



WBA0887_v1.0

Proposta de resolução do Desafio Profissional

Frameworks front end

Anderson da Silva Marcolino





Caso – Selecionando um *Framework Front End*

O gerenciamento de equipes em empresas de desenvolvimento de aplicações web não é tarefa fácil. Ser gerente vai além de selecionar tecnologias para serem adotadas no desenvolvimento. Devem ser considerados fatores humanos, a curva de aprendizagem para o uso de determinados recursos e a identificação dos recursos físicos (*hardware*) disponíveis para hospedar tais aplicações. Sem contar a necessidade de se adotar e criar projetos *front end* atrativos.

Nesse cenário, coloque-se no papel do gerente de uma equipe de desenvolvimento de aplicações web no contexto do *front end*. Considere os perfis e os conhecimentos (apresentados no Quadro 1) dos funcionários que estão à disposição para trabalhar em seu projeto.





Caso – Selecionando um *Framework Front End*

Quadro 1 – Perfil

Identificação	Papel	Conhecimentos	Observações
1	Desenvolvedor	Java, banco de dados relacional, banco de dados não relacional.	Trabalhou em projetos <i>back end</i> .
2	Desenvolvedor	JavaScript, HTML, CSS e PHP.	Trabalhou com desenvolvimento de e-commerce. Dificuldade em aprender novas tecnologias.
3	Designer	CSS, Bootstrap, Material.	Trabalhou com o desenvolvimento de interfaces para aplicativos móveis.
4	Analista de Qualidade	Testes caixa preta.	-
5	Desenvolvedor/Designer	PHP, HTML, CSS, Ajax e JQuery.	Dificuldade de trabalho em equipe.

Fonte: elaborado pelo autor.





Caso – Selecionando um *Framework Front End*

Considerado as informações do Quadro 1, monte três equipes para trabalhar adotando os *frameworks front end*: a) Vue.js; b) Angular; e c) React. Atenda às regras a seguir para a criação de cada equipe e, ao final, justifique a seleção de cada membro, relacionando sua escolha ao *framework* adotado:

- Garanta que sua equipe possua ao menos um desenvolvedor e um design.
- Considere as especificidades do *framework* na sua seleção e as apresente em sua justificativa. Considere elementos como: curva de aprendizagem; linguagem relacionada ao *framework*; tipos de recursos que podem ser utilizados; e a possível evolução da sua aplicação, se novas demandas forem solicitadas por clientes.





Caso – Selecionando um *Framework Front End*

- Lembre-se de que sua equipe é reduzida e alguns desenvolvedores trabalharão em mais de um projeto. Isso implica o balanceamento de conhecimentos prévios dos membros da equipe com os conhecimentos necessários a se aprender.
- Considere as observações como essenciais para a formação das equipes por *framework*.
- Garanta que todos os recursos humanos estejam envolvidos em, pelo menos, uma das três equipes.
- Justifique e relacione tecnologias que os membros conhecem e as que serão necessárias para o uso dos *frameworks*.



► Solução Prática

Para criarmos a nossa demanda, com uma equipe para cada um dos *frameworks front end*, recomenda-se iniciar pela criação de um quadro que apresente as especificidades de cada uma dessas tecnologias, como consta no Quadro 2.





Caso – Selecionando um *Framework Front End*

Quadro 2 – Especificidades dos *frameworks front end*

Framework	Especificidades técnicas	Observações
Vue.js	Padronização para desenvolvimento, fórum para dúvidas, tamanho intermediário da biblioteca, renderização rápida de projetos, linguagem JavaScript.	Como é menos conhecido, pode dificultar o processo de aprendizagem, além de sua documentação ser predominantemente em inglês.
Angular	Não segue um padrão específico, não possui fórum oficial da comunidade, maior tamanho da biblioteca, renderização mais lenta de projetos, linguagem TypeScript e JavaScript, melhor divisão dos arquivos do projeto.	Documentação vasta, mas a quantidade de versões e a existência do Angular 1 podem gerar confusão no momento de se estudar.
React	Padronização para desenvolvimento, fórum para dúvidas, menor tamanho da biblioteca, renderização rápida de projetos, linguagem JavaScript, JSX.	Documentação ampla e boa em português, porém o <i>framework</i> não tem boa divisão dos arquivos de seus componentes, o que dificulta seu uso.



Fonte: elaborado pelo autor.

➤ Solução Prática

Tendo como base o Quadro 2, podemos criar um conjunto inicial de recursos humanos para cada uma das tecnologias:

Vue.js: Desenvolvedor (Identificador 1), Designer (Identificador 3) e Analista de Qualidade (Identificador 4).

Angular: Designer (Identificador 3), Analista de Qualidade (Identificador 4) e Desenvolvedor/Designer (Identificador 5).

React: Desenvolvedor (Identificador 1), Designer (Identificador 3), Analista de Qualidade (Identificador 4) e Desenvolvedor/Designer (Identificador 5).



Solução Prática

Com base nessa seleção, podemos descrever a justificativa para cada escolha e permitir um refinamento de cada equipe. Para facilitar a visualização, podemos criar um quadro como o Quadro 3.





Caso – Selecionando um *Framework Front End*

Quadro 3 – Membros por *frameworks front end*

Membro	Vue.js	Angular	React	Justificativa
Desenvolvedor (1)	Sim	Não	Sim	Considerando que o Angular apresenta uma divisão maior dos arquivos, a curva de aprendizagem pode ser maior. Além disso, o React apresenta conceitos diferentes de separação de responsabilidades por meio do JSX, e o Desenvolvedor 1 não tem conhecimento em JavaScript e TypeScript, justificando-se sua alocação no Vue.js e no React, pois ambos são em JavaScript, o que pode dinamizar seu processo de aprendizagem.
Desenvolvedor (2)	Não	Não	Sim	A decisão principal da alocação do Desenvolvedor 2 é a sua dificuldade em aprender novas tecnologias. Como o Angular tem várias versões e o Vue.js apresenta grande parte da documentação em inglês e mais restrita, indica-se a alocação na equipe de React. Adicionalmente, seus conhecimentos o tornarão apto a trabalhar com o <i>framework</i> mais rapidamente.
Designer (3)	Sim	Sim	Sim	Como o designer é peça-chave na criação de interfaces e todos os frameworks são de <i>front end</i> , ele é alocado nas três equipes. Um ponto importante a se destacar é que não foi considerada a carga horária de cada recurso. Nesse caso, recomenda-se sua alocação em dois <i>frameworks</i> apenas; no framework que sobrar, é possível alocar o Desenvolvedor/Design (5).
Analista de Qualidade (4)	Sim	Sim	Sim	Considerando que o Analista de Qualidade é o responsável por realizar testes de caixa preta, o que é pertinente pro contexto de projetos <i>front end</i> , ele foi alocado em todas as equipes.
Desenvolvedor/ Designer (5)	Não	Sim	Sim	Pensando estrategicamente, o Desenvolvedor/Designer (5) foi alocado em dois projetos, com o objetivo de integrá-lo à equipe e permitir que possa reduzir suas dificuldades de trabalho em equipe. Adicionalmente, ele possui pouco conhecimento em JavaScript, visto que sabe utilizar apenas bibliotecas desenvolvidas em tal linguagem, como o Ajax e JQuery; logo, ele poderá se familiarizar com TypeScript no Angular e o JavaScript e JSX no React. Finalmente, por ser também designer, sua participação poderá ser útil para suprir necessidades específicas na ausência do Designer (3).

Fonte: elaborado pelo autor.



Conclusões

Com base nessa divisão e nas justificativas, é importante destacar algumas especificidades dos *frameworks* que não interferiram na seleção dos recursos:

- A renderização de projetos e o tamanho das bibliotecas dos *frameworks* não deverão influenciar na escolha de recursos, mas sim na decisão de projetos de cada solução a ser desenvolvida. Para isso, uma análise com o cliente solicitante deverá ser realizada para definir qual tecnologia escolher.
- O fórum de comunidade é um fator importante, mas não decisivo, visto que soluções como o *Stack Overflow* – site em que desenvolvedores tiram dúvidas sobre problemas no desenvolvimento – são ferramentas que concentram vários questionamentos e respostas às dúvidas.



➤ Conclusões

- Um dos fatores mais relevantes foram os recursos, que possibilitam uma aprendizagem mais rápida, somada ao conhecimento prévio dos desenvolvedores.
- Desse modo, temos uma divisão realista considerando nossos recursos e os *frameworks*.



Bons estudos!

