

Aplicação de Controlo e Gestão de **Obra**

8220337 - Hugo Guimarães

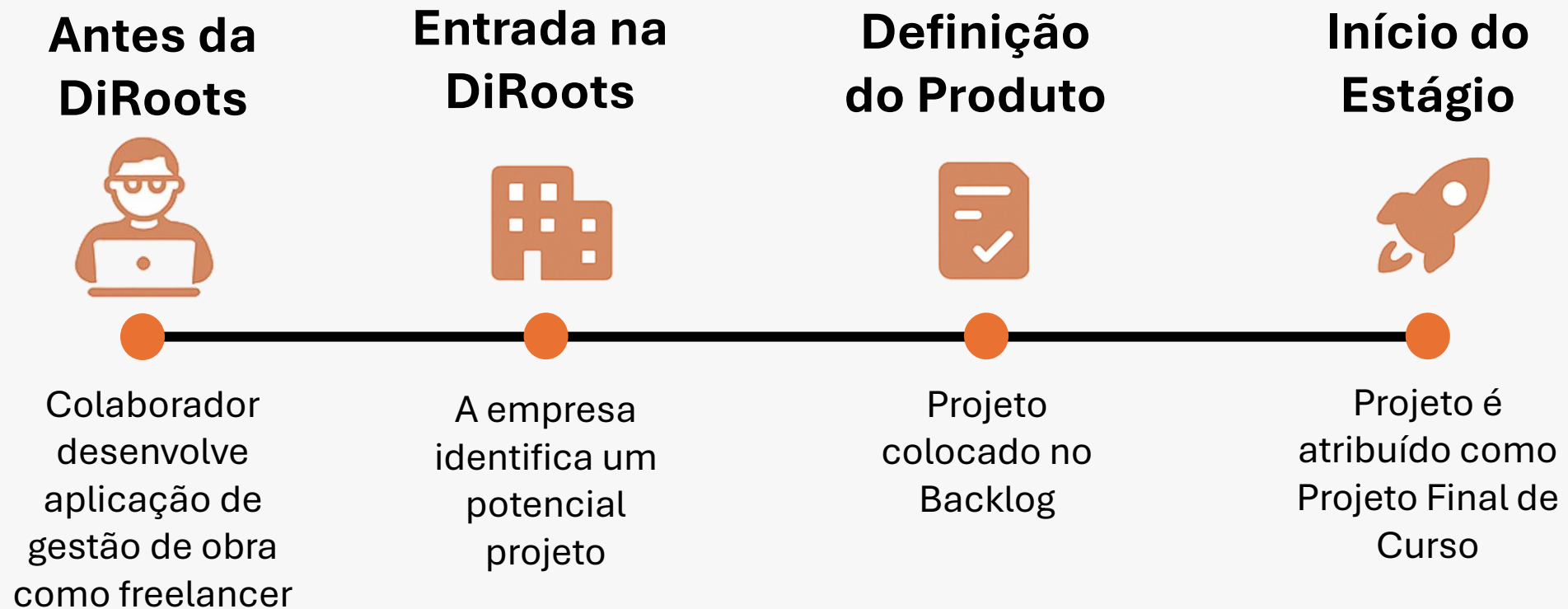
Orientador: Bruno Oliveira

Entidade de Acolhimento

- 7 anos de atividade;
- Trabalha do setor AEC (Arquitetura, Engenharia e Construção)
- Presta serviços de consultoria BIM
- Desenvolve:
 - Software à medida
 - Aplicações Web / Cloud
 - Plugins



Como Surgiu o Projeto?



Objetivo Geral

Desenvolver uma aplicação Web que seja **modular** e **escalável** para **controle e gestão de obra**, de modo a suprir as necessidades de uma empresa do setor AEC

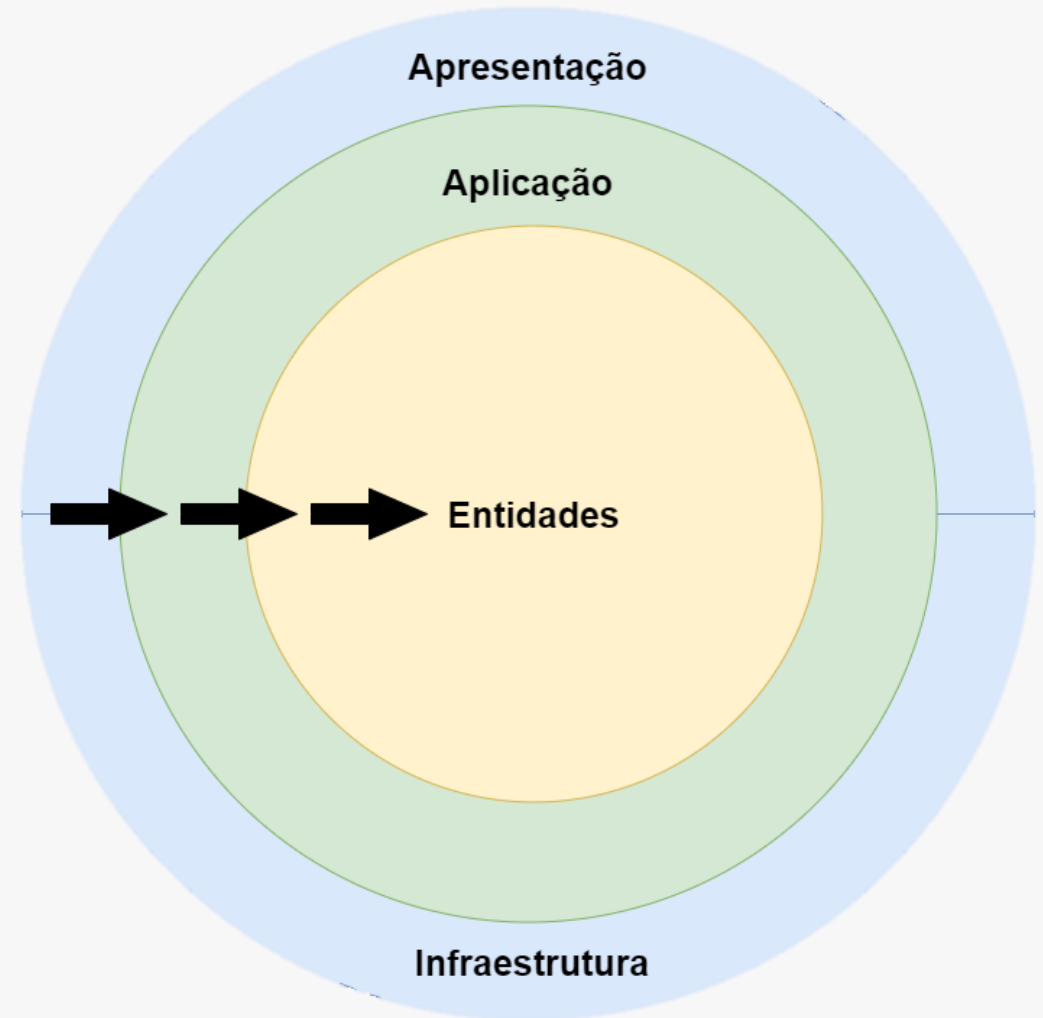
Objetivos Específicos

- Aplicar padrões arquiteturais internos da empresa;
- Implementar uma arquitetura em microserviços organizados por domínio de negócio
- Desenvolver funcionalidades para:
 - Recursos Humanos
 - Planeamentos de Projetos
 - Custos e Clientes
 - Processamento de salários
 - Gestão operacional

Principais Arquiteturas

Arquitetura Limpa

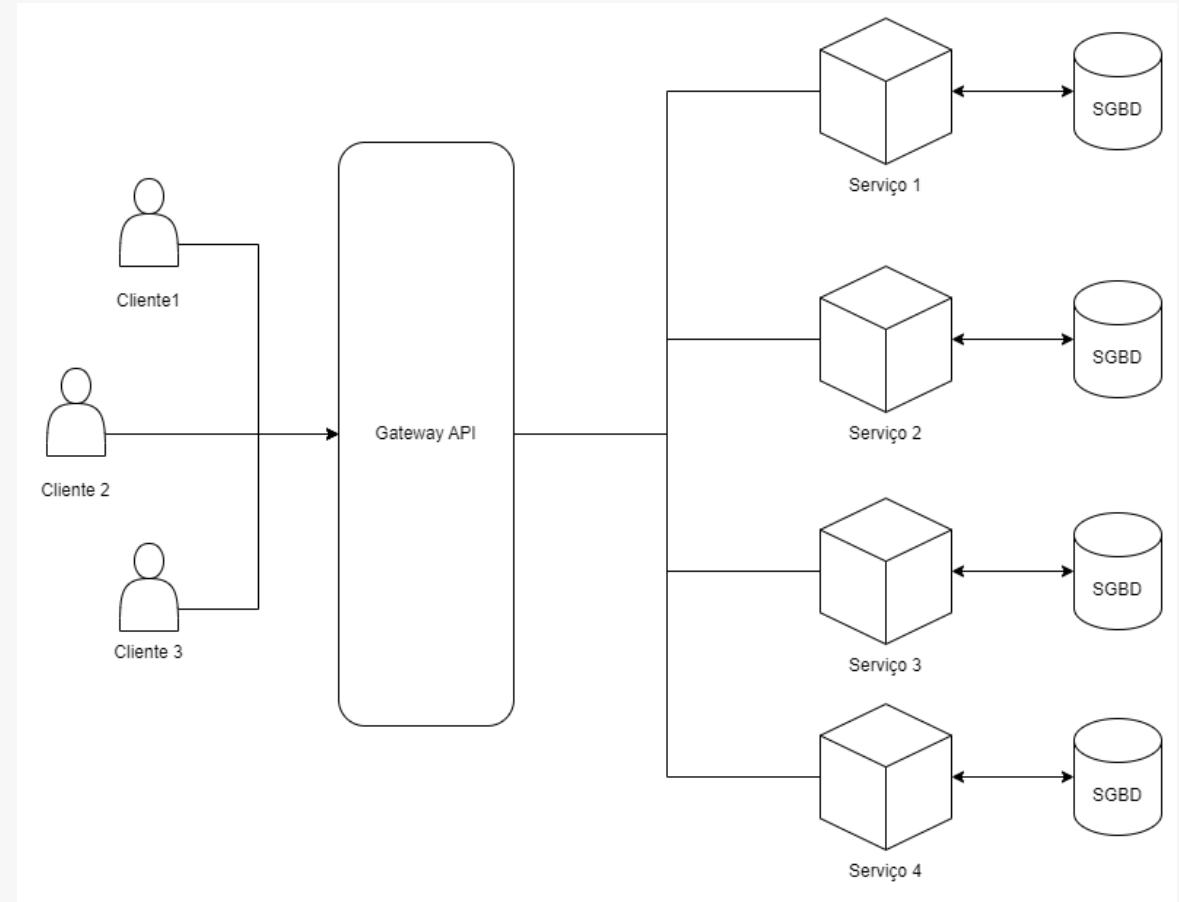
- Organiza o projeto em **camadas independentes**
- Fluxo de dependências vai sempre de fora para dentro (regra da dependência)
- Utilizado para tornar o **código mais limpo e fácil de manter**



Principais Arquiteturas

Arquitetura Microserviços

- Sistema dividido em **várias partes independentes**
- Facilita a **escalabilidade** e a **manutenção**;
- Foi utilizada de modo a garantir **modularidade**
 - separação clara dos domínios do negócio



Principais Padrões



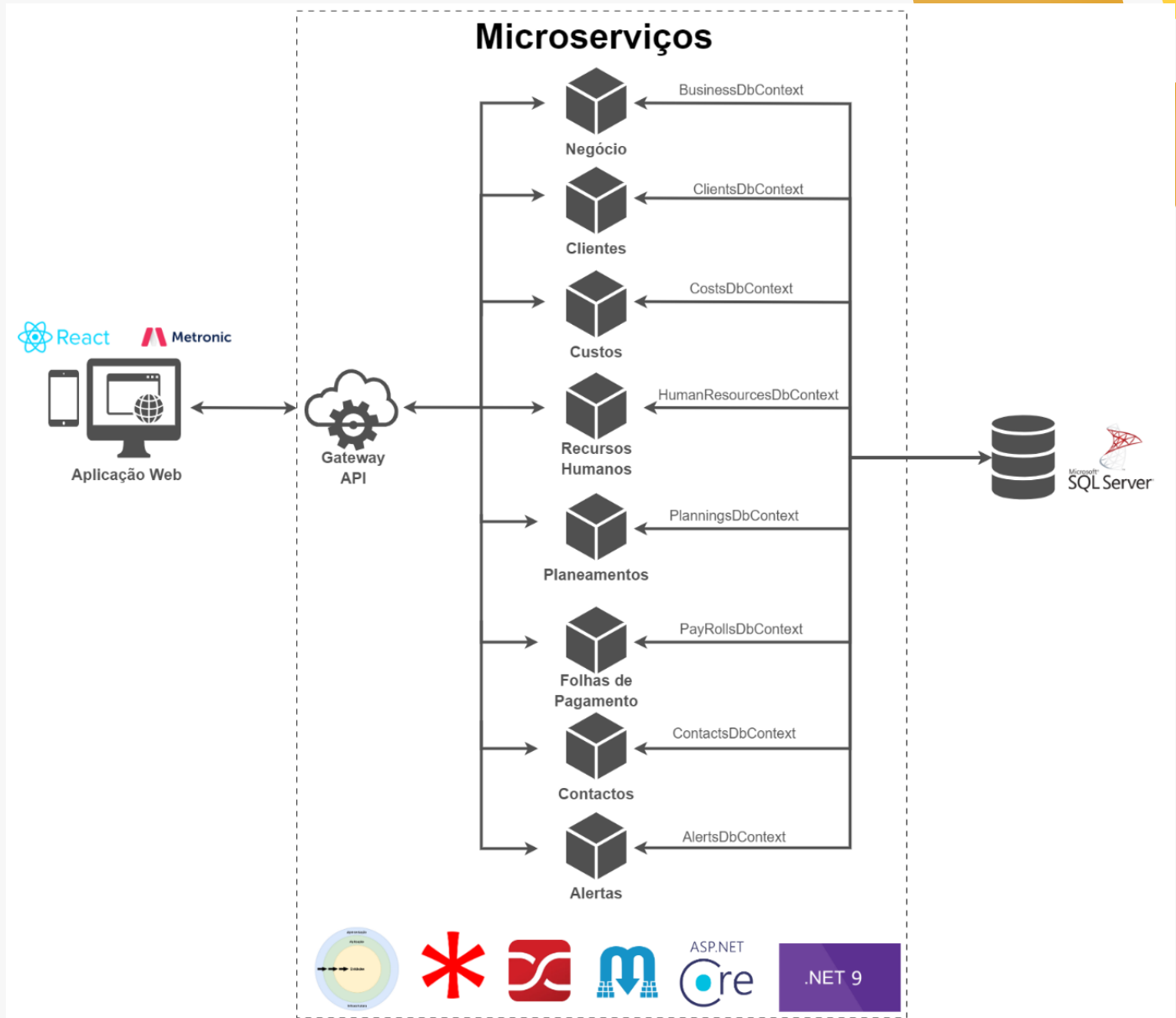
CQRS

- Separa operações de leitura e escrita
- Mantém o sistema **organizado**;
- Facilita a **escalabilidade** e **aplicação de regras**

Mediator

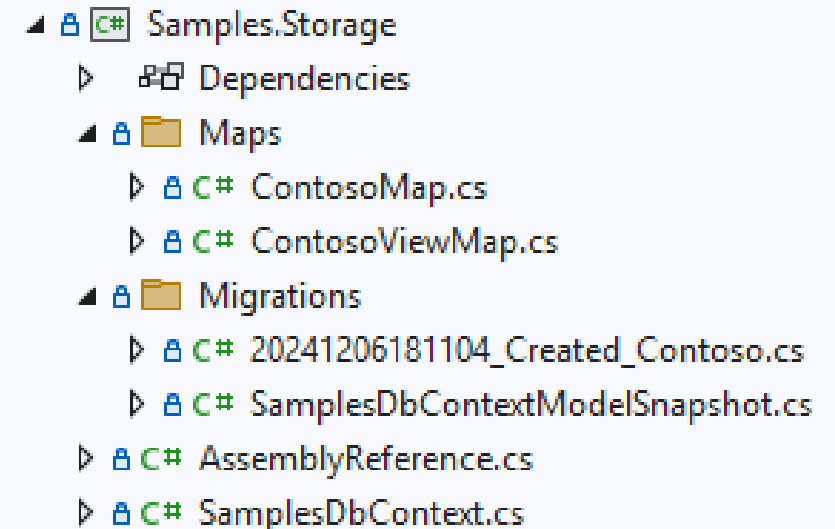
- **Centraliza** a comunicação entre componentes;
- Evita chamadas diretas entre classes,
- Torna o sistema **mais limpo** e **fácil de alterar** no futuro

Arquitetura do Sistema



Garantia da Integridade Referencial

- Não existem chaves estrangeiras entre tabelas de serviços diferentes
 - Relacionamentos são simulados ao nível da aplicação
- Garantida por validadores do **Fluent Validation**
- Leitura dos dados é realizada a partir de views
 - Views não SQL, mas modelos de leitura com relacionamentos simulados



```

Samples.Storage
├── Dependencies
├── Maps
│   ├── ContosoMap.cs
│   └── ContosoViewMap.cs
├── Migrations
│   ├── 20241206181104_Created_Contoso.cs
│   └── SamplesDbContextModelSnapshot.cs
├── AssemblyReference.cs
└── SamplesDbContext.cs

```

Como os serviços comunicam entre si?



- Comunicação baseada em eventos
- Sem sistemas de mensagem externa
 - Ex: RabbitMQ, Kafka
- Funcionam a partir de **notificações** no MediatR
 - Permite que múltiplas partes interessadas reajam a um acontecimento no sistema
- Garantem **desacoplamento** e facilita **manutenção futura**

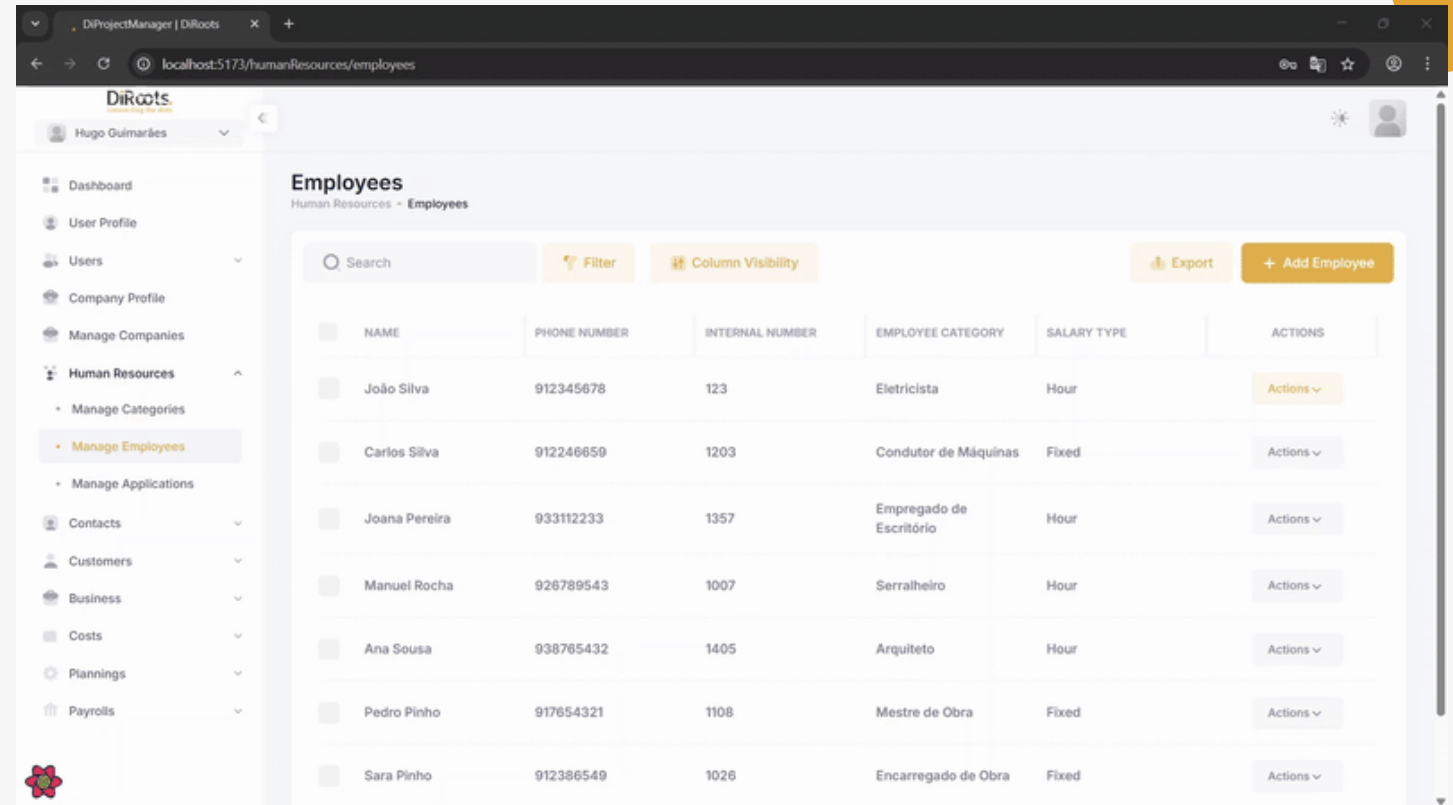
Metodologia e Abordagem de Trabalho

- Divisão por domínios de negócio
- Adoção de templates internos
- Abordagem iterativa e incremental
 - Permitiu feedback contínuo do supervisor
 - Alinhamento com os objetivos da empresa



Resultados Alcançados

- Microserviços finalizados e visíveis no Frontend:
 - Recursos Humanos
 - Negócio
 - Planeamentos
 - Contactos
 - Clientes
- Exportação para Excel;
- Suporte a internacionalização (i18n)
- Automatização de tarefas repetitivas:
 - I18n Helper
 - CodeCare



NAME	PHONE NUMBER	INTERNAL NUMBER	EMPLOYEE CATEGORY	SALARY TYPE	ACTIONS
João Silva	912345678	123	Eletricista	Hour	Actions
Carlos Silva	912246659	1203	Condutor de Máquinas	Fixed	Actions
Joana Pereira	933112233	1357	Empregado de Escritório	Hour	Actions
Manuel Rocha	926789543	1007	Serralheiro	Hour	Actions
Ana Sousa	938765432	1405	Arquiteto	Hour	Actions
Pedro Pinho	917654321	1108	Mestre de Obra	Fixed	Actions
Sara Pinho	912386549	1026	Encarregado de Obra	Fixed	Actions

Dificuldades e como foram superadas

Dificuldades



Falta de documentação



Compreender o template do Bakcend



Entender a aplicação da arquitetura microserviços e o padrão CQRS

Soluções



Explorar o Código



Reuniões com o supervisor



Prototipagem

Trabalho Futuro

- **Dashboards**
 - Visualização clara de indicadores custo
- **Relatórios**
 - Automatização da geração de relatórios salariais para envio contabilístico
- **Alertas**
 - Notificações proativas para eventos críticos ou prazos.

Conclusões



Conclusões Técnicas

- Aplicação está funcional;
- Todas as entidades base estão criadas;
- Está pronta para evoluir e escalar para diferentes contextos do setor AEC

Reflexão Técnica

- Aprofundei competências em práticas e tecnologias de desenvolvimentos modernas;
- Compreendi a importância de uma arquitetura limpa e modular

Reflexão Pessoal

- Consegui desenvolver a minha capacidade de adaptação;
- Experiência prática que me preparou para o mercado de trabalho



Obrigado pela Atenção