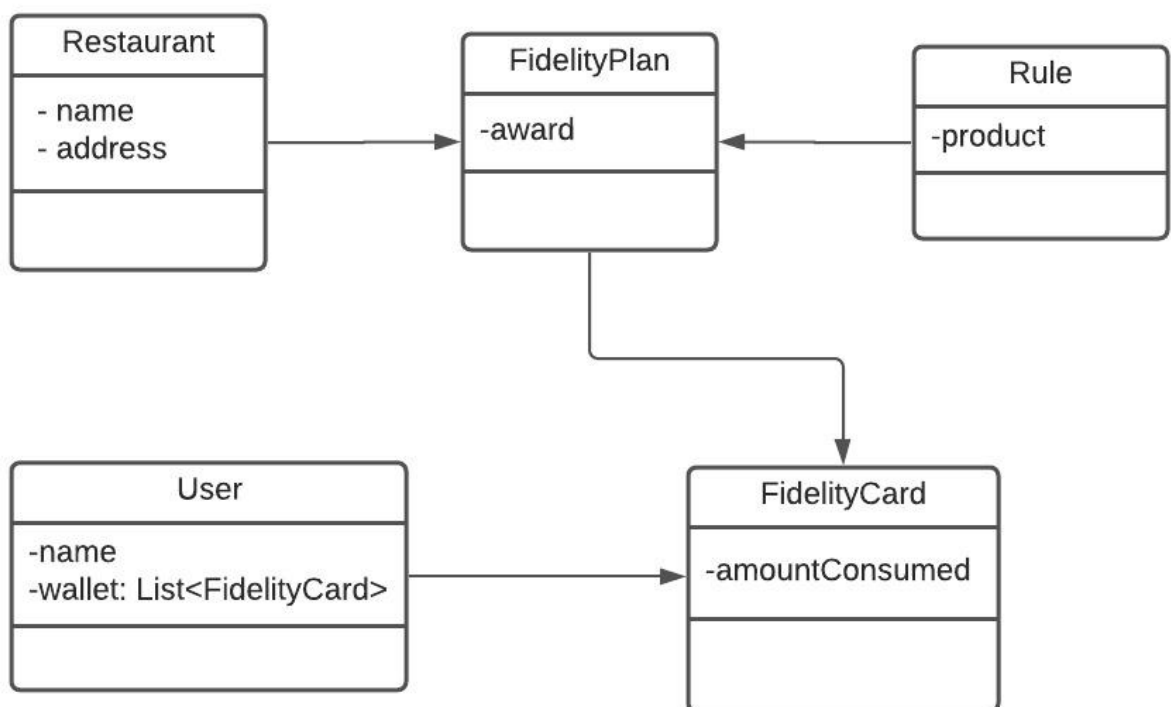


1. Visão arquitetural

As visões arquiteturais são os possíveis modos como as pessoas que desempenham papéis diferentes dentro do processo de desenvolvimento de software vêem o problema, tendo diversas perspectivas a fim de melhor identificar o contexto a ser desenvolvido. Ademais, cada visão é registrada utilizando um determinado modelo, como o desenvolvimento, lógico, físico, processo entre outros. No Fidelicia será abordado o modelo de visão arquitetural lógico.

1.1 Visão lógica

Uma visão lógica mostra as abstrações fundamentais do sistema como objetos ou classes. Nesse tipo de visão, deve ser possível relacionar os requisitos do sistema com as suas entidades.



- Restaurante (Restaurant)

Responsável por cadastrar e manter os dados de um restaurante que utilizará do sistema para o gerenciamento de seus cartões de fidelidade. Um

restaurante pode definir vários planos de fidelidades e suas regras desde que sejam diferentes.

- **Plano de fidelidade (FidelityPlan)**

É responsável por armazenar as informações referente a relação entre Restaurante e Cartão de Fidelidade, gerindo as regras que permitem ao usuário obter os benefícios do plano de fidelidade. Essa classe possui a Regra e o prêmio que o usuário pode receber ao cumprir os requisitos estabelecidos na regra. Um restaurante pode estipular mais de um plano de fidelidade com regras diferentes.

- **Regra (Rule)**

Responsável por definir quais os requisitos que o plano de fidelidade irá aplicar para que um usuário seja beneficiado com a promoção.

- **Cartão de fidelidade (FidelityCard)**

Este objeto possui a função de registrar a evolução do usuário no cumprimento das regras de um plano de fidelidade. Ao ser completado todos os requisitos da regra, o cartão torna-se elegível para solicitação do prêmio (award) vinculado ao plano de fidelidade.

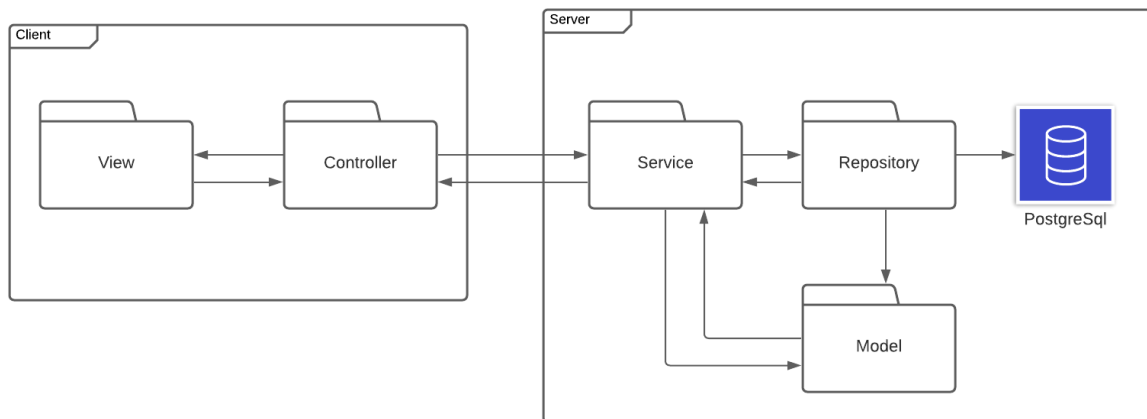
- **Usuário (User)**

O usuário é responsável por realizar um cadastro na aplicação e manter dados de seus “cartões” de fidelidade que estão cadastrados, tendo como objetivo juntar selos para receber promoções ou benefícios. O usuário possui uma lista de cartões de fidelidade aqui chamada de carteira (wallet). Ele pode possuir cartões fidelidade de vários restaurantes ou vários de um mesmo restaurante desde que possuam planos de fidelidade diferentes.

2.Design Arquitetural

O design arquitetural é responsável por tratar as definições de organização, distribuição e comunicação dos componentes que integram o estilo arquitetural.

No contexto do MVC o design arquitetural tem como responsabilidade definir o comportamento dos componentes Model, View, Controller.



A **View** é o componente responsável por interagir com o usuário na parte do cliente e contém os formulários de interação, onde o usuário consegue enviar dados para a aplicação.

O **Controllers** é responsável por receber os dados e informações da view e enviar para o componente de Model devidamente tratado.

O **Service** é a classe que processa as requisições que o controller precisa. Ele quem conhece a lógica de qual model que atenderá o esperado e é ela que instancia o repositório.

O **Repository** é quem lida com as chamadas ao banco de dados. É o repositório que possui os métodos para criação, leitura, atualização e exclusão de dados no banco de dados e as demais variações que podem ser necessárias ao projeto utilizando como base os dados tratados pelo model.

Por fim, a **Model** é a parte que contém as regras de negócio da aplicação e consegue definir como esses dados devem ser armazenados.