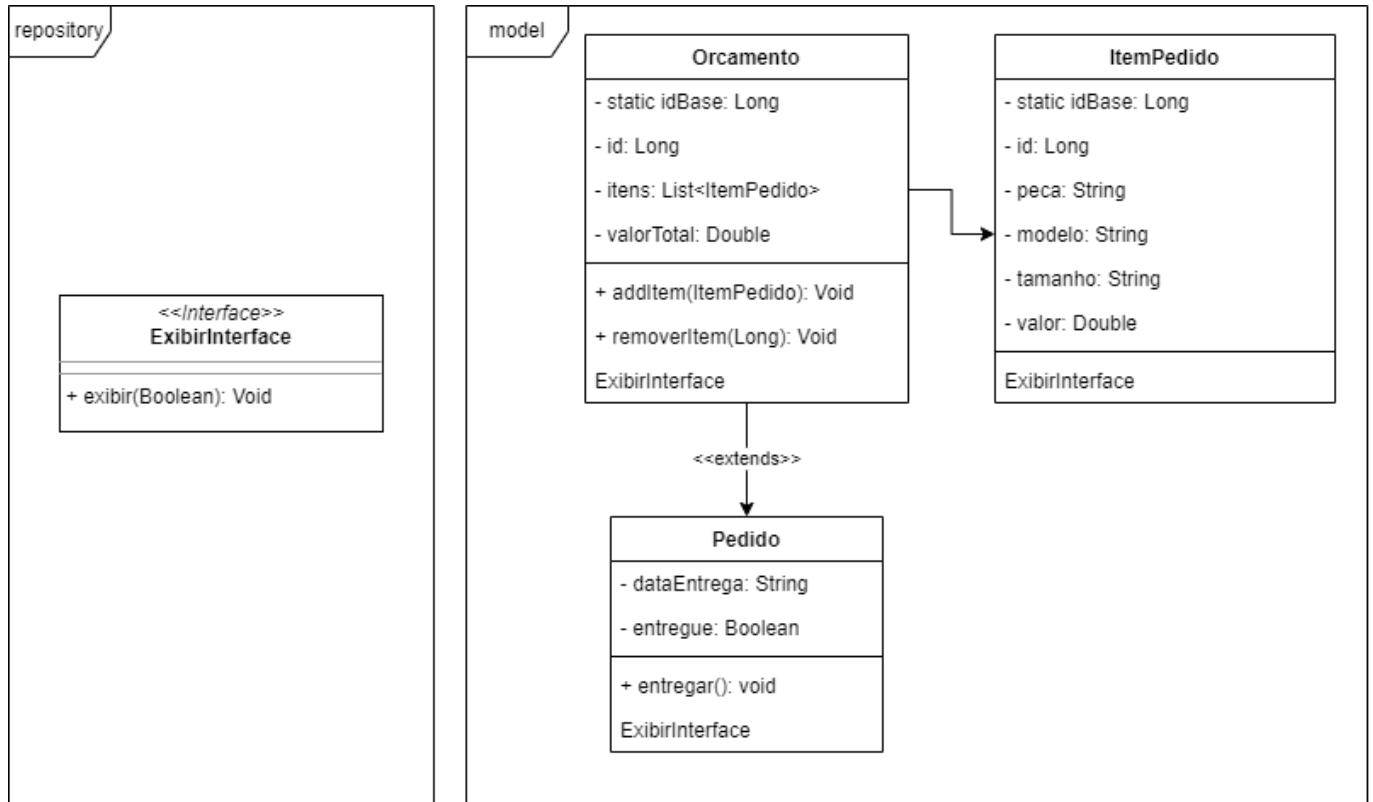


## Prova Prática

**Sistema de pedidos:** Vocês deverão implementar um sistema de pedidos e orçamentos simplificado. Nesse sistema o usuário poderá gerar orçamentos ou pedidos onde cada um deles pode conter diversos itens. O diagrama a seguir descreve como o sistema deve ser implementado.



Explicação das classes do sistema:

- A classe **Orcamento** descreve um orçamento de um pedido. O construtor dessa classe deve ser vazio e todos os valores devem ser alterados automaticamente ou através de métodos.
  - Essa classe só deve possuir getters e nenhum setter.
  - O id deve ser adicionado automaticamente ao criar um orçamento, usando o `idBase`.
- A classe **Pedido** descreve um pedido e herda de **Orcamento** por isso tudo para **Orcamento** é válido para **Pedido**.
  - Somente a `dataEntrega` pode ser alterada por um setter.
  - O construtor do pedido só recebe a `dataEntrega` e inicia o `entregue` com `false` automaticamente.
- A classe **ItemPedido** representa um item do pedido.
  - O construtor dessa classe deve ter todos os atributos menos o `id`.
  - O `id` deve ser adicionado automaticamente ao criar um item, usando o `idBase`.
  - Deve ter todos os getters e setter, exceto o setter para o `id`.

Explicação das funcionalidades do sistema:

- A classe Orcamento deve poder adicionar itens, ou remover itens, durante a adição ou remoção de um item o valor total do pedido deve ser alterado somando ou subtraindo o valor do item.
- A remoção do item deve ser feita pelo id do item.
- A classe Pedido deve permitir que o pedido seja alterado para entregue tornando o atributo entregue para verdadeiro.
- A interface ExibirInterface tem o método exibir que deve exibir os dados de cada classe na tela.
- O método exibir recebe um booleano que determina se a exibição será completa ou reduzida (true para completa e false para reduzida).

#### Exceções:

- Se o pedido já estiver entregue o método entregar deve gerar uma exceção do tipo Exception com a mensagem: "O pedido já foi entregue!".
- Se o id não for encontrado na remoção de item, deve seguir uma exceção do tipo RuntimeException com a mensagem: "O item não foi encontrado!".
- Não esqueça de tratar a exceção e exibir a mensagem na tela.

#### Conteúdo de main:

- Criar um objeto do ItemPedido.
- Exibir ItemPedido completo.
- Exibir ItemPedido reduzido.
- Criar Orçamento.
- Adicionar o item criado no orçamento.
- Exibir orçamento completo.
- Exibir orçamento reduzido.
- Adicionar outro item ao orçamento.
- Exibir orçamento completo.
- Remover o último item que foi adicionado.
- Exibir orçamento completo.
- Tentar remover um item que não existe no orçamento.
- Exibir orçamento completo.
- Criar pedido
- Adicionar o item criado no pedido.
- Exibir pedido completo.
- Exibir pedido reduzido.
- Entregar pedido.
- Exibir pedido reduzido.
- Tentar entregar o mesmo pedido novamente.
- Exibir pedido reduzido.

Exemplo de exibição:

Item completo

```
Item do pedido: 0
- Peça: Bermuda
- Modelo: Jogador
- Tamanho: G
- Valor: R$30.0
```

Item reduzido

```
- Bermuda, Jogador, G, R$30.0
```

Orçamento completo

```
Orçamento: 0
Itens:
- Bermuda, Jogador, G, R$30.0
- Calça, Jogguer, M, R$50.0
Valor Total: R$80.0
```

Orçamento reduzido

```
- 0, R$30.0
```

Pedido completo

```
Pedido: 1
Itens:
- Bermuda, Jogador, G, R$30.0
- Calça, Jogguer, G, R$60.0
Data de Entrega: R$15/06/2023
Situação: Não entregue
Valor Total: R$90.0
```

Pedido reduzido

```
- 1, 15/06/2023, Entegue, R$90.0
```