

Micro *frontends* numa aplicação de pré-contabilidade

NICOLE SILVA

Índice

Introdução

- Contexto
- Problema
- Objetivo
- Metodologia

Tecnologias

Solução

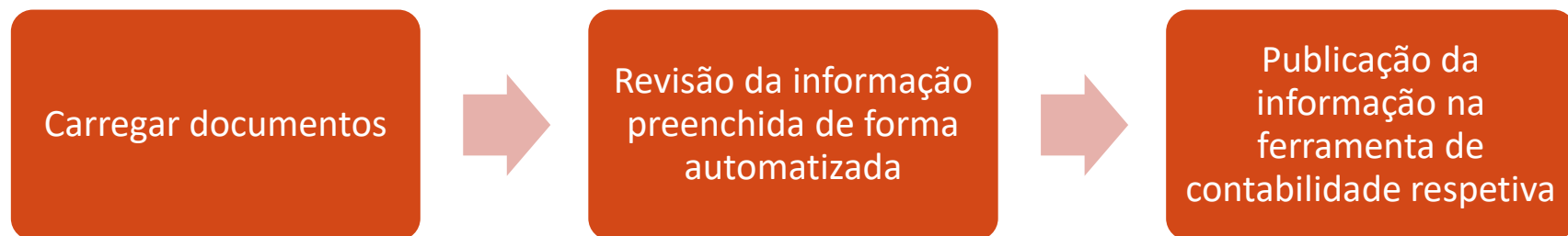
- Requisitos
- Design
- Implementação
- Implantação

Experimentação e Avaliação

Conclusões

Introdução | Contexto



Produto de Pré-contabilidade:



- O contabilista presta serviços de contabilidade a diferentes clientes/empresas;
- Cada empresa usa uma ferramenta de contabilidade diferente para registo/publicação das suas faturas;
- Com o produto Basecone, o contabilista:
 - Recebe as faturas dos seus clientes;
 - Carrega os documentos na Basecone;
 - Revê a informação preenchida pelo OCR (Reconhecimento Ótico de Caracteres);
 - Regista/publica cada fatura de cada cliente na respetiva ferramenta de contabilidade.

“Devolvemos tempo ao contabilista”

Welcome

 Pre-Accounting SYSTEM 

This is a Pre-Accounting Tool for Accountants

Automated data entry to speed up processes

Introdução | Problema

- Não é possível realizar *Continuous Integration/Continuous Delivery* (integração e entrega contínuas);
- Dependências entre equipas;
- Necessidade de comunicação constante entre equipas para resolução de problemas;
- Duração do tempo de entrega ao cliente é demorada.

A empresa necessita de adquirir

- Mais conhecimento e mais experiência no uso de micro frontends
 - Identificando uma solução viável para a implementar.



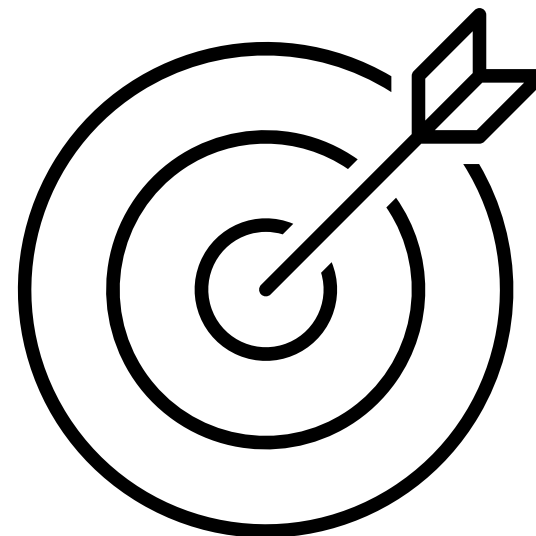
Introdução | Objetivo

Explorar uma solução baseada em micro *frontends* que traga benefícios:

- Autonomia das equipas
- *Deployability*
- Manutenibilidade

Sem penalizar:

- Performance



Introdução | Metodologia

- Recolha de informação dos principais desafios da empresa;
- Criação de um projeto representativo do atual;
- Desenvolver uma prova de conceito;
- Comparação do *frontend* monolítico com os micro *frontends*.



Tecnologias

Flexibilidade para lidar com *frameworks* diferentes.

Não recebe atualizações desde dezembro de 2019.

Necessidade de código adicional para tratar de resolução de conflitos de versões, partilhar dependências...

Em Desenvolvimento.
O pacote *npm* ainda está numa versão beta.

Dependência do *Webpack*.
Complexidade da configuração inicial.

Tecnologias

Single-SPA

Mooa

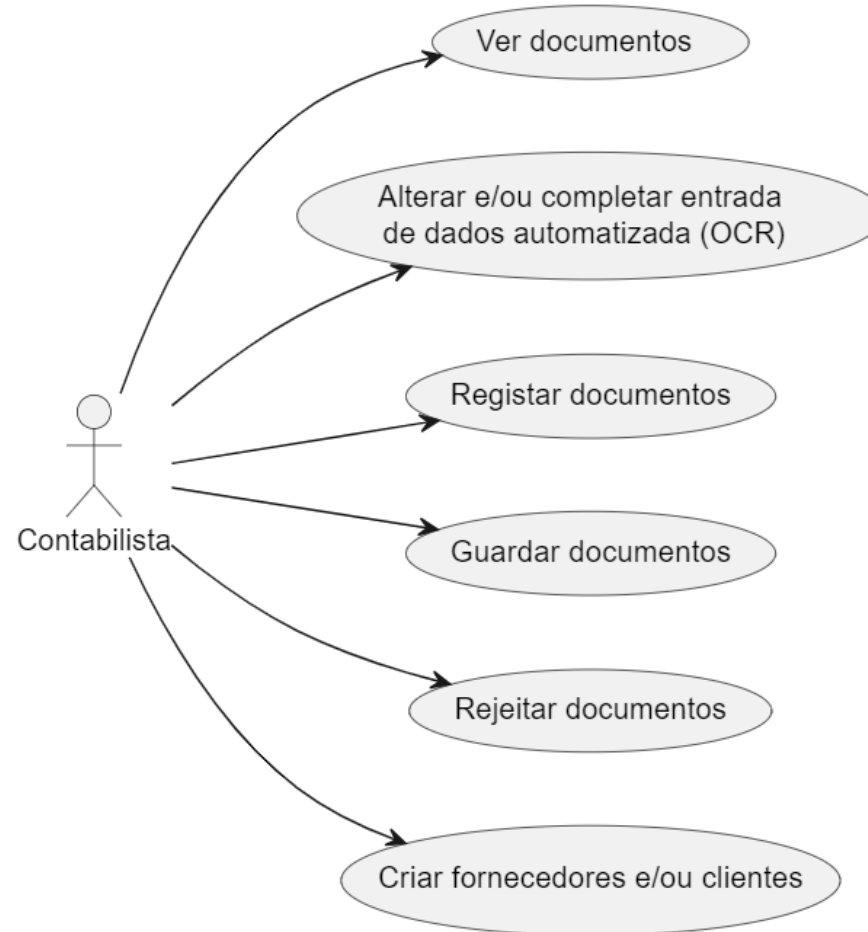
Import Maps

Native
Federation

Module
Federation

Escolhida

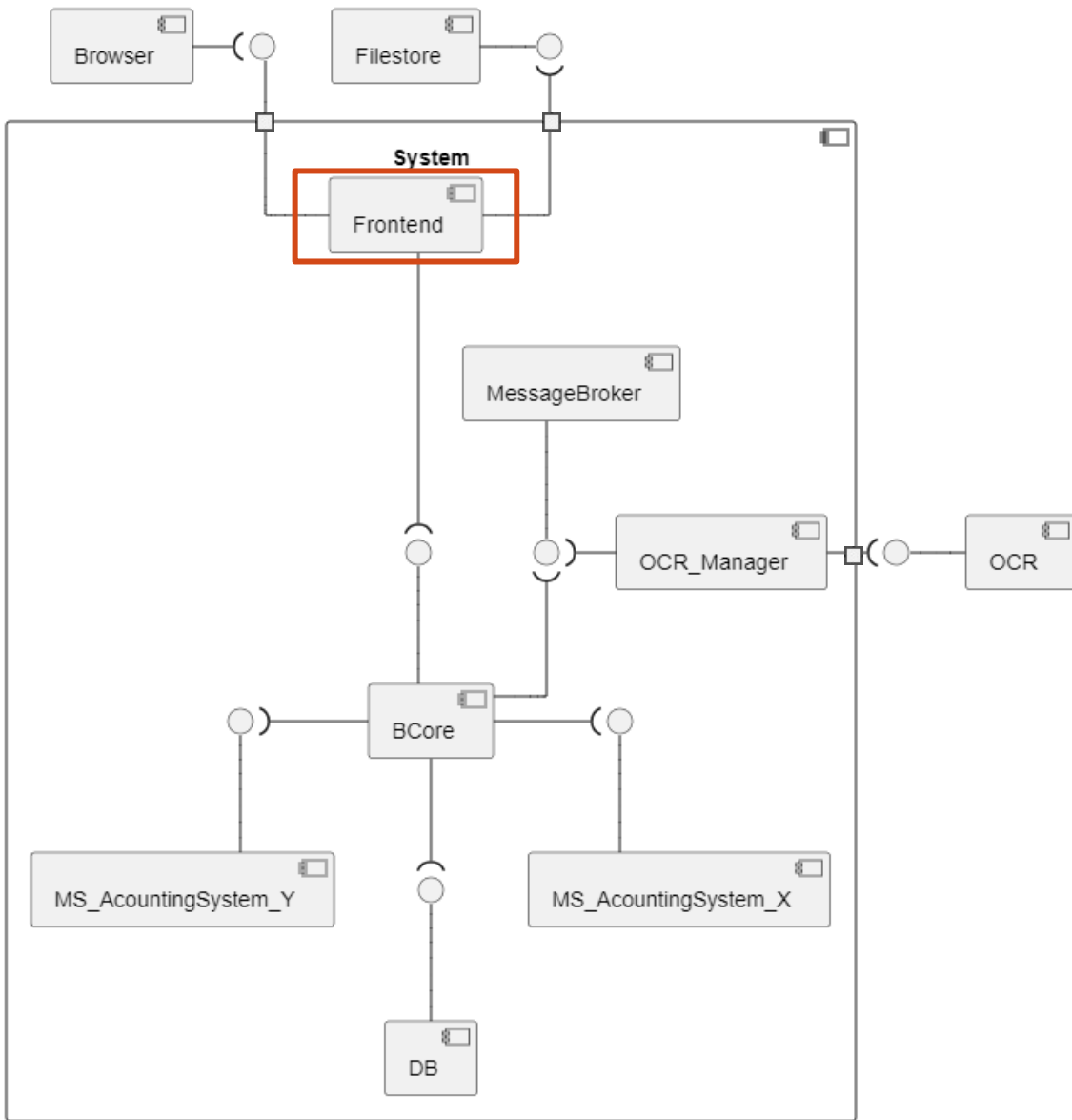
Solução | Requisitos



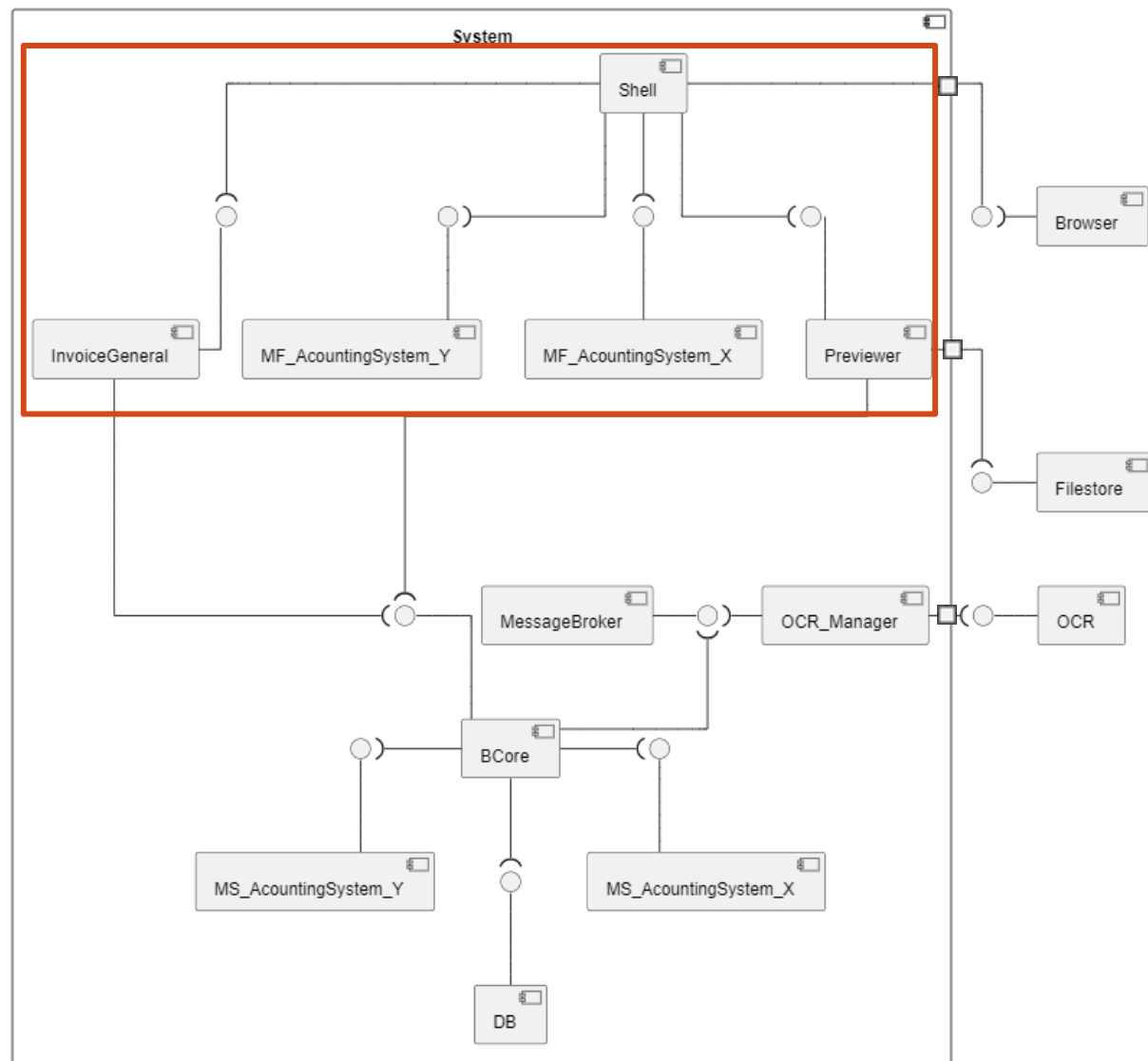
Solução | Requisitos

Requisitos Não Funcionais

- Divisão em micro *frontends*;
- Maior facilidade:
 - No suporte de mais de uma ferramenta de contabilidade;
 - Na adição de novas funcionalidades;
- Manter o desempenho da solução;
- Melhorar:
 - Autonomia das equipas;
 - Deployability;
 - Manutenibilidade.



Solução | Design



Solução | Design

Solução | Implementação

FRONTEND MONOLÍTICO

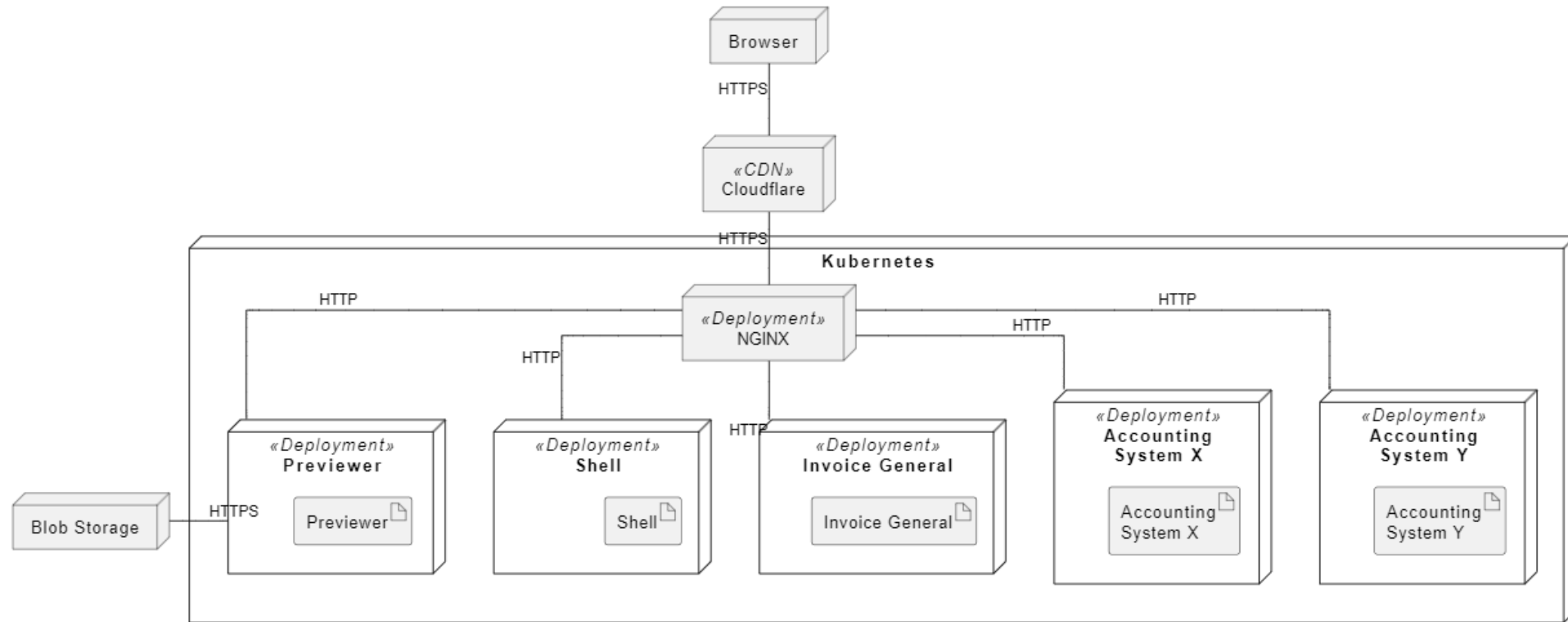
```
vatValueCalculation(value: any, id): void {  
  let vatValue =  
    this.systemForm.controls["accountingSystem"].value === "tokyo"  
      ? this.invoiceLines.value[id]["amount"] * value  
      : this.invoiceLines.value[id]["quantity"] *  
        this.invoiceLines.value[id]["unitPrice"] *  
        value;  
  this.invoiceLines.at(id).get("vatAmount").setValue(vatValue);  
}
```

MICRO FRONTENDS

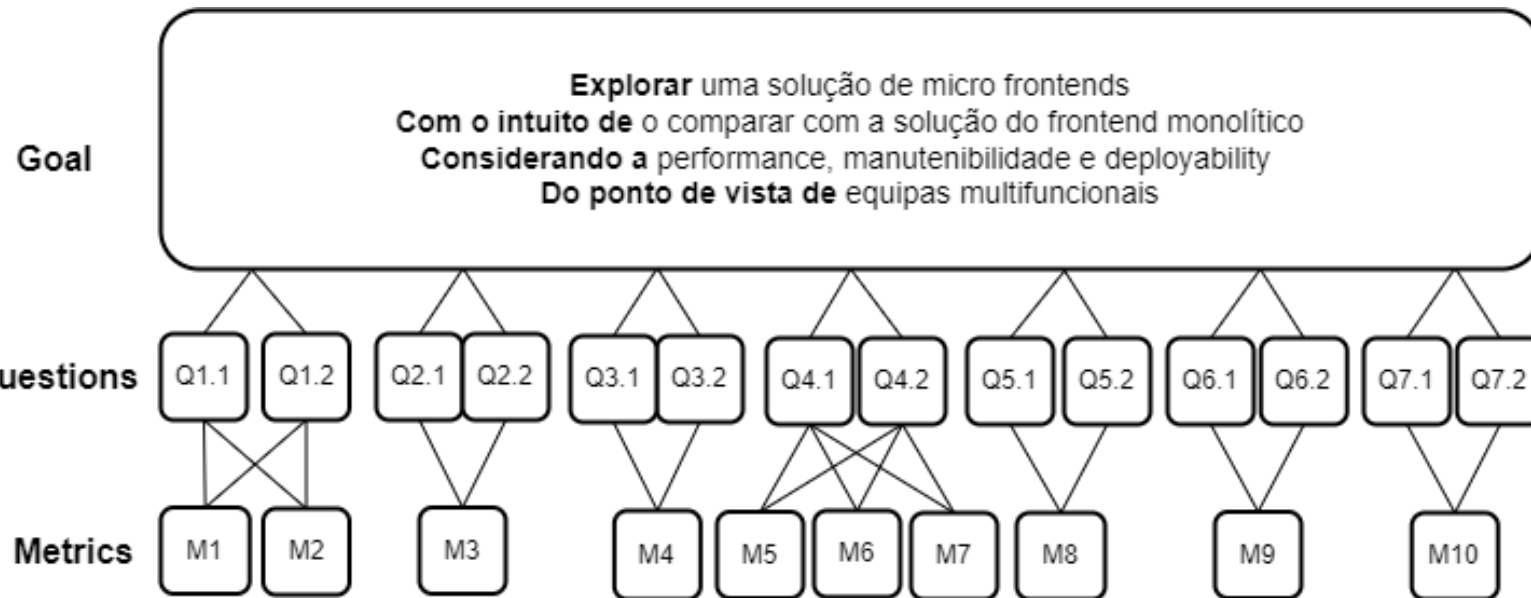
```
vatValueCalculation(value: any, id): void {  
  let vatValue =  
    this.invoiceLines.value[id]['quantity'] *  
    this.invoiceLines.value[id]['unitPrice'] *  
    value;  
  this.invoiceLines.at(id).get('vatAmount').setValue(vatValue);  
}
```

```
vatValueCalculation(value: any, id): void {  
  let vatValue = this.invoiceLines.value[id]['amount'] * value;  
  
  this.invoiceLines.at(id).get('vatAmount').setValue(vatValue);  
}
```

Solução | Implantação



Experimentação e Avaliação



*.1 - Solução de microfrontend

*.2 - Solução de frontend monolítico

Q1 - Quão difícil é manter o código-fonte?

Q2 - Quão complexo é o código-fonte?

Q3 - Quantas formas há para suportar diferentes versionamentos de dependências?

Q4 - Como é o desempenho do projeto?

Q5 - Quanto tempo demora a realizar a build da solução?

Q6 - Quanto tempo demora a implantar a solução?

Q7 - Oferece isolamento de implantação?

M1 - Maintainability rating

M2 - Cognitive Complexity

M3 - Cyclomatic Complexity

M4 - Formas diferentes

M5 - FCP (First Contentful Paint)

M6 - LCP (Largest Contentful Paint)

M7 - TBT (Total Blocking Time)

M8 - Building Time

M9- Deployment Time

M10- Número de Pipelines

Experimentação e Avaliação

Manutenibilidade

- Classificação da Manutenibilidade;
 - Relacionada com o débito técnico;
- Complexidade Cognitiva;
 - Quantifica a dificuldade de compreender o código;
- Complexidade Ciclomática;
 - Número mínimo necessário de testes para cobrir o código.

	Maintainability rating	Cognitive Complexity	Cyclomatic Complexity
Frontend Monolítico	A	23	80
InvoiceGeneral	A	17	69
Previewer	A	4	44
AccountingSystemX	A	3	28
AccountingSystemY	A	3	28
Shell	A	1	32

Experimentação e Avaliação

Versatilidade de Gestão de Dependências e Versões | Frontend Monolítico

- Carrega as versões no *package.json*

```
"dependencies": {  
  "@angular/animations": "^14.0.0",  
  "@angular/common": "^14.0.0",  
  "@angular/compiler": "^14.0.0",  
  "@angular/core": "^14.0.0",  
  "@angular/forms": "^14.0.0",  
  "@angular/platform-browser": "^14.0.0",  
  "@angular/platform-browser-dynamic": "^14.0.0",  
  "@angular/router": "^14.0.0",  
  "@azure/storage-blob": "^12.14.0",  
  "bootstrap": "^5.2.2",  
  "bootstrap-icons": "^1.9.1",  
  "ngx-doc-viewer": "^15.0.1",  
  "rxjs": "~7.5.0",  
  "zone.js": "~0.11.4"  
},
```

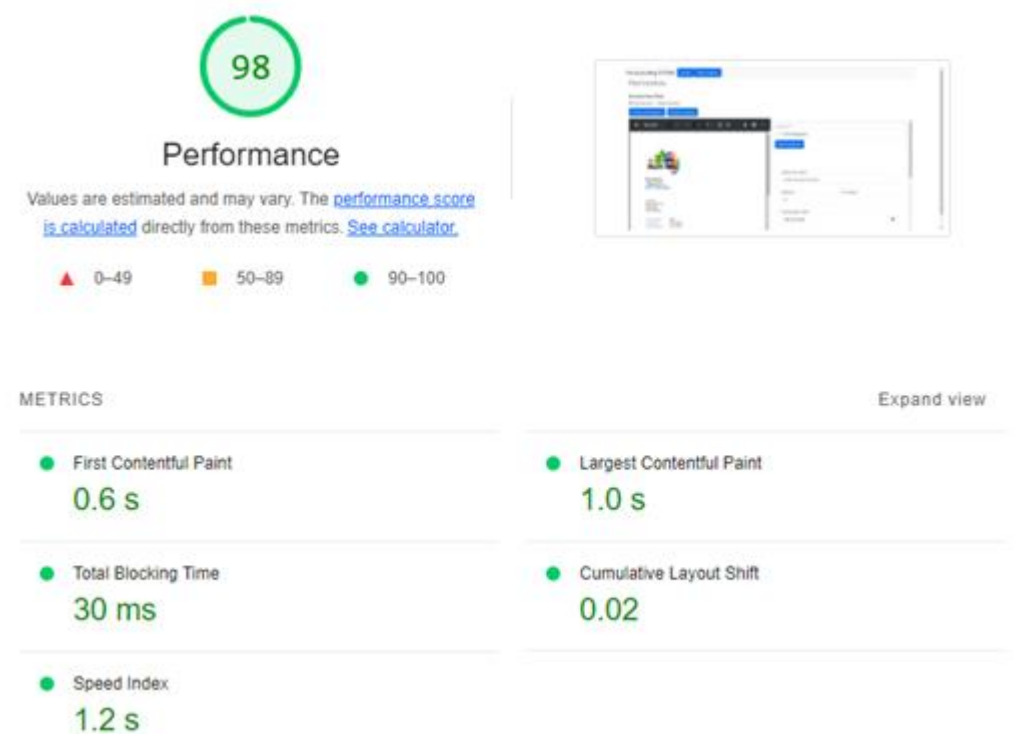
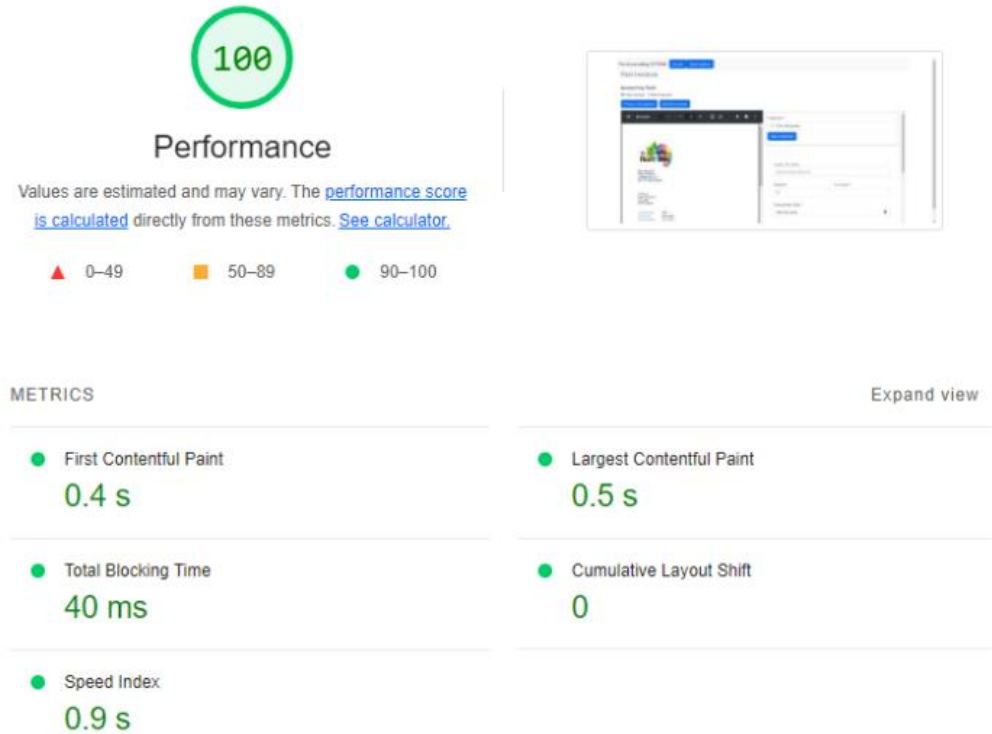
Experimentação e Avaliação

Versatilidade de Gestão de Dependências e Versões | Micro Frontends

- Por omissão, carrega:
 - A versão compatível maior;
 - Várias versões quando há incompatibilidade;
- Uso das versões e dependências do *package.json*;
- *singleton:true*;
 - Versões compatíveis;
 - A dependência é carregada uma vez;
 - Versões incompatíveis;
 - Surge na consola:
 - Um aviso;
 - Um erro (*strictVersion: true*) → *fail fast*;
- Sem *singleton* definido;
 - Não partilha essas dependências com os restantes *micro frontends*.

```
shared: {  
  ...shareAll({ singleton: true, strictVersion: true, requiredVersion: 'auto' }),  
},
```

```
shared: share({  
  "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },  
})
```



Experimentação e Avaliação

PERFORMANCE

Experimentação e Avaliação

Deployability

- Duração da *Build*;
 - Teste de Shapiro-Wilk;
 - Teste Wilcoxon rank-sum (Não paramétrico);
- Duração do *Deploy*;
 - Teste de Shapiro-Wilk;
 - *T-test* (Distribuição normal);
- Implantações Independentes;
 - Isolamento da implantação;
 - Releases diárias;
 - Integrações e entregas contínuas.

	Build	Deploy
Frontend Monolítico	1m 48s	13s
InvoiceGeneral	2m 49s	12s
Previewer	2m 35s	12s
AccountingSystemX	2m 10s	7s
AccountingSystemY	2m 14s	9s
Shell	2m 0s	8s

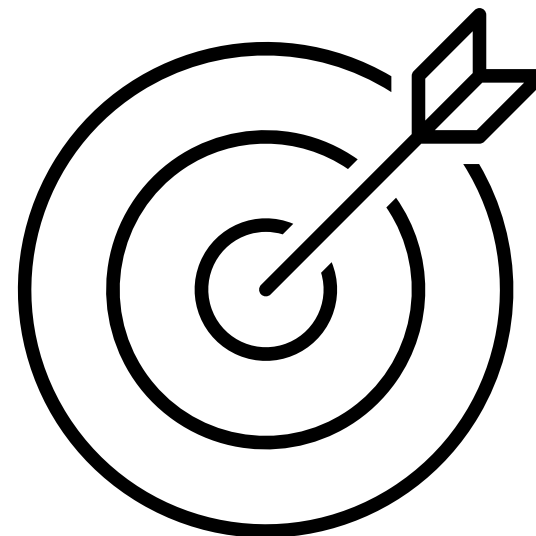
Conclusões | Objetivo Alcançado

Explorar uma solução baseada em micro *frontends* que traga benefícios:

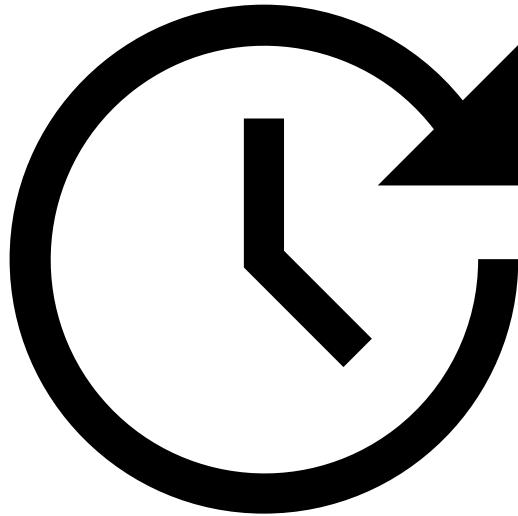
- Autonomia das equipas ✓
- *Deployability* ✓
- Manutenibilidade ✓

Sem penalizar:

- Performance ✓



Conclusões | Trabalho Futuro



- Algumas métricas usadas tiveram resultados similares;
- Necessidade de rever o tempo de *build*;
- Uso de outras métricas que analisem outros aspetos da autonomia das equipas;
- Partilha de informação com a empresa;
- Planos para iniciar a sua implementação;
- Reanálise da tecnologia mais adequada.

Micro *frontends* numa aplicação de pré-contabilidade

NICOLE SILVA