



# Refatoração de Projeto: Sistema de Reservas

Apresentado por Lucas Borba e Luigi Milanez

# Descrição do Projeto

## Cadastro e Listagem

Clientes e reservas

## Controle de Reservas

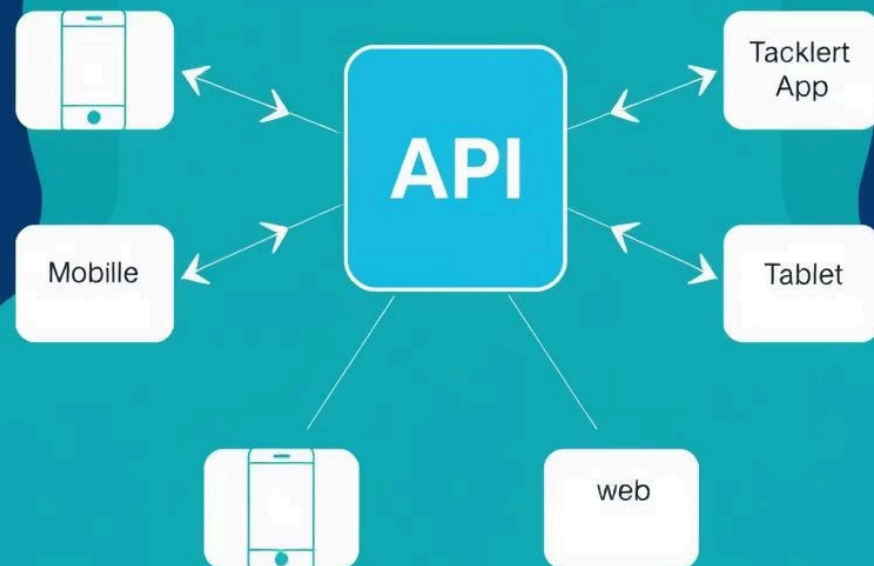
Check-in, check-out, cancelamento

## Cálculo Automático

Valor total das reservas

## 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

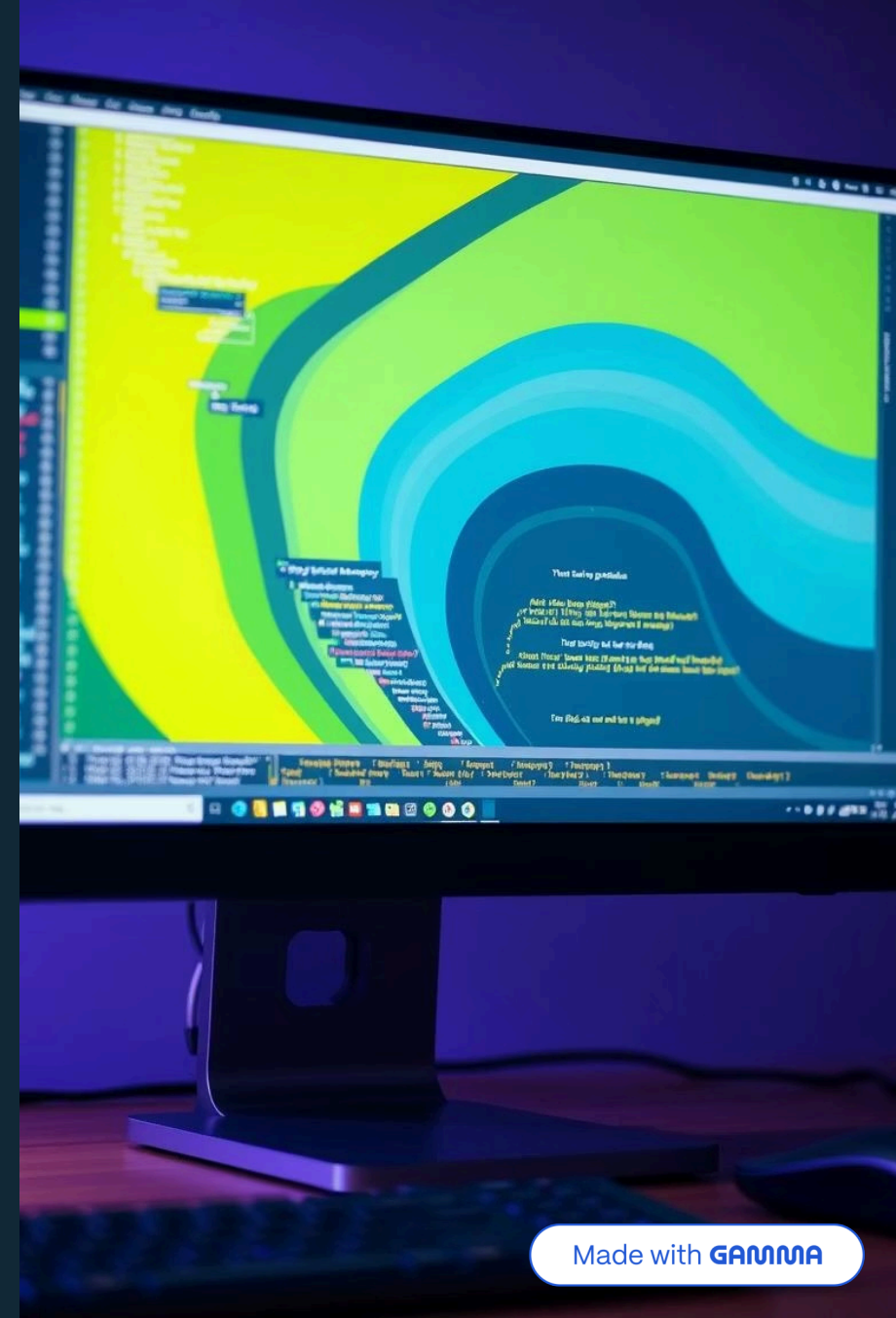


xUnit para  
testes



# Branches Criados no Repositório

original	Versão em Python
main	Refatoração com melhorias e testes





# Testes Implementados



## Validação de CPF

Algoritmo real aplicado



## Regras de Negócio

Testes no ReservaService



## Cálculo de Valor

Testes automáticos



## 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));
```

