

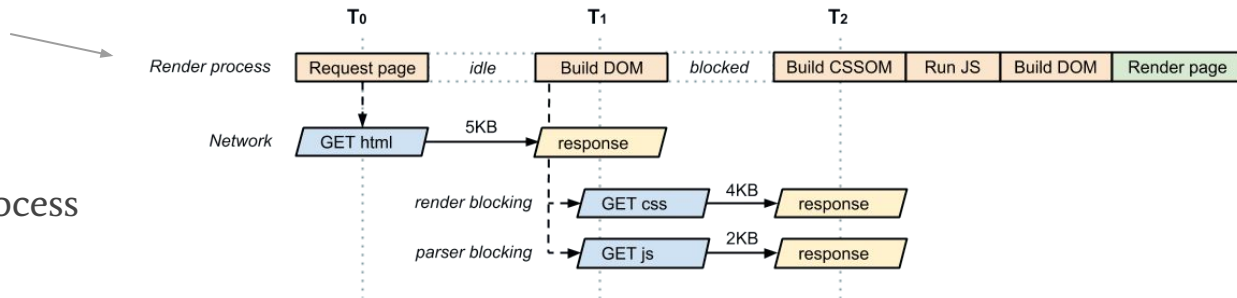


## CASE

**Bruno Orcajo | 40 anos | 3 filhas | SP -> Florianópolis**

1. Para um contexto de um portal vertical com grande acesso onde a performance de carregamento, juntamente com uma experiência fluida de carregamento e navegação, quais os melhores padrões, técnicas e framework que você usaria, e justifique.

- Critical rendering path
  - Defer / async
  - Resource hints
- Context(support) and Process
  - Can I use
- Over-engineering
- Gzip
- LS/SS (client)
- Minify
- CDN (assets)

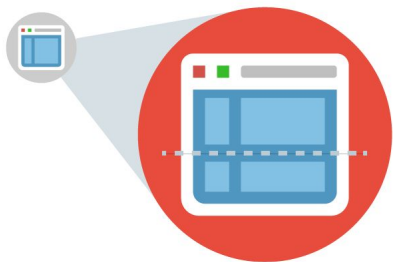


“Reduzir 50% no tempo de resposta no servidor(\$) reduz em média 10-15% no frontend. Ajustes no front podem ter um ganho de performance em média 40-60%”

Ilya Grigorik - web performance engineer at Google

1. Para um contexto de um portal vertical com grande acesso onde a performance de carregamento, juntamente com uma experiência fluida de carregamento e navegação, quais os melhores padrões, técnicas e framework que você usaria, e justifique.

- Images
  - Optimize
  - Webp (~<26% average)
  - Base64
  - Srcset/Picture (media query)
- **Prioritize above the fold content**
- UX
  - Skeleton
  - Lazy Load
  - Pagination
  - Infinite scroll



“O melhor requisição é a requisição não feita”

1. Para um contexto de um portal vertical com grande acesso onde a performance de carregamento, juntamente com uma experiência fluida de carregamento e navegação, quais os melhores padrões, técnicas e framework que você usaria, e justifique.

- REACT

- Code splitting
  - Webpack (splitChunksPlugin)
  - Suspense/Lazy load
- Dev Tools (Profiling comp)
  - Memo
  - useMemo
  - useCallback
- Virtualize Long Lists
- Avoid Reconciliation
- Server side rendering
- Optimization of 3rd parties (webpack)

- Best practices

- Keep components small
- Use fragment
- Avoid inline functions
- Debounce

- Improvements

- Page speed insights
- Lighthouse
- 3rd party

**2. Dentro de um contexto de metodologia ágil, onde todas as entregas são de extrema importância para o negócio, quais padrões, técnicas e frameworks de desenvolvimento de front você usaria, e justifique porque eles consomem menos tempo no processo de desenvolvimento**

- Architecture
  - Styled Components
  - File organization
- Type checking
  - Typescript
  - Proptypes
- ESLint
- Editor config
- Doc. components
  - Storybook
- Test

3. Para um portal vertical de grande acesso, onde a performance, a conversão, a relevância e a experiência do usuário são os pontos mais importantes, qual a ordem dos seguintes pontos você julga mais importante dentre eles:

- a. Performance de carregamento.
- b. Semântica.
- c. Fidelidade com o protótipo

1. Performance de carregamento
2. Semântica
  - SEO
  - Relevant information
  - Better markup
  - Maintenance
3. Fidelidade com o protótipo
  - Context