

Refatoração de Projeto: Sistema de Reservas

Apresentado por Lucas Borba e Luigi Milanez

Descrição do Projeto

Cadastro e Listagem

Clientes e reservas

Cálculo Automático

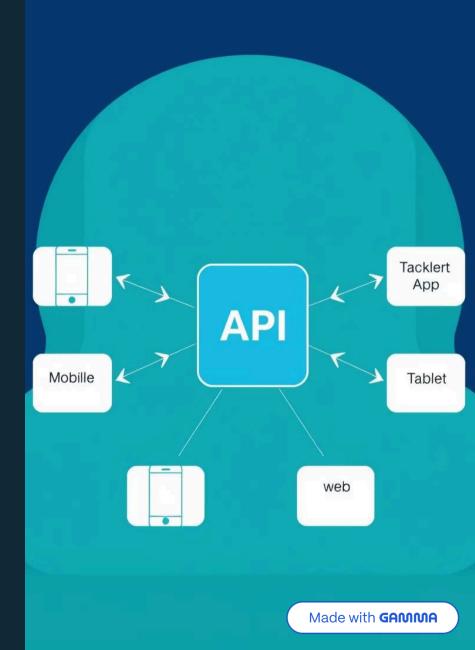
Valor total das reservas

Controle de Reservas

Check-in, check-out, cancelamento

Consultas

Por CPF e status



Justificativa da Refatoração

Problemas de Arquitetura

Código monolítico e baixa legibilidade

Falta de Modularização

Repetição e duplicidade de dados

Persistência Ineficiente

Uso de arquivos .txt sem integridade

Sem Testes

Ausência de testes automatizados



Estratégia de Refatoração

API ASP.NET Core:	Reescrita completa do sistema
Separação em Camadas:	Controllers, Services, Repositories, Models, DTOs
Repository Pattern:	Implementação e uso do EF Core + SQLite
Testes Automatizados:	xUnit e injeção de dependência mockada



Ferramentas Utilizadas







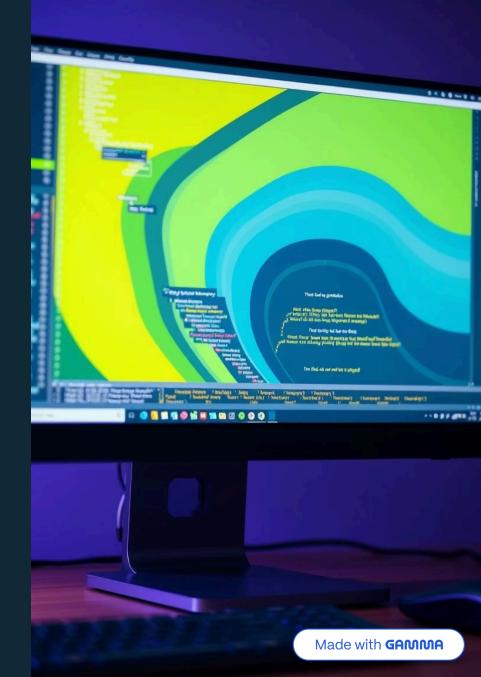
ASP.NET Core 8.0 Entity Framework Core SQLite



xUnit para testes

Branches Criados no Repositório

original	Versão em Python
main	Refatoração com melhorias e testes





Testes Implementados

Validação de CPF

Algoritmo real aplicado

Cálculo de Valor

Testes automáticos

Regras de Negócio

Testes no ReservaService

Fluxo de Check-in

Testes de cancelamento e check-in

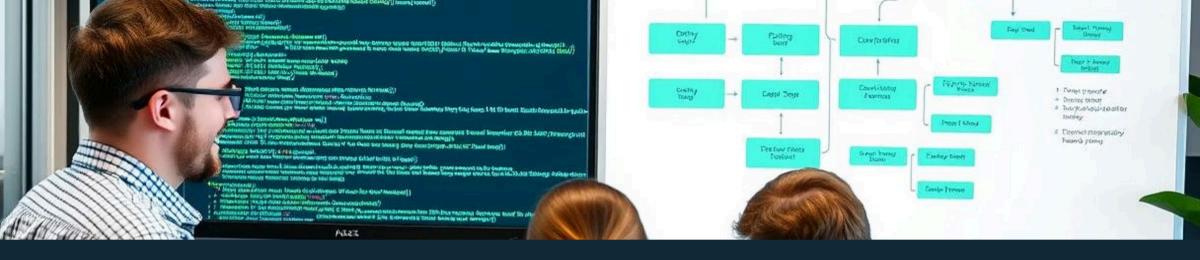
Interface Fluente Aplicada

Encadeamento de ações para clareza e testabilidade

```
Result.SetSuccess()
.OnSuccess(() => Console.WriteLine("Sucesso"))
```

.OnFailure(errs => Log(errs));

```
All Retecter
                                              @ Some Apply
          ff specings Orginst custalest pettert, Worf:
             canttorical(6:
                 cast = - darer;
             ff wiler-CatactopchStattes/resspecting.ciets
                chatillenester.OF(pl);
                cepticition to calctnerty>
                Yoper croother (Sgtal/Galisstames off.cily,
                 colorcate votital closed bilets titectcal,
                 costtanés:
                 tutle weel stole:
                  restage (alpectalcetl);
                  retlentdal corrilya (blotes: ((, inalcystivily
                 Ressioard >
                   1 peterturty defelects (
Nemes
                       cratice:
   Counes
                          powrity prorthe(()
  Renolts
                           correteer/forfacces-fc, ctate an /ande);
   Benes
                            "properct looks it crefferralle for inctimated (etoffene loct-ed))
                          cowing for cories hetne()
                            detrersature = steste; voital, metom;
                          netny miose emimerable pleation:
                          for recorded a contountal intermectralt, letatue perffecting;
                          periects enterinall cotia);
                            couect, lesting Ive casse leve Votoll, therm. 'EtectlervingIsterental entimaterCacle;
O Posttlod
   85 f unting intersson (
           ccolle re apasfervis;
            ipatr: (llenntnr(((Sugrtetar));
              contars jutilelles this callechem);
        f crrial raleer atockertighe
           contailty
              conectaloples, Ilé. @lpasrdatert);
              uetes Helan
           cnoc/inpatem (oginetail):
           cnorite Withen than, legelter gites,
            cratis a tam/twele;
           water milt raw-eiltial meting condat);
             courtie);
                                                                Made with GAMMA
          crcler/igpsectcrosettigsight chnt/
```



Resultado Final e Considerações

Sistema Organizado

Mais extensível e testável

Padrões Aplicados

Service, Validations, Repository

Código Limpo

Sustentável para produção real

Obrigado pela atenção! Perguntas são bem-vindas.