|  |  |
| --- | --- |
|  |  |
|  |  |

**Licenciatura em E.C.G.M.**

Escola Superior de Tecnologia e Gestão

*Instituto Politécnico de Viana do Castelo*

Unidade curricular de

Engenharia de Software

Relatório de projeto (Trabalho Prático)

2023/2024

*Gestão de Campeonato de Futebol*

25968 João Paulo Martins Novo

Sumário

[1. Introdução 1](#_Toc169563654)

[2. Apresentação do negócio 2](#_Toc169563655)

[2.1 Modelação dos Processos de negócio 2](#_Toc169563656)

[2.1.1 Processo 1 – Gestão de Clubes 3](#_Toc169563657)

[2.1.2 Processo 2 – Gestão de Equipas 4](#_Toc169563658)

[2.1.3 Processo 3 – Gestão de Jogadores 5](#_Toc169563659)

[2.1.4 Processo 4 – Gestão de Jogos 6](#_Toc169563660)

[3. Levantamento de Requisitos 7](#_Toc169563661)

[3.1 Requisitos Funcionais 7](#_Toc169563662)

[3.2 Requisitos não funcionais 7](#_Toc169563663)

[4. Design e Modelação 8](#_Toc169563664)

[4.1 Modelo de casos de uso 8](#_Toc169563665)

[4.2 Modelo de classes do Domínio 12](#_Toc169563666)

[4.3 Diagramas de transição de estados 14](#_Toc169563667)

[5. Desenho e Implementação da BD 15](#_Toc169563668)

[5.1 Modelação de dados 15](#_Toc169563669)

[5.2 Modelo Físico da Base de dados 17](#_Toc169563670)

[6. Arquitetura e Design da Aplicação 18](#_Toc169563671)

[6.1. Arquitetura e Tecnologias utilizadas 18](#_Toc169563672)

[6.2. Principais características da Aplicação 18](#_Toc169563673)

[7. Conclusão e Trabalho Futuro 18](#_Toc169563674)

# Introdução

Neste relatório, apresentamos o resultado do trabalho prático desenvolvido no âmbito da unidade curricular de Engenharia de Software. O objetivo principal consiste na conceção e implementação de um sistema de software dedicado à gestão de um campeonato de futebol. Ao longo deste documento, descreveremos os passos seguidos no processo de desenvolvimento, desde a análise dos requisitos até aos modelos do sistema desenvolvidos.

# Apresentação do negócio

O sistema que estamos a propor vai ajudar na gestão do campeonato de futebol, vai permitir adicionar, editar e eliminar clubes, que terá equipa, também será possível fazer o mesmo com os jogadores de cada equipa. Assim, teremos toda a informação organizada sobre quem joga em cada equipa. Além disso, vamos integrar a função de adicionar árbitros e estádios para cada jogo. Isso vai permitir que os responsáveis atribuam árbitros e estádios aos jogos e, se necessário, façam alterações. Essas funcionalidades vão facilitar a organização do campeonato, desde a formação das equipas até a gestão dos jogos.

## Modelação dos Processos de negócio

* Gestão de Clubes
* Gestão de Equipas
* Gestão de Jogadores
* Gestão de Jogos

Os processos vão ser naturalmente similares devido a sua natureza de serem operações CRUD (create-read-update-delete).

## Processo 1 – Gestão de Clubes

* + Início: O processo começa quando é necessário criar, editar ou eliminar uma equipa.
  + Ações:  
    - Criação de Clube: Um administrador insere o nome do Clube, do estádio e do Presidente.
    - Edição de Clube: Um administrador pode alterar o nome, o estádio e o nome do Presidente.
    - Eliminação de Equipa: Um administrador remove um club do campeonato.
  + Fim: O processo termina quando a operação de gestão de equipas é concluída com sucesso.

*Uma imagem com texto, diagrama, captura de ecrã, Paralelo

Descrição gerada automaticamente*

*Diagrama do Processo de Gestão de Clubes*

## Processo 2 – Gestão de Equipas

* + Início: O processo começa quando é necessário criar, editar ou eliminar uma equipa.
  + Ações:  
    - Criação de Equipa: Um administrador insere o nome da equipa ao clube que pertence e também os jogadores á equipa.
    - Edição de Equipa: Um administrador pode alterar o nome da equipa e os jogadores que estão na equipa
    - Eliminação de Equipa: Um administrador remove uma equipa do clube.
  + Fim: O processo termina quando a operação de gestão de equipas é concluída com sucesso.

Uma imagem com texto, diagrama, captura de ecrã, Paralelo

Descrição gerada automaticamente  
*Diagrama do Processo de Gestão de Equipa*

## Processo 3 – Gestão de Jogadores

* + Início: O processo começa quando é necessário adicionar, editar ou remover um jogador de uma equipa.
  + Ações:  
    - Adição de Jogador: Um administrador insere as informações do jogador, como nome, posição e estatísticas.
    - Edição de Jogador: Um administrador pode alterar as informações de um jogador existente.
    - Remoção de Jogador: Um administrador remove um jogador da equipa.
  + Fim: O processo termina quando a operação de gestão de jogadores é concluída com sucesso.

*Uma imagem com texto, diagrama, captura de ecrã, Esquema

Descrição gerada automaticamente  
Diagrama do Processo de Gestão de Jogador*

## Processo 4 – Gestão de Jogos

* + Início: O processo começa quando é necessário agendar, designar árbitros ou atribuir estádios a um jogo.
  + Ações:  
    - Agendamento de Jogo: Um administrador define a data, hora e local do jogo.
    - Designação de Árbitros: Um administrador seleciona os árbitros para o jogo.
    - Atribuição de Estádios: Um administrador aloca um estádio para a realização do jogo.
  + Fim: O processo termina quando todas as operações relacionadas à gestão da partida são concluídas com sucesso.

Uma imagem com texto, diagrama, captura de ecrã, Esquema

Descrição gerada automaticamente  
*Diagrama do Processo de Gestão de Jogo*

# Levantamento de Requisitos

## Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| Número | Requisito Funcional | Nível de Prioridade |
| RF1 | Como administrador, quero poder criar clubes para registar no sistema. | Alta Prioridade |
| RF2 | Como administrador, quero ter a capacidade de editar informações dos clubes, como nome, estádio e presidente. | Alta Prioridade |
| RF3 | Como administrador, quero poder eliminar clubes existentes do sistema | Alta Prioridade |
| RF4 | Como utilizador, quero visualizar informações sobre os clubes, como nome, estádio e presidente | Alta Prioridade |
| RF5 | Como administrador, quero poder criar equipas associadas a um clube existente. | Alta Prioridade |
| RF6 | Como administrador, quero ter a capacidade de editar informações das equipas, como nome e jogadores. | Alta Prioridade |
| RF7 | Como administrador, quero poder eliminar equipas existentes do sistema. | Alta Prioridade |
| RF8 | Como utilizador, quero visualizar informações sobre as equipas, como nome e jogadores. | Alta Prioridade |
| RF9 | Como administrador, quero poder adicionar novos jogadores a uma equipa existente. | Alta Prioridade |
| RF10 | Como administrador, quero ter a capacidade de editar informações dos jogadores, como nome e posição. | Alta Prioridade |
| RF11 | Como administrador, quero poder eliminar jogadores existentes de uma equipa. | Alta Prioridade |
| RF12 | Como utilizador, quero visualizar informações sobre os jogadores, como nome e posição. | Alta Prioridade |
| RF13 | Como administrador, quero poder criar jogos, especificando a data, hora e local. | Alta Prioridade |
| RF14 | Como administrador, quero ter a capacidade de designar árbitros para os jogos agendados. | Alta Prioridade |
| RF15 | Como administrador, quero poder atribuir estádios para os jogos agendados. | Alta Prioridade |
| RF16 | Como utilizador, quero visualizar informações sobre os jogos agendados, como data, hora, equipas e estádio. | Alta Prioridade |

## Requisitos não funcionais

|  |  |
| --- | --- |
| Número | Requisito Não Funcional |
| RNF1 | O sistema deve permitir que haja diferentes tipos de Users (Admin/User). |
| RNF2 | O sistema tem de ter uma interface simples e intuitiva. |

# Design e Modelação

## A diagram of a person's diagram Modelo de casos de uso

|  |  |
| --- | --- |
| **Use Case ID:** | *UC1* |
| **Use Case Name:** | Gestão de Jogos |
| **Relevant requirements:** | * O sistema deve permitir que o administrador crie listagens de jogos. * O sistema deve validar os detalhes do jogo inseridos pelo administrador. * O sistema deve guardar os novos detalhes do jogo na base de dados. |
| **Primary actor:** | Administrador |
| **Pre-conditions:** | * O administrador deve estar logado no sistema. * O administrador deve ter privilégios para criar listagens de jogos. |
| **Post-conditions:** | * Uma nova listagem de jogo é criada no sistema. * O administrador recebe uma confirmação de que o jogo foi criado com sucesso. |
| **Basic Flow or Main Scenario:** | 1. O administrador entra no sistema 2. O administrador navega até à secção "Criar Jogo". 3. O administrador introduz todos os detalhes necessários do jogo. 4. O sistema valida os detalhes introduzidos para garantir a completude e correção. 5. O administrador submete o pedido de criação do jogo. 6. O sistema guarda o novo jogo na base de dados. 7. O sistema exibe uma mensagem de confirmação ao administrador de que o jogo foi criado com sucesso. |
| **Extensions or Alternate Flows:** | 4a. Se a validação falhar.   1. O sistema exibe uma mensagem de erro. 2. O administrador corrige os erros e submete novamente. 3. Retorna ao passo 4. |
| **Exceptions:** | Se houver uma falha na base de dados durante a submissão:   1. O sistema exibe uma mensagem de erro indicando que o jogo não pôde ser criado. 2. O administrador é aconselhado a tentar novamente mais tarde. |

A diagram of a person's work flow

Description automatically generated

|  |  |
| --- | --- |
| **Use Case ID:** | *UC2* |
| **Use Case Name:** | Gestão de Jogadores |
| **Relevant requirements:** | * O sistema deve permitir que o administrador crie perfis de jogadores. * O sistema deve validar os detalhes do jogador inseridos pelo administrador. * O sistema deve guardar os novos perfis de jogador na base de dados. |
| **Primary actor:** | Administrador |
| **Pre-conditions:** | * O administrador deve estar logado no sistema. * O administrador deve ter privilégios para criar perfis de jogadores. |
| **Post-conditions:** | * Um novo perfil de jogador é criado no sistema. * O administrador recebe uma confirmação de que o jogador foi criado com sucesso. |
| **Basic Flow or Main Scenario:** | 1. O administrador entra no sistema. 2. O administrador navega até à secção "Criar Jogador". 3. O administrador introduz todos os detalhes necessários do jogador. 4. O sistema valida os detalhes introduzidos quanto à sua completude e correção. 5. O administrador submete o pedido de criação do jogador. 6. O sistema guarda o novo jogador na base de dados. 7. O sistema exibe uma mensagem de confirmação de que o jogador foi criado com sucesso. |
| **Extensions or Alternate Flows:** | 4a. Se a validação falhar:   1. O sistema exibe uma mensagem de erro. 2. O administrador corrige os erros e submete novamente. 3. Retorna ao passo 4. |
| **Exceptions:** | Se houver uma falha na base de dados durante a submissão:   1. sistema exibe uma mensagem de erro indicando que o jogo não pôde ser criado. 2. O administrador é aconselhado a tentar novamente mais tarde. |

A diagram of a person's work flow

Description automatically generated

|  |  |
| --- | --- |
| **Use Case ID:** | *UC3* |
| **Use Case Name:** | Gestão de Equipas |
| **Relevant requirements:** | * O sistema deve permitir que o administrador crie perfis de equipas. * O sistema deve validar os detalhes da equipa inseridos. * O sistema deve guardar os novos perfis de equipa na base de dados. |
| **Primary actor:** | Administrador |
| **Pre-conditions:** | * O administrador deve estar logado no sistema. * O administrador deve ter privilégios para criar perfis de equipas. |
| **Post-conditions:** | * Um novo perfil de equipa é criado no sistema. * O administrador recebe uma confirmação de que a equipa foi criada com sucesso. |
| **Basic Flow or Main Scenario:** | 1. O administrador entra no sistema. 2. O administrador navega até à secção "Criar Equipa". 3. O administrador introduz todos os detalhes necessários da equipa. 4. O sistema valida os detalhes introduzidos. 5. O administrador submete o pedido de criação da equipa. 6. O sistema guarda a nova equipa na base de dados. 7. O sistema exibe uma mensagem de confirmação de que a equipa foi criada com sucesso. |
| **Extensions or Alternate Flows:** | 4a. Se a validação falhar:   1. sistema exibe uma mensagem de erro. 2. administrador corrige os erros e submete novamente. 3. Retorna ao passo 4. |
| **Exceptions:** | Se houver uma falha na base de dados durante a submissão:   1. sistema exibe uma mensagem de erro indicando que o jogo não pôde ser criado. 2. O administrador é aconselhado a tentar novamente mais tarde. |

A diagram of a person's diagram

Description automatically generated

|  |  |
| --- | --- |
| **Use Case ID:** | *UC4* |
| **Use Case Name:** | Gestão de Clubes |
| **Relevant requirements:** | * O sistema deve permitir que o administrador crie perfis de clubes. * O sistema deve validar os detalhes do clube inseridos. * O sistema deve guardar os novos perfis de clube na base de dados. |
| **Primary actor:** | Administrador |
| **Pre-conditions:** | * O administrador deve estar logado no sistema. * O administrador deve ter privilégios para criar listagens de clubes. |
| **Post-conditions:** | * Um novo perfil de clube é criado no sistema. * O administrador recebe uma confirmação de que o jogo foi criado com sucesso. |
| **Basic Flow or Main Scenario:** | 1. O administrador entra no sistema. 2. O administrador navega até à secção "Criar Clube". 3. O administrador introduz todos os detalhes necessários do clube. 4. O sistema valida os detalhes introduzidos. 5. O administrador submete o pedido de criação do clube. 6. O sistema guarda o novo clube na base de dados. 7. O sistema exibe uma mensagem de confirmação de que o clube foi criado com sucesso. |
| **Extensions or Alternate Flows:** | 4a. Se a validação falhar:   1. sistema exibe uma mensagem de erro. 2. administrador corrige os erros e submete novamente. 3. Retorna ao passo 4. |
| **Exceptions:** | Se houver uma falha na base de dados durante a submissão:   1. sistema exibe uma mensagem de erro indicando que o jogo não pôde ser criado. 2. O administrador é aconselhado a tentar novamente mais tarde. |

## Modelo de classes do Domínio

*Uma imagem com texto, diagrama, Paralelo, Esquema

Descrição gerada automaticamente*

* Classes e Atributos  
  + Classe Estádio
    - Atributos: id\_estadio, nome, localização.
    - Descrição: Representa os estádios onde os jogos são realizados,   
      cada um com um identificador único, nome e localização específica.
  + Classe Clube
    - Atributos: id\_clube, nome, id\_estadio, presidente
    - Descrição: Lista os clubes participantes no campeonato, onde cada clube está associado a um estádio e tem um nome e um presidente.
  + Classe Arbitro
    - Atributos: id\_arbitro, nome, idade
    - Descrição: Representa os árbitros, onde tem o seu identificador único, nome e idade.
  + Classe Equipa
    - Atributos: id\_equipa, nome, id\_clube
    - Descrição: Lista as equipas que competem no campeonato, onde cada equipa tem o seu nome e clube.
  + Classe Jogador
    - Atributos: id\_jogador, nome, idade, id\_equipa
    - Descrição: Representa os jogadores, onde tem o seu identificador único, nome e idade e a equipa que representam.
  + Classe Jogo
    - Atributos: id\_jogo, data, hora, id\_estadio, id\_arbitro
    - Descrição: Representa os jogos agendados no campeonato, especificando a data, hora, o estádio e o árbitro do jogo.
* Relações entre as Classes  
  + Estádio e Clube: Cada clube está associado a um único estádio.
  + Clube e Equipa: Um clube pode ter várias.
  + Equipa e Jogador: Cada equipa inclui diversos jogadores.
  + Jogo, Arbitro e Estádio: Cada jogo ocorre em um estádio específico e é arbitrado por um árbitro específico.
  + Jogo e Equipa: Cada jogo relaciona diretamente duas equipas específicas.

## Diagramas de transição de estados

*Diagrama de transição do estado do Jogo*

*Uma imagem com diagrama, Esquema, texto, Desenho técnico

Descrição gerada automaticamente*

# Desenho e Implementação da BD

## Modelação de dados

Uma imagem com texto, diagrama

Descrição gerada automaticamenteEste é o modelo E-R construído para a base de Dados.

*Modelo E-R*

Este é o modelo relacional da Base de Dados

*Uma imagem com texto, diagrama, Paralelo, número

Descrição gerada automaticamente*

*Modelo Relacional da Base de Dados*

Depois normalizamos o modelo relacional segundo as 3 Formas Normais:

* Primeira Forma Normal (1NF):  
  + Cumprida - Todos os atributos têm valores atômicos.
* Segunda Forma Normal (2NF):  
  + Cumprida - Cada tabela tem uma chave primária que a identifica de forma única a cada registo, e todos os atributos não-chave dependem da totalidade da chave primária.
* Terceira Forma Normal (3NF):  
  + Cumprida - Todos os campos não-chave dependem diretamente das chaves primárias, sem dependências transitivas.

*Mas devido ao facto de apenas existir duas equipas que disputam um jogo, não é a melhor abordagem para lidar com a relação.*

*Uma imagem com texto, diagrama, Paralelo, número

Descrição gerada automaticamente*

*Modelo Relacional “Normalizado”*

## Modelo Físico da Base de dados

Inicialmente, a base de dados foi configurada utilizando o MVC com contas individuais. Este processo gerou automaticamente uma base de dados, um db \_context que serve como meu ORM (Object-Relational Mapping). Posteriormente, foram criados os Models das classes, nos quais adicionei anotações de dados. Estas anotações envolvem a utilização de keywords que são colocadas acima das variáveis para definir as propriedades do banco de dados, como restrições e tipos de dados. Para integrar as modificações no banco de dados, utilizei o comando 'add-migrations'. Finalizei este processo utilizando o comando 'update-database', que analisa o script gerado e atualiza a base de dados de acordo com os Models definidos.

Depois peguei no código gerado e criei igual no SQL Server.

*Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente  
Print do SQL Server da Base de Dados*

As tabelas AspNet(…) são as geradas automaticamente ao criar o projeto MVC com contas individuais.

# Arquitetura e Design da Aplicação

## 6.1. Arquitetura e Tecnologias utilizadas

Optei por utilizar o ASP.NET Core MVC, um framework fornecido pela Microsoft, que permite a criação de aplicações. A implementação do sistema com contas individuais foi realizada através do Identity Framework, integrado ao ASP.NET Core.

## 6.2. Principais características da Aplicação

A principal característica da aplicação é a sua abrangência, possuindo contas para utilizadores e administradores. Inclui todas as funcionalidades CRUD necessárias, além de uma página pública que fornece todas as informações essenciais para que os utilizadores possam acessar.

# Conclusão e Trabalho Futuro

Este relatório detalhou o processo de desenvolvimento de um sistema de gestão de campeonatos de futebol, desde a conceção inicial até a modelação da base de dados. O sistema foi desenhado para a administração de clubes, equipes, jogadores e jogos, com capacidades de criar, editar e elimina, garantindo uma gestão do campeonato. Através da implementação de requisitos funcionais e não funcionais, o projeto garante usabilidade e que seja operacional.

Concluindo esta segunda etapa do projeto, posso afirmar que foi um período desafiador, principalmente pela necessidade de me adaptar e dominar uma nova tecnologia. Apesar das dificuldades iniciais, encontrei um tutorial que realmente me ajudou a entender e realizar este trabalho prático.