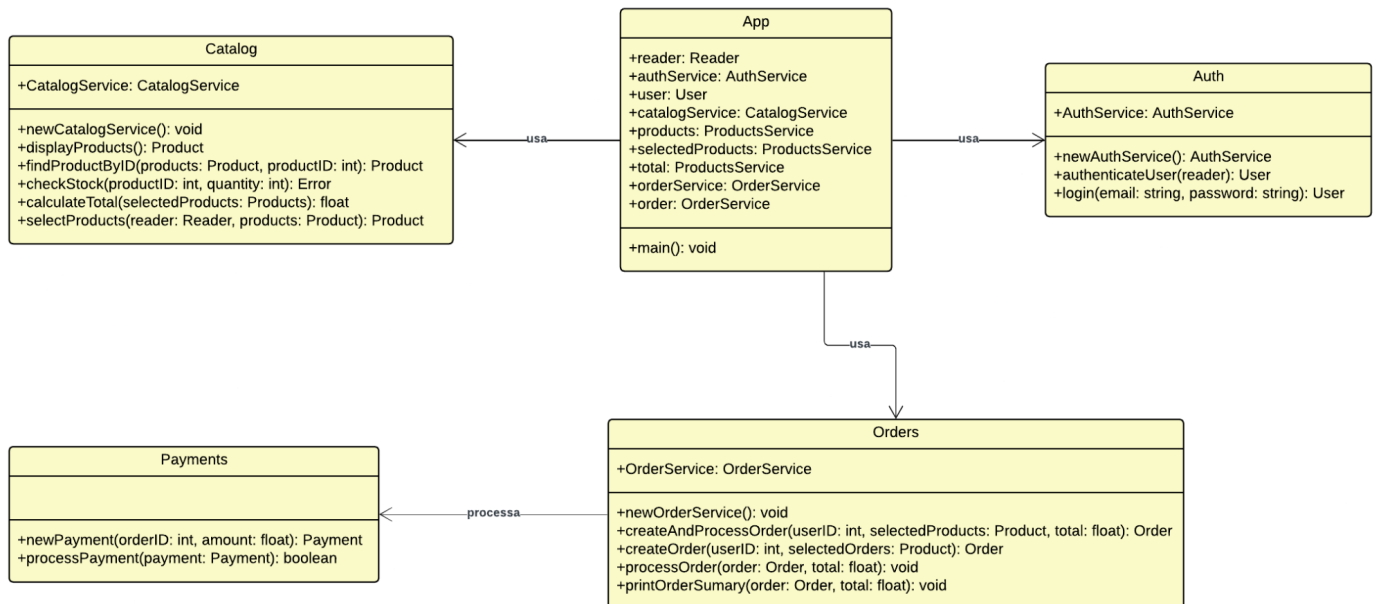


## Alunos:

- Marcos Vinicius Bueno Prestes - RA: 2465760
- Micael Ribeiro Rocha - RA: 2454424

### 1) Diagrama da aplicação:



### 2) Descrição do sistema:

O conteúdo principal do sistema baseia-se na seguinte disposição de arquivos:

- O Arquivo "app.go" representa o main da aplicação.
  - func main: Inicia e utiliza os microsserviços de autenticação, catálogo de produtos e realização de pedidos.
- A pasta models visa identificar os arquivos que definem o tipo das instâncias utilizadas na aplicação. Esta pasta contém os arquivos "order.go", "payment.go", "product.go" e "user.go".
  - order.go: Define a estrutura do tipo pedido (Order) com os atributos ID, Products e State.
  - payment.go: Define a estrutura de um tipo pagamento (Payment) com os atributos ID, Amount, e Status.
  - product.go: Define um tipo produto (Product) com os atributos ID, Name, Price, e Stock.
  - user.go: Define o tipo usuário (User) com os atributos como ID, Name, Email, e Password.

- A pasta `services` identifica os microsserviços feitos para integrar as funcionalidades da aplicação que são utilizadas em `“app.go”`. Os micro serviços desenvolvidos foram `“auth/auth.go”`, `“catalog/catalog.go”`, `“orders/orders.go”` e `“payments/payments.go”`.
  - `auth/auth.go`: Responsável pela autenticação do usuário e liberação do acesso ao serviço. Seus métodos são `NewAuthService` (cria uma nova instância de `AuthService`), `AuthenticateUser` (solicita ao usuário seu e-mail e senha para autenticação) e `Login` (simula o login com os dados pré-definidos).
  - `catalog/catalog.go`: Responsável por gerenciar o catálogo de produtos do serviço. Seus métodos são `NewCatalogService` (cria uma nova instância de `CatalogService`), `DisplayProducts` (lista os produtos disponíveis), `FindProductByID` (busca um produto presente no catálogo pelo ID), `CheckStock` (verifica a disponibilidade de um produto no estoque), `CalculateTotal` (realiza o cálculo do valor total dos produtos selecionados) e `SelectProducts` (permite a seleção de produtos e cálculo do total da compra).
  - `orders/orders.go`: Responsável por gerenciar os pedidos de compra. Seus métodos são `NewOrderService` (cria uma nova instância de `OderService`), `CreateAndProcessOrder` (cria e processa um pedido), `CreateOrder` (gera um pedido com os produtos selecionados), `ProcessOrder` (processa o pedido chamando o microsserviço de pagamentos) e `PrintOrderSummary` (exibe o resumo do pedido).
  - `payments/payments.go`: Responsável por simular o processo de pagamento dos pedidos. Seus métodos são `NewPayment` (cria uma nova instância de `Payment`) e `ProcessPayment` (simula o processamento de pagamento com três tentativas, utilizando números aleatórios).