

IMPLEMENTAÇÃO DE UM SERVIDOR GRPC EM UMA APLICAÇÃO JAVA PARA CADASTRO DE PARTIDAS DE FUTEBOL

Felipe Moreira da Paz¹, Eduardo Siqueira Souto², Rafael Savignon Rezende³

¹Campus Cachoeiro de Itapemirim – Instituto Federal do Espírito Santo (Ifes)
Caixa Postal 727 – 29311970 – Cachoeiro de Itapemirim – ES – Brazil

felipemp2012@hotmail.com¹, eduardosouto2014@gmail.com², rafael.savignon68@gmail.com³

Resumo. *Com base no sistema desenvolvido durante a realização da Disciplina de Programação Orientada a Objetos 2, com o intuito de criar um sistema para o gerenciamento de partidas de futebol, foi proposta pelo professor João Paulo de Brito Gonçalves, na disciplina de Sistemas Distribuídos, a criação de um servidor gRPC com aplicabilidade e que tivesse comunicação com o cliente.*

Abstract. *This paper explores the development of a system during the Programming Object-Oriented 2 course, aimed at creating a solution for managing football matches. The proposal, presented by Professor João Paulo de Brito Gonçalves in the Systems Distributed course, suggests the creation of a gRPC server with practical applicability and seamless communication with the client.*

1. Introdução

- Primeira Versão do Sistema de Cadastro de Partidas:

O presente projeto, um Sistema de Cadastro de Partidas, foi concebido como resultado do esforço empreendido durante a realização da Disciplina de Programação Orientada a Objetos 2. O seu propósito principal é proporcionar um eficiente gerenciamento de partidas de futebol, abrangendo tanto os processos prévios ao jogo quanto aqueles que ocorrem durante a sua realização.

O sistema é estruturado em duas fases distintas: o cadastro prévio, que contempla o registro de informações relevantes antes do início da partida, e os procedimentos inerentes ao decorrer do jogo. Como parte essencial do projeto, foram estabelecidas regras de negócio, como a limitação de dois cartões amarelos e um cartão vermelho por jogador, e a necessidade de que ambos os times realizem suas operações no mesmo horário.

No âmbito tecnológico, o projeto adota a linguagem Java, empregando a biblioteca JavaFX para a construção das interfaces do usuário, e a biblioteca JFoenix para aprimorar a experiência visual. A arquitetura do software é fundamentada nos padrões MVC (Model-View-Controller) e DAO (Data Access Object), proporcionando uma organização eficiente e modular.

Quanto ao armazenamento de dados, o projeto integra o PostgreSQL como sistema gerenciador de banco de dados, assegurando uma base robusta e confiável para o gerenciamento das informações. Além disso, para a geração de relatórios, adotou-se a tecnologia Jasper Reports, contribuindo para a produção de documentação clara e objetiva sobre as atividades do sistema.

Este projeto representa não apenas a aplicação prática dos conhecimentos adquiridos na disciplina, mas também uma solução funcional e inovadora para o gerenciamento eficaz de partidas de futebol, alinhando-se às exigências e desafios do ambiente esportivo contemporâneo.

- Implementação do Servidor gRPC no Sistema de Cadastro de Partidas:

Enraizado na estrutura sólida do sistema desenvolvido na Disciplina de Programação Orientada a Objetos II, uma nova fase de inovação foi empreendida por meio da integração de um servidor gRPC (gRPC Remote Procedure Call). Este avanço tecnológico visa otimizar a comunicação entre o servidor e o cliente, elevando a eficiência e a escalabilidade do sistema de gerenciamento de partidas de futebol.

A implementação do servidor gRPC representa uma extensão natural do projeto original, transcendendo, quando necessário, as limitações linguísticas associadas ao Java. Esta adaptação é realizada para alinhar o sistema aos princípios e protocolos avançados do gRPC, proporcionando uma comunicação mais eficaz e moderna em comparação com a abordagem anterior baseada em sockets.

No âmbito funcional, o servidor gRPC não se restringe apenas à transmissão de informações sobre os membros do grupo. Além dessa funcionalidade central, ele incorpora aprimoramentos substanciais, oferecendo serviços adicionais que enriquecem a interação entre o cliente e o sistema de cadastro de partidas. Esta expansão funcional não apenas atende às exigências iniciais do projeto, mas também introduz dinâmicas mais

sofisticadas, como a capacidade de transmitir e processar dados sobre os times vencedores de partidas.

Particularmente, essa implementação permite a exibição dos dados dos times vencedores em uma tabela específica. Através de uma Table Column, os resultados das partidas são apresentados de maneira clara e organizada, com a especificação do time vencedor com base nos gols marcados tanto pelo time da casa quanto pelo time visitante. Essa funcionalidade proporciona uma visão mais detalhada e acessível do desempenho das equipes, contribuindo para uma experiência de usuário mais enriquecedora.

2. Metodologia

2.1. Primeira Versão:

- Primeira Versão do Sistema de Cadastro de Partidas:

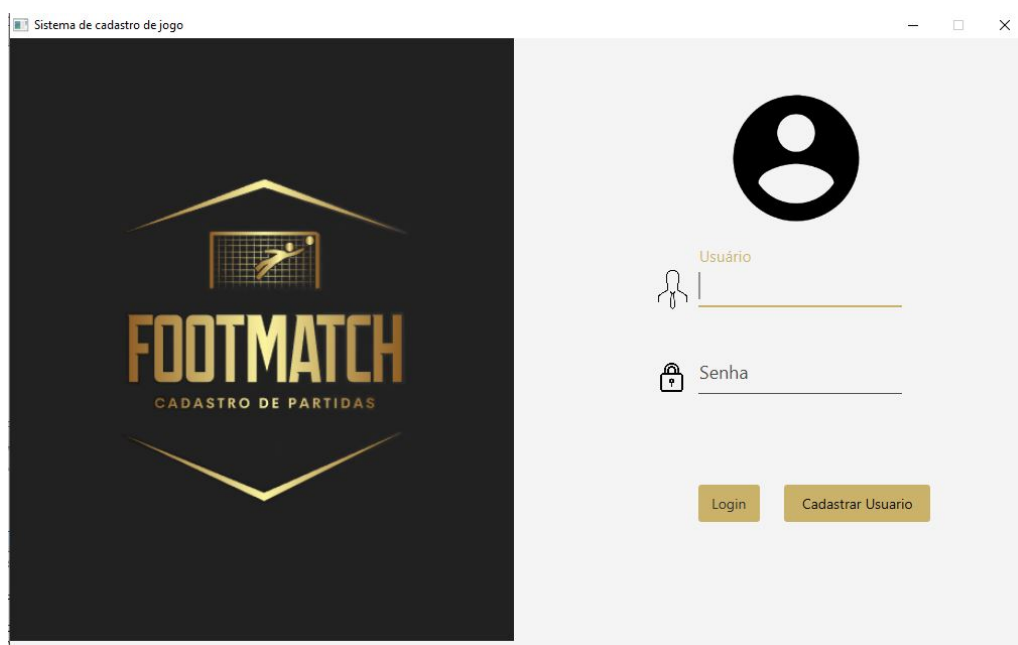


Figure 1. Tela Inicial

Tela de Cadastro para login no sistema, com criação do usuário no banco de dados.

Tela Cadastro Sistema

Footmatch
CADASTRO DE PARTIDAS

Cadastro Usuario

Email

Nome Completo

Nome de Usuario

Senha

Cadastrar

Figure 2. Tela Cadastro

Tela de Cadastro para login no sistema, com criação do usuário no banco de dados.



Figure 3. Tela Main

Tela de Menu do Sistema com cadastros, Processos e Graficos.

Nome	CPF
Sandro Meira Ricci	12345678273
Wilton Pereira Sampaio	13452333433
Raphael Claus	14569002342
jao	232332
rafael	12334
felipe	1233
JaumB	123

Detalhes do Arbitro

Codigo: 5

Nome: jao

CPF: 232332

Data de Nascimento: 2023-06-23

Nacionalidade: brasil

Formação: argelia

Inserir Alterar Remover

Figure 4. Cadastro Arbitro
Tela de Cadastro de Arbitro da partida.

Nome	ID

Detalhes

ID Time 3

Jogadores ▼

Pais Espanha

Liga La Liga

Tecnico Carlo Ancelotti

Ano Fundação 1902-03-06

Inserir Alterar Remover

Figure 5. Cadastro Equipe
Tela de Cadastro de Equipes das partida.

FOOTMATCH
CADASTRO DE PARTIDAS

Cartões

Inserir Alterar/Remover Listar

Inserir Cartão

Arbitro: Sandro Meira Ricci

Nome Jogador: Lionel Messi

Jogo: Barcelona x Real Madrid

Cor Cartão: amarelo

Quantidade: 1

Data do cartão:

Confirmar

Figure 8. Processo Cartão

Tela de Processo Cartão, aonde é registrado os cartões que estão acontecendo em jogo.



Figure 9. Grafico cartões

Grafico cartões, grafico que informa cartões aplicados em partida.

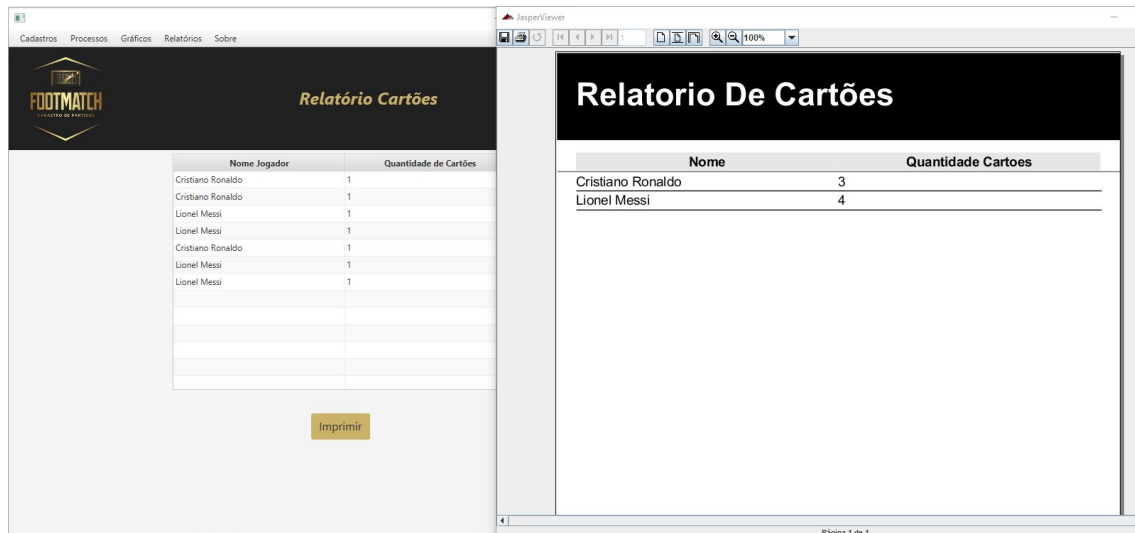


Figure 10. Relatório Cartoes

Relatório Cartões, Relatório de cartões por jogadores, aonde gera relatorios em PDF.

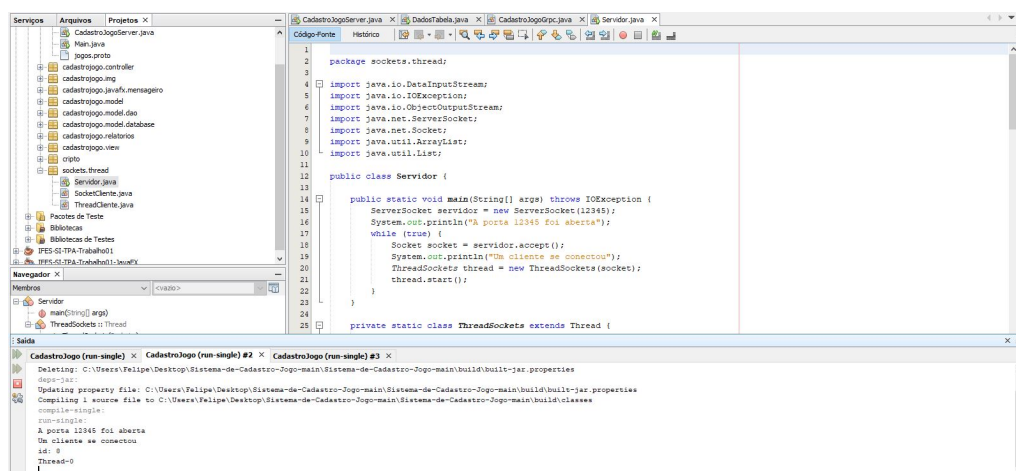


Figure 11. Servidor Socket

Executando o Servidor Socket para a Tela de Integrantes.



Figure 12. Tela Integrantes
Integrantes do sistema

2.2. Segunda Versão do Projeto utilizando um servidor gRPC:

- Configurando o ambiente de desenvolvimento:

Primeiro para implementação do servidor gRPC no projeto, foi necessário fazer o download do projeto, que pode ser encontrado em: <https://github.com/Felipepz/Sistema-de-Cadastro-Jogo>.

Após isso baixo os pacotes do protoc que estava localizado em : <https://drive.google.com/file/d/1jtIFiyuSjLq3bCTTfoeFAP6ZZ9PKHtIK/view?us> extrailo e executar.

Depois foi adicionado na pasta de preferencia um executavel do protobuf para java, localizado em: <https://drive.google.com/file/d/1UxwqBabrGC6gJleaRtYcBoqVIW4EB0Nx/view>.

E foi feito o download dos jar pelo link :https://drive.google.com/file/d/1jzWLwKztLbc_8c05vip9utQ6Sg4dJk_w/view?us

Com eles baixado foi necessário configurar as variáveis de ambiente no Path com o diretório bin do protoc como na imagem abaixo.

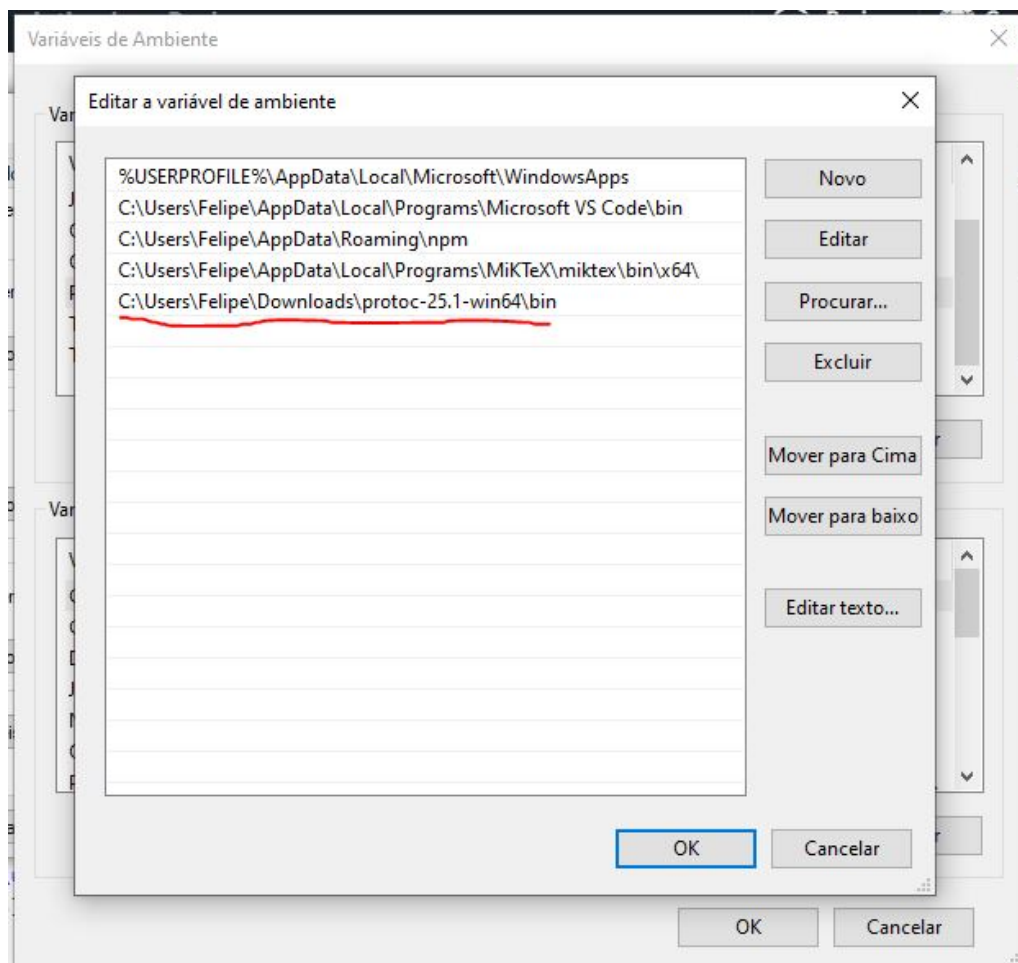


Figure 13. Variaveis de ambiente

Após isso é possível executar o comando `protoc --version` para ver se o `protoc` está configurado de forma certa em sua máquina.

Foi utilizado o NetBeans 8.2 + jdk do java 8 e o Postgres versão atualizada. O NetBeans pode ser encontrado em: https://archive.org/details/jdk-8u111-nb-8_2

E o Postgres pode ser encontrado em <https://www.postgresql.org/download/>

- Implementação

Após configurado todo o ambiente foi necessário abrir o projeto e adicionar os jar citados anteriormente, levando em questão que a pasta com os jar tem que estar em seu projeto.

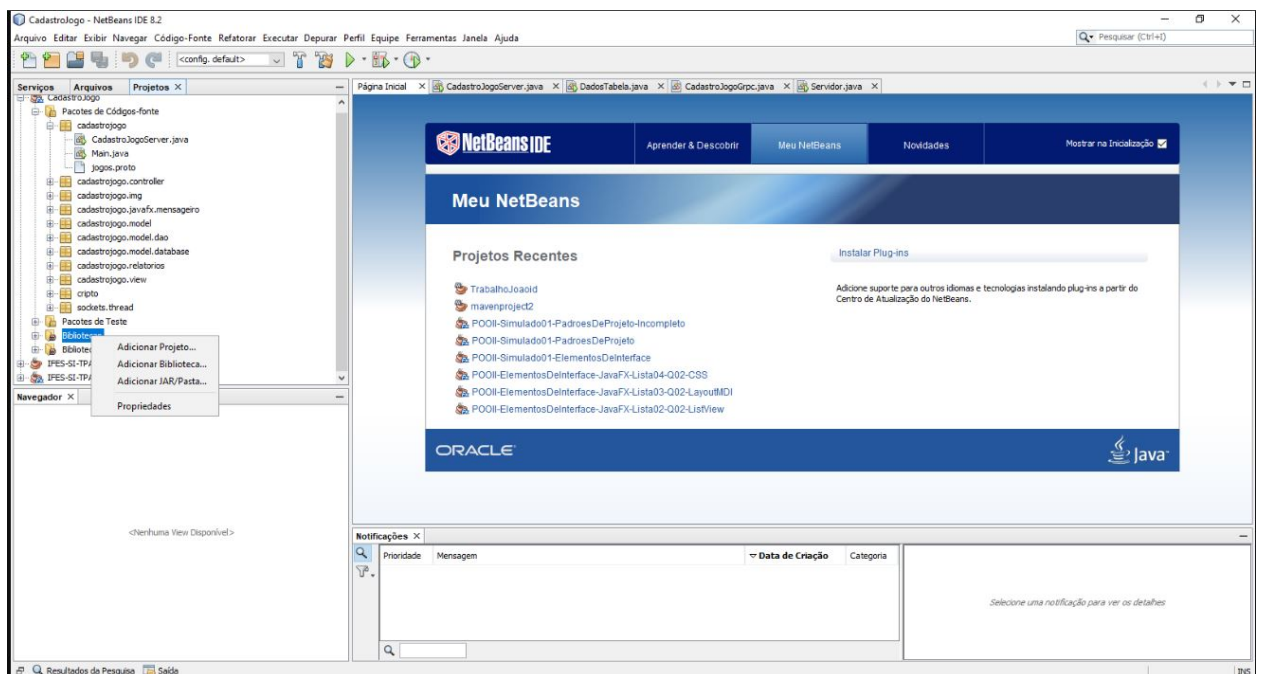


Figure 14. JAR
Selecione adicionar JAR

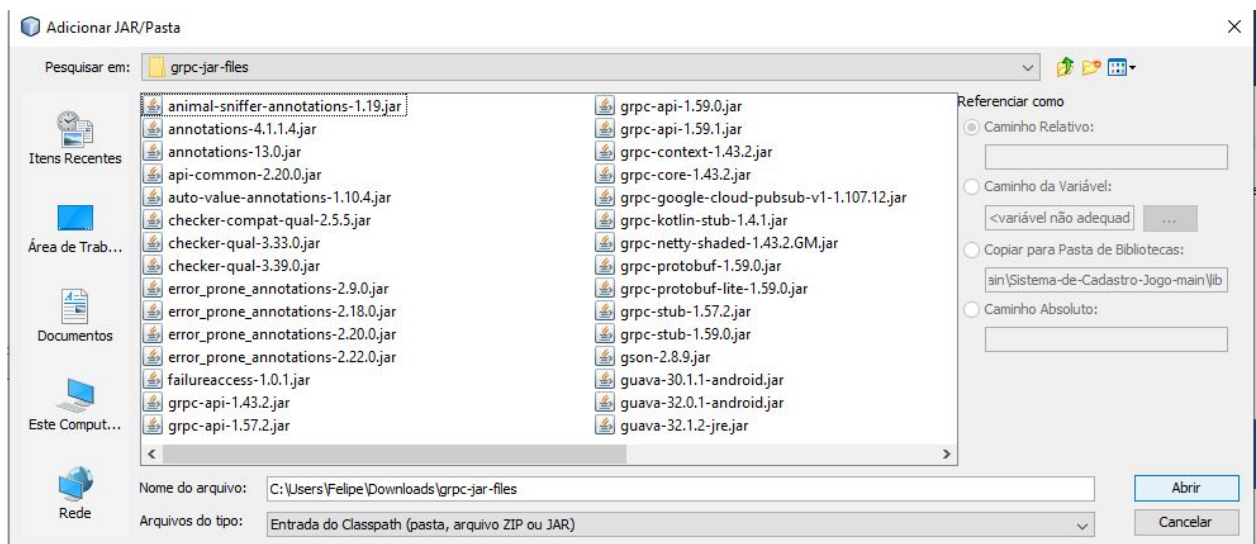


Figure 15. JAR Files
Selecione todos os JAR

Com os jar já adicionados é necessário a criação do arquivo.proto, abaixo será mostrado o jogos.proto que foi utilizado para criação dos pacotes gRPC para aplicação.

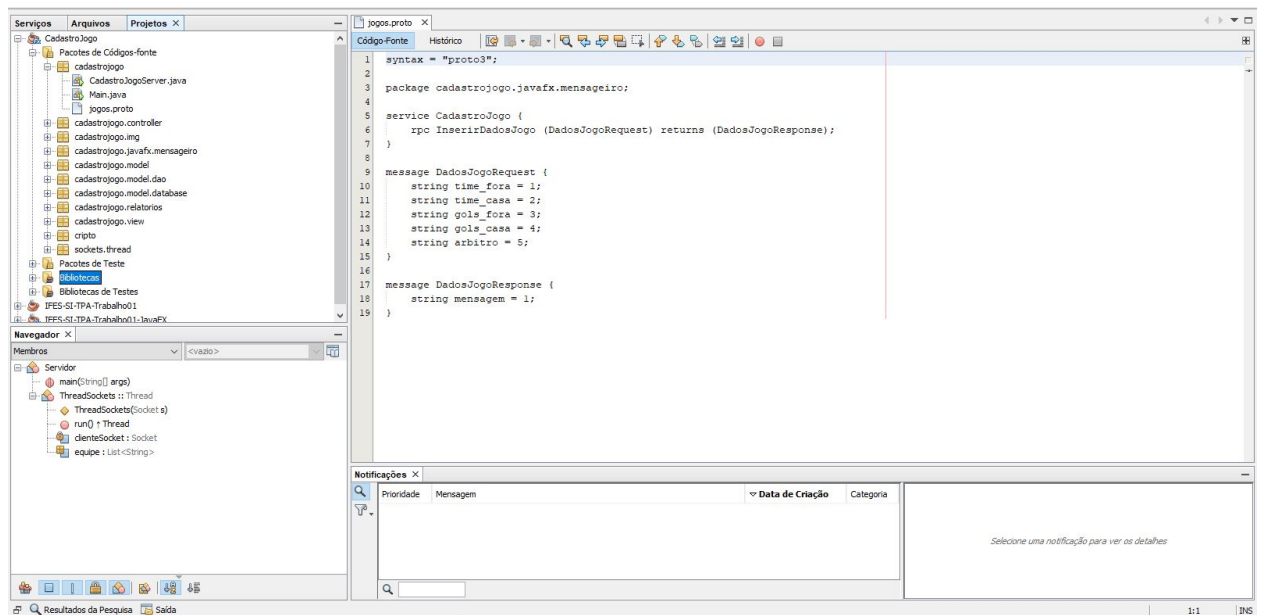


Figure 16. JAR

Arquivo jogos.proto

Com arquivo proto criado é necessário criar os pacotes com o comando protoc aonde em Diretorio seria especificado o diretorio do arquivo protoc citado anteriormente e mudar o arquivo.proto para o nome do seu arquivo proto. O comando seria o comando a seguir:

```
protoc --java_out=. --grpc_out=. --plugin=protoc-gen-grpc=Diretorio
\protoc-gen-grpc-java-1.59.1-windows-x86_64.exe arquivo.proto
```

Com o comando anterior executado é criado os pacotes CadastroJogoGrpc.java e Jogos.java como mostrado abaixo abaixo:

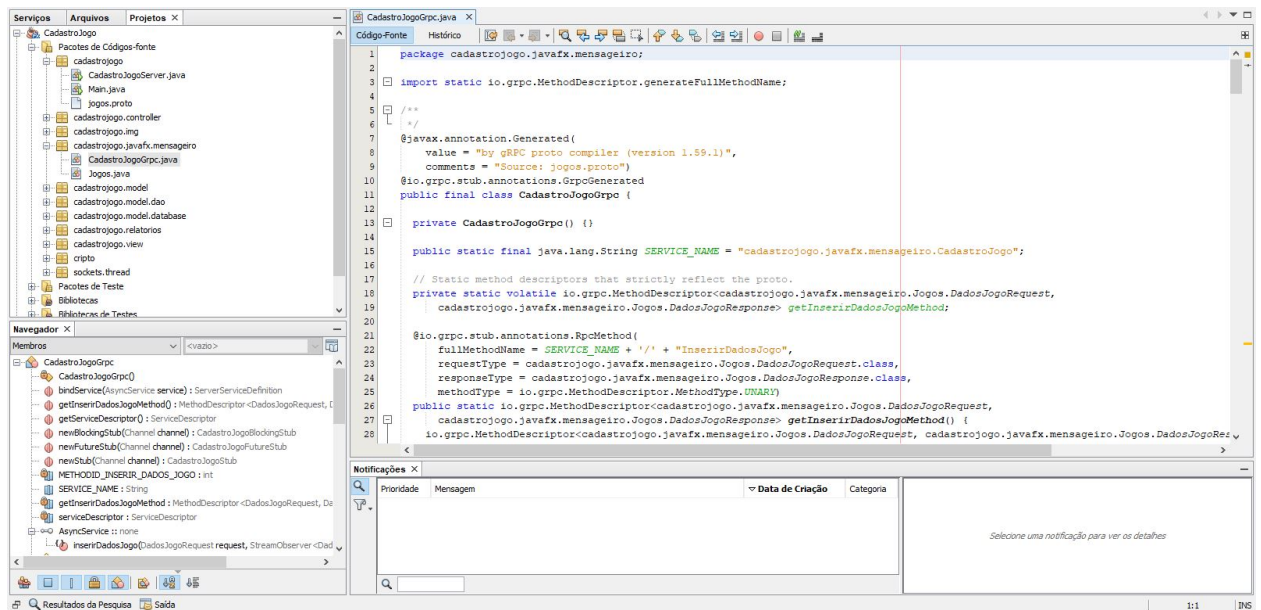
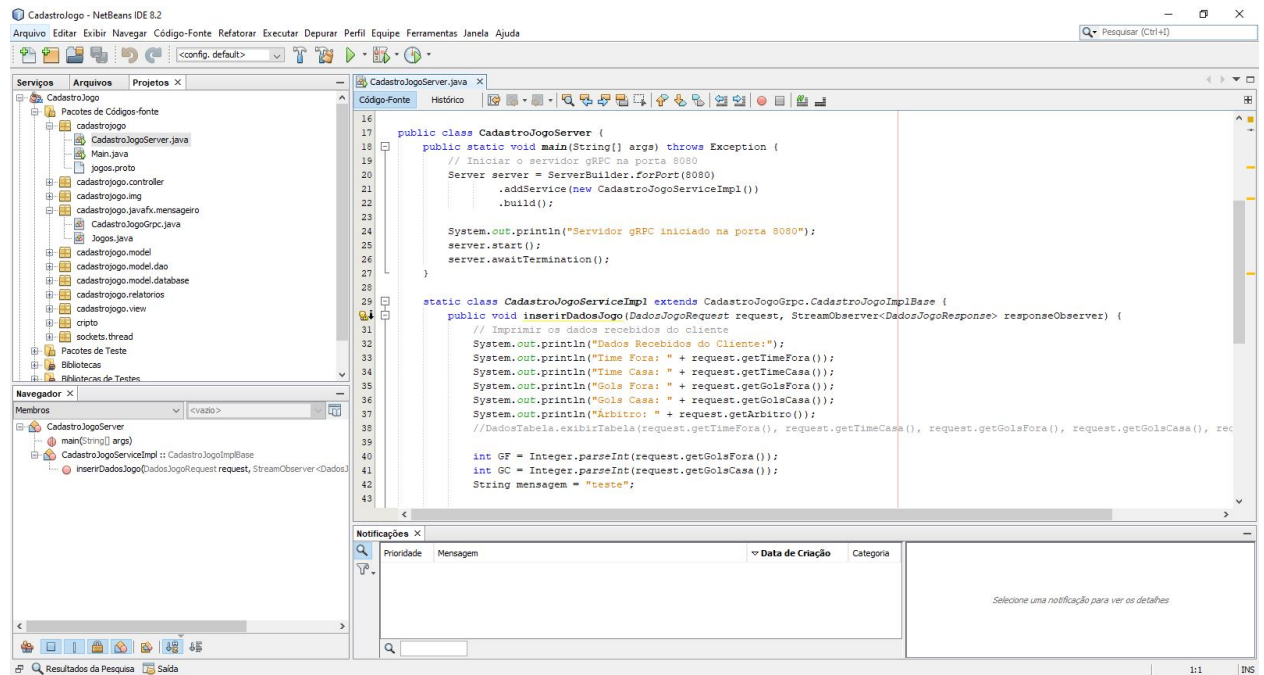


Figure 17. Pacotes Criados
Pacotes Criados

Após a criação dos pacotes, fizemos dois métodos para podermos chegar no resultado que estipulamos. O primeiro método simplesmente faz uma consulta na tabela "pJogo" do banco de dados e salva os dados obtidos no tableview, dados esses que também são passados para o servidor GRPC através do segundo método, método esse que tem como retorno o nome do time vencedor ou "empate" caso nenhum time tenha vencido.



**Figure 18. Main gRPC
Codigo Servidor gRPC**

Dentro do Servidor GRPC usamos uma simples estrutura de blocos de decisão para encontrar o vencedor ou informar um "empate", estrutura essa que pode ser conferida na imagem abaixo:

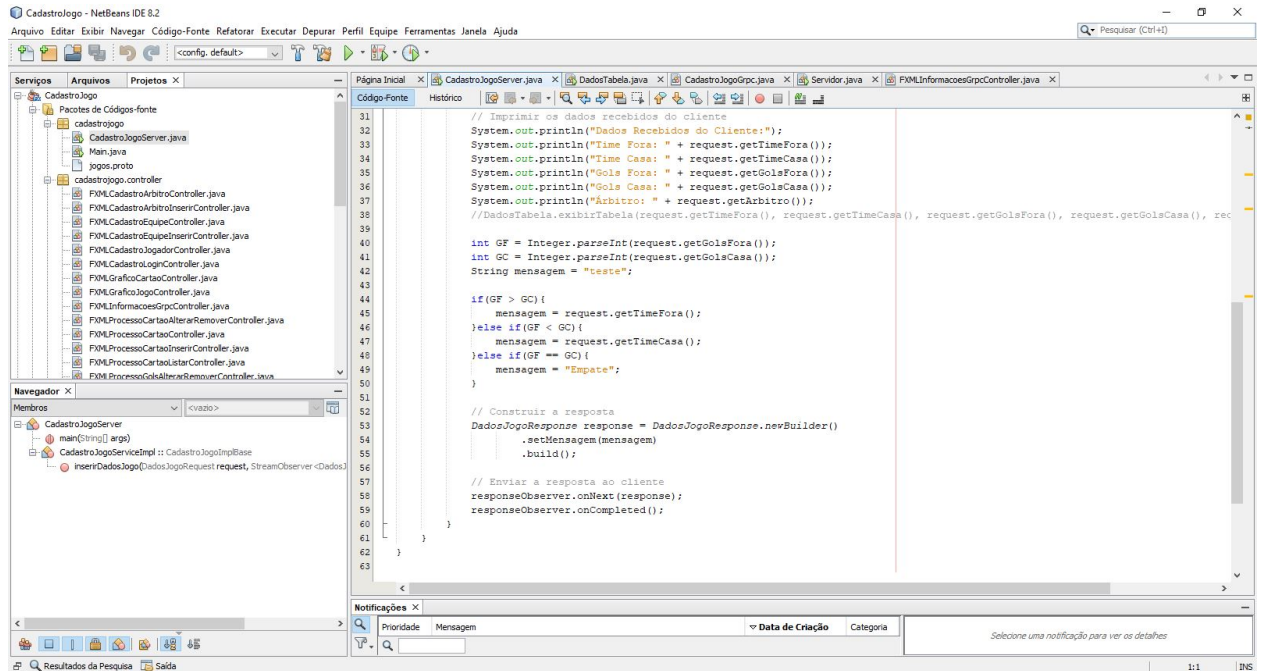


Figure 19. Main gRPC
Codigo Servidor gRPC

O método abaixo se trata do método responsável por enviar os dados de cada jogo para o servidor GRPC e informar a resposta do servidor (Time vencedor ou "Empate"), resposta essa que também ficara salva no tableview

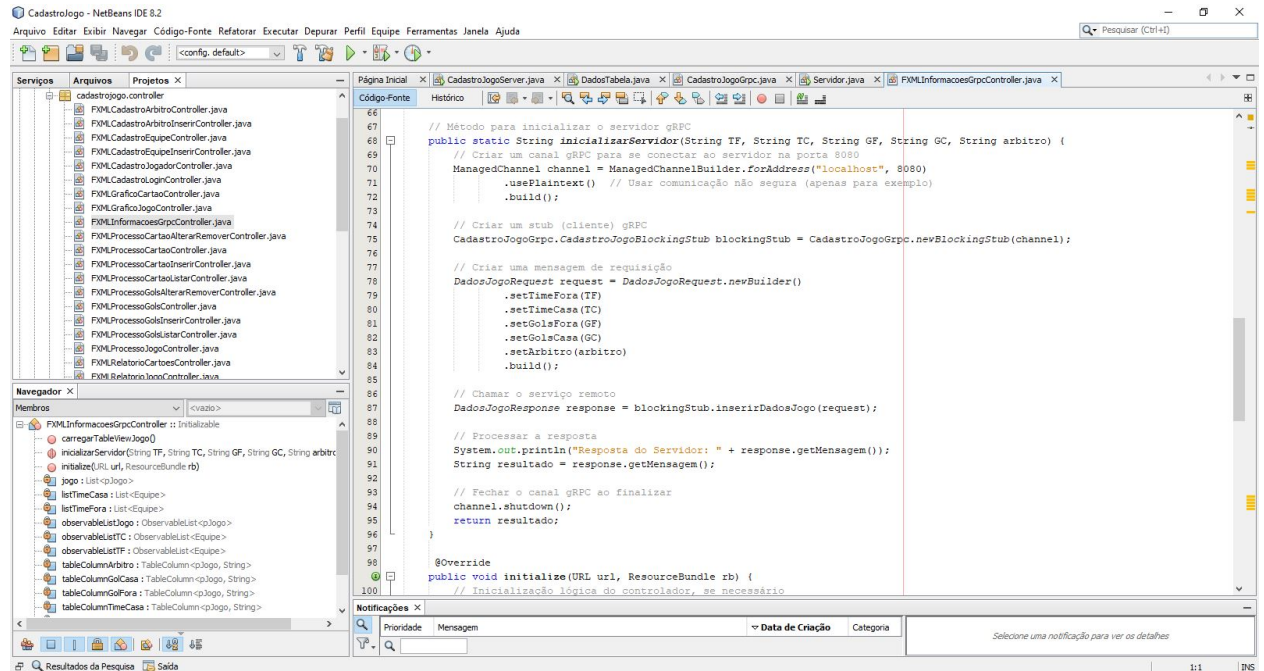


Figure 20. Controller
Controller gRPC

O método abaixo se trata do método responsável por realizar a consulta no banco de dados e salvar os dados no tableview, após isso ele chama o método responsável por enviar esses dados para o servidor GRPC e salva o retorno do método na coluna "Time Vencedor" do tableview

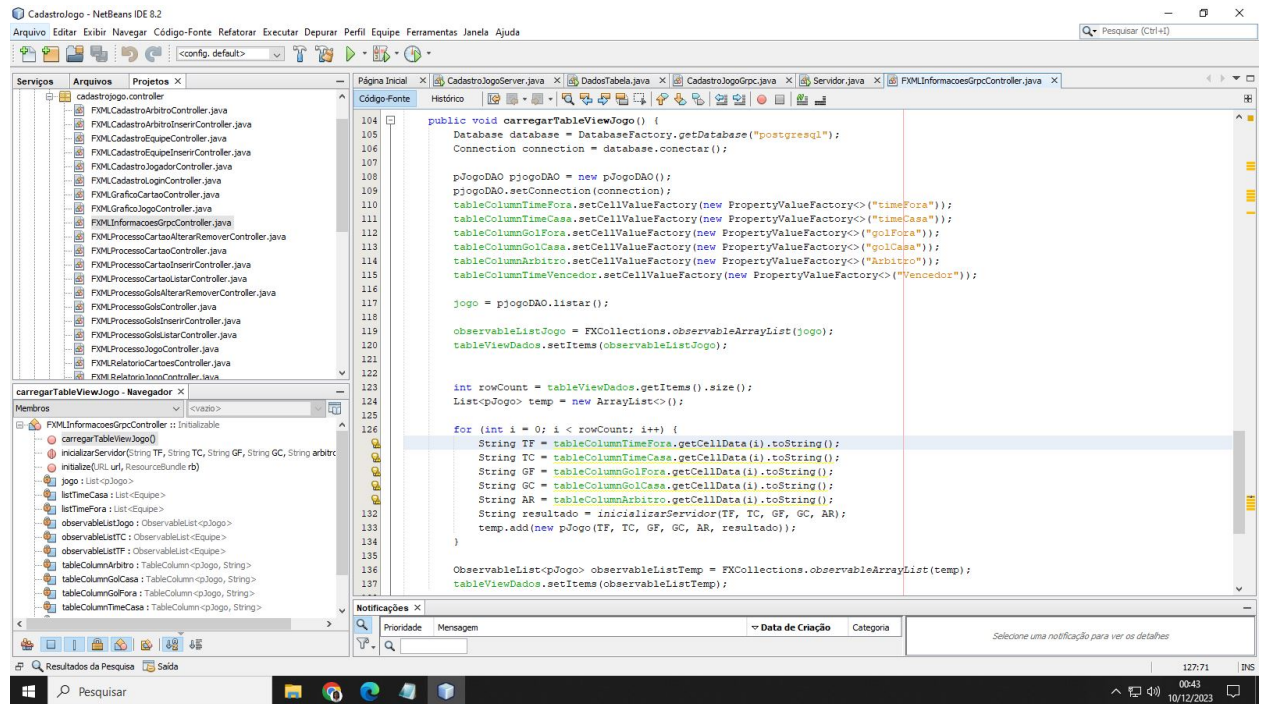


Figure 21. Controller
Controller gRPC

Nas duas imagens abaixo é possível ver como nossa tableview foi representada a nível de código, bem como as listview's necessárias para preencher as colunas com seus respectivos dados. Essa configuração se faz necessária pelo fato de estarmos usando javafx

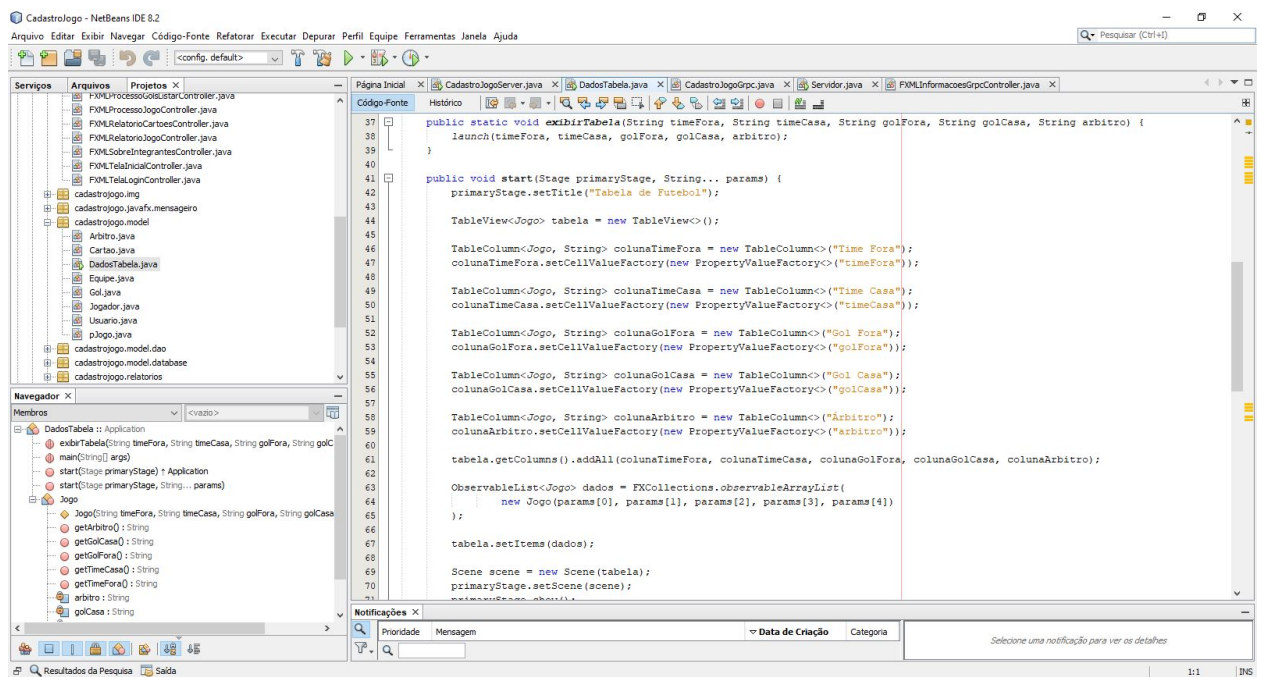


Figure 22. Dados tabela

Dados tabela

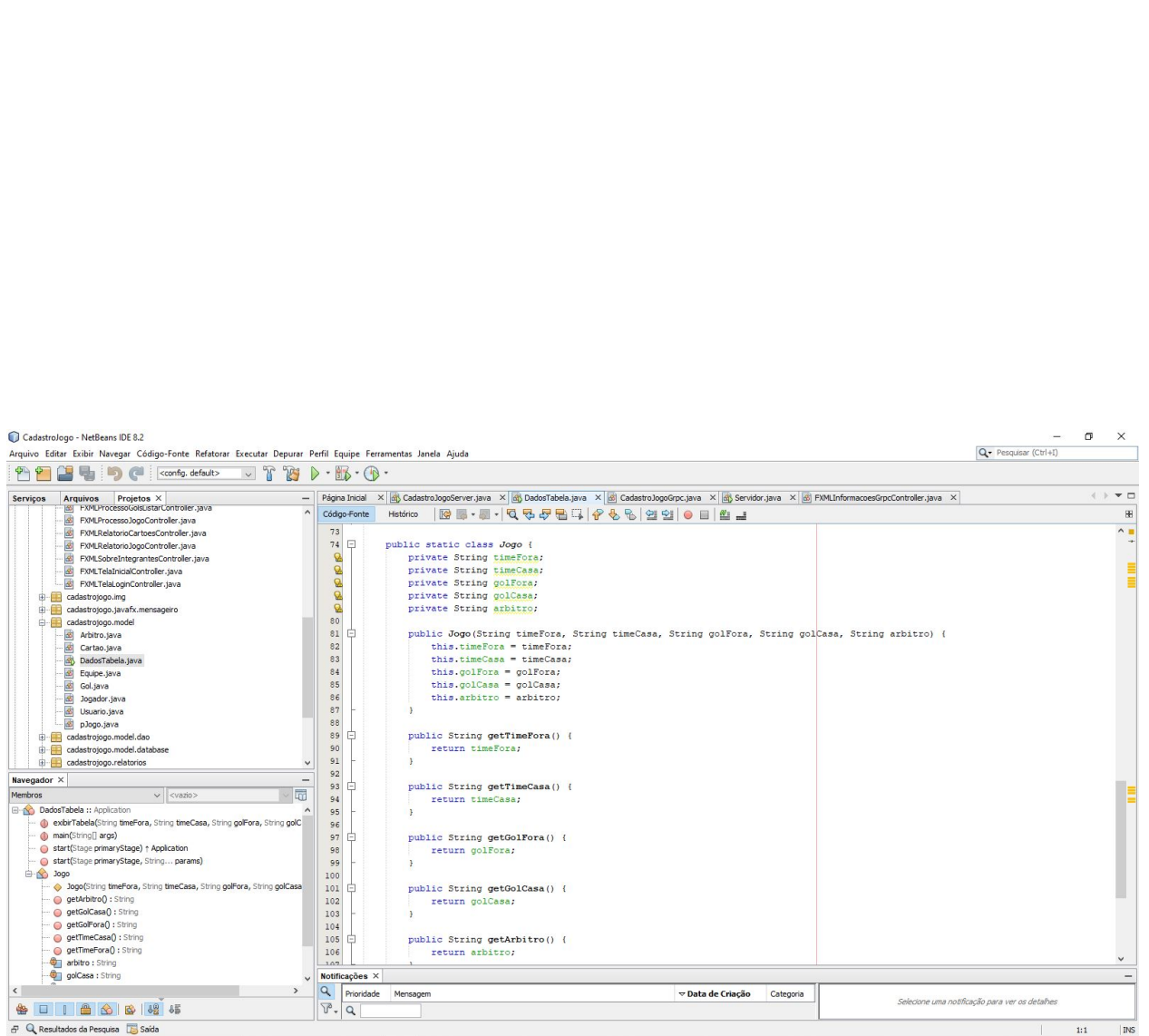
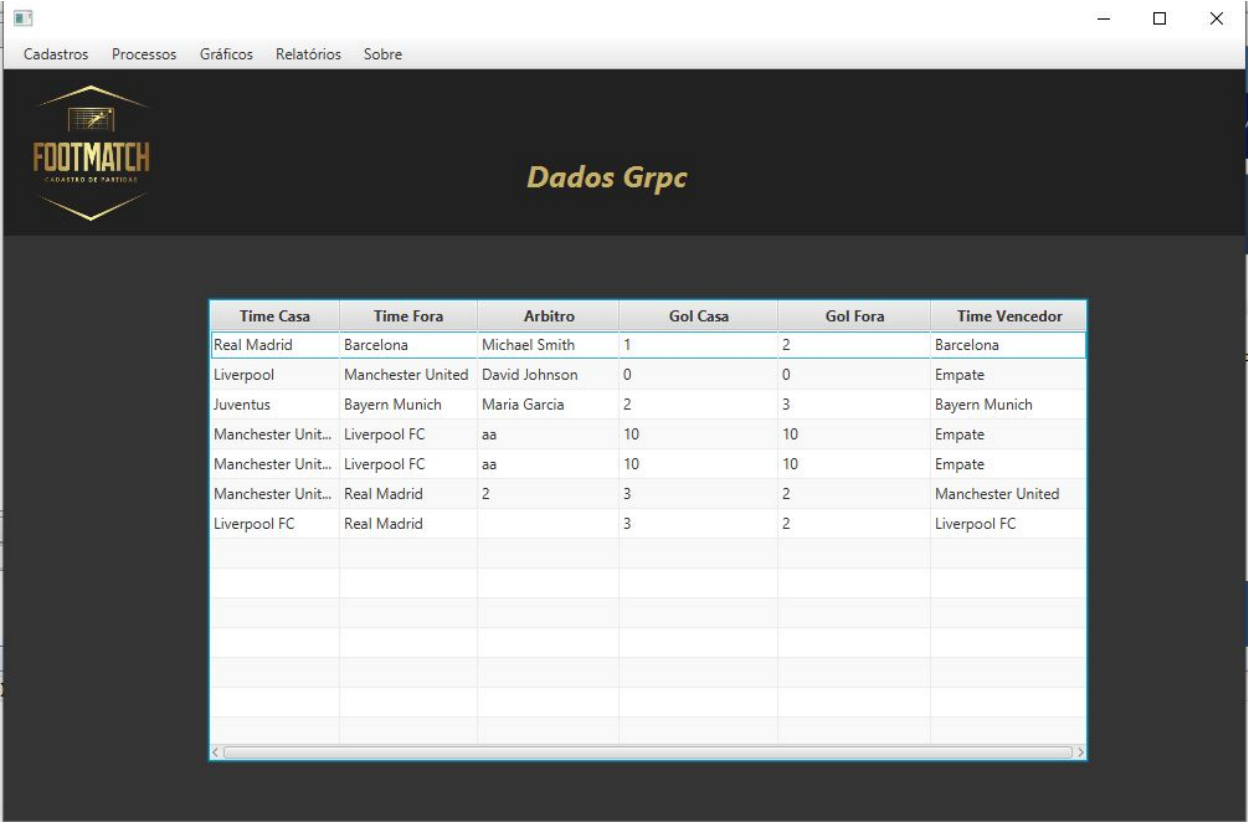


Figure 23. Dados tabela2

Dados tabela2, Segue a explicação anterior

Na imagem abaixo é possível verificar o layout da nossa tableview na interface gráfica junto de alguns dados de exemplo:



Time Casa	Time Fora	Arbitro	Gol Casa	Gol Fora	Time Vencedor
Real Madrid	Barcelona	Michael Smith	1	2	Barcelona
Liverpool	Manchester United	David Johnson	0	0	Empate
Juventus	Bayern Munich	Maria Garcia	2	3	Bayern Munich
Manchester Unit...	Liverpool FC	aa	10	10	Empate
Manchester Unit...	Liverpool FC	aa	10	10	Empate
Manchester Unit...	Real Madrid	2	3	2	Manchester United
Liverpool FC	Real Madrid		3	2	Liverpool FC

Figure 24. Time Vencedor gRPC
Time Vencedor gRPC

3. Conclusão

A transição da comunicação inicial, baseada em servidores Thread-Socket, para a implementação do gRPC representou uma significativa evolução no desempenho e eficiência do sistema. Na primeira versão, embora o modelo Thread-Socket tenha demonstrado eficácia em um contexto inicial, suas limitações em termos de escalabilidade e consumo de recursos tornaram-se evidentes à medida que o projeto crescia.

A decisão de adotar o gRPC na segunda versão proporcionou melhorias substanciais, resultando em chamadas de comunicação mais ágeis e tempos de resposta mais rápidos. Embora o sistema atual não apresente uma base de dados sólida de alto fluxo, a flexibilidade e desempenho oferecidos pelo gRPC atenderam de maneira satisfatória às necessidades dos integrantes do projeto, retornando como resposta, os times vencedores de cada partida. Essa transição não apenas abordou as limitações identificadas na abordagem anterior, mas também estabeleceu uma base sólida para futuras expansões e adaptações do sistema.

A eficiência do gRPC, aliada à sua capacidade de lidar com cenários mais complexos, destacou-se como uma escolha acertada, refletindo o compromisso da equipe em buscar soluções inovadoras para otimizar a comunicação e desempenho do sistema.

3.1. Código da Aplicação Final

<https://drive.google.com/file/d/1-MosrnlhCNdKsXuqbuVjQIU3l2IyXPQ/view>