

Geração dinâmica de páginas a partir de dados armazenados no servidor

Segundo release

No segundo *release*, devem ser incluídas dinâmicas, tanto do lado cliente como do lado servidor da aplicação por meio do *JavaScript* (JS). No caso do servidor, a tecnologia a ser usada é compulsoriamente o **NodeJS** (<https://nodejs.org/>) acrescido dos módulos **express** e **ejs**.

Observações iniciais:

- *Por mais que seja tentador solucionar e desenvolver o projeto usando outras ferramentas de produtividade, como os diversos frameworks disponíveis, a proposta da disciplina é trabalhar com o mínimo de recursos intermediários possível e compartilhar o conhecimento entre os alunos.*
- *Exemplos com o uso do de JS, express e ejjs serão disponibilizados no Solar e discutidos em sala.*

1. Desenvolva soluções com as tecnologias sugeridas para os seguintes propósitos:

- a) Gerar dinamicamente as diferentes seções do documento HTML a partir de **dados armazenados no servidor**, considerando **usuários e/ou perfis distintos**. Podem utilizar arquivos armazenados no servidor (sistema de arquivos) com estruturas JSON. **Não é obrigatório** o uso de um Sistema Gerenciador de Banco de Dados (SGBD), **mas os grupos são livres para adotar SGBDs relacionais ou orientados a objeto, caso desejem**.
- b) Organizar os dados de tal maneira que um mesmo **template** possa ser usado para diferentes usuários e/ou situações. Os *templates* podem ser definidos usando o EJS, conforme exemplos fornecidos, ou usar módulos semelhantes a partir do NodeJS.
- c) A carga de conteúdo deve contemplar tanto requisições HTTP **síncronas, a partir de formulários e hiperlinks**, como **assíncronas, a partir de JS CLIENTE** (uso da API Fetch ou do objeto XMLHttpRequest – **Não usar recursos intermediários fornecidos por frameworks!!!**).

Obs.: Como há dependência da natureza do projeto específico de cada grupo, é impossível especificar a priori, neste enunciado, as funcionalidades para cada uma dessas modalidades. Assim, é responsabilidade do grupo a forma de implementar e demonstrar as duas abordagens.

- d) Nesta versão, a alternância entre os arquivos CSS definidos no *release 1* deve ser implementada por meio de solução interativa.
- e) Como recurso de acessibilidade adicional, pede-se também implementar a configuração de tamanho das fontes por meio de botões (A+ e A-). A última configuração (tamanho de fonte e CSS) definida pelo usuário deve ser armazenada **no cliente**. Você pode utilizar *cookies* ou outro recurso como o [Web Storage do JS](#) (não guardar a configuração como dado de sessão!).
- f) O desenvolvimento deve ser relatado em *slides* e o progresso apresentado em sala de aula na data especificada para a entrega. **Todos os componentes do grupo devem estar presentes e cada um deve apresentar a sua contribuição individual!**

DATA DA APRESENTAÇÃO DO *RELEASE 2*: Ver Calendário