

Especificação do **TRABALHO 2**
(Versão 1)

Nesta disciplina temos por objetivo compreender o funcionamento de uma aplicação (ou serviço) web desde o back-end (parte da aplicação executada na máquina do servidor) até o front-end (parte da aplicação executada na máquina do cliente), bem como a API (interface de programação) através da qual ambas as partes se comunicam. No primeiro bimestre você utilizou a linguagem de programação C# para implementar tanto o back-end de um serviço quanto a API REST de acesso ao mesmo. Sua API foi testada através de um simulador de requisição como o Thunder Client.

Neste segundo bimestre seu objetivo será implementar um front-end para seu serviço, isto é, você deverá utilizar HTML, CSS, Javascript e AngularJS (sendo este último opcional) para desenvolver uma página web através da qual a API possa ser utilizada pelo usuário final. A página deve funcionar como uma aplicação única (SPA), isto é, ela deverá utilizar os recursos da linguagem Javascript (como AJAX, Fetch API ou Web Forms) para fazer requisições ao back-end, além de atualizar seu próprio design em tempo real para acomodar as respostas recebidas.

O design das páginas é livre, mas o trabalho será avaliado pela correção da interface gráfica e pela correção da comunicação com a API. Não é necessário implementar sistema de login ou de sessão, o usuário poderá acessar diretamente a página do serviço. Todo o serviço pode ser realizado em uma única página, porém, se você preferir, o CRUD das entidades pode ser realizado em páginas separadas (no máximo uma página extra para cada entidade). Exemplo: se o seu serviço é um sistema de locação de filmes, você pode usar a página inicial para fazer todas as operações relativas ao serviço (locar filme, devolver filme, renovar locação, consultar locações etc.), além de outras duas páginas, uma para o CRUD de filmes e outra para o CRUD de clientes. Entretanto, cada uma dessas páginas deve ser capaz de listar suas entidades sem mais redirecionamentos, ou seja, os elementos da página com a lista de entidades devem ser criados e/ou atualizados dinamicamente no front-end.

Os trabalhos deverão ser preferencialmente realizados pelas mesmas equipes do primeiro trabalho, mantendo também os mesmos temas. Caso haja alterações nas equipes e/ou nos temas, o professor deve ser comunicado. A defesa do trabalho ocorrerá durante a aula especificada no calendário do Blackboard da turma. A equipe deverá subir o código em um repositório GIT qualquer e terá cerca de 10 minutos para explicar seu funcionamento e executá-lo durante a avaliação, demonstrando todos os métodos solicitados. Apenas os membros presentes no dia da defesa serão pontuados e a pontuação de cada membro será dada individualmente para garantir que todos tenham participado no desenvolvimento. Caso o dia especificado coincida com um recesso, o professor combinará com a turma alternativas para a defesa.

Dúvidas devem ser encaminhadas ao email do professor ou postadas no Blackboard. Aplicações que não cumpram todos os requisitos do trabalho serão pontuadas apenas parcialmente.