

Engenharia Disciplinada de Softwares

Projeto de Bloco – TP 5

Professor Armenio Cardoso

Vinicius Dornelas Braz
22/08/2022

Link para repositório: <https://github.com/braz-vinicius/venturahr>

- Primeiro protótipo funcional do sistema:

R. Link para o repositório:

<https://github.com/braz-vinicius/venturahr>.

Demonstração funcional em gif animado:

<https://github.com/braz-vinicius/venturahr/raw/master/docs/demos/Demo%20-%20Login%20and%20Profile.gif>

- Incremento da construção anterior com mais casos de uso.

R. Foram implementados os casos de uso *Criar Conta* (RF001) e *Alterar Dados da Conta* (RF002), conforme demonstrado na animação disponível em

<https://github.com/braz-vinicius/venturahr/raw/master/docs/demos/Demo%20-%20Login%20and%20Profile.gif>.

- Justifique a escolha da tecnologia utilizada para o desenvolvimento do projeto e compare com outra opção que poderia ser utilizada, enumerando uma lista de pelo menos 3 prós e contras.

R. O seguinte stack tecnológico foi selecionado:

- **Backend:** Será utilizado a linguagem de programação C# e o framework de desenvolvimento .NET 5.
- **Banco de dados:** Será utilizado banco de dados relacional SQL Server.
- **Frontend (Web):** O frontend será construído utilizando uma SPA (Single Page Application) com uso de JavaScript, HTML e CSS. Além disso serão utilizados os frameworks Vue e Bootstrap.

A justificativa de escolha para as tecnologias supracitadas dá-se em função da prévia experiência do time com o referido stack, além do robusto ecossistema de plugins, add-ons e componentes fornecidos por elas.

Comparativamente poderiam ter sido utilizadas outras stacks, como por exemplo Java no Backend e Frontend utilizando Spring MVC, utilizando JSP (Java Server Pages).

Analisando os trade-offs deste stack com a solução escolhida podemos citar:

Pros:

- Utilização de um stack único Java, propiciando menor fragmentação de tecnologias e menor curva de aprendizado.
- Integração seamless do frontend com o Backend, facilitando a implementação de funcionalidades mais complexas, como por exemplo Autenticação e Autorização.
- Facilidade na criação de visões e formulários dinâmicos processados no servidor, sem necessidade de utilizar uma API separada.
- A solução pode funcionar em navegadores sem suporte a JavaScript.

Contras:

- Maior acoplamento do frontend com o Backend, dificultando implementações multi-dispositivo como por exemplo clientes Web, Android e iOS.
- Limitados a criação de interfaces de usuário mais simples, comparadas com a utilização de um SPA.
- Em muitos casos será necessário recompilar a solução para realizar alterações no frontend, enquanto que no caso de SPAs a alteração reflete instantaneamente na solução, sem necessidade de recompilar o código-fonte (Hot Reloading).

- Elaborar um Plano de Projeto com o que pretende entregar nas próximas 4 iterações.

R. Plano de Projeto:

Ator – Caso de Uso	Complexidade	Iteração
Usuário – Criar Conta	2	1
Usuário – Alterar Conta	2	1
Usuário – Painel de Estatísticas	3	2
Empresa – Publicar Vaga	2	2
Empresa – Listar Vaga	2	2
Empresa – Pesquisar Vaga		3
Candidato – Responder Vaga	3	3
Candidato – Listar Vagas Respondidas	3	3
Empresa – Consultar Ranking	3	4
Empresa – Consultar Dados do Candidato	3	4