



Universidade Federal do Ceará Campus Crateús

Ciência da Computação

Projeto Integrador IV

Modelo Arquitetural do Jogo Show do Milhão

ADALBERTO FELIPE PINHEIRO CHAVES	471718
DIEFESSON DE SOUSA SILVA	471842
LUIS FELIPE DOMINGOS VIEIRA TORRES	473253
NATANAEL VIEIRA BARBOZA	391028
VICTOR GABRIEL MARTINS DE O. LOIOLA	471718

Introdução	3
Arquitetura do Show do Milhão	4
Modelo Arquitetural	4
Definição das Componentes	4
Conclusões	5

Introdução

Nossa arquitetura utiliza dois padrões de arquitetura: a Arquitetura em Camadas, e a Arquitetura Cliente-Servidor. Na arquitetura em Camadas usamos o padrão MVC (Model-View-Controller) para organizar o sistema, de forma a manter a segurança nas operações e acesso a dados, e a manter a coesão das componentes, a fim de facilitar a manutenção e evolução futura do código. Já na Arquitetura Cliente-Servidor, todos os tipos de usuários se comunicaram com o servidor do Show do Milhão para ter acesso às funcionalidades e aos dados do aplicativo. Dentro do servidor, haverá as classes do Controller, do Model e do DAO, além do Banco de Dados do aplicativo, enquanto a parte do View estará nos dispositivos e nos computadores dos usuários.

Arquitetura em camadas é um dos padrões arquiteturais mais usados, desde que os primeiros sistemas de software de maior porte foram construídos nas décadas de 60 e 70. Em sistemas que seguem esse padrão, as classes são organizadas em módulos de maior tamanho, chamados de camadas. As camadas são dispostas de forma hierárquica, como em um bolo. Assim, uma camada somente pode usar serviços — isto é, chamar métodos, instanciar objetos, estender classes, declarar parâmetros, lançar exceções, etc. — da camada imediatamente inferior.

Arquitetura do Show do Milhão

Modelo Arquitetural

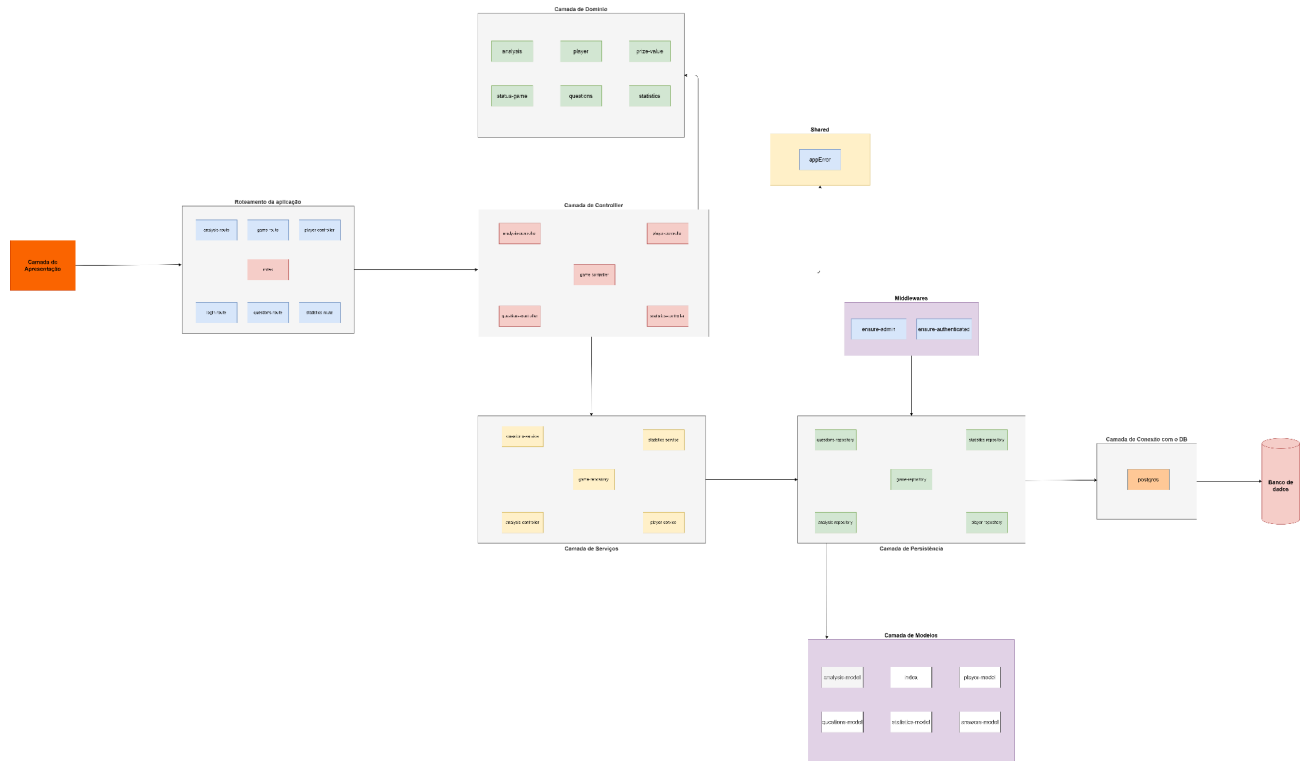


Figura 1 - Modelo Arquitetural do Aplicativo Show Do Milhão
(https://app.diagrams.net/#G1xxTuB2-9DR_hK7LIQJSGewuezRd6DwPQ)

Definição das Componentes

- Front-end(View)
- Controller: Ele é utilizado para lidar com a ligação da apresentação(front-end) com a camada de serviços.
- Service: A ideia dos Services é separar as regras de negócio, regras da aplicação e regras de apresentação para que as mesmas possam ser facilmente testadas e reutilizadas em outras partes do seu programa.
- Models: São os modelos para aplicação em termos de entidades.
- Repository: Repositories são responsáveis pela comunicação com os dados que os Services precisam.
- Routes: Camada que cuida o direcionamento seguro, correto e coeso das requisições feitas a aplicação. Podemos citar o uso de Routes quando navegamos entre telas conforme o alguns eventos, como clique em botão.

Conclusões

Não podemos encerrar sem demonstrar as vantagens da arquitetura em camada, que vão desde o particionamento da complexidade do desenvolvimento até a manutenção.

A maior vantagem é poder trocar certos componentes, tornando as partes da aplicação independentes. Poder fazer alterações em uma parte sem afetar as demais.

Também é útil para que vários desenvolvedores (não necessariamente programadores) possam trabalhar na mesma aplicação, de forma que cada um possa trabalhar nas áreas de sua especialidade. Tornando mais prático o processo de testes.