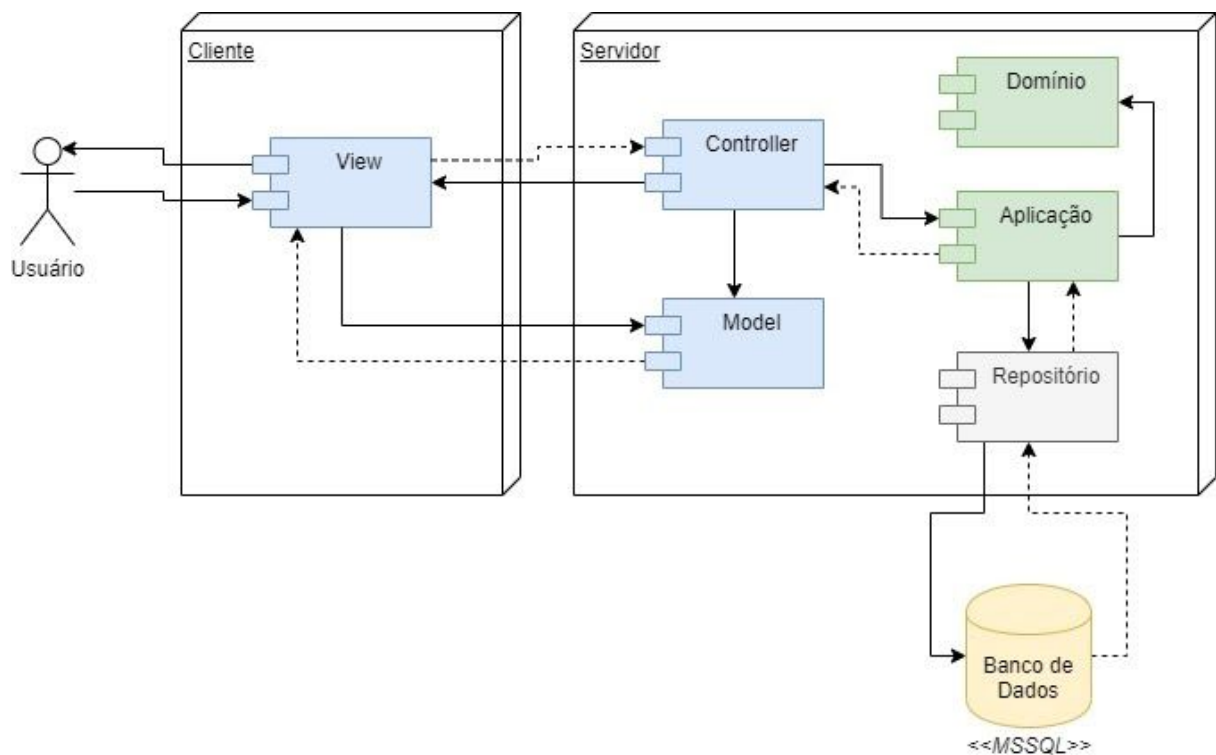


Estilo Arquitetural

1 - Introdução

Este documento irá abordar o estilo arquitetural escolhido, para o desenvolvimento do projeto da disciplina de Desenvolvimento de Software para Persistência, bem como seus componentes, e porque a equipe decidiu que seria o melhor estilo para a elaboração desse sistema.

2 - Modelo



3 - Componentes

3.1 - Componentes do Cliente

Componente	View
Descrição	A View é o componente responsável pela interação entre o sistema e o usuário. Neste modelo, considera-se a View como "passiva", ou seja, não contém lógica de negócio. A View apenas implementa a interface, que contém os elementos necessários

	para que o Model seja capaz de inscrever-se para receber notificações acerca das interações do usuário; e atualizar os dados que são exibidos na View.
Responsabilidades	Exibir uma interface que permita exibir informação e receber entradas do usuário. Implementar a interface, permitindo: <ul style="list-style-type: none"> • Atualizar os dados que são exibidos para o usuário. • Inscrever-se para receber notificações sobre as interações do usuário.
Tecnologias	Html, Css (Bootstrap) e Javascript(Jquery)

3.2 - Componentes do Servidor

Componente	Model
Descrição	Contém as propriedades que a View vai obter ou definir, e funções para cada operação que pode ser feita pelo usuário em cada view. Encapsulamento da parte lógica e dos dados referente à interface com usuário.
Responsabilidades	<ul style="list-style-type: none"> • Disponibilizar para a View uma lógica de apresentação. • Implementa propriedades e comandos, para que a View possa preencher seus controles e notifica a mesma, caso haja alteração de estado. • Pode implementar a lógica de validação, para garantir a consistência dos dados da View. • encapsula a lógica e os dados referentes à interface com usuário. • Realiza requisições para api de acordo com as solicitações feitas pelo usuário. • Recebe a resposta da api e envia para o ViewModel. • Herda os atributos que estão na camada de Domínio de acordo com sua responsabilidade. • Pode realizar comunicação com a camada de Aplicação.
Tecnologias	C#

Componente	Controller
Descrição	O responsável por receber todas as requisições do usuário. Seus métodos chamados actions são responsáveis por uma página, controlando qual Model usar e qual View será mostrado ao usuário.
Responsabilidades	<ul style="list-style-type: none"> • Receber requisições dos usuários. • Controlador de Views.

	<ul style="list-style-type: none"> • Pode realizar comunicação com a camada de Aplicação. • Atualiza o estado das Views.
Tecnologias	C#

Componente	Domínio
Descrição	É a réplica da entidade que está no Banco de Dados de acordo com sua responsabilidade. Deve possuir os mesmos atributos
Responsabilidades	<ul style="list-style-type: none"> • Encapsula atributos que estão nas entidades do Banco de Dados. • Usado para transformar os dados que estão no Model em dados que a camada de aplicação irá entender para realizar as devidas operações
Tecnologias	C#

Componente	Aplicação
Descrição	Realiza todas as chamadas de operações a serem realizadas no banco de dados: Inserção, Alteração, Exclusão e Consulta, fazendo uma ponte entre o Controller e o Repositório
Responsabilidades	<ul style="list-style-type: none"> • Comunicação entre Controller e Repositório • Formata os dados de uma maneira que o Repositório irá entender e processar para enviar para o banco de dados. • Envia a solicitação dos comandos de Inserção, Alteração, Exclusão e Consulta ao Repositório
Tecnologias	C#

Componente	Repositório
Descrição	Interface de comunicação da camada de aplicação com o Banco de Dados
Responsabilidades	<ul style="list-style-type: none"> • Estabelecer comunicação entre a Aplicação e o Data base • Implementar todos os mecanismos necessários para enviar e receber dados do banco.
Tecnologias	C#

Data Base (Banco de dados)
Cuida do armazenamento dos dados do sistema. O banco utilizado para este fim deverá ser o Microsoft SQL Server