UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIA E TECNOLOGIAS (CCET)
COORDENAÇÃO DO CURSO DE ENGENHARIA DA COMPUTAÇÃO
PROF.DR THALES LEVI AZEVEDO VALENTE

ANTONIO LISTER AZEVEDO SOUSA
GABRIEL FERNANDO CARNEIRO
KELY SOUZA DA SILVA
LUCAS TEIXEIRA MATOS
VICTOR COELHO DA SILVA

FUTAPP:
GERENCIADOR DE CLUBES DE FUTEBOL

São Luís - MA
2023

Prof.Dr. THALES LEVI AZEVEDO VALENTE

FUTAPP:
GERENCIADOR DE CLUBES DE FUTEBOL

Documentação sobre um software apresentado como parte dos requisitos para aprovação na disciplina Projeto e Desenvolvimento de Software, Turma 01, respectivamente, do período 2023.2 ministrado pelo Prof. Dr. Thales Levi Azevedo Valente.

São Luís - MA

2023

SUMÁRIO

1 Introdução

O futebol, assim como outras organizações culturais que constroem suas identidades a partir daquilo que expressam, tem sido influenciado por um processo contínuo de mercantilização. Percebe-se, neste sentido, que a referida atividade, caracterizada enquanto um esporte organizado no âmbito de clubes e associações, parece estar migrando de uma instituição fundamentada em valores e tradições para um modelo de negócio, que enfatiza critérios de eficiência, rentabilidade e competitividade (RODRIGUES, 2009), um bom exemplo é surgimento da Sociedades Anônimas do Futebol, criada pela Lei 14.193/2021, permitindo que clubes de futebol pudessem se transformados em empresas, destacando-se grandes clube do futebol Brasileiro, como Cruzeiro Esporte Clube, Clube de Regatas Vasco da Gama e o Botafogo de Futebol e Regatas.

Com o passar do tempo e o aumento dos retornos financeiros, o esporte vem se desenvolvendo e se modificando de forma que passa a ser encarado como um negócio lucrativo para as partes envolvidas: clubes, jogadores, empresários, patrocinadores, sócios e cidades. Atualmente, o futebol é muito mais que um esporte, há toda uma indústria por trás de um jogo.

A busca por eficiência e organização em uma empresa é um desafio constante, não seria diferente para um clube de futebol, pois acima de tudo, clubes de futebol são empresas gigantescas, que geram uma abundância de receitas por ano.

2 Objetivos

O principal objetivo desta aplicação é ajudar o gerenciamento de um clube de futebol, onde é possível armazenar dados como nome do clube, cidade e estádio, além disso, também é armazenado os jogadores do clube e treinador, no caso dos

jogadores, é importante destacar que existem dois tipos de transferência: empréstimo e definitivo.

3 Requisitos funcionais

Conforme Ian Sommerville (2011), os requisitos funcionais são uma representação das atividades que o sistema deve realizar, bem como uma descrição das interações esperadas entre o sistema e o ambiente. Essa perspectiva detalhada fornece uma base sólida para o desenvolvimento e a validação eficaz do software, assegurando que as funcionalidades essenciais sejam compreendidas e atendidas.

Com base na definição dada anteriormente, elaboraram-se os requisitos funcionais necessários para o desenvolvimento da aplicação FUTAPP, listados abaixo:

RF01. Efetuar Login:

- O sistema deve permitir que os usuários façam login usando credenciais válidas (nome de usuário e senha);
- Deve haver um sistema de recuperação de senha que permita aos usuários redefinirem suas senhas caso as esqueçam.

RF02. Gerenciar clubes:

- O sistema deve permitir o cadastro dos dados do clube, incluindo informações como nome, cidade e estádio;
- O sistema deve permitir a atualização das informações do clube;
- O sistema deve permitir a exclusão do clube do banco de dados;
- O sistema deve permitir a atualização dos dados do clube no banco de dados.

RF03. Gerenciar jogadores:

- O sistema deve permitir o cadastro de novos jogadores, incluindo informações como nome, posição, pé, camisa, nacionalidade e tipo de transferência(empréstimo ou definitivo);
- O sistema deve permitir a atualização das informações dos jogadores;
- O sistema deve permitir a exclusão de jogadores do banco de dados;
- O sistema deve permitir atualização de jogadores no banco de dados.

4 Requisitos não funcionais

Seguindo a visão de Roger S. Pressman (2006), os requisitos não funcionais são intrinsecamente ligados às qualidades específicas que um software deve possuir. Estes requisitos não se limitam apenas às funcionalidades do sistema, mas também incorporam restrições adicionais que são essenciais para a performance, segurança, usabilidade e outras características que moldam a qualidade global do software.

Com base na definição dada anteriormente, elaboraram-se os requisitos não-funcionais necessários para o desenvolvimento da aplicação FUTAPP, listados abaixo:

RNF01. Tempo de Resposta

- O sistema deve responder rapidamente às solicitações do administrador, levando no máximo 5 segundos, garantindo tempos de resposta curtos para ações comuns.

RNF02- Clareza

- O sistema deve fornecer feedback claro e mensagens de erro compreensíveis para orientar os usuários em caso de problemas.

RNF03. Manutenibilidade

- O código-fonte deve ser bem organizado e seguir as melhores práticas de programação para facilitar a manutenção e a extensibilidade do sistema.

RNF04. Documentação:

- O sistema deve ter uma documentação abrangente, incluindo manual do usuário, guia de instalação e documentação técnica para facilitar o entendimento e a manutenção do sistema.

RNF05- Interface

- A interface do usuário deve ser intuitiva e fácil de usar, exigindo o mínimo de treinamento para os usuários.

5 Regras de negócio

Regras de negócio são uma nova categoria de requisitos do sistema que representam decisões sobre como executar o negócio, e são caracterizadas pela orientação do negócio e sua tendência às mudanças (Rosca et al., 1997).

Abaixo estão listadas algumas regras de negócio deste projeto:

- **RN01:** Os jogadores devem ser contratados por meio de transferências do tipo empréstimo ou definitivo.
- **RN02:** no tipo transferência – empréstimo é necessário informar a duração do empréstimo e a situação de obrigatoriedade de compra.
- **RN03:** O código de referência de cada jogador é gerado automaticamente e deve ser único no sistema..

6 Estrutura de projeto

A estrutura do projeto pode ser organizada em camadas lógicas, seguindo uma arquitetura que promove a modularidade, reusabilidade e facilidade de manutenção. Uma abordagem comum é a arquitetura de três camadas: apresentação, lógica de negócios e persistência de dados.

- `model`: Classes que representam as entidades do sistema (Clube, Jogador, Treinador, Transferência).
- `service`: Lógica de negócios e operações CRUD.
- `dao`: Acesso a dados (interfaces e implementações).
- `exception`: Exceções personalizadas.
- `util`: Classes utilitárias, como gerador de código de referência.
- `presentation`: Classes relacionadas à interface do usuário.

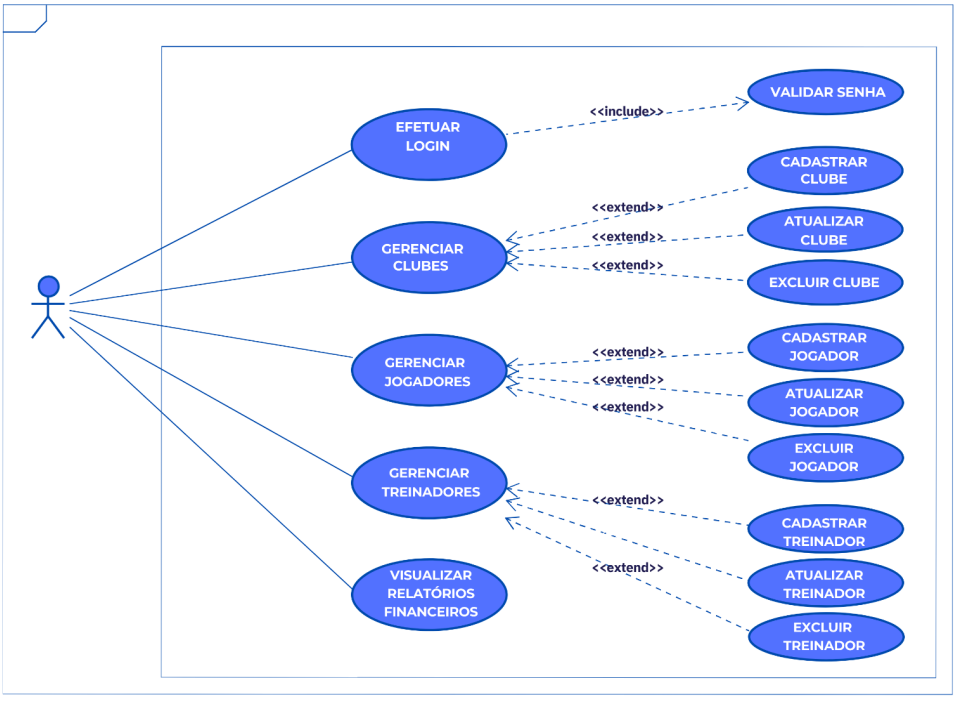
7 Modelagem Funcional: Casos de uso

A modelagem funcional é uma abordagem na engenharia de software que se concentra na representação das funcionalidades e comportamentos de um sistema, destacando as interações entre seus componentes (JACOBSON et al. 2006)

Diagrama de Casos de Uso

Uma representação gráfica que ilustra as interações entre os atores (usuários,outros sistemas ou entidades externas) e os casos de uso em um sistema. Tem o objetivo de ilustrar em um nível alto de abstração quais elementos externos interagem com as funcionalidades do sistema (Booch et al. 2006).

Figura 1: Diagrama de Casos de Uso - futAPP



Fonte: Própria Autoria, 2023

Caso de Uso: Efetuar Login

Ator Principal: Usuário (Administrador ou qualquer usuário autenticado)

Objetivo: Permitir que o usuário acesse o sistema por meio de credenciais válidas.

Fluxo Principal:

1. O usuário acessa a página de login do sistema.
2. O sistema exibe campos para inserção de nome de usuário e senha.
3. O usuário insere seu nome de usuário e senha.
4. O usuário solicita a autenticação no sistema.
5. O sistema valida as credenciais do usuário.
6. Se as credenciais são válidas, o sistema concede acesso ao usuário e registra a sessão.
7. O sistema redireciona o usuário para a página principal do sistema.

Fluxo Alternativo:

- No passo 6, se as credenciais são inválidas, o sistema exibe uma mensagem de erro.
- O usuário pode tentar novamente inserindo as credenciais corretas.

Requisitos Especiais:

- O sistema deve ter um mecanismo de bloqueio de conta após um número específico de tentativas de login mal-sucedidas para evitar ataques de força bruta.
- Senhas devem ser armazenadas de forma segura e criptografadas no banco de dados.

Pré-condições:

- O sistema está operacional.
- O usuário tem um nome de usuário e senha válidos.

Pós-condições:

- Se a autenticação for bem-sucedida, o usuário tem acesso às funcionalidades autorizadas.
- Se a autenticação falhar, o usuário permanece na página de login.

Caso de Uso: Gerenciar Jogadores

Ator Principal: Administrador

Objetivo: Permitir que o administrador cadastre, atualize e exclua informações sobre jogadores no sistema.

Fluxo Principal:

1. O administrador acessa a funcionalidade de gerenciamento de jogadores no sistema.
2. O sistema exibe as opções para cadastrar, atualizar ou excluir jogadores.
3. Cadastrar Jogador
    1. O administrador seleciona a opção de cadastrar jogador.
    2. O sistema exibe um formulário para inserção das informações do novo jogador (nome, posição, pé, camisa, nacionalidade, tipo de transferência).
    3. O administrador preenche os campos e confirma o cadastro.
    4. O sistema valida os dados e armazena o novo jogador no banco de dados.
4. **Atualizar Jogador:**
    1. O administrador seleciona a opção de atualizar jogador.
    2. O sistema exibe a lista de jogadores cadastrados.
    3. O administrador seleciona o jogador a ser atualizado.
    4. O sistema exibe um formulário preenchido com as informações atuais do jogador.
    5. O administrador realiza as atualizações necessárias (por exemplo, posição ou camisa) e confirma.
    6. O sistema valida os dados atualizados e os armazena no banco de dados.
5. **Excluir Jogador:**
    1. O administrador seleciona a opção de excluir jogador.
    2. O sistema exibe a lista de jogadores cadastrados.
    3. O administrador seleciona o jogador a ser excluído.
    4. O sistema confirma a exclusão após a confirmação do administrador.
    5. O sistema remove o jogador do banco de dados.

Fluxo Alternativo:

- Se houver erro na validação dos dados em qualquer uma das etapas de cadastro ou atualização, o sistema exibe mensagens de erro e retorna ao ponto correspondente no fluxo principal.

Requisitos Especiais:

- O sistema deve garantir que o número da camisa seja único para cada jogador.
- Informações sensíveis, como salário ou dados médicos, podem ser adicionadas como requisitos futuros, dependendo da necessidade do sistema.

Pré-condições:

- O administrador está autenticado no sistema.
- Pode haver jogadores já cadastrados no sistema.

Pós-condições:

- As informações dos jogadores são atualizadas no sistema conforme as ações do administrador.

Caso de Uso: Gestão de Transferências

Ator Principal: Administrador

Objetivo: Permitir que o administrador registre e atualize as transferências de jogadores, incluindo informações sobre empréstimos e transferências definitivas.

Fluxo Principal:

1. O administrador acessa a funcionalidade de gestão de transferências no sistema.
2. O sistema exibe as opções para registrar uma nova transferência ou atualizar informações de transferências existentes.
3. Registrar Transferência:
    1. O administrador seleciona a opção de registrar nova transferência.
    2. O sistema exibe um formulário para inserção das informações da transferência (jogador, tipo de transferência, clube de destino).
    3. O administrador preenche os campos, incluindo detalhes específicos, como duração do empréstimo e situação de obrigatoriedade de compra.
    4. O administrador confirma o registro da transferência.

5. O sistema valida os dados e armazena as informações da transferência no banco de dados.

4. **Atualizar Transferência:**
    1. O administrador seleciona a opção de atualizar informações de transferência.
    2. O sistema exibe a lista de transferências registradas.
    3. O administrador seleciona a transferência a ser atualizada.
    4. O sistema exibe um formulário preenchido com as informações atuais da transferência.
    5. O administrador realiza as atualizações necessárias, como a mudança de clube de destino, duração do empréstimo, ou situação de obrigatoriedade de compra, e confirma.
    6. O sistema valida os dados atualizados e os armazena no banco de dados.

Fluxo Alternativo:

- Se houver erro na validação dos dados em qualquer uma das etapas de registro ou atualização, o sistema exibe mensagens de erro e retorna ao ponto correspondente no fluxo principal.

Requisitos Especiais:

- O sistema deve verificar a elegibilidade da transferência com base nas regras de negócio, como a situação de obrigatoriedade de compra em transferências por empréstimo.
- A atualização de transferências deve ser cuidadosamente validada para garantir a consistência dos dados.

Pré-condições:

- O administrador está autenticado no sistema.
- Pode haver jogadores e clubes cadastrados no sistema.

Pós-condições:

- As informações da transferência são registradas ou atualizadas no sistema conforme as ações do administrador.

Caso de Uso: Gerenciar Clubes

Ator Principal: Administrador

Objetivo: Permitir que o administrador cadastre, atualize e exclua informações sobre clubes no sistema.

Fluxo Principal:

1. O administrador acessa a funcionalidade de gerenciamento de clubes no sistema.
2. O sistema exibe as opções para cadastrar, atualizar ou excluir clubes.
3. **Cadastrar Clube:**
    1. O administrador seleciona a opção de cadastrar clube.
    2. O sistema exibe um formulário para inserção das informações do novo clube (nome, cidade, estádio).
    3. O administrador preenche os campos e confirma o cadastro.
    4. O sistema valida os dados e armazena o novo clube no banco de dados.
4. **Atualizar Clube:**
    1. O administrador seleciona a opção de atualizar clube.
    2. O sistema exibe a lista de clubes cadastrados.
    3. O administrador seleciona o clube a ser atualizado.
    4. O sistema exibe um formulário preenchido com as informações atuais do clube.
    5. O administrador realiza as atualizações necessárias (por exemplo, cidade ou estádio) e confirma.
    6. O sistema valida os dados atualizados e os armazena no banco de dados.
5. **Excluir Clube:**
    1. O administrador seleciona a opção de excluir clube.
    2. O sistema exibe a lista de clubes cadastrados.
    3. O administrador seleciona o clube a ser excluído.
    4. O sistema confirma a exclusão após a confirmação do administrador.
    5. O sistema remove o clube do banco de dados.

Fluxo Alternativo:

- Se houver erro na validação dos dados em qualquer uma das etapas de cadastro ou atualização, o sistema exibe mensagens de erro e retorna ao ponto correspondente no fluxo principal.

Requisitos Especiais:

- O sistema deve garantir que o nome do clube seja único no sistema.
- Informações adicionais, como histórico do clube ou proprietário, podem ser adicionadas como requisitos futuros, dependendo da necessidade do sistema.

Pré-condições:

- O administrador está autenticado no sistema.
- Pode haver clubes já cadastrados no sistema.

Pós-condições:

- As informações dos clubes são atualizadas no sistema conforme as ações do administrador.

Caso de Uso: Gerenciar Treinador

Ator Principal: Administrador

Objetivo: Permitir que o administrador cadastre, atualize e exclua informações sobre o treinador do clube no sistema.

Fluxo Principal:

1. O administrador acessa a funcionalidade de gerenciamento de treinadores no sistema.
2. O sistema exibe as opções para cadastrar, atualizar ou excluir treinadores.
3. **Cadastrar Treinador:**
    1. O administrador seleciona a opção de cadastrar treinador.
    2. O sistema exibe um formulário para inserção das informações do novo treinador (nome, cargo).
    3. O administrador preenche os campos e confirma o cadastro.
    4. O sistema valida os dados e armazena o novo treinador no banco de dados.
4. **Atualizar Treinador:**
    1. O administrador seleciona a opção de atualizar treinador.
    2. O sistema exibe a lista de treinadores cadastrados.
    3. O administrador seleciona o treinador a ser atualizado.
    4. O sistema exibe um formulário preenchido com as informações atuais do treinador.
    5. O administrador realiza as atualizações necessárias (por exemplo, cargo) e confirma.
    6. O sistema valida os dados atualizados e os armazena no banco de dados.
5. **Excluir Treinador:**
    1. O administrador seleciona a opção de excluir treinador.
    2. O sistema exibe um formulário com os dados do treinador cadastrado.
    3. O sistema confirma a exclusão após a confirmação do administrador.
    4. O sistema remove o treinador do banco de dados.

Fluxo Alternativo:

- Se houver erro na validação dos dados em qualquer uma das etapas de cadastro ou atualização, o sistema exibe mensagens de erro e retorna ao ponto correspondente no fluxo principal.

Requisitos Especiais:

- O sistema deve garantir que o nome do treinador seja único no sistema.
- Informações adicionais, como histórico ou conquistas do treinador, podem ser adicionadas como requisitos futuros, dependendo da necessidade do sistema.

Pré-condições:

- O administrador está autenticado no sistema.
- Pode haver treinadores já cadastrados no sistema.

Pós-condições:

- As informações dos treinadores são atualizadas no sistema conforme as ações do administrador.

Caso de Uso: Exibir Relatórios Financeiros

Ator Principal: Usuário Autorizado

Objetivo: Permitir que usuários autorizados visualizem relatórios financeiros do clube de futebol.

Fluxo Principal:

1. O usuário autorizado acessa a funcionalidade de relatórios financeiros no sistema.
2. O sistema exibe opções para visualizar diferentes tipos de relatórios (receitas, despesas, lucros, etc.).
3. O usuário seleciona o tipo de relatório desejado.
4. O sistema gera o relatório com base nas informações disponíveis no banco de dados.
5. O usuário visualiza o relatório na interface do sistema.

Fluxo Alternativo:

- Se não houver dados disponíveis para gerar o relatório selecionado, o sistema exibe uma mensagem indicando a ausência de informações.

Requisitos Especiais:

- O sistema deve ter um mecanismo de segurança para garantir que apenas usuários autorizados possam acessar os relatórios financeiros.
- Os relatórios devem ser gerados de forma clara e compreensível, utilizando gráficos ou tabelas quando apropriado.

Pré-condições:

- O usuário autorizado está autenticado no sistema.

Pós-condições:

- O usuário visualiza o relatório financeiro desejado.

8 Modelagem de Classes de análise

8.1 Diagrama de Classes

O diagrama de classes, um componente essencial da UML, é um diagrama estrutural que revela a composição interna das entidades envolvidas. É possivelmente o diagrama mais utilizado na UML (BOOCH, 2006).

- Ele evolui com o sistema e pode ter diferentes perspectivas, como:
- Na Análise: Identificamos objetos no domínio do problema;
- No Projeto: Pensamos em objetos para a solução;

Segundo Fowler (2003), o modelo de análise representa as classes no domínio do negócio em questão. Não leva em consideração restrições inerentes à tecnologia a ser utilizada na solução de um problema. Já o modelo de projeto, é obtido através da adição de detalhes ao modelo anterior conforme a solução de software escolhida.

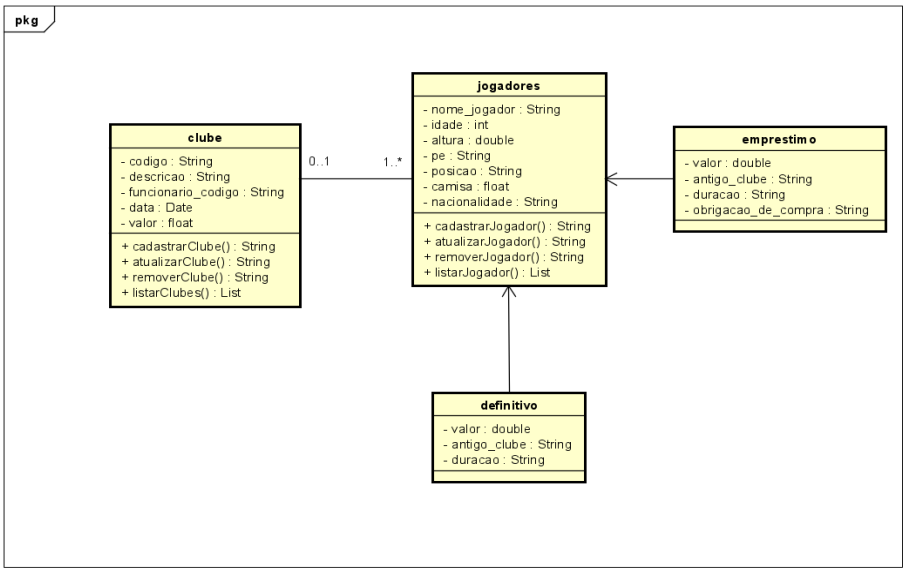
9 Modelagem de Classes de projeto

Em conformidade com os princípios estabelecidos por Ivar Jacobson, a fase de projeto é um estágio crítico onde os métodos são adicionados e o Modelo

Conceitual é refinado, culminando na criação do Diagrama de Classes. Essa abordagem, preconizada por Jacobson, não apenas aprimora a estrutura do sistema, mas também proporciona uma base sólida para a implementação eficaz de sistemas complexos.

Na Figura 2 consta o diagrama de classes desenvolvido para o projeto. Foram adotadas quatro classes: clube, jogadores, definitivo e empréstimo.

Figura 2: Diagrama de classes de projeto.



Fonte: Própria Autoria, 2023

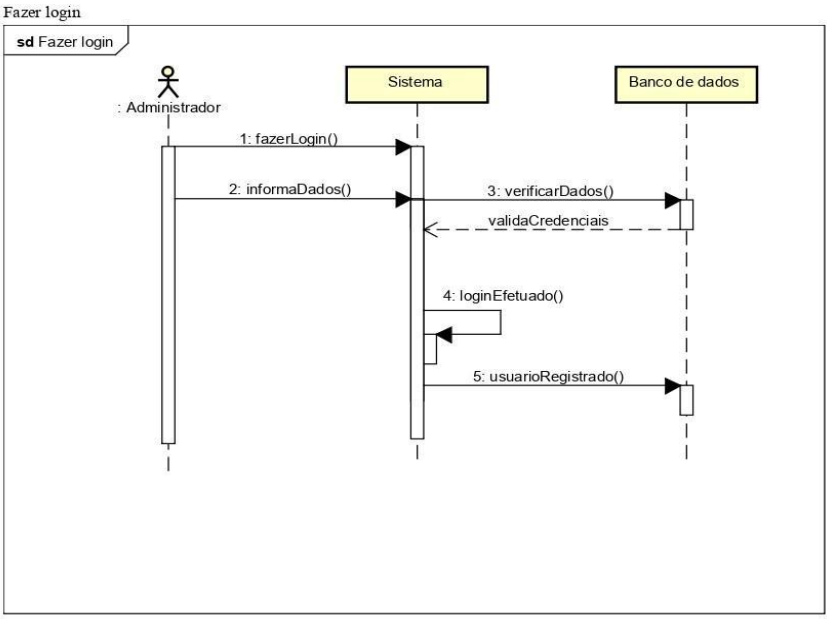
10 Modelagem de interação e classe de projeto

Em consonância com as orientações de Ivar Jacobson (2006), uma autoridade em engenharia de software e coautor da UML, a modelagem de interação e o design de classes encontram expressão eficaz na utilização de diagramas de sequência. Estes diagramas, empregados nesta etapa, oferecem uma representação detalhada e passo a passo das interações de um objeto em um único caso de uso, contribuindo assim para uma compreensão abrangente do sistema em desenvolvimento.

Grady Booch (2006), enfatiza que o diagrama de sequência é uma ferramenta gráfica valiosa para representar a interação temporal entre objetos em um sistema. Sua utilidade é evidenciada na capacidade de visualizar de forma clara e compreensível o fluxo de mensagens entre objetos em resposta a eventos específicos, proporcionando assim uma visão dinâmica crucial durante o processo de modelagem.

Na ilustração abaixo tem-se o diagrama de sequência efetuar login:

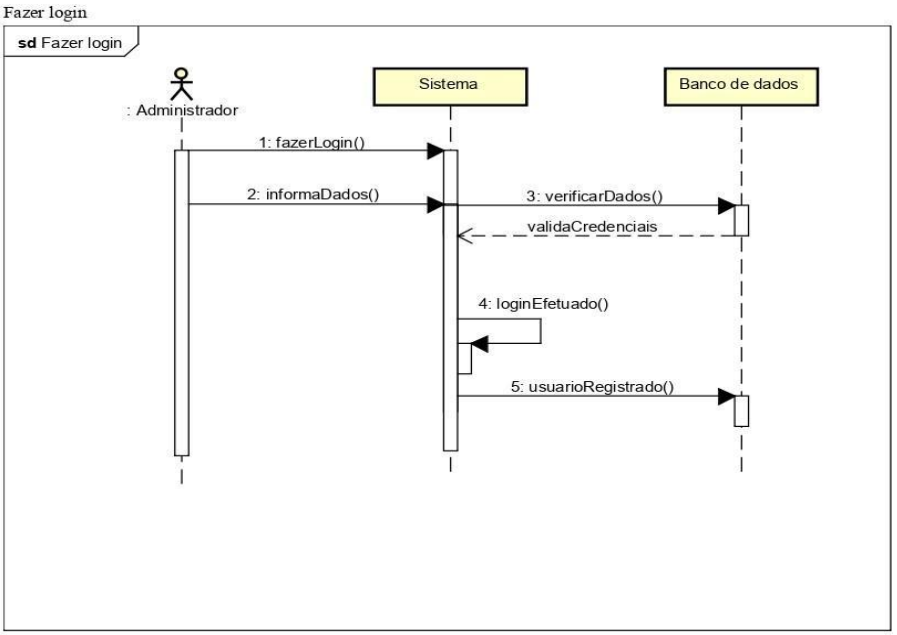
Figura 3: Diagrama de sequência-fazer login



Fonte: Própria Autoria, 2023

Na imagem abaixo tem-se o diagrama de sequência da ação cadastrar jogador:

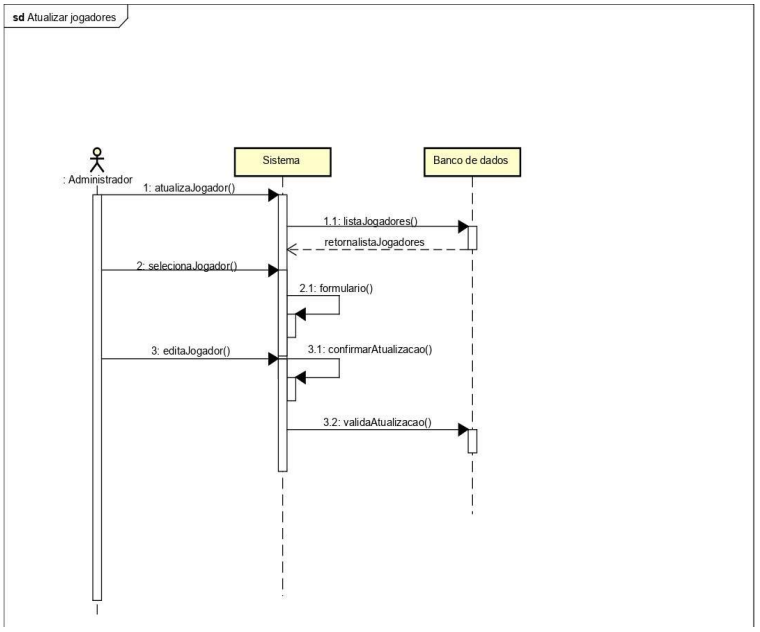
Figura 4: diagrama de sequência



Fonte: Autoria Própria, 2023

Destaca-se o diagrama de sequência referente a atualização de jogadores, ilustrado na Figura 5:

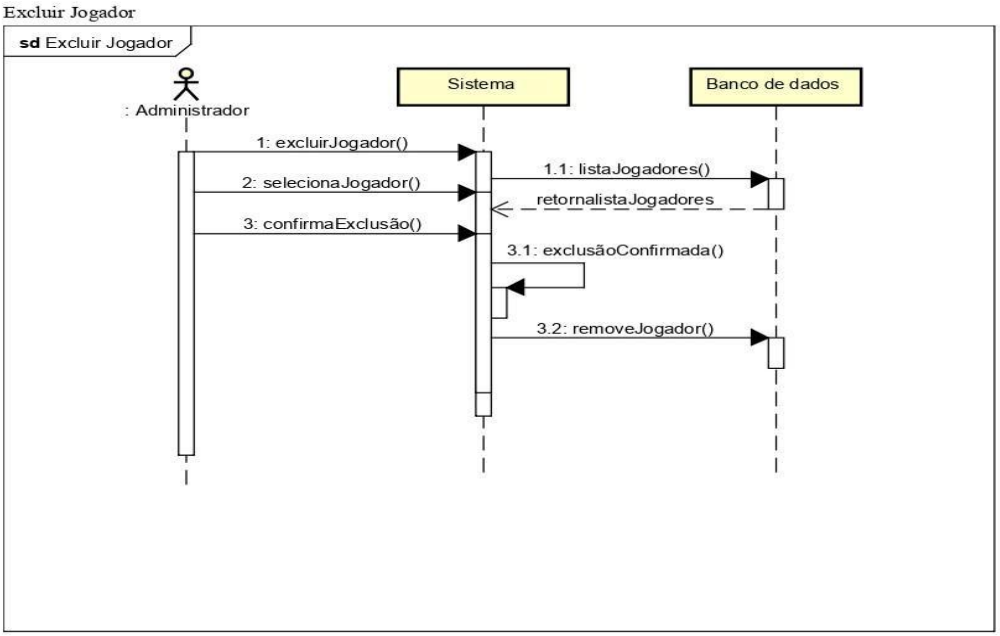
Figura 5: Diagrama de sequência-atualizar jogador



Fonte: Autoria própria, 2023

Na ilustração abaixo tem-se o diagrama de sequência - excluir jogador.

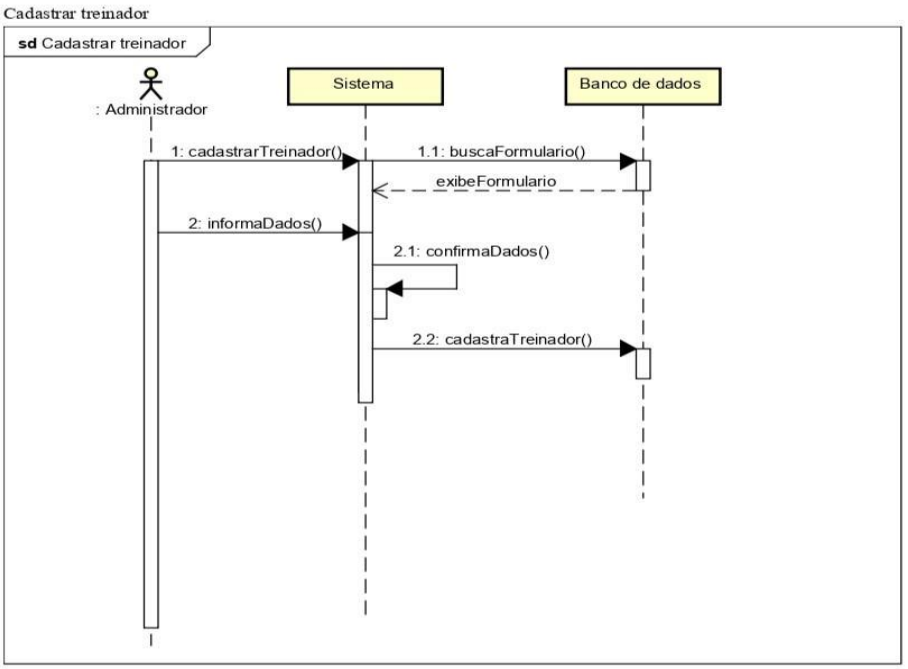
Figura 6: Diagrama de sequência-excluir jogador



Fonte: Autoria própria, 2023

Quanto ao gerenciamento de treinadores, tem-se o seguinte diagrama de sequência:

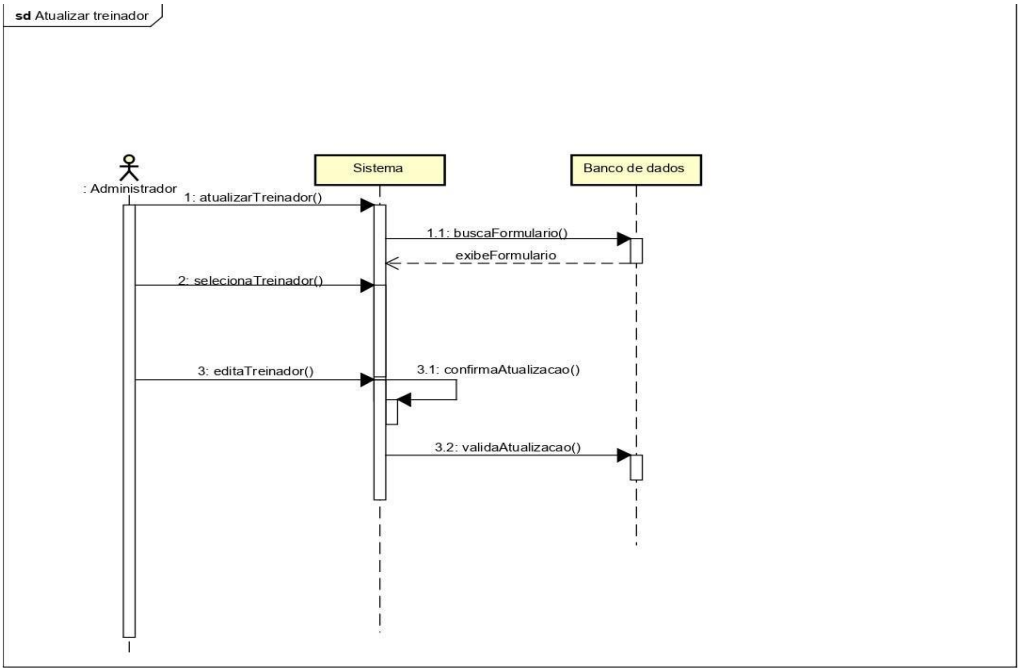
Figura 7: Diagrama de sequência-cadastrar treinador



Fonte Autoria própria, 2023

Assim, para a atualização de treinador, tem-se o seguinte diagrama:

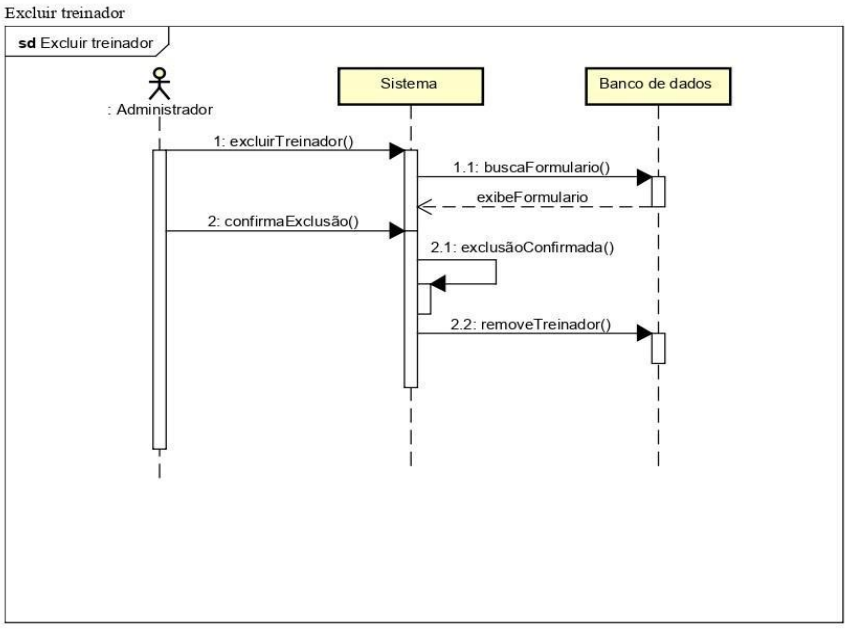
Figura 8: diagrama de sequência atualizar treinador



Fonte: Autoria própria, 2023

Para o método de exclusão referente ao técnico, tem-se seguinte diagrama, conforme Figura 9:

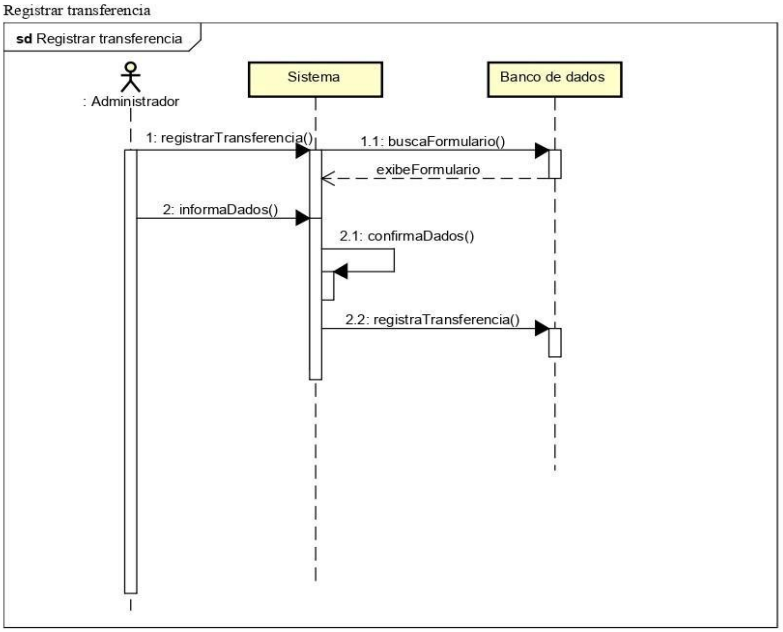
Figura 9: Diagrama de sequência-excluir treinador



Fonte: Autoria própria, 2023.

Os últimos dois referem-se a geração de relatórios e o gerenciamento de transferências, inicialmente tem-se abaixo, na Figura 10, o diagrama de sequência referente ao gerenciamento de transferência:

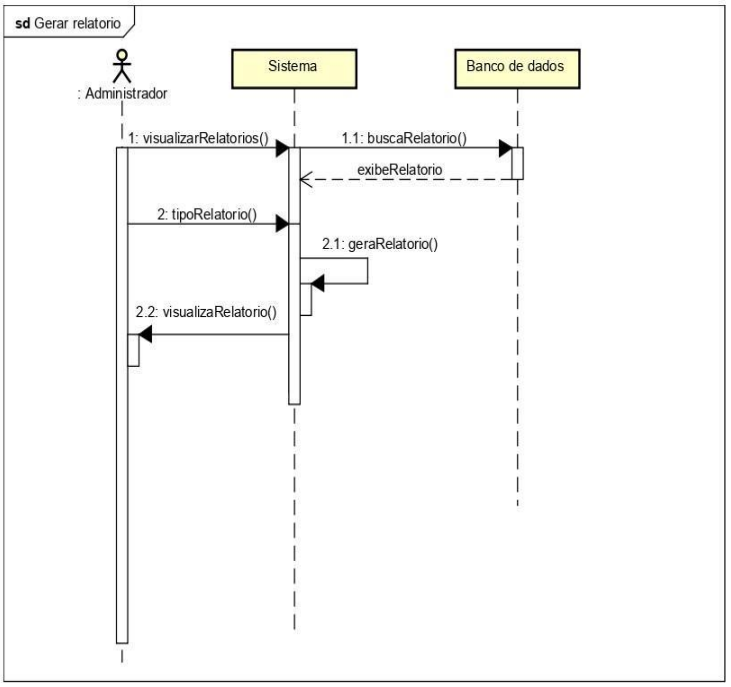
Figura 10: Diagrama de sequência-registrar transferência



Fonte: Autoria própria, 2023

Por fim, na imagem abaixo tem-se o diagrama de sequência sobre a geração de relatórios:

Figura 11: Diagrama de sequência-gerar relatório



Fonte: Própria autoria, 2023

11 Modelagem de estados de atividade

Os diagramas de atividades são uma técnica para descrever lógica de procedimento, processo de negócio e fluxo de trabalho. De várias formas eles desempenham um papel semelhante aos fluxogramas, mas a principal diferença

entre eles e a notação de fluxograma é que os diagramas suportam comportamento paralelo.

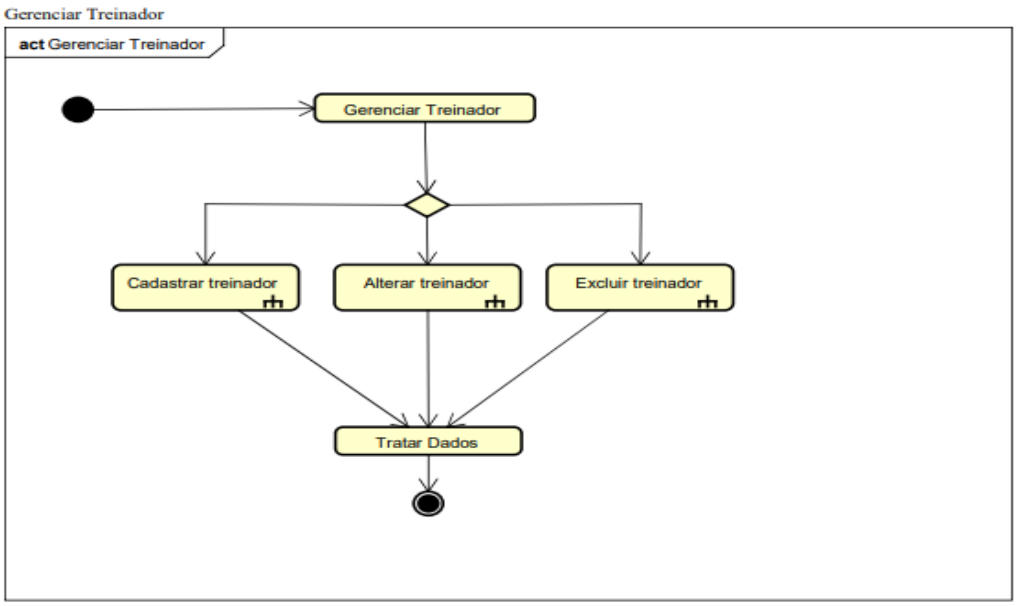
O diagrama de atividades permite que quem está seguindo o processo escolha a ordem na qual fazer as coisas, em outras palavras, ele simplesmente determina as regras essenciais de sequência que se deve seguir. Isso é importante para a modelagem de negócios, pois os processos frequentemente ocorrem em paralelo. Isso também é útil para algoritmos concorrentes, nos quais linhas de execução independentes podem fazer coisas em paralelo.

As ações podem ser decompostas em subatividades. Pode-se pegar a lógica de entrega e representar a granularidade das atividades no diagrama de atividade através de sub atividades. As sub-atividades são usadas para descrever passos mais detalhados dentro de uma atividade principal. Isso ajuda a decompor a lógica de uma tarefa complexa em partes mais gerenciáveis e compreensíveis.

No caso do diagrama de atividade para gerenciar treinadores, cada ação pode ser decomposta em sub-atividades para refletir os passos específicos envolvidos. Isso torna o diagrama mais detalhado e fácil de entender.

Na próxima página ilustra-se o diagrama de atividade sobre o gerenciamento de treinador:

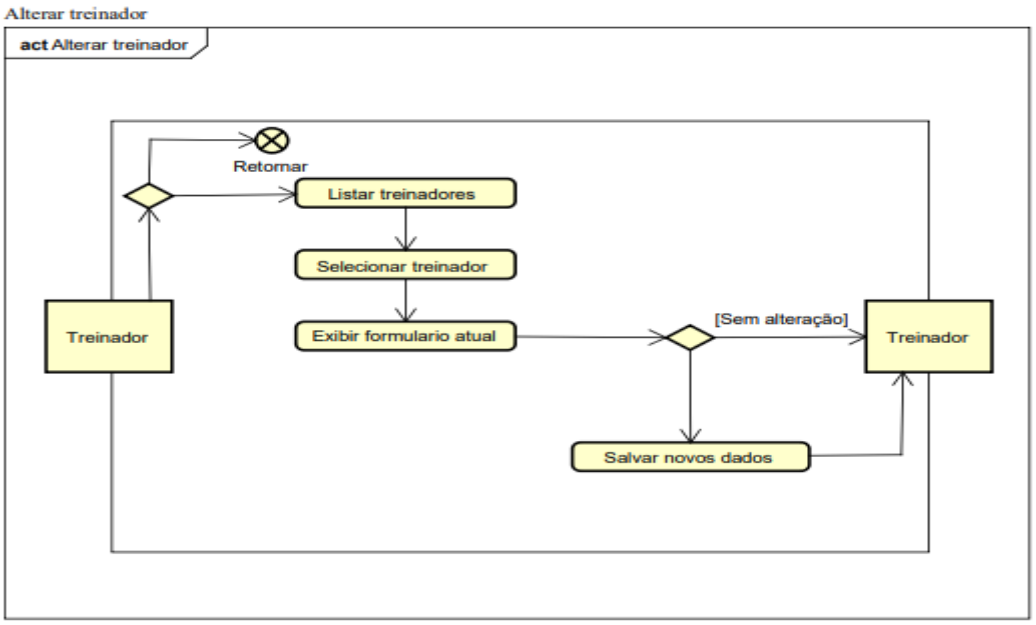
Figura 12: Diagrama de atividade-gerenciar treinador



Fonte: Autoria própria, 2023

Podemos dividir as sub-rotinas e assim absorver melhor os passos específicos daquela rotina. Vejamos isso no exemplo de alterar treinador.

Figura 13: Diagrama de subatividade - alterar treinador



Fonte. Autoria própria

Temos nos nossos diagramas vários exemplos como esse, usando com o interesse de ser o mais claro possível quanto às funcionalidades que o sistema deva apresentar.

12 Implementação

A versão atual do projeto consiste na utilização da versão 17 do Java, IDE Eclipse e a biblioteca JotaOpine.

A implementação de um projeto em Java 17 utilizando a IDE Eclipse e a biblioteca JotaOpine oferece uma abordagem robusta e eficiente para o desenvolvimento de aplicativos Java. A combinação do poder da linguagem Java 17, as ferramentas avançadas fornecidas pela IDE Eclipse e a funcionalidade adicional fornecida pela biblioteca JotaOpine resulta em uma solução escalável e de alta qualidade.

12.1 Configuração do Ambiente de Desenvolvimento

Antes de iniciar o projeto, é crucial configurar um ambiente de desenvolvimento adequado. Certifique-se de ter o Java 17 JDK instalado em sua máquina e configure o Eclipse para utilizar essa versão.

12.2 Criação do Projeto no Eclipse:

Utilize o Eclipse para criar um novo projeto Java. Escolha a opção "Java Project" e configure-o para utilizar a versão 17 do Java. Organize a estrutura do projeto conforme as melhores práticas, dividindo-o em pacotes lógicos e criando classes para representar os diferentes componentes.

12.3 Integração da Biblioteca JotaOpine:

A biblioteca JotaOpine oferece recursos valiosos para a implementação de opiniões e avaliações em um aplicativo. Integre a biblioteca ao seu projeto, adicionando-a ao classpath. Isso pode ser feito manualmente ou utilizando ferramentas de gerenciamento de dependências como o Maven ou Gradle

13 Conclusão

Este projeto de sistema de gerenciamento de clubes de futebol foi desenvolvido com base em práticas consolidadas de gestão esportiva e em princípios de engenharia de software, visando atender às demandas crescentes no cenário esportivo, proporcionando uma ferramenta robusta e intuitiva para a gestão eficiente de clubes. Com foco em requisitos funcionais, não funcionais e regras de negócio, o sistema abrange desde o cadastro de clubes, jogadores e treinadores até a gestão de transferências e a visualização de relatórios financeiros. O compromisso com a clareza da interface, tempos de resposta rápidos e boas práticas de programação reflete a busca pela excelência e usabilidade.

Inspirados por conceitos amplamente reconhecidos na área de gestão esportiva, como os discutidos por renomados especialistas como David K. Stotlar (2005), buscamos estabelecer um conjunto robusto de funcionalidades que atendam às complexas demandas do cenário esportivo contemporâneo.

Ao longo do processo de desenvolvimento, orientamo-nos pelos princípios de Ian Sommerville em engenharia de software, visando não apenas atender aos

requisitos funcionais específicos, mas também garantir que o sistema seja altamente responsivo, claro em sua interface e mantenha uma estrutura de código sustentável para facilitar futuras expansões e manutenções.

A integração de conceitos fundamentais derivados das melhores práticas em gestão esportiva e engenharia de software proporcionou uma base sólida para a construção deste sistema. Esse esforço visa não apenas oferecer um instrumento de gerenciamento eficaz para clubes de futebol, mas também contribuir para a evolução contínua dos métodos de gestão e tecnologias aplicadas ao universo esportivo.

Neste cenário, encaramos este projeto como uma tentativa de aplicar teorias consolidadas de gestão esportiva, adaptando-as de maneira sinérgica com as práticas modernas de engenharia de software. A abordagem meticulosa e a consideração cuidadosa das necessidades específicas do ambiente esportivo refletem o compromisso com a excelência e a inovação na interseção entre esporte e tecnologia. Este trabalho almeja contribuir para a contínua evolução e aprimoramento dos sistemas de gerenciamento esportivo no contexto dinâmico e desafiador do esporte profissional moderno.

14 Reconhecimentos e direitos autorais

Autores: Antônio Lister, Gabriel Fernando, Kely Souza, Lucas Teixeira, Victor Coelho.

contato:lister.wd@hotmail.com;gabrielfernandocarneiro798@gmail.com;
kely.souza@discente.ufma.br;lucas.teixeira@discente.ufma.br;
victor.coelho@discente.ufma.br.

data última versão: 04/12/2023

versão: 1.0

outros repositórios: https://github.com/kelyazuos?tab=repositories

Agradecimentos: Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO VIOLAÇÃO. EM HIPÓTESE ALGUMA OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM UMA AÇÃO DE CONTRATO, ATO ILÍCITO OU DE OUTRA FORMA, DECORRENTE DE, OU EM CONEXÃO COM O MATERIAL OU O USO OU OUTRAS NEGOCIAÇÕES NO MATERIAL.

Referências

BOOCH, Grady , RUMBAUGH, James, JACOBSON, Ivar; UML: Guia do Usuário. 2. ed. Campus, 2006

Roger Pressman - Engenharia de Software - MacGraw Hill, 2006

SOMMERVILLE, I. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STOTLAR, David K DUALIB, Carla. Como desenvolver planos de marketing esportivo de sucesso. São Paulo: Idéia e Ação, 2005