

# Relatório Técnico

## Plano de Trabalho (previsto vs. realizado)

Previsto:

Separei o projeto em tarefas (levantamento, definição de arquitetura, implementação do domínio, API microserviço e documentação) com um total estimado de 13 horas.

Realizado: Toda a implementação seguiu o planejamento sem desvios, pois defini com arquitetura modular e utilizando as boas práticas (SOLID).

Nota: O mapeamento dos requisitos e a escolha de tecnologias maduras (MongoDB, RabbitMQ, .NET Core 8) garantiram a fluidez do desenvolvimento.

## Tecnologias Utilizadas

Linguagem: C# (.NET Core 8).

Framework: ASP.NET Core para a API REST e Worker (BackgroundService) para integração com RabbitMQ.

Banco de Dados: MongoDB (utilizando o driver oficial do MongoDB para .NET).

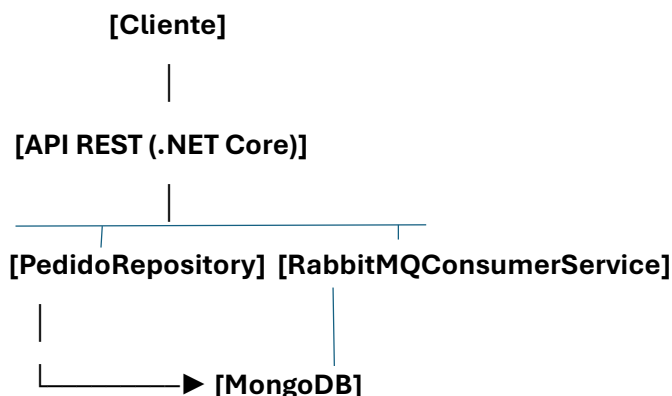
Mensageria: RabbitMQ (biblioteca RabbitMQ.Client).

IDE: Visual Studio 2022.

SO: Desenvolvimento realizado em ambiente Ubuntu.

## Diagrama da Arquitetura

A Arquitetura é composta por 3 camadas:



API Rest expoe os endpoints para consultas de pedidos.

O RabbitMQConsumerService consome mensagem da fila e grava na base de dados.

As camadas compartilham o mesmo repositório.

## **Implantação da Solução**

Servidor de Aplicação hospeda a API REST e o serviço de consumo RabbitMQ, .NET e Containers.

MondoDB armazena os documentos.

RabbitMQ fila de mensageria de pedidos.

## **Repositórios (GIT e Docker)**

GIT: [haroldoselin/btgpactual: Repo](#)

DOCKER HUB: [Docker Hub](#)