

Seminário: Arquitetando uma aplicação em micro-frontends

Fernando Rodrigues de Moraes

fr.moraes@unesp.br



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Conteúdo do Projeto

- Introdução e Motivação para utilização de micro-frontends
- Fundamentos de micro-frontends
- Arquitetando uma aplicação
- Implementação no projeto final
- Resultados obtidos e resultados esperados

Introdução a Micro-frontends

- Modularização de uma aplicação frontend em pequenas partes, divididas em pequenas aplicações frontend
- Inspirada diretamente por micro-serviços
- Ao passo em que reduz a complexidade de um software (e de sua regra de negócio), aumenta a necessidade de automação, monitoramento e comunicação.

Introdução a Micro-frontends

- Arquitetar um software em micro-frontends, como toda decisão arquitetural, deve partir da motivação para solucionar um ou mais problemas.
- Praticamente exige uma cultura semelhante a de micro-serviços para ser adotada
- Uma má implementação podem gerar transtornos e até a falha arquitetural de um software

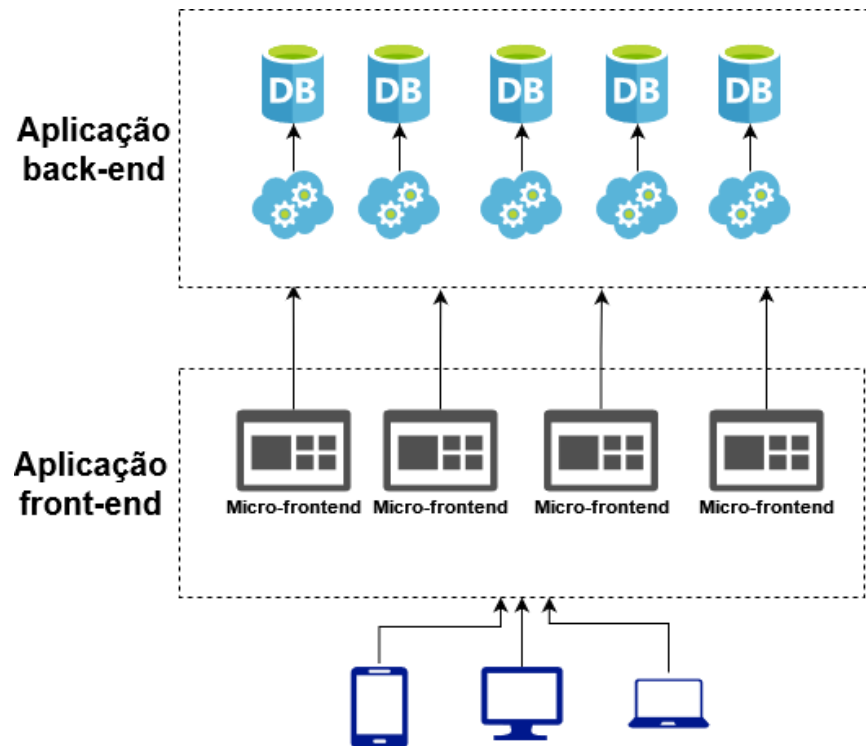
Motivação para implementação de Micro-frontends

- A alta modularização do software em micro-frontends traz benefícios em diversas aplicabilidades
 - Projetos em larga escala, domínio extenso
 - Times grandes prestando manutenção ao software
 - Refatoração de um monolito frontend
 - Exige uma cultura DevOps bem difundida

Motivação para implementação de Micro-frontends

- A má implementação de micro-frontends pode gerar problemas ao arquitetar e desenvolver um software
 - Aumento da complexidade do software
 - A não adesão de uma cultura de DevOps pode gerar dificuldades na implementação e manutenção
 - A aplicação da arquitetura em projetos que não há necessidade pode ser uma má ideia

Fundamentos e Princípios de Micro-frontends



Fundamentos e Princípios de Micro-frontends



Princípios de micro-serviços do livro - Building Microservices (O'Reilly)

Fundamentos e Princípios de Micro-frontends

- Entrega independente
 - Em uma aplicação arquitetada em micro-frontends, é possível que equipes entreguem um micro-frontend por vez
 - Redução no risco das entregas
- Isolamento de Falhas
 - Com cada módulo separado, falhas em uma aplicação micro-frontend são isoladas dos outros módulos

Fundamentos e Princípios de Micro-frontends

- Cultura de automação
 - Ao passo em que uma aplicação é dividida em micro-aplicações, aumenta a complexidade de *setup* e entrega da aplicação
 - Ao escalar, se torna fundamental automatizar para reduzir tempo de entrega e de desenvolvimento
- Altamente monitorável
 - Cada micro-frontend é independente e pode ser monitorado de forma independente, facilitando a detecção de falhas e usabilidade do usuário

Fundamentos e Princípios de Micro-frontends

- Modelado acerca de um domínio
 - Assim como em micro-serviços, o domínio também se torna modularizado
 - Portanto, descentralizando a governança empresarial
- Governança descentralizada
 - Facilidade em equipes diferentes com responsabilidade sob diferentes micro-frontend's
 - Domínio guiado para equipes, sem impactar outras equipes

Fundamentos e Princípios de Micro-frontends

- Detalhes de implementação abstraídos
 - Diferentes equipes não precisam conhecer detalhes de certas implementações
 - Fundamenta uma boa implementação de um "*core*" da aplicação
 - Acelera o desenvolvimento com domínio bem fundamentado

Arquitetando uma aplicação micro-frontend em Flutter

- Flutter
 - Utiliza linguagem Dart
 - Framework para desenvolvimento multi-plataforma
 - iOS, Android
 - Web
 - Windows, Linux, macOS

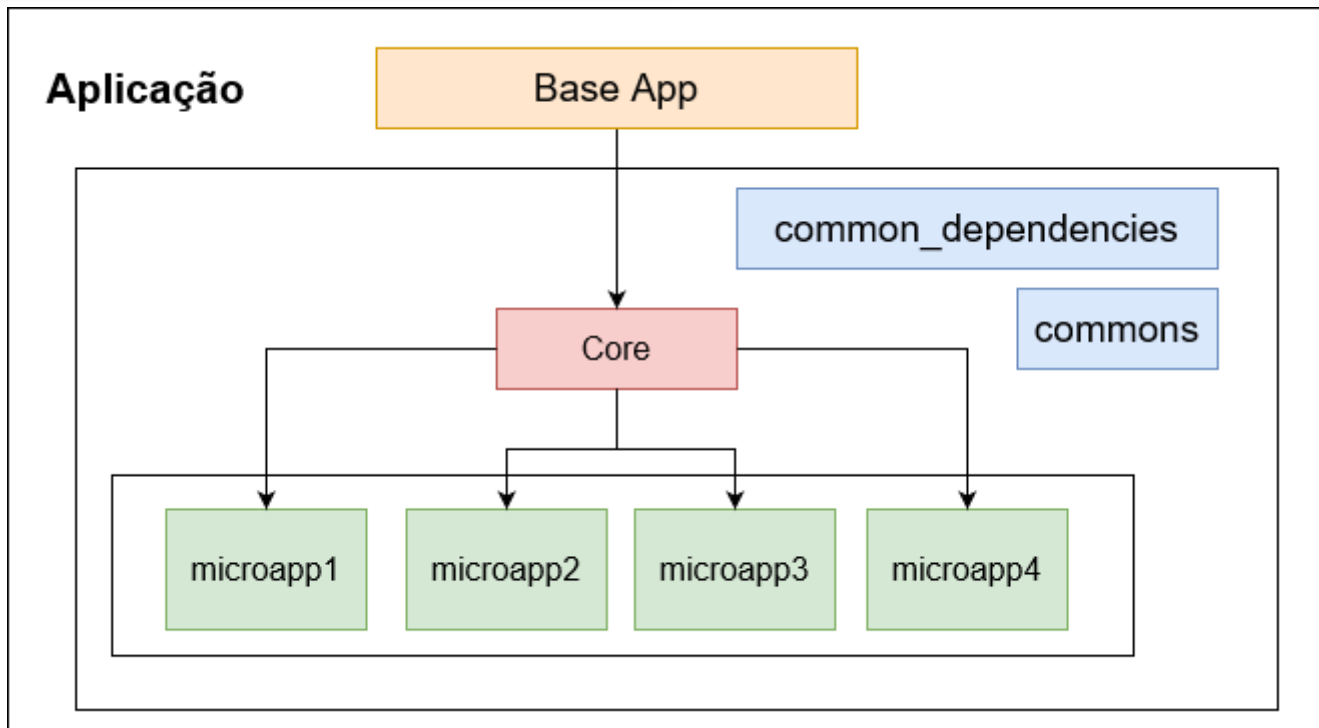
Arquitetando uma aplicação micro-frontend em Flutter

- Micro-app responsável por classes e métodos comuns
- Micro-app responsável por dependências em comuns
- Core responsável pela comunicação entre micro-apps
- Entrypoint da aplicação
- Módulos da aplicação se tornam micro-frontends

Arquitetando uma aplicação micro-frontend em Flutter

- Micro-app responsável por classes e métodos comuns
- Micro-app responsável por dependências em comuns
- Core responsável pela comunicação entre micro-apps
- Entrypoint da aplicação
- Módulos da aplicação se tornam micro-frontends

Arquitetando uma aplicação micro-frontend em Flutter



Visão geral da estrutura da arquitetura

Arquitetando uma aplicação micro-frontend em Flutter

➤ Base App

- Aplicação base, servindo apenas como ponto de entrada da execução da aplicação

```
void main() => runApp(  
  ModularApp(  
    module: AppModule(),  
    child: const AppWidget(),  
  ),  
);
```

Arquitetando uma aplicação micro-frontend em Flutter

➤ common

- Responsável por domínio em comum, elementos visuais, classes e métodos
- Domínio de usuários/autenticação que deve ser consumido em diversos módulos
- Um botão/ícone que é utilizado em diversas telas
- Métodos úteis de formatação de datas, monetárias, etc

Arquitetando uma aplicação micro-frontend em Flutter

- `common_dependencies`
 - Dependências de pacotes comuns entre aplicações
 - Modular (Modularização de aplicação e injeção de dependência)
 - MobX (Gerência de estado)
 - Dartz (Dependência para programação funcional no Flutter)
 - `Build_runner` (Dependência para geração automática de código)

Arquitetando uma aplicação micro-frontend em Flutter

➤ Core

➤ Responsável pela comunicação dos micro-frontends e tema da aplicação

```
class AppWidget extends StatelessWidget {  
  const AppWidget({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp.router(  
      title: 'MAppBank',  
      theme: ThemeData(primarySwatch:  
Colors.blue),  
      routeInformationParser:  
Modular.routeInformationParser,  
      routerDelegate:  
Modular.routerDelegate,  
    );  
  }  
}
```

```
class AppModule extends Module {  
  @override  
  final List<Bind> binds = [];  
  
  @override  
  final List<ModularRoute> routes = [  
    ModularRoute(Modular.initialRoute, module:  
LoginModule()),  
    ModularRoute('/home/', module: HomeModule()),  
    ModularRoute('/area_pix/', module: PixModule()),  
    ModularRoute('/extrato/', module:  
ExtratoModule()),  
    ModularRoute('/meus_cartoes/', module:  
MeusCartoesModule())  
  ];  
}
```

Arquitetando uma aplicação micro-frontend em Flutter

- Identificando micro-frontends
 - Separando por domínio
 - Extrato da conta
 - Área de PIX
 - Pagar (Boleto)
 - Recarga de celular
 - Cadastro e consulta de cartões
 - Perfil/Dados do usuário



Arquitetando uma aplicação micro-frontend em Flutter

➤ Versionamento por micro-frontend

```
microapp1:  
  git:  
    url: git://github.com/fernandormoraes/microapp1.git  
    ref: v1.0.0
```

➤ Integração Contínua

- Testar integração dos micro-frontends
- Realizar o build da aplicação

➤ Entrega contínua

- Realizar a entrega da aplicação atualizada e testada

Implementação no projeto final

➤ Demonstração

Artigo para análise crítica

Daojiang Wang, DongMing Yang, Huan Zhou, Ye Wang, Daocheng Hong, Qiwen Dong, Shubing Song,
A Novel Application of Educational Management Information System based on Micro Frontends, Procedia Computer Science, Volume 176, 2020, Pages 1567-1576, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2020.09.168>.



Referências

L. Mezzalana and O. M. C. Safari, Building Micro-Frontends. O'Reilly Media, Incorporated, 2021. [Online]. Available: <https://books.google.com.br/books?id=zu9ZzgEACAAJ>