

IFPR – INSTITUTO FEDERAL DO PARANÁ

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS – TADS

Calebe Augusto Santana de Oliveira

Single Page Application (SPA), Multiple Page Application (MPA), Client-Side e Server-Side.

Foz do Iguaçu/PR

SPA – SINGLE-PAGE APPLICATION

Um SPA é uma aplicação web que roda em uma única página, se assemelhando a um aplicativo desktop ou um mobile, são leigamente chamadas de “páginas ajax”, um bom exemplo que gosto de usar é o Gmail do Google, ele é um SPA, a navegação na aplicação rola toda em uma única página e todo o conteúdo é carregado de uma vez ou obtido dinamicamente (ou seja, via requisições Ajax).

A aplicação SPA pode ser construída de diversas formas, a mais comum é ser auxiliado por um framework Javascript como Angular ou Vue.js, ambos possuem sistemas de rotas e clientes HTTP para fazer requisições a recursos externos (uma API, por exemplo).

VANTAGENS

SPA é rápido, por se tratar de uma aplicação em que todos os recursos HTML, CSS e JavaScript são carregados de inicio de uma só vez, a redução de trafego servidor – cliente também é reduzida, como toda a renderização da página fica no lado cliente, isso acaba poupando o servidor desse serviço, muitas vezes o desenvolvimento acaba se tornando mais simples e curto, justamente por ser SPA, e ser vantajoso para ser usado em aplicações, seu back-end pode ser reutilizado em muitos desenvolvimentos mobiles. Se bem feito, pode até ser utilizado sem conexão.

DESVANTAGENS

Uma das maiores desvantagens do SPA é a dificuldade de fazer otimizações SEO, e também comparado ao método “tradicional”, SPA é menos seguro. Necessita também que o browser cliente esteja com Javascript ativo.

FRAMEWORKS/BIBLIOTECAS

Angular: Criada pelos desenvolvedores do Google, o Angular é uma plataforma de desenvolvimento de interfaces front-end usando principalmente JavaScript. Além de ser uma plataforma open source, tem uma forte comunidade que só tende a crescer, dentre os principais, podem destacar-se componentes, templates, diretivas,

roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a de executar os testes unitários de uma aplicação.

Vue: Sendo também uma plataforma library de desenvolvimento de interfaces, a diferença do VueJs para outros frameworks é a praticidade de se usar, sua principal função é a de componentes reativos para interfaces. Muito semelhante ao framework ReactJs, acabou se saindo melhor em alguns testes.

MPA – Multiple-Page Application

Conhecido por manter a forma “tradicional” de páginas web, a cada mudança de conteúdo e reloads dados na página, uma nova tela é carregada e renderizada no browser, na maior parte das vezes, são projetos maiores, que demandam mais funcionalidades tanto do servidor quanto do cliente, o que acaba tornando mais complexo e demorado de ser desenvolvido. A cada clique dado em um site MPA, uma requisição é feita ao servidor e é devolvida por ele.

VANTAGENS

Sendo perfeito para usuários que precisam de um “guia” para onde ir na aplicação, mas uma das melhores vantagens desse modelo de páginas web, é a ótima SEO(Search Engine Optimization), o desenvolvedor pode otimizar o site com palavras-chave e meta tags em cada página individual da aplicação, o que te dá uma melhor chance de visualização e busca.

DESVANTAGENS

Uma das grandes desvantagens desse modelo é a forte acoplação front-beck-end, o que torna o tratamento de alguns domínios mais complexo, e também é claro o longo tempo de desenvolvimento e complexidade desse modelo, que por possuir mais funcionalidades e usabilidade, acaba se tornando um processo que precisa de mais atenção e tempo.

FRAMEWORKS/BIBLIOTECAS

Bootstrap: Um dos frameworks mais populares no meio front-end, o Bootstrap facilita a vida do desenvolvedor e muito, responsável por manter a responsividade da

aplicação (mobile), ele possui uma variedade de plug-ins e componentes ótimos para estilização de sites, sem que seja preciso digitar uma linha CSS, uma das funcionalidades mais legais também é o sistema de grid de layout que o framework conta, pode ser integrado com qualquer linguagem de programação, tem uma vasta comunidade de usuários na web e possui uma documentação muito simples e prática.

Laravel: Sendo um framework desenvolvido especialmente para linguagem PHP, é totalmente open source e free. Permitindo que você trabalhe de forma estruturada e rápida, possui uma enorme comunidade e uma equipe de desenvolvedores totalmente ativa, o Laravel utiliza composer para gerenciar suas dependências, trabalha também, com um ótimo sistema de rotas que fazem o mapeamento pela URL para alguma funcionalidade dentro da sua aplicação. Trabalha também com o Blade, um excelente compilador de templates, conhecido por sua flexibilidade, o Blade não restringe o uso do PHP puro utilizado junto a syntaxe do template.

CLIENT-SIDE

Como o próprio nome diz, cliente-side(lado-cliente) é todo o material que vai rodar diretamente no seu browser, quem processa e renderiza toda a informação e o conteúdo é o browser, não sendo necessário o servidor processar nenhuma tarefa, quando se fala em aplicações client-side, a primeira coisa que vem na cabeça de muitos é JavaScript e AJAX. Isso acontece porque a aplicação deixa de necessitar de tarefas e processos que na maioria das vezes são feitos dentro do servidor, e passa a executa-las todas dentro do próprio browser.

VANTAGENS

Obviamente, por se tratar de uma aplicação que roda toda no lado cliente, acaba muita das vezes se tornando aplicações mais leves, que rodam com mais facilidade e economizam sua banda-larga, pois não corre o risco de inflar o tráfego ao servidores, justamente por não haver processos externos. Sendo ao mesmo tempo uma desvantagem, a maioria das aplicações client-side utilizam JavaScript, que é compatível com qualquer tipo de navegador, o que da ao usuário a sensação de rapidez e leveza, muito semelhante aos apps desktop.

DESVANTAGENS

Como dito anteriormente, usar JavaScript pode se tornar um problema se o usuário não tiver essa opção habilitada dentro do browser, muitas pessoas desabilitam por

questões de segurança, o que torna também esse tipo de aplicação insegura, justamente por rodar no próprio computador cliente. Outro grande problema, são os otimizadores de busca SEO, não são nada amigáveis com o Javascript, o que pode fazer com que sua aplicação não seja tão “visível” nos buscadores.

FRAMEWORKS/BIBLIOTECAS

Skeleton: Muito parecido com o Bootstrap, Skeleton é um framework baseado em CSS e JavaScript, usado para criação de sites responsivos, conta com uma grande diversidade de componentes, como layout de grid, todo desenvolvido com linguagens de fácil entendimento, é também muito adaptável ao design e a interface do seu site.

jQuery: Um dos frameworks mais populares para client-side de Javascript, o que ele faz é basicamente compactar várias linhas de código em uma única função, ou até efetuar requisições AJAX, além disso, é totalmente open source e conta com uma vasta comunidade de usuários.

SERVER-SIDE

Server-side ou back end é o lado servidor. Em aplicações que possuem mais dinamismo e interatividade, são usados códigos back end para que possam ser gerados diferentes dados quando necessário, sites podem realçar conteúdos que são mais relevantes baseado na preferencia dos usuários, ainda podem armazenar informações pessoais e dados de usuários para realizar pagamentos, por exemplo.

Tudo isso funciona com solicitações HTTP do lado cliente(front end) feita pelos navegadores para o servidor. O servidor(back end) processa e valida as informações e retorna uma resposta para o cliente.

VANTAGENS

Uma das vantagens do server-side é a renderização de páginas dinâmicas no browser, já que alguns browsers não suportam Javascript. Se bem projetado, um servidor pode aumentar a velocidade de carregamento do seu site, pode também, oferecer uma maior segurança para dados pessoais de usuários como cartões de crédito e contas bancárias, perfeito para o desenvolvimento de e-commerces e sites de negócios. Outra vantagem também é que pelo script ficar armazenado todo no lado servidor, fica menos vulnerável a invasões e clonagens de hackers.

DESVANTAGENS

Em contrapartida, sites que necessitam de um grande aplicação e que pesam no tráfego da rede podem necessitar de um equipamento de rede mais sofisticado, que demanda mais investimento, como servidores dedicados e até mesmo cloud hosting. Precisam também de uma base de dados para armazenar todas as informações e apresentá-las dinamicamente, que precisam constantemente estar realizando backups. Com a constante visita e interação dos usuários à aplicação, muitas requisições ao servidor são feitas ao mesmo tempo, o que pode causar uma lentidão no site.

FRAMEWORKS/BIBLIOTECAS

Django: Django é um framework web gratuito e open source, escrito em Python. Utilizando o padrão de projeto MTV (Model, Template e View), conta com uma variedade de componentes muito úteis para construção de um site. Uma das principais características é o ORM (Mapeamento Objeto-Relacional) que serve para que possamos gerar tabelas no banco sem a necessidade de utilizar SQL, possibilitando a criação de formulários automaticamente através dos modelos de dados.

Express: O Express é um framework para aplicações web do Node Js mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e móvel. Oferece também um ótimo desempenho para requisições e rotas URL. Baseado no número de empresas de alto perfil que utilizam o Express, o número de pessoas contribuindo com o código fonte e que proveem suporte do mesmo, é sim considerado um framework popular.

