

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO DE COMPUTAÇÃO  
ENGENHARIA DE SOFTWARE

DANIELA RODRIGUES COUTO COSTA RA: 2454297  
MARCELO GREGÓRIO VIEIRA RA: 2454408

**LISTA 3 - Arquitetura Baseada em Micro Serviços**

ARQUITETURA DE SOFTWARE

CORNÉLIO PROCÓPIO

2025

## Objetivo do projeto:

Projetar uma simulação de arquitetura de microserviços para um sistema de gerenciamento de pedidos simulando a interação entre os serviços de forma local, utilizando métodos e objetos.

## Visão geral do projeto (Diagrama de Microserviços):

O projeto simula o padrão de microserviços a partir de uma API (local) que implementa os microserviços e gerencia a comunicação entre eles, sendo que cada micro serviço possui um papel distinto e é dependente dos outros microserviços.

O usuário(user) na Imagem 1 acessa diretamente a API, posto que não existe uma interface ou front-end personalizada.

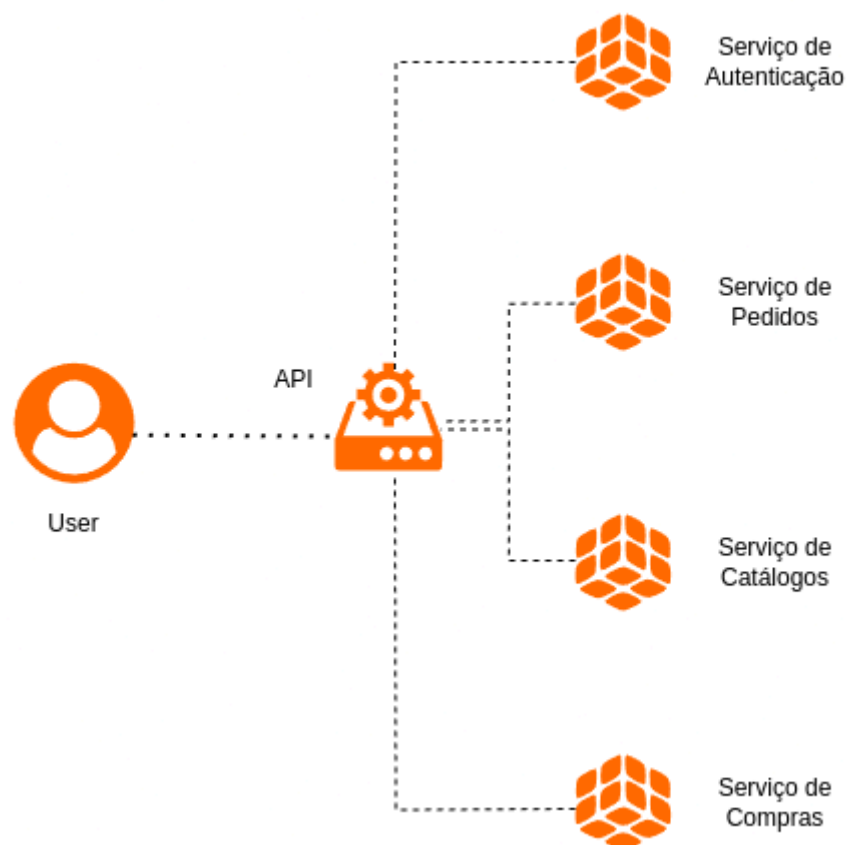


Imagem 1 :Representação da comunicação entre serviços

## Classes do sistema:

O sistema é dividido em sete classes principais como representado na imagem 2 abaixo. As classes com funcionalidade são: Pagamento, Catalogo, Pedidos e Autenticação, todas as classes de microsserviços são implementadas pelo App e realizam comunicações por meio dele.

As classes complementares são utilizadas para dar coesão e maior simplicidade, permitindo que as classes de microsserviços não sofram uma sobrecarga de funcionalidades e mantenham sua atomicidade. A classe User é responsável pela criação de novos usuários com os atributos necessários para a utilização do sistema. A classe Carrinho é diretamente ligada às classes User e Pedidos, o carrinho é responsável por armazenar os itens do catálogo que um usuário deseja adquirir, sendo que cada usuário possui um carrinho que pode conter diversos itens, além disso cada carrinho possui um Pedido atrelado, o qual armazena o usuário que realizou seu pedido, seu carrinho e o status do Pedido, que pode ser alterado caso o usuário realize ou não a compra.

A classe ReadInput é responsável por ler as entradas de dados realizadas pelo teclado, com o objetivo de simular uma interface via terminal da API.

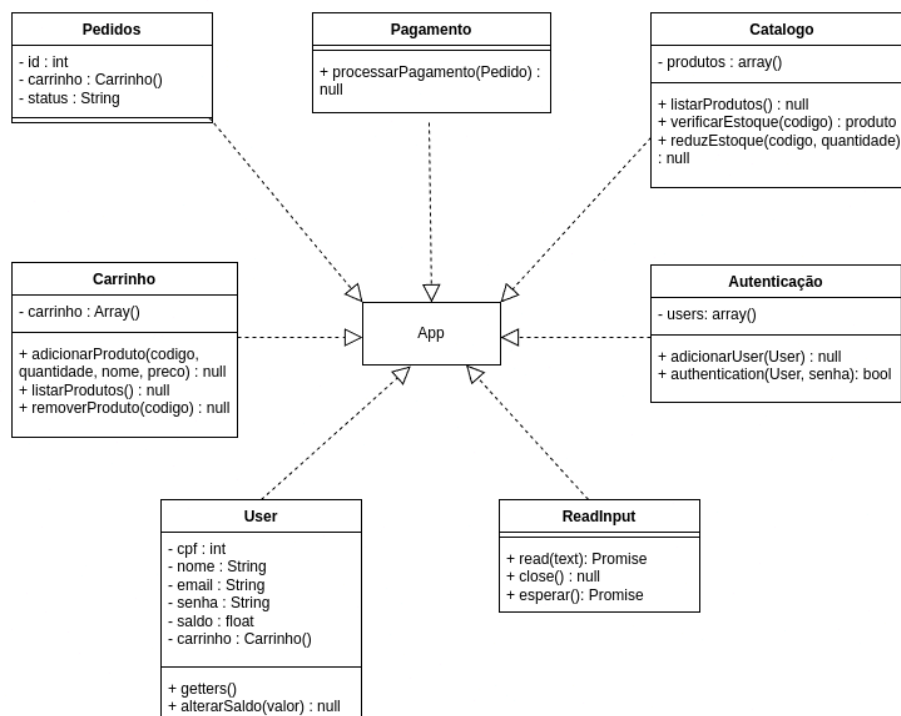


Imagem 2: Diagrama de classes do sistema