

Disciplina: Métodos Avançados de Programação

Professora: Sabrina

Grupo:

Anna Caroline, Mat: 202080226

Daniel Ribeiro, Mat: 212080431

Henrique José, Mat: 202080080

João Pedro, Mat: 192080199

João Vitor, Mat: 211080080

Luana Nóbrega, Mat: 201080478

Relatório: Marketplace

Relatório Marketplace

MAP - Métodos Avançados de Programação

Grupo: Anna Caroline, Daniel Ribeiro, Henrique José, João Vitor, João Pedro e Luana Nóbrega

Índice

Índice.....	2
Relatório de entregas.....	3
1. Release 01 (04/06/23).....	3
1.1. Diagrama de classe.....	3
1.2. Lista de padrões utilizados.....	3
2. Release 02 (18/06/23).....	5
2.1. Diagrama de Classe.....	5
2.2. Lista de padrões utilizados.....	5
3. Release 03 (29/06/23).....	7

Relatório de entregas

Esse relatório contém informações de acordo com o que foi desenvolvido em cada release, juntamente com observações e alterações, conforme necessário.

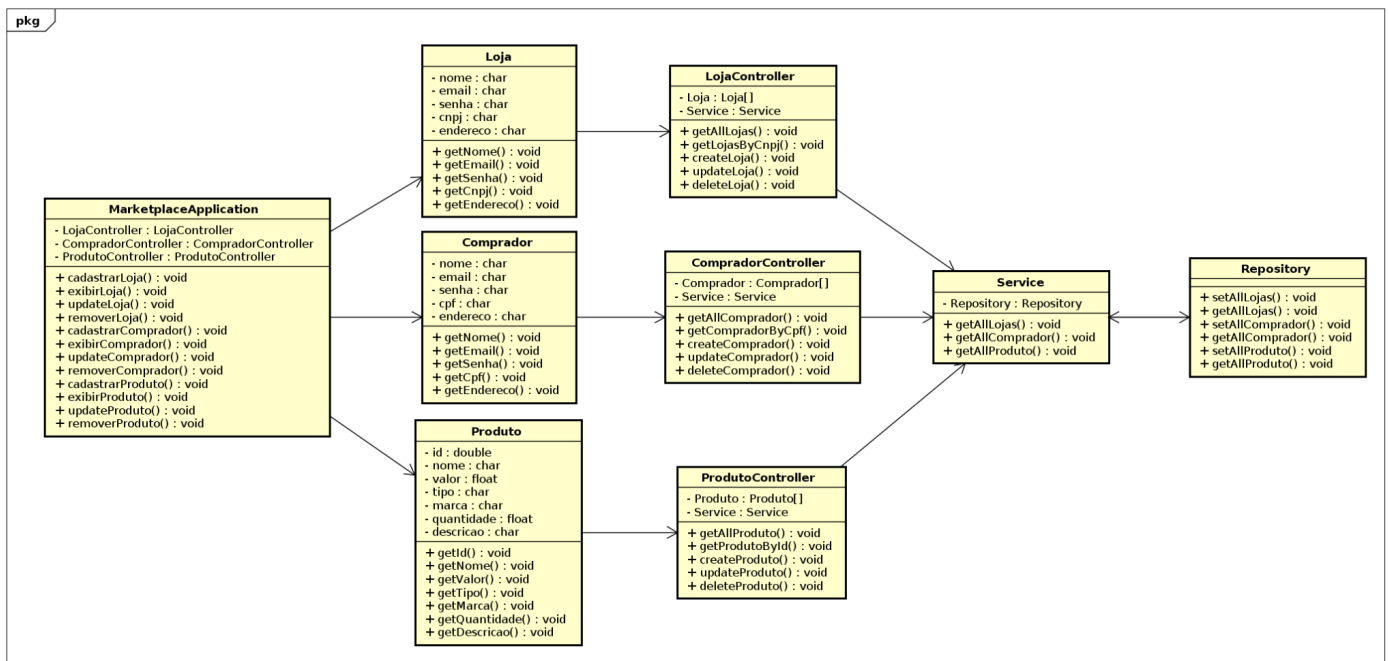
1. Release 01 (04/06/23)

Repositório: <https://github.com/jv-guimaraes/marketplace/releases/tag/v1.0.0>

Vídeo de apresentação:

https://drive.google.com/file/d/1-I_X76Rr5ni1xY_GC5L4oSQkHi7w5QAr/view?usp=sharing

1.1. Diagrama de classe



1.2. Lista de padrões utilizados

Padrões:

- Singleton
- Fachada
- Visitor
- Repository

O padrão de projeto singleton é aplicado na classe **MarketplaceApplication** do sistema de marketplace. Essa classe permite que tenha uma acesso global para as outras instância, sendo esta o ponto de acesso conhecido entre elas. O **MarketplaceApplication** é

Relatório Marketplace

MAP - Métodos Avançados de Programação

Grupo: Anna Caroline, Daniel Ribeiro, Henrique José, João Vitor, João Pedro e Luana Nóbrega

uma abordagem que foi aplicada no sistema de marketplace para separar a lógica de acesso aos dados das operações de CRUD (Create, Read, Update, Delete).

Os métodos da classe MarketplaceApplication são responsáveis por chamar os métodos correspondentes nas classes com padrões de fachada, no projeto foram identificados com o sufixo “controller”, sendo estes: LojaController, CompradorController e ProdutoController. Essas fachadas são responsáveis por lidar com a manipulação dos objetos Loja, Comprador e Produto, respectivamente, incluindo a validação, busca, atualização e exclusão dos dados.

Por exemplo, o método cadastrarLoja cria um objeto Loja com os dados fornecidos e chama o método createLoja do LojaController para persistir a loja no sistema. Esse método é responsável por cadastrar uma nova loja no marketplace. O cliente não precisa conhecer os detalhes da implementação complexa envolvida na criação e persistência da loja, pois a fachada fornece uma interface simples e direta para realizar essa operação.

A classe de serviço que é o Service, se utiliza do padrão visitor, o qual permite a separação do algoritmo do objeto que são manipulados, este é responsável pela validação dos dados, busca no banco de dados, atualização e exclusão dos registros correspondentes.

Outro padrão de projeto utilizado é o Repository e tem como objetivo fornecer uma camada de abstração entre o código de negócio e a forma como os dados são persistidos, permitindo uma maior flexibilidade e facilidade na manipulação dos dados.

Para aplicar o padrão Repository, foi criada uma entidade do sistema, Repository. Essas interfaces definem os métodos de acesso aos dados para cada entidade, como busca, criação, atualização e exclusão. As interfaces de repositório descrevem as operações que podem ser realizadas nas entidades, sem especificar a forma como essas operações serão implementadas. Isso permite uma maior separação de responsabilidades, pois a lógica de negócio não precisa se preocupar com os detalhes de como os dados são persistidos.

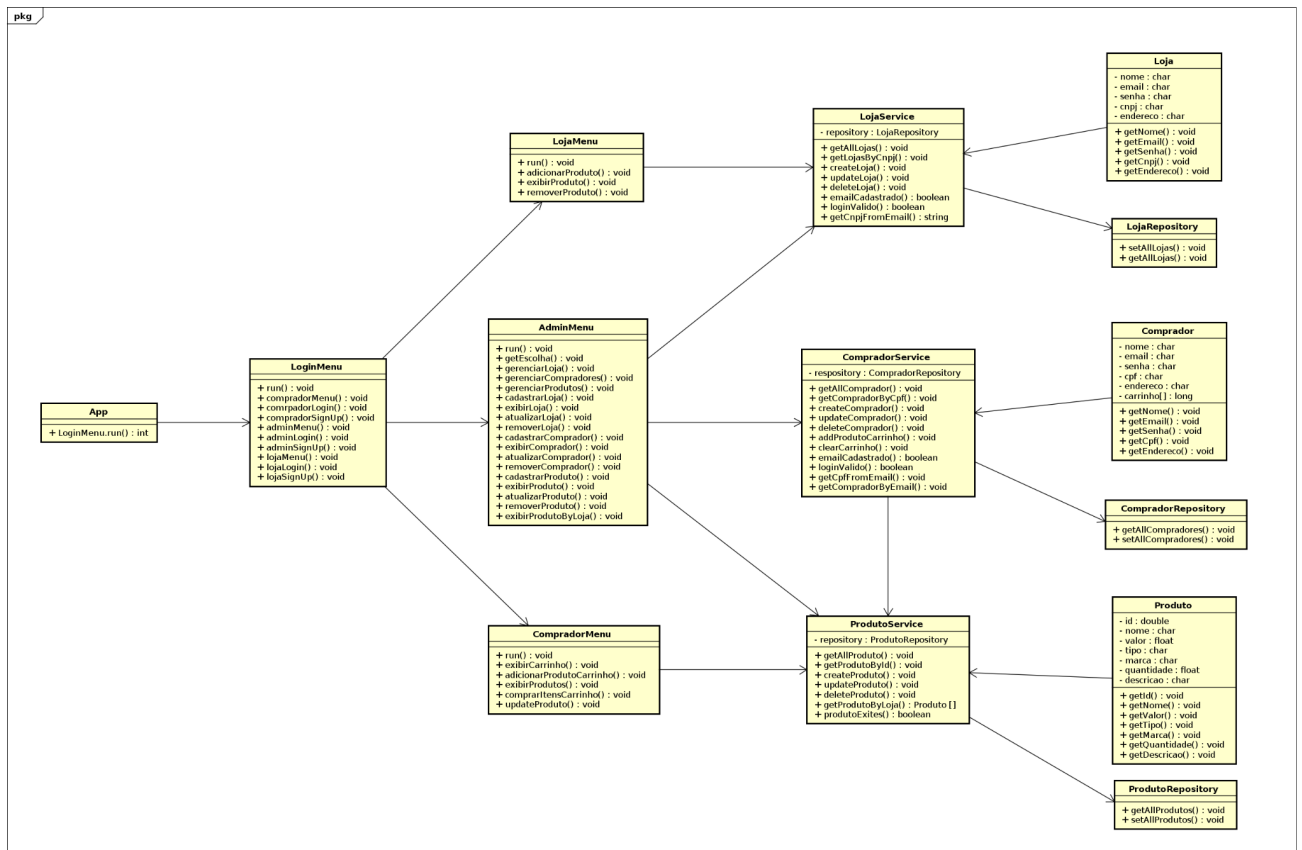
Essa separação da lógica de acesso aos dados em interfaces de repositório e classes concretas traz benefícios significativos para o sistema, por exemplo permitindo que a camada de acesso aos dados seja modificada sem afetar o restante do código.

2. Release 02 (18/06/23)

Repositório: <https://github.com/jv-guimaraes/marketplace/releases/tag/v1.0.0>

Vídeo de apresentação:

2.1. Diagrama de Classe



2.2. Lista de padrões utilizados

- Singleton
- Fachada
- Visitor
- Repository

Na classe LoginMenu, foi utilizado o padrão fachada, o qual será utilizado para direcionar o tipo de login realizado, se foi feito pelo Comprador, Loja ou Admin.

O padrão singleton, por sua vez, encontra-se, agora, no AdminMenu, onde serão realizadas chamadas de gerenciamento e manutenção de Loja, Comprador e Produto, esse está ligada aos três essenciais services.

Os services se mantêm como padrões visitors, pois permite a separação do algoritmo do objeto que são manipulados, este é responsável pela validação dos dados de login, busca no banco de dados, atualização e exclusão dos registros correspondentes.

Relatório Marketplace

MAP - Métodos Avançados de Programação

Grupo: Anna Caroline, Daniel Ribeiro, Henrique José, João Vitor, João Pedro e Luana Nóbrega

O padrão repository utilizado, ainda se mantém com a função de banco, agora com os novos requisitos, está sendo utilizado como “carrinho de compras” onde é feito o armazenamento dos produtos selecionados e efetuada a compra.

Relatório Marketplace

MAP - Métodos Avançados de Programação

Grupo: Anna Caroline, Daniel Ribeiro, Henrique José, João Vítor, João Pedro e Luana Nóbrega

3. Release 03 (29/06/23)