportar um banco de dados. Usa linguagens como PHP, Java, Python, etc.
   Quando o servidor recebe um pedido do cliente, processa-o e gera uma resposta. A resposta inclui o código HTML, CSS e JavaScript que é necessário para gerar a página da web, sendo depois enviada a resposta de volta ao cliente.

Parte II

1. No primeiro caso temos getElement porque estamos a especificar apenas um elemento e não um conjunto. O id define apenas um elemento. Não sendo possível identificar vários.
   Já no caso dos restantes, podemos identificar vários elementos, pois podemos associar o mesmo nome e classe a vários elementos.

   Ficheiros na pasta -> getElement.js

2. A estrutura JSON é hierárquica, com o curso no topo e as UCs aninhadas dentro dele.
   Cada UC tem um nome, uma sigla e um objeto docente.
   O objeto docente tem um nome e um email.
   Esta estrutura é flexível e pode ser facilmente expandida para acomodar mais informações, como carga horária, descrições de UC, etc.

   A estrutura XML também é hierárquica, com o curso no elemento principal e as UCs aninhadas dentro do elemento UCs.
   Cada UC tem elementos Nome, Sigla e Docente.
   O elemento Docente tem elementos Nome e Email.
   Esta estrutura é mais verbose do que a estrutura JSON, mas é mais fácil de ler por humanos.

Parte III

1. A diferença está na fonte da imagem, a primeira opção vai a um endereço web e utiliza uma imagem presente. A segunda opção utiliza uma imagem local, presente nas diretorias da página web.

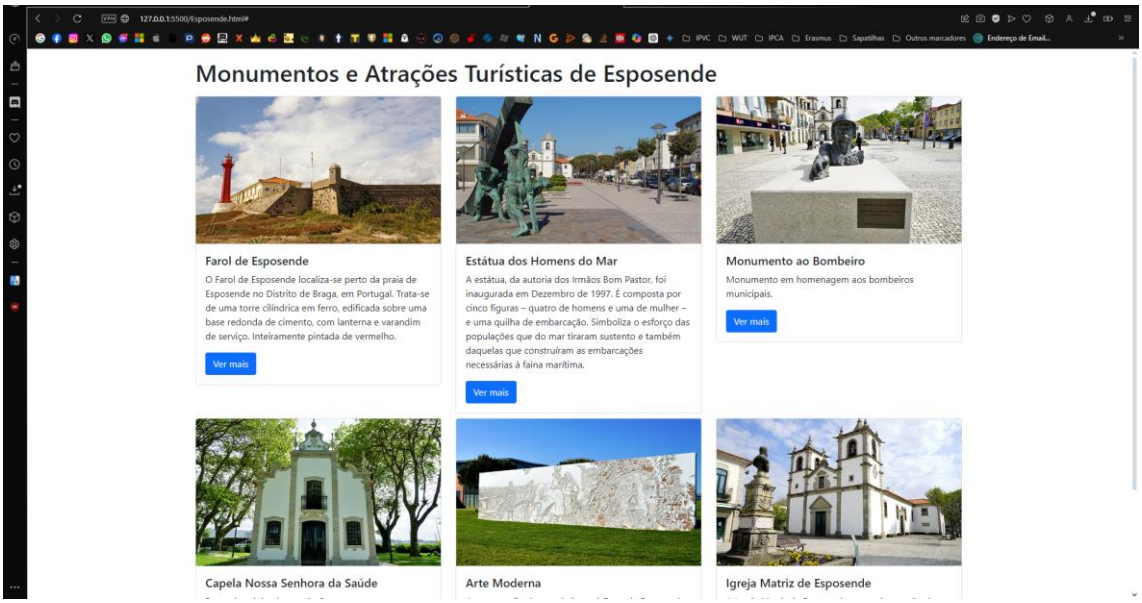
2. O atributo define a linguagem do documento para português.

Parte IV

1. **Ficheiros na pasta -> index.html e estilos.css**

Parte V

1. **Ficheiros em pasta -> Esposende.html**



Parte VI

1. **Ficheiro na pasta -> routes.js**

2. Import Express Router for creating modular routes.
   Import product controller functions from the controllers directory.
   Import authentication middleware from the auth directory.
   Define GET route at root path for fetching all products using controller function.
   Define GET route with dynamic ID parameter for fetching a product by ID using controller function.
   Define POST route at '/create' path for product creation using controller function.
   Define PUT route at '/update' path for product update using controller function.
   Define DELETE route with dynamic ID parameter for product deletion using controller function.
   Export the products router for use in other parts of the application.

3. **Ficheiro na pasta -> products.js**