Relatório Projeto
LP2

Arturo Fonseca Souza[1], **Lucas Agnez Lima**[2]

[1]Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)
Natal – RN – Brazil

Abstract. *The project consists of a desktop application where the user is able to register sports matches, search for athletes and view match results. Athletes, national teams and matches are saved in .csv files, where they are read and rewritten in case of any changes made by the client.*

Resumo. *O projeto consiste em uma aplicação desktop onde o usuário é capaz de registrar partidas de esportes, pesquisar por atletas e visualizar resultados de partidas. Os atletas, seleções e partidas são salvos em arquivos .csv, onde são lidos e reescritos em caso de alguma alteração feita pelo cliente.*

1. Introdução

A aplicação busca um modo de armazenar partidas de esportes, podendo depois acessa-las pelo esporte a qual ela pertence. Além de partidas, o aplicativo também permite que busque por atletas, onde é possível ver informações sobre ele, como peso, altura, data de nascimento e a sua posição na seleção. Atualmente os esportes disponíveis são Futebol, Volei e Basquete, mas é possível a adição de mais esportes em versões futuras.

2. Metodologia de Desenvolvimento

2.1. Etapas do Projeto

Em sua primeira versão, a aplicação tinha apenas as classes modelo dos esportes, partidas, atletas e seleções. Não existia um Banco para as seleções e cada classe de Esporte continha um array de partidas do seu respectivo esporte.

Já em sua segunda versão, introduzimos bancos para armazenar todos os atletas e todas as partidas, porém, todos os dados só existiam durante a execução do programa e eram perdidos ao reinicia-lo. Nessa versão também demos início à interface gráfica. O padrão de projeto utilizado foi o MVC (Model-View-Controller), com separação de classes em pacotes de Controle, Visão e Modelo .

Já na terceira e final versão, usamos de manipulação arquivos para ter um registro permanente das partidas e dos atletas. Nessa nos desvencilhamos um pouco do padrão MVC, visto que a ferramenta de interface gráfica que escolhemos, Java Swing, não comporta adequadamente esse padrão. Logo, juntamos a lógica de controle as classes de visão, ficando mais próximo do padrão MDI (Multiple Document Interface). Nessa versão também foi removido o esporte Tenis, pois a grande diferença na estrutura de suas partidas em relação aos outros esportes estava se mostrando um desafio. Por fim, a interface gráfica também foi finalizada e integrada na versão final do programa.

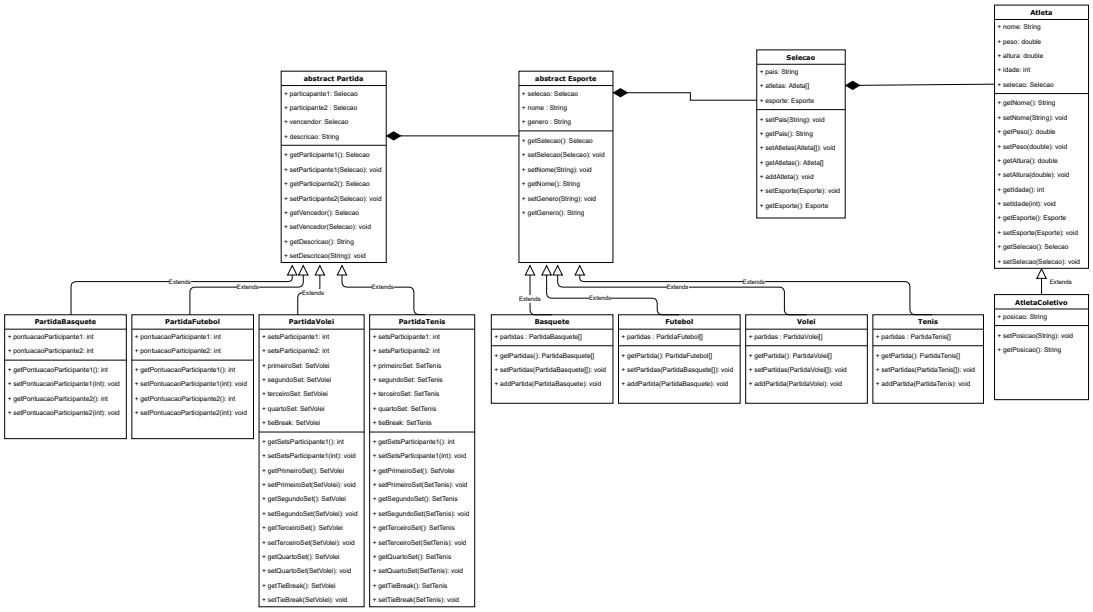

Figura 1. Diagrama de Classes Inicial

2.2. Descrição das Classes

Classes de Modelo

As classes de modelo estão contidas em 4 pacotes: ”esportes”, ”partida”, ”atletas”e ”tablemodel”. No primeiro pacote se encontra a classe abstrata ”Esporte”, cujos atributos são ”nome”, ”genero”e ”ID”, e as classes ”Basquete”, ”Futebol”e ”Volei”. Os atributos de Esporte representam o nome do esporte, o gênero (masculino ou feminino) e seu ID no banco de dados, o que será explicado mais a fundo nas classes de controle. Já os atributos das classes filhas, são Arrays de partidas de seu respectivo esporte.

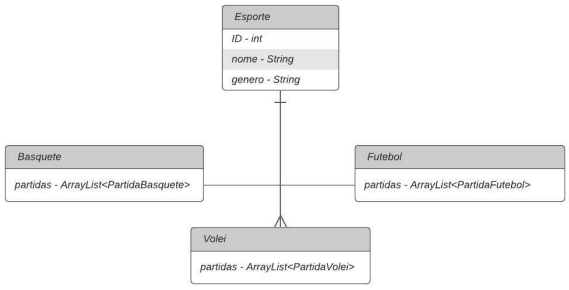

Figura 2. Diagrama de Classes de esportes

Já o pacote ”partidas”é onde se encontram as classes ”Partida”, ”PartidaBasquete”, ”PartidaFutebol”e ”PartidaVolei”. Na classe pai são armazenados os participantes de uma partida, a data em que ocorreu, seu vencedor e uma descrição da partida. Nas classes Futebol e Basquete, temos a pontuação de cada time, já na classe Volei, pela diferença

da estrutura do esporte, temos a pontuação de cada set jogado e quantos sets cada time ganhou.

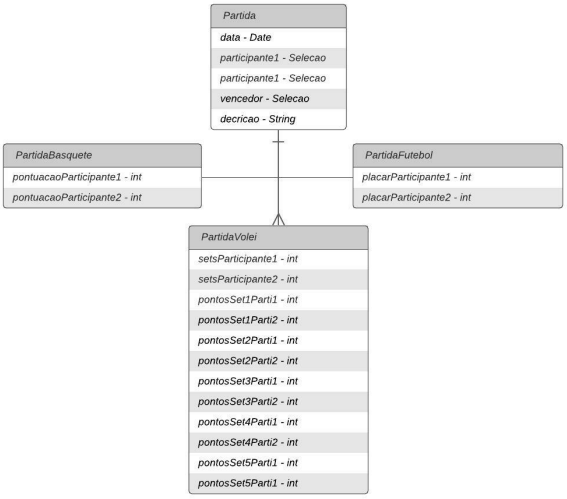

Figura 3. Diagrama de Classes de partidas

O pacote ”atletas”é onde estão as classes ”Atleta”, ”AtletaColetivo”e ”Selecao”. Em ”Atleta”estão as informações básicas de um atleta (i.e nome, peso, data de nascimento, etc), enquanto a classe ”AtletaColetivo”extende ”Atleta”e adiciona a posição que ele joga em seu time e a seleção a qual participa. Cada seleção contém um ID, um array de atletas, o nome de seu esporte e o nome do pais que representa.

Por fim, o pacote ”tablemodel”é onde são armazenados o modo em que tabelas serão exibidas pelas classes de visão. O modo que elas operam será exibido junto as classes de visão.

Classes de Controle

Como deixamos de seguir o padrão MVC, as classes de controle consistem unicamente na classe ”BancoSelecao”que realiza a persistência dos dados em arquivos do tipo CSV, assim como sua consulta. Tentamos mimetizar a estrutura de bancos de dados relacionais como MySQL, com tabelas com registros que fazem referência a registros de outras tabelas, através de chaves estrangeiras, isto é, IDs únicos que identificam registros e que são referenciados em outros registros em tabelas externas.

A classe ”BancoSelecao”segue o padrão Singleton, fazendo com que só exista uma instância dela durante toda a execução do programa, usada excessivamente nas classes de visão. Seu construtor, que é chamado assim que o programa inicia sua execução, é encarregado que buscar todos registro armazenados nos arquivos e colocar na memória do programa. Todos registro acabam sendo armazenados no atributo ”selecoes”, um conjunto de seleções que contém esportes, atletas e partidas em seus atributos.

A partir dessa coleta inicial dos arquivos, toda busca de dados é feita partindo do atributo ”selecoes”, já armazenado na memória, junto com atributos auxiliares que possuem as partidas dos esportes diretamente.

Classes de Visão

Primeiramente, a TelaPrincipal é a classe onde se encontram os 3 funcionamentos principais da aplicação: buscar partidas, buscar atletas, e armazenar partidas. Existem 3 botões nessa classe representando os 3 funcionamentos, e ao apertá-los, as outras telas serão chamadas.

Ao buscar partidas, o programa executará a tela ”TelaEscolherEsporteProcurarPartidas”que mostrará as opções dos esportes disponíveis e ao selecionar algum, será mostrada uma tabela com as informações básicas das partidas desse esporte.

Quando o botão de adicionar partidas é pressionado, você é levado a tela de selecionar esporte, similar a buscar partidas. Ao selecionar um esporte você será levado a tela da Figura 4.

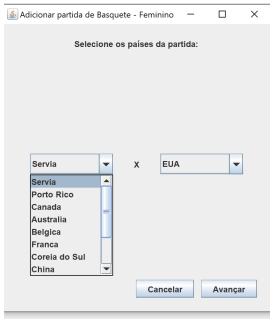

Figura 4. tela de selecionar os países

Onde você deve selecionar os países participantes. Após isso você terá de preencher a pontuação de cada seleção, a data e o horário que jogo ocorreu.

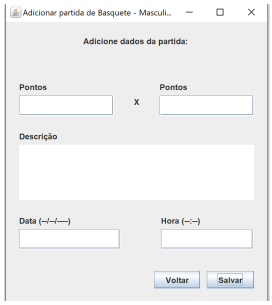

Figura 5. tela de cadastro de partida

Por fim, na opção de buscar atleta, você será levado a uma tela de busca onde deve inserir o nome que deseja buscar em uma caixa de texto e todos os atletas cadastrados que contém a String que você inseriu serão exibidos em uma tabela, junto de suas informações básicas.

Nas classes de visão a comunicação com o banco de dados fica principalmente no método ”actionPerformed”, onde também é realizado verificações e lançamentos de erro. Exceções são tratadas e mensagens de erro são exibidas caso algum campos esteja inválido ou vazio, por exemplo.

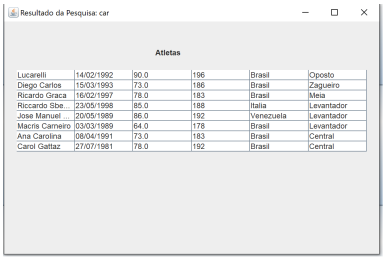

Figura 6. resultado de uma pesquisa

3. Dificuldades Encontradas

Inicialmente gastamos bastante tempo tentando entender que padrão de projeto usar, visto que tentamos utilizar estritamente o padrão MVC com Java Swing, o que não é muito recomendado.

A maior dificuldade encontrada foi a manipulação de arquivos para gerar o banco de dados, e a integração do banco de dados com a interface visual. Muitos bugs surgiam por pequenos erros que passavam despercebidos facilmente, o que gerava grandes problemas na interface gráfica.

Outra dificuldade foi o modo que o banco de dados foi estruturado, pois complicava mais que o necessário o acesso de algumas ações que deviam ser simples.

Acabamos não tendo tempo para desenvolver uma agenda esportiva propriamente dita com um calendário e listas de partidas próximas como foi planejado inicialmente, existindo apenas a função de registrar e visualizar partidas que ocorreram.

4. Conclusão

O projeto se mostrou mais desafiador que inicialmente pensado mas estamos muito satisfeitos com o resultado. Após conseguirmos terminar o banco de dados e remover todos os bugs durante o desenvolvimento a aplicação se mostrou muito eficiente em sua proposta e interessante de se utilizar.

Com a versão atual pronta, é fácil enxergar muitas adições possíveis a serem feitas, como maior variedade de buscas/exibição das partidas, maior polimento nas interfaces, suporte a mais esportes, inclusão de eventos futuros com uma biblioteca de calendário, dentre outros, mas todos podem ser facilmente inseridos com o modelo atual.

5. Referências

https://docs.oracle.com/javase/tutorial/uiswing/[1]
https://www.geeksforgeeks.org/[2]