

DHIEGO COLETTI CAMOLESI

SISTEMA DE SÚMULA ELETRÔNICA

ASSIS
2009

SISTEMA DE SÚMULA ELETRÔNICA

DHIEGO COLETTI CAMOLESI

Trabalho de Conclusão de Curso apresentado ao Instituto municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: _____

Analisador (1): _____

Analisador (2): _____

DHIEGO COLETTTO CAMOLESI

SISTEMA DE SÚMULA ELETRONICA

Trabalho de Conclusão de Curso apresentado ao Instituto municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador:

Área de concentração:

**Assis
2009**

RESUMO

Neste trabalho, será desenvolvido um software direcionado para ligas de futebol, tanto amadoras como profissionais. Destaca-se a necessidade de informatizar a tecnologia ligada à área de sistemas de informação na execução de um jogo, pois até então o controle é manual.

Este sistema será de fundamental importância para auxiliar a comissão de arbitragem e a federação no total controle de todos os dados e movimentações ocorrentes no campeonato. A linguagem escolhida para o desenvolvimento do software foi o Borland delphi 7.

ABSTRACT

In this work, developed of a software will be directed to football leagues, both amateur and professional. It is the need to computerize the technology linked to the area of information systems in the execution of a game, because until then the control is manual.

This system will be of fundamental importance to help the arbitration committee and the federation in total control of all data and movements occurring in the championship. The language chosen for the development of software was the Borland Delphi 7.

LISTA DE FIGURAS

Figura 1 Exemplo de Atores	5
Figura 2 Exemplo de Use Case	6
Figura 3 Relacionamento entre atores e caso de uso	6
Figura 4 Relacionamento entre atores	6
Figura 5 Diagrama de Sequência	7
Figura 6 Exemplo de classe	8
Figura 7 Ambiente de desenvolvimento Borland Delphi 7	9
Figura 8 Object Inspector	10
Figura 9 Form	10
Figura 10 Code Editor	11
Figura 11 Navicat de MySql	12
Figura 12 Tela Inicial	47
Figura 13 Tela Inicial - Cadastro	48
Figura 14 Cadastro de Jogadores	48
Figura 15 Súmula	49
Figura 16 Súmula CBF	50

SUMÁRIO

1. INTRODUÇÃO	1
2. PROCESSO DE PRODUÇÃO	2
2.1. RECURSOS	2
3. INTRODUÇÃO A UML	3
3.1 VISÕES	4
3.2 MODELOS DE ELEMENTOS	4
3.3 MECANISMOS GERAIS	4
3.4 DIAGRAMAS	4
3.4.1 Diagrama Use case (Caso de Uso)	5
3.4.2 Diagramas de Seqüência	7
3.4.3 Diagrama de Classes	8
9. BORLAND DELPHI 7	9
4.1 OBJECT INSPECTOR	10
4.2 FORM (FORMULÁRIO)	10
4.3 CODE EDITOR (EDITOR DE CÓDIGO)	11
5. BANCO DE DADOS MYSQL	12
6. PROJETO	13
6.1 LISTA DE EVENTOS	13
6.2. RELAÇÃO DOS ATORES	14
6.2.1 Federação	14
6.2.2 Comissão de Arbitragem	14
6.3. USE CASES	15
6.3.1 USE CASE CADASTRAR CLUBES	15
6.3.2 USE CASE CADASTRAR JOGADORES	16
6.3.3 USE CASE CADASTRAR ÁRBITROS	17
6.3.4 USE CASE CADASTRAR DIVISÕES	18
6.3.5 USE CASE CADASTRAR GRUPOS	19
6.3.6 USE CASE CADASTRAR ESTÁDIOS	20
6.3.7 USE CASE LANÇAR SÚMULA	21
6.3.8 USE CASE LANÇAR JOGADOR SÚMULA	23

6.3.9 USE CASE LANÇAR GOLS.....	24
6.3.10 USE CASE LANÇAR SUSPENSÃO.....	25
6.3.11 USE CASE EMITIR RELATÓRIO DE CLASSIFICAÇÃO.....	26
6.3.12 USE CASE EMITIR RELATÓRIO DE SALDO DE GOLS.....	27
6.3.13 USE CASE EMITIR RELATÓRIO DE GOLS MARCADOS.....	28
6.3.14 USE CASE EMITIR RELATÓRIO DE GOLS SOFRIDOS.....	29
6.3.15 USE CASE EMITIR RELATÓRIO DE ARTILHEIRO.....	30
7. DIAGRAMA DE SEQUÊNCIA.....	31
7.1 DIAGRAMA DE SEQUÊNCIA INCLUIR CLUBES.....	31
7.2 DIAGRAMA DE SEQUÊNCIA ALTERAR CLUBES.....	31
7.3 DIAGRAMA DE SEQUÊNCIA EXCLUIR CLUBES.....	32
7.4 DIAGRAMA DE SEQUÊNCIA PESQUISAR CLUBES.....	32
7.5 DIAGRAMA DE SEQUÊNCIA INCLUIR ÁRBITROS.....	33
7.6 DIAGRAMA DE SEQUÊNCIA ALTERAR ÁRBITROS.....	33
7.7 DIAGRAMA DE SEQUENCIA EXCLUIR ÁRBITROS.....	34
7.8 DIAGRAMA DE SEQUENCIA PESQUISAR ÁRBITROS.....	34
7.9 DIAGRAMA DE SEQUENCIA INCLUIR DIVISÃO.....	35
7.10 DIAGRAMA DE SEQUENCIA ALTERAR DIVISÃO.....	35
7.11 DIAGRAMA DE SEQUENCIA EXCLUIR DIVISÃO.....	36
7.12 DIAGRAMA DE SEQUENCIA PESQUISAR DIVISÃO.....	36
7.13 DIAGRAMA DE SEQUENCIA INCLUIR GRUPOS.....	37
7.14 DIAGRAMA DE SEQUENCIA ALTERAR GRUPOS.....	37
7.15 DIAGRAMA DE SEQUENCIA EXCLUIR GRUPOS.....	38
7.16 DIAGRAMA DE SEQUENCIA PESQUISAR GRUPOS.....	38
7.17 DIAGRAMA DE SEQUENCIA INCLUIR JOGADORES.....	39
7.18 DIAGRAMA DE SEQUENCIA ALTERAR JOGADORES.....	39
7.19 DIAGRAMA DE SEQUENCIA EXCLUIR JOGADORES.....	40
7.20 DIAGRAMA DE SEQUENCIA PESQUISAR JOGADORES.....	40
7.21 DIAGRAMA DE SEQUENCIA INCLUIR ESTÁDIOS.....	41
7.22 DIAGRAMA DE SEQUENCIA ALTERAR ESTÁDIOS.....	41
7.23 DIAGRAMA DE SEQUENCIA EXCLUIR ESTÁDIOS.....	42

7.24 DIAGRAMA DE SEQUENCIA PESQUISAR ESTÁDIOS	42
7.25 DIAGRAMA DE SEQUENCIA DE RELATÓRIOS	43
7.26 DIAGRAMA DE SEQUENCIA JOGADOR SÚMULA.....	44
7.27 DIAGRAMA DE SEQUENCIA DE SÚMULA.....	45
8. DIAGRAMA DE CLASSES	46
9. IMPLEMENTAÇÃO	47
9.1 TELAS INICIAL	47
9.2 CADASTRO	48
9.3 SÚMULA.....	49
17. SÚMULA CONVENCIONAL.....	50
18. CONCLUSÃO	51
19. REFERÊNCIAS BIBLIOGRÁFICAS	52

1. INTRODUÇÃO

Assistindo jogos de futebol, é possível perceber que a tecnologia ligada à área de sistemas de informação não é contemplada na execução de um jogo. Quando o jogo termina, todos os dados são fichados em uma súmula onde será enviada para a federação ter o controle de tudo o que aconteceu na partida.

Um software de súmula eletrônico pode tornar mais prático e eficiente o trabalho da comissão de arbitragem, sendo que todas as ocorrências das partidas podem ser digitalizadas, automatizando assim, todo o processo que até então era manual.

Este software tem como objetivo melhorar o processo de trabalho da comissão de arbitragem e o armazenamento dos dados das partidas de determinado campeonato. Assim, todos os dados recorrentes das partidas armazenados no banco de dados podem ser disponibilizados para técnicos que podem acompanhar o que ocorre nos jogos dos adversários, a imprensa e torcedores.

Com a dificuldade do controle manual de súmulas em partidas de futebol, muitas vezes, ocasionando erros no processo de armazenamento, perdendo informações importantes. Surgiu a necessidade de informatizar o sistema que controla todos os dados das partidas, assim, com a tecnologia existente, podemos armazenar e disponibilizar para terceiros todas as informações imersas em um campeonato de futebol.

Como o futebol é uma paixão nacional, em diversas cidades do país existem ligas e federações que organizam campeonatos amadores. As ligas e as federações podem ser consideradas um grande nicho de mercado, pois grande parte destas, não possui tal tecnologia.

Pensando no sucesso deste projeto, pode-se pensar em sua comercialização para a federação paulista de futebol e até quem sabe para a confederação brasileira, tornando-o um grande produto.

Portanto, este projeto tem como objetivo colaborar com a comissão de arbitragem, automatizando o sistema de súmula, ou seja, todos os dados da partida serão armazenados no banco de dados, tendo o total controle da federação.

2. PROCESSO DE PRODUÇÃO

Para a análise do projeto, será utilizada a Linguagem de Modelagem Unificada (UML), sendo assim, uma programação orientada a objeto. Para o desenvolvimento escolheu-se a linguagem de programação Delphi, pois, essa tecnologia nos permite uma fácil formatação de janelas de entrada de dados, propiciando ao usuário um simples entendimento dos mecanismos do sistema.

2.1. RECURSOS

- Software:

Windows XP como sistema operacional

Borland Delphi7 como linguagem de programação

MySQL como banco de dados

- Hardware:

Um microcomputador com processador Pentium 4, 2800 MHZ, 1 GByte de memória RAM.

3. INTRODUÇÃO A UML

“UML” abreviação de Linguagem de Modelagem Unificada (Unified Modeling Language). Criada por três desenvolvedores de sistemas Grady Booch, James Rumbaugh e Ivar Jacobson, a UML é uma linguagem de modelagem que auxilia na visualização do desenho e a comunicação entre os objetos na projeção de um sistema orientado a objeto.

O conceito da UML veio para padronizar as metodologias do desenvolvimento de sistemas orientados a objetos, baseando-se em diagramas que são modelados e classificados focando os aspectos essenciais do contexto.

A UML quando unida com um processo de produção de software torna-se um padrão para o desenvolvimento de sistemas. Um processo é dividido em fases e atividades, sendo existentes cinco fases no sistema de desenvolvimento de software.

- **Análise de requisitos**

Onde as necessidades do usuário serão levantadas.

- **Análise**

As classes do sistema serão definidas e descritas no diagrama de classes.

- **Design**

Implementação da análise, onde serão feitos os ajustes finais para começar a programação.

- **Programação**

Na programação o projeto “sai do papel” e começa ser desenvolvido em um código de linguagem Orientada a Objeto.

- **Teste**

Fase final, onde o sistema é testado e passara pelo teste de aceitação dos usuários.

A ordem de execução dessas fases não deve ser seguida a regra, mas sim de modo que, os problemas ao surgirem sejam solucionados em determinada fase sem interferir nas fases anteriores ou posteriores.

Além das cinco fases de desenvolvimento, existem quatro partes que compõem a UML, cada uma delas com seus componentes, onde esse conjunto especifica e exemplifica a definição de um sistema.

As partes que compõem a UML são: visões, modelos de elementos, mecanismos gerais e diagramas.

3.1. VISÕES

As visões representam as descrições do software que está sendo modelado. Como todas as informações necessárias para a descrição de um software são incapazes de serem representadas em um único diagrama, cada visão é descrita por abstrações, constituindo em uma série de diagramas.

3.2. MODELOS DE ELEMENTOS

São os conceitos utilizados na modelagem e construção dos diagramas. Utilizando programação orientada a objeto, suas notações são representadas nos modelos de elementos por objetos, classes, mensagens, pacotes, componentes, relacionamentos entre classes, dependências e herança.

3.3. MECANISMOS GERAIS

São usados os mecanismos gerais para anexar ou anotar informações adicionais aos diagramas. Para adicionar semântica aos elementos ou constatar as instâncias envolvidas na relação, ornamentos gráficos são anexados aos modelos de elementos. Outros mecanismos gerais são as notas, que são utilizadas para adicionar informações complementares ao diagrama.

3.4. DIAGRAMAS

Nos diagramas as abstrações do software são representadas graficamente, ou seja, todas as informações necessárias para sua descrição serão representadas por diagramas. Existem nove tipos de diagramas utilizados pela UML: diagrama de

use case, de classes, de objeto, de colaboração, de atividade, de estado, de componente, de execução e de seqüência.

Os diagramas facilitam a representação e visualização dos componentes do software, ajudando a compreender a arquitetura do projeto em desenvolvimento.

3.4.1. Diagrama Use case (Caso de Uso)

Para Jacobson (1992) apud LARMAN Craig (2000), página 68 “Um caso de uso é um documento narrativo que descreve a seqüência de eventos de um ator (um agente externo) que usa um sistema para completar um processo”. O modelo caso de uso funciona como um contrato entre o desenvolvedor e o cliente, nele o cliente pode constatar se todas as suas exigências e necessidades estão sendo aplicadas no software. Portanto, o diagrama use case auxilia a interação entre o analista e o cliente, definindo assim, em sua modelagem as ações e especificações que o sistema deve ser capaz de executar.

Os requisitos funcionais do software são representados no diagrama por atores, use cases e relacionamentos entre estes elementos.

Atores: São usuários do software, eles interagem com o sistema representando papéis desempenhados por usuários humanos ou sistemas computacionais. Sendo a representação de uma entidade externa, a interação dos atores com o sistema é feita através da troca de mensagens. Um ator é representado por um boneco ou por um retângulo (vide Figura 1), e as entidades externas que ele representa podem ser:

- Pessoas
- Dispositivos de saída
- Hardware
- Software



Figura 1: Exemplo de Atores.

Use cases: Representa as funcionalidades do software, descrevem o contexto mostrando classes/objetos, seus relacionamentos e sua interação. Uma use case é representada por uma elipse (vide Figura 2).

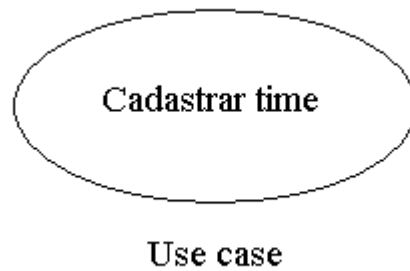


Figura 2: Exemplo de Use case.

Relacionamentos: São relações externas ao sistema que definem sua funcionalidade do ponto de vista do usuário. Exemplo de relacionamento entre atores e caso de uso (vide Figura 3) e relacionamento entre atores (vide Figura 4).

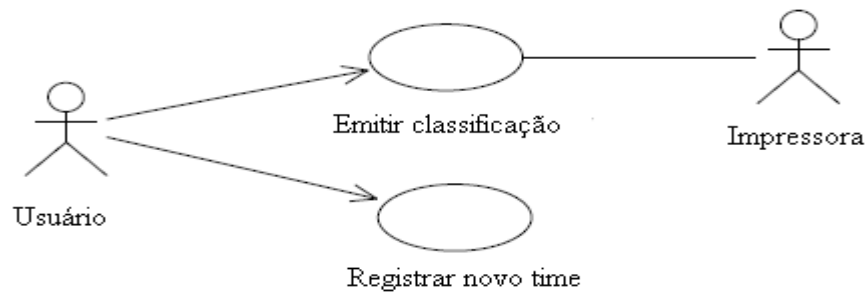


Figura 3: Relacionamento entre atores e caso de uso.



Figura 4: Relacionamento entre atores.

3.4.2. Diagramas de Sequência

Consiste em um diagrama que mostra a dinâmica dos objetos ao longo do tempo para a realização de determinada operação pelo software. O diagrama de sequências mostra através da interação entre os objetos, operações que ocorrerão no sistema em fases específicas do processo de execução.

Na representação de um diagrama de seqüências, a UML utiliza a seguinte notação:

- Linhas verticais tracejadas representam o tempo de vida dos objetos (lifeline);
- Retângulos com seus nomes sublinhados representam os objetos
- Setas horizontais rotuladas com as mensagens representam a interação entre os objetos, ou seja, as mensagens passadas entre eles.
- A posição vertical mostra a ordem relativa das mensagens no tempo, sendo assim, a primeira mensagem aparece no topo;
- Anotações podem ser utilizadas para complementar informação ao diagrama;

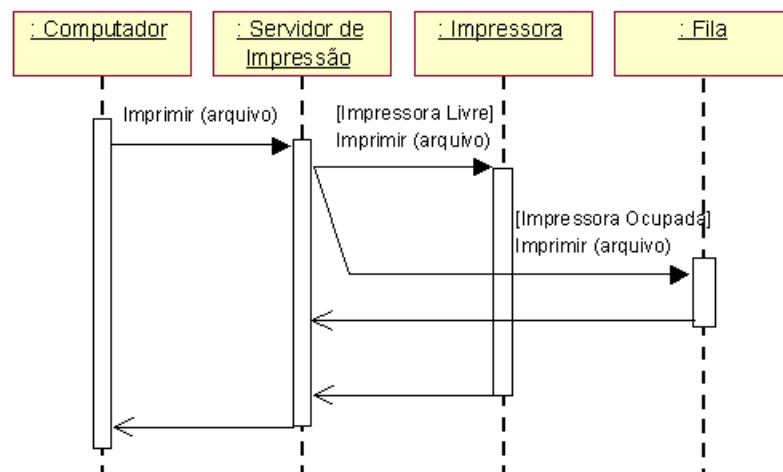


Figura 5: Diagrama de Sequência – Servidor de impressão.

3.4.3. Diagrama de Classes

O diagrama de classes demonstra as classes, ou seja, atributos e métodos de um conjunto de objetos que compõem o software e seus relacionamentos entre si. Por não mostrar a troca de mensagens entre as classes, somente os relacionamentos, ele é um diagrama “estático”, por essa razão, o diagrama pode ser utilizado em qualquer etapa do ciclo de vida do sistema.

Um diagrama de classe é composto por entidades e relacionamentos, cada entidade contendo classes que são representadas por um retângulo com seu nome no topo e abaixo os atributos e operações em dois compartimentos dentro do retângulo (vide Figura 6). Tendo as classes identificadas e descritas, o diagrama é composto pelo relacionamento entre as classes.

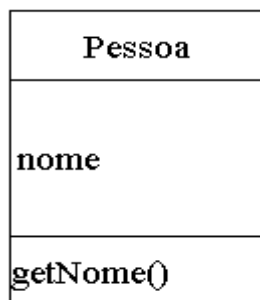


Figura 6: Exemplo de classe.

4. BORLAND DELPHI 7

O ambiente de desenvolvimento escolhido para a implementação do software é o Delphi7, desenvolvido pela Borland Internacional Inc., é uma ferramenta de fácil aprendizado, acesso e interação entre as ferramentas que combina códigos totalmente compiláveis, ferramentas visuais e tecnologia para a composição de base de dados escaláveis. Voltada exclusivamente para desenvolvimento em plataforma Windows e aplicações Cliente/Servidor, possui todas as ferramentas necessárias para criação de interfaces e implementação de projetos (Vide Figura 7).

Inicializando o Delphi, temos a sua IDE (Ambiente de Desenvolvimento Integrado). Na janela superior, observa-se a barra do menu principal contendo a barra de títulos com o nome Default Project1, a barra de menus, uma caixa de ferramentas, a paleta de componentes contendo os elementos que serão utilizados para trabalhar com a aplicação e o form.

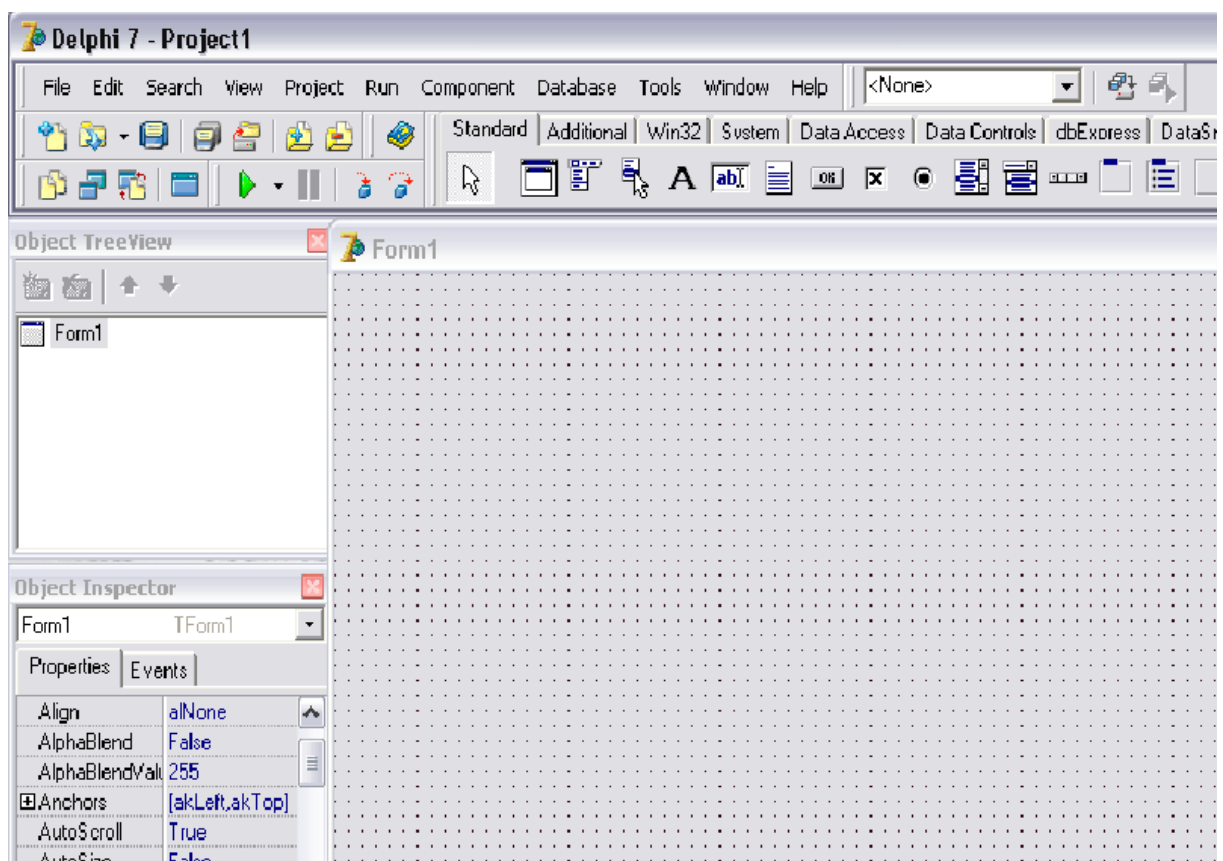


Figura 7: Ambiente de desenvolvimento Borland Delphi 7.

4.1. OBJECT INSPECTOR

A janela Object Inspector é uma ferramenta composta de duas páginas: Properties e Events, que nos permite acesso direto as propriedades e eventos relacionados a um determinado componente.

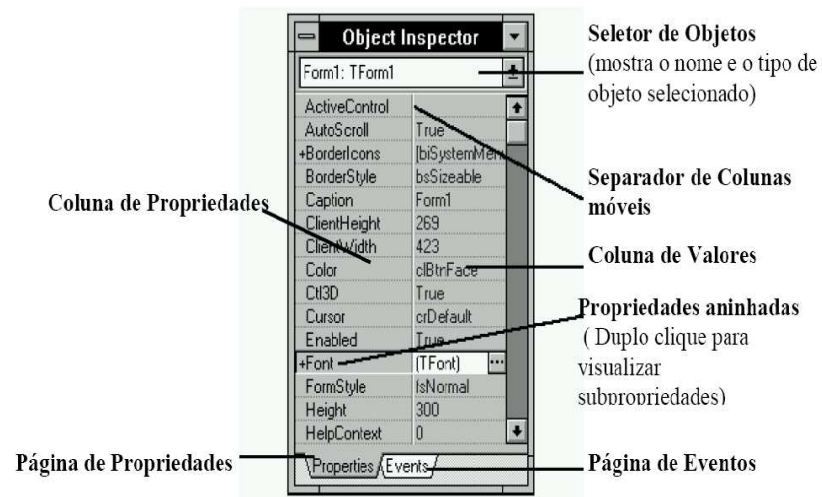


Figura 8: Object Inspector.

4.2. FORM (FORMULÁRIO)

O delphi nos oferece o Form (formulário), uma janela criada automaticamente, utilizada para construção da interface com o usuário, onde serão inseridos os componentes da aplicação.

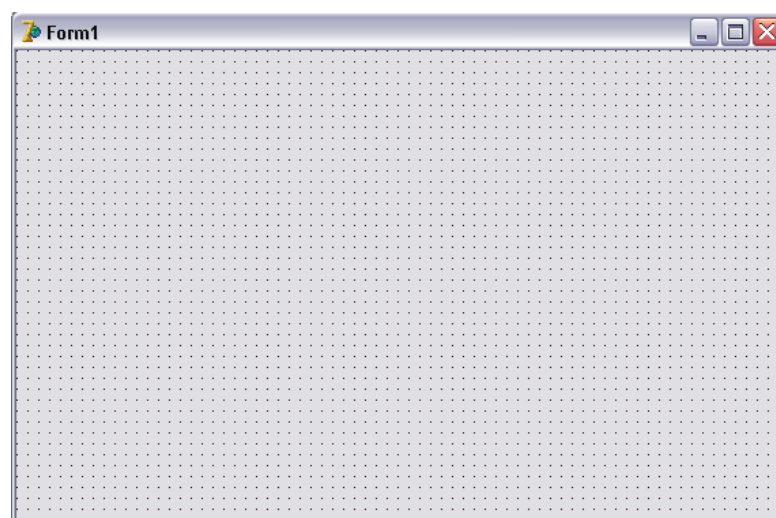


Figura 9: Form.

4.3. CODE EDITOR (EDITOR DE CÓDIGO)

Por trás do Form existe a janela Code Editor, onde será feita a edição dos códigos.

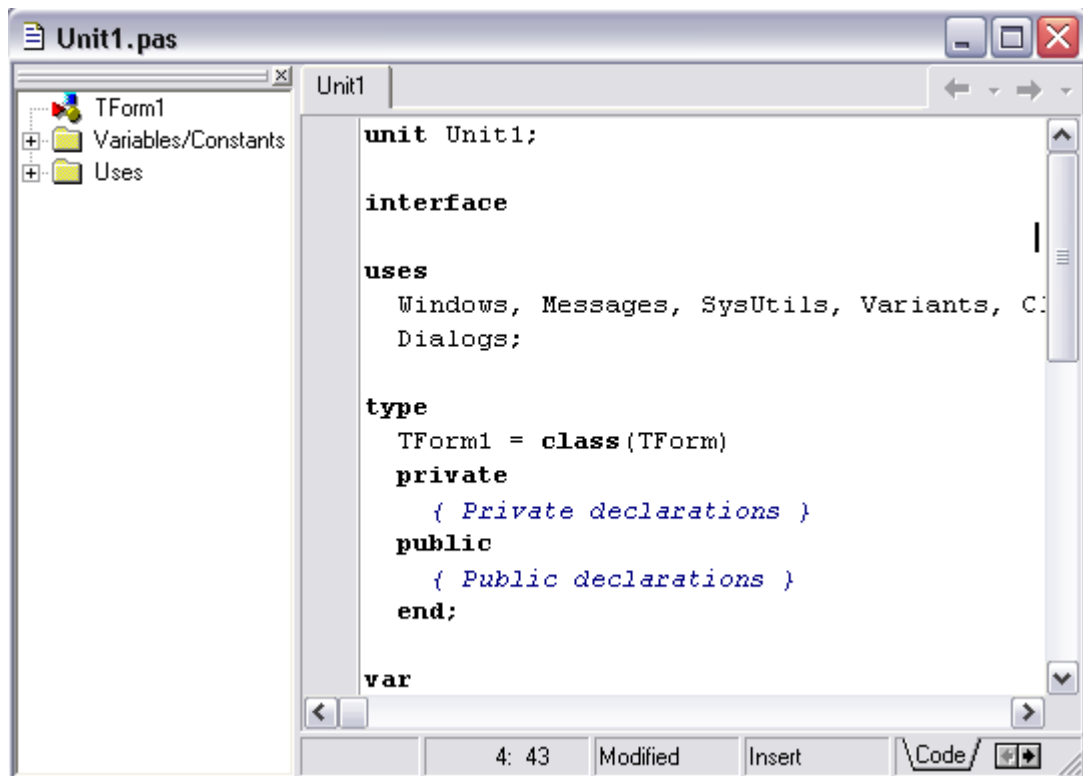


Figura 10: Code Editor.

5. BANCO DE DADOS MYSQL

O MySQL foi desenvolvido pela empresa TCX em 1996, mas atualmente é controlado pela companhia MySQL AB. É um sistema gerenciador de banco de dados relacional com código fonte aberto, considerado um dos bancos de dados relacionais mais rápidos do mercado. Tem como principais características, foco em facilidade de administração com baixo consumo de recursos do hardware, rápido tempo de conexão e servidor Multithreaded, ou seja, um único comando *mysqld* atende a todos os clientes.

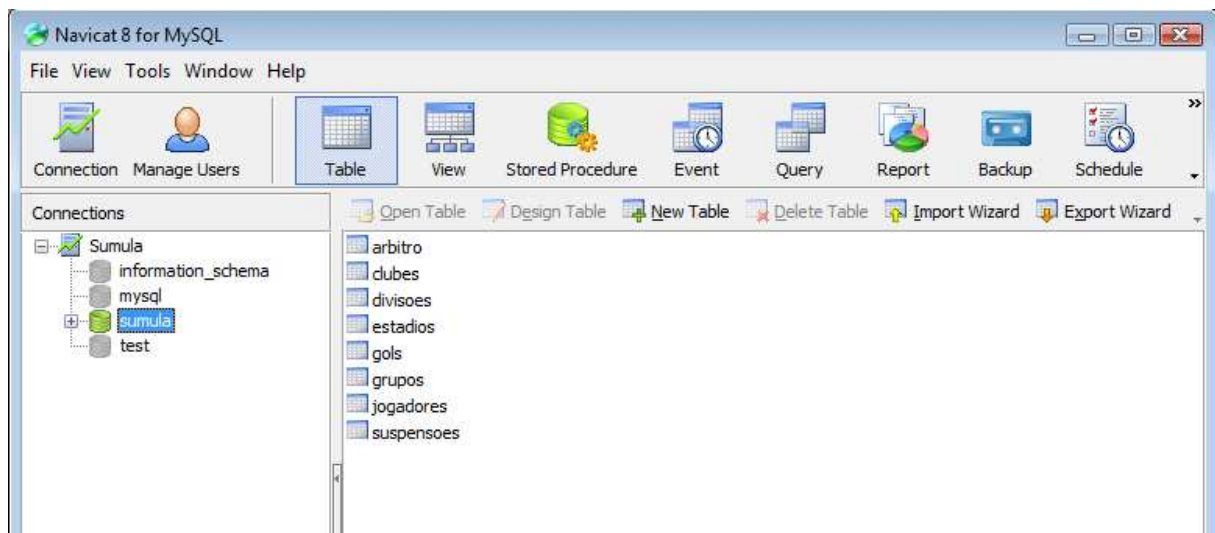


Figura 11: Navicat de MySql.

6. PROJETO

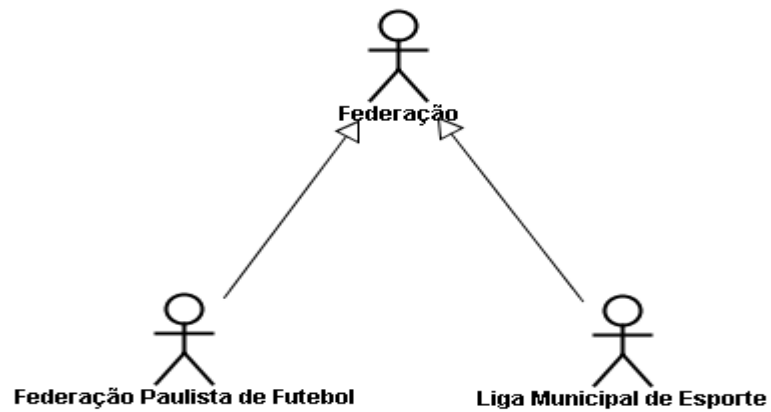
A partir desse tópico começa a abordagem da análise do projeto, ou seja, a lista de eventos, os diagramas e a apresentação das telas assim como manual do usuário.

6.1. LISTA DE EVENTOS

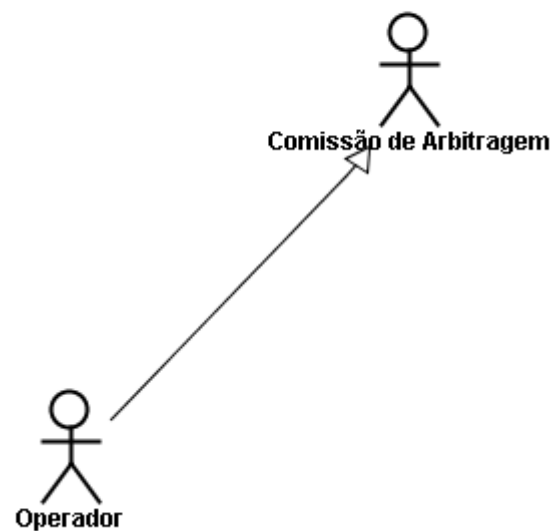
- 1) Cadastrar clubes
- 2) Cadastrar jogadores
- 3) Cadastrar árbitros
- 4) Cadastrar divisões
- 5) Cadastrar grupos
- 6) Cadastrar estádios
- 7) Lançar súmula
- 8) Lançar jogador súmula
- 9) Lançar gols
- 10) Lançar suspensão
- 11) Emitir relatório de classificação
- 12) Emitir relatório de saldo de gols
- 13) Emitir relatório de gols marcados
- 14) Emitir relatório de gols sofridos
- 15) Emitir relatório de artilheiro

6.2. RELAÇÃO DOS ATORES

6.2.1. Federação



6.2.2. Comissão de Arbitragem



6.3. USE CASES

6.3.1. Use Case Cadastrar Clubes



Caso de Uso: Cadastrar Clubes.

Ator: Federação.

Finalidade: Cadastrar os clubes que irão disputar o campeonato.

Descrição: Os clubes enviam seus dados para a federação que por sua vez os cadastra no software.

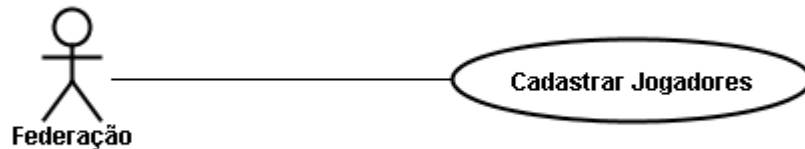
Ação do ator:

1. A federação recebe a relação dos clubes inscritos no campeonato.
2. Cadastra o nome e o código de cada clube.
3. Cadastra a cidade.
4. Cadastra o estádio no qual o clube administra seus jogos.

Resposta do software:

1. Mostra os dados dos clubes.
2. Mostra a divisão e o grupo do campeonato que o clube esta inscrito.
3. Retorna gols pró e gols contra.
4. Retorna o saldo de gols.
5. Retorna os pontos ganhos pelo clube.

6.3.2. Use Case Cadastrar jogadores



Caso de Uso: Cadastrar Jogadores.

Ator: Federação.

Finalidade: Cadastrar os jogadores que disputarão o campeonato em seus respectivos clubes.

Descrição: Os clubes fornecem para a federação os dados de seus jogadores para serem cadastrados no software.

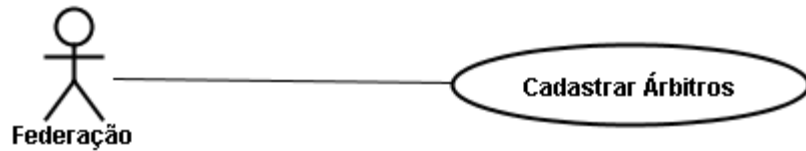
Ação do ator:

1. A federação recebe os dados dos jogadores de cada clube.
2. Cadastra o código e o nome dos jogadores.
3. Cadastra a data de nascimento dos jogadores.
4. Cadastra a posição dos jogadores.

Resposta do software:

1. Mostra os dados dos jogadores.
2. Mostra o clube no qual cada jogador esta inscrito.
3. Mostra os gols marcados.
4. Mostra os gols sofridos.
5. Mostra o número de cartões amarelos.
6. Mostra o número de cartões vermelhos.

6.3.3. Use Case Cadastrar Árbitros



Caso de Uso: Cadastrar Árbitros.

Ator: Federação.

Finalidade: Cadastrar os árbitros do campeonato.

Descrição: A federação cadastra os dados dos árbitros no software.

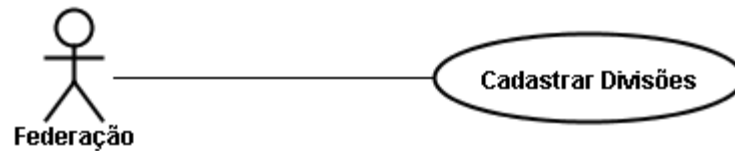
Ação do ator:

1. A federação recebe os dados dos árbitros.
2. Cadastra o código e o nome dos árbitros.
3. Cadastra a data de nascimento dos árbitros.
4. Cadastra a federação na qual os árbitros estão filiados.
5. Cadastra a profissão dos árbitros.

Resposta do software:

1. Mostra os dados dos árbitros.

6.3.4. Use Case Cadastrar Divisões



Caso de Uso: Cadastrar Divisões.

Ator: Federação.

Finalidade: Cadastrar no software as divisões pertencentes ao campeonato.

Descrição: A federação cadastra cada divisão pertencente a determinado campeonato, podendo ocorrer de só haver uma divisão.

Ação do ator:

1. Cadastra o código e o nome da divisão.

Resposta do software:

1. Mostra os dados das divisões.
2. Mostra os clubes pertencentes a divisão.

6.3.5. Use Case Cadastrar Grupos



Caso de Uso: Cadastrar Grupos.

Ator: Federação.

Finalidade: Se houver fase de grupos, informar os clubes pertencentes a cada grupo.

Descrição: A federação divide os clubes em grupos iguais, onde jogarão um contra o outro dentro do grupo.

Ação do ator:

1. Cadastra o código e o nome do grupo.

Resposta do software:

1. Mostra os dados dos grupos.
2. Mostra os clubes pertencentes aos grupos.

6.3.6. Use Case Cadastrar Estádios



Caso de Uso Cadastrar Estádios.

Ator: Federação.

Finalidade: Cadastrar os estádios nos quais serão disputados os jogos do campeonato.

Descrição: A federação cadastra no software os estádios com seus respectivos clubes mandantes.

Ação do Ator:

1. Cadastra código do estádio.
2. Cadastra nome do estádio.
3. Cadastrar a cidade.
4. Cadastrar a capacidade do estádio.

Resposta do software:

1. Mostra o nome do estádio.
2. Mostra a cidade em que se localiza.
3. Mostra a capacidade total de pessoas.

6.3.7. Use Case Lançar Súmula



Caso de Uso: Lançar Súmula.

Ator: Comissão de arbitragem.

Finalidade: Proporcionar a federação, todos os dados de cada partida.

Descrição: A comissão de arbitragem registra todas as informações decorrentes de cada partida em uma súmula que será enviada para a federação.

Ação do ator:

1. Cadastra o código da súmula.
2. Cadastra o código do clube mandante.
3. Cadastra o código do clube visitante.
4. Cadastra o código do estádio.
5. Cadastra o código do juiz da partida.
6. Cadastra o código dos bandeirinhas da partida.
7. Cadastra a data da partida.
8. Informa os gols e o tempo de jogo.
9. Cadastra as ocorrências da partida.
10. Faz o fechamento da súmula.

Resposta do software:

1. Mostra os clubes que estão em disputa.
2. Mostra o árbitro e os bandeirinhas da partida.
3. Mostra o estádio da partida.
4. Mostra o placar da partida.
5. Mostra o tempo de jogo.
6. Mostra as ocorrências da partida.
7. Fecha a súmula e os dados decorrentes da partida serão processados.

6.3.8. Use Case Lançar Jogador Súmula



Caso de Uso: Lançar Jogador Súmula.

Ator: Comissão de arbitragem.

Finalidade: Controlar os cartões recebidos pelos jogadores.

Descrição: A comissão de arbitragem registra no software os cartões recebidos pelos jogadores. Ao final da partida é feita a validação dos cartões, quando um cartão vermelho é emitido, causa a suspensão imediata do jogador na próxima partida e os cartões amarelos são acumulativos, quando determinado jogador recebe o terceiro cartão amarelo no campeonato recebe a suspensão.

Ação do ator:

1. Cadastra a partida.
2. Cadastra o jogador penalizado.
3. Informa o tipo de cartão aplicado.

Resposta do software:

1. Mostra o jogador penalizado e o tipo de cartão.
2. Valida os cartões.

6.3.9. Use Case Lançar Gols



Caso de Uso: Lançar Gols.

Ator: Comissão de arbitragem

Finalidade: Relatar os gols ocorrente em cada partida assim como o nome do jogador que o fez e o tempo de jogo.

Descrição: A comissão de arbitragem registra no software cada gol ocorrido em determinada partida, contendo informações de que time marcou, o jogador e o tempo de jogo.

Ação do ator:

1. Cadastra o gol para o time que o fez.
2. Cadastra o jogador que marcou o gol.
3. Cadastra o tempo de jogo que ocorreu o gol.

Resposta do software:

4. Mostra o placar.
5. Mostra o jogador que fez o gol.
6. Mostra o tempo de jogo que ocorreu o gol.

6.3.10. Use Case Lançar Suspensão



Caso de Uso: Lançar Suspensão.

Ator: Comissão de arbitragem.

Finalidade: Suspender os jogadores penalizados.

Descrição: Após a validação no Jogador Súmula, os cartões são processados e as punições são efetuadas.

Ação do ator:

1. Cadastra o jogador.

Resposta do software:

1. Mostra o jogador suspenso.
2. Mostra a data da suspensão.

6.3.11. Use Case Emitir Relatório de Classificação



Caso de Uso: Emitir Relatório de Classificação.

Ator: Federação.

Finalidade: Disponibilizar para federação a classificação do campeonato.

Descrição: A cada rodada do campeonato, são somados os pontos ganho por cada equipe, proporcionando a visão da tabela de classificação.

Ação do ator:

1. Atualiza os pontos ganhos por cada equipe, no caso de derrota, nenhum ponto será cadastrado.

Resposta do software:

1. Soma os pontos.
2. Soma os gols marcados.
3. Soma os gols sofridos.
4. Subtrai gols marcados por gols sofridos, obtendo o saldo de gols.
5. Soma mais um (+ 1) ao número de jogos.
6. Mostra a classificação do campeonato.

6.3.12. Use Case Emitir Relatório de Saldo de Gols



Caso de Uso: Emitir Relatório de Saldo de Gols.

Ator: Federação.

Finalidade: Obter o saldo de gols de cada equipe, ou seja, o numero de gols marcados menos o numero de gols sofridos.

Descrição: Ao término de cada rodada são lançados no software o resultado de todos os jogos pela federação, o sistema emite o saldo de gols de cada clube.

Ação do ator:

1. Cadastra no software o resultado de cada partida.

Resposta do software:

1. Subtrai o número de gols marcados pelo número de gols sofridos
2. Mostra o saldo de gols

6.3.13. Use Case Emitir Relatório de Gols Marcados



Caso de Uso: Emitir Relatório de Gols Marcados.

Ator: Federação

Finalidade: Obter o número de gols marcados por cada clube no campeonato.

Descrição: Ao término de cada rodada, tendo os resultados dos jogos cadastrados pela federação, o software emite o número de gols marcados por cada clube.

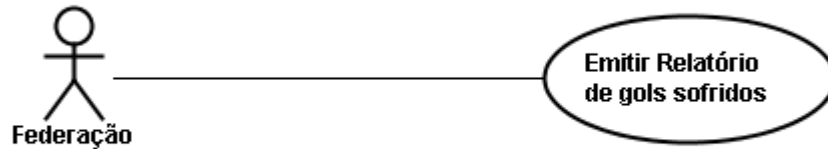
Ação do ator:

1. Cadastra o número de gols marcados por cada clube.

Resposta do software:

1. Soma o número de gols marcados na ultima rodada com os gols já cadastrados no sistema.
2. Mostra o número de gols marcados por cada clube.

6.3.14. Use Case Emitir Relatório de Gols Sofridos



Caso de Uso: Emitir Relatório de Gols Sofridos.

Ator: Federação

Finalidade: Obter o número de gols sofridos por cada clube no campeonato.

Descrição: Ao término de cada rodada, tendo os resultados dos jogos cadastrados pela federação, o software emite o número de gols sofridos por cada clube.

Ação do ator:

1. Cadastra o número de gols sofridos por cada clube.

Resposta do software:

1. Soma o número de gols sofridos na ultima rodada com os gols sofridos já cadastrados no software.
2. Mostra o número de gols sofridos por cada clube.

6.3.15. Use Case Emitir Relatório de Artilheiro



Caso de Uso: Emitir Relatório de Artilheiro.

Ator: Federação.

Finalidade: Informar o nome do jogador que marcou o maior número de gols no campeonato.

Descrição: A cada rodada, os resultados de todas as partidas são cadastrados no software, assim como os nomes dos jogadores que marcaram os gols. O sistema soma o número de gols marcados por cada jogador.

Ação do ator:

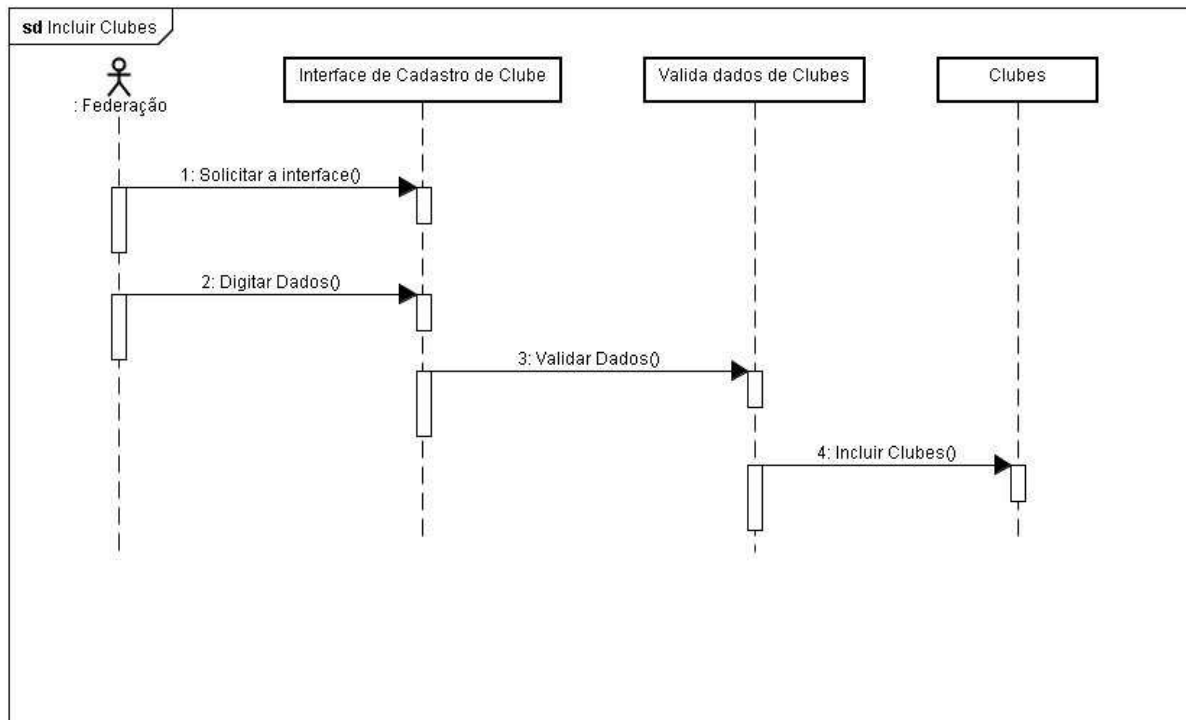
1. Cadastra o gol marcado para o respectivo jogador que o fez.

Resposta do software:

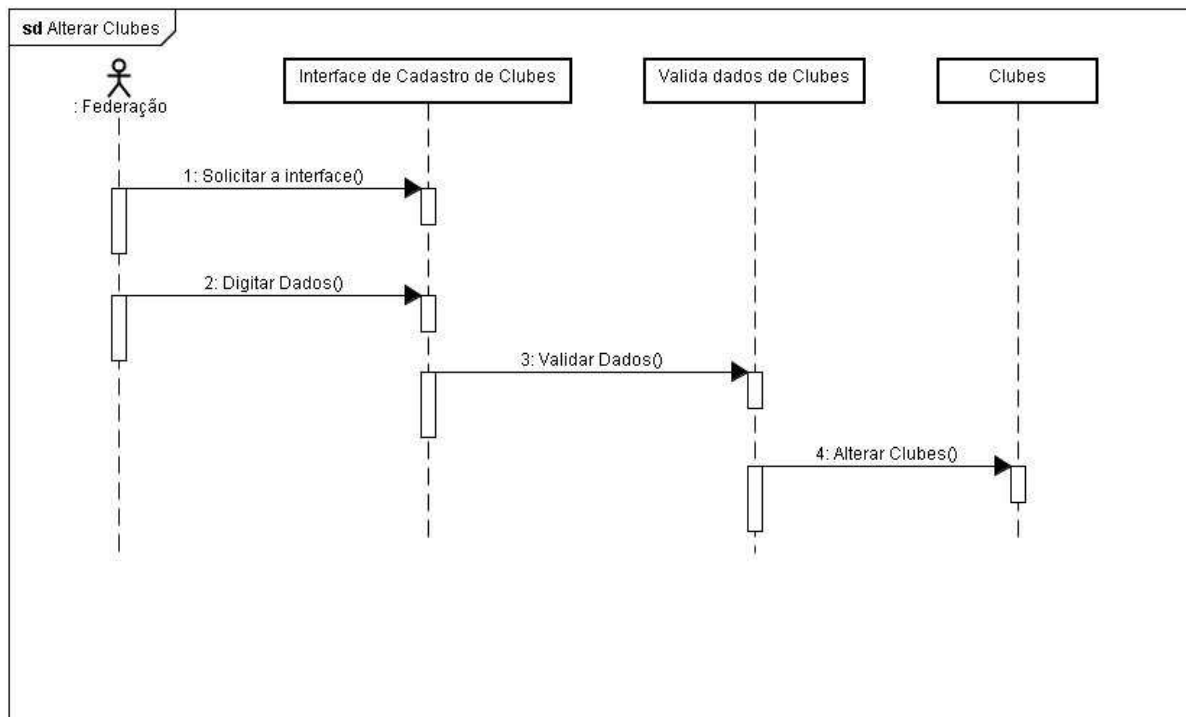
1. Soma os gols marcados por cada jogador
2. Mostra o numero de gols marcados de cada jogador do campeonato.

7. DIAGRAMAS DE SEQUÊNCIA

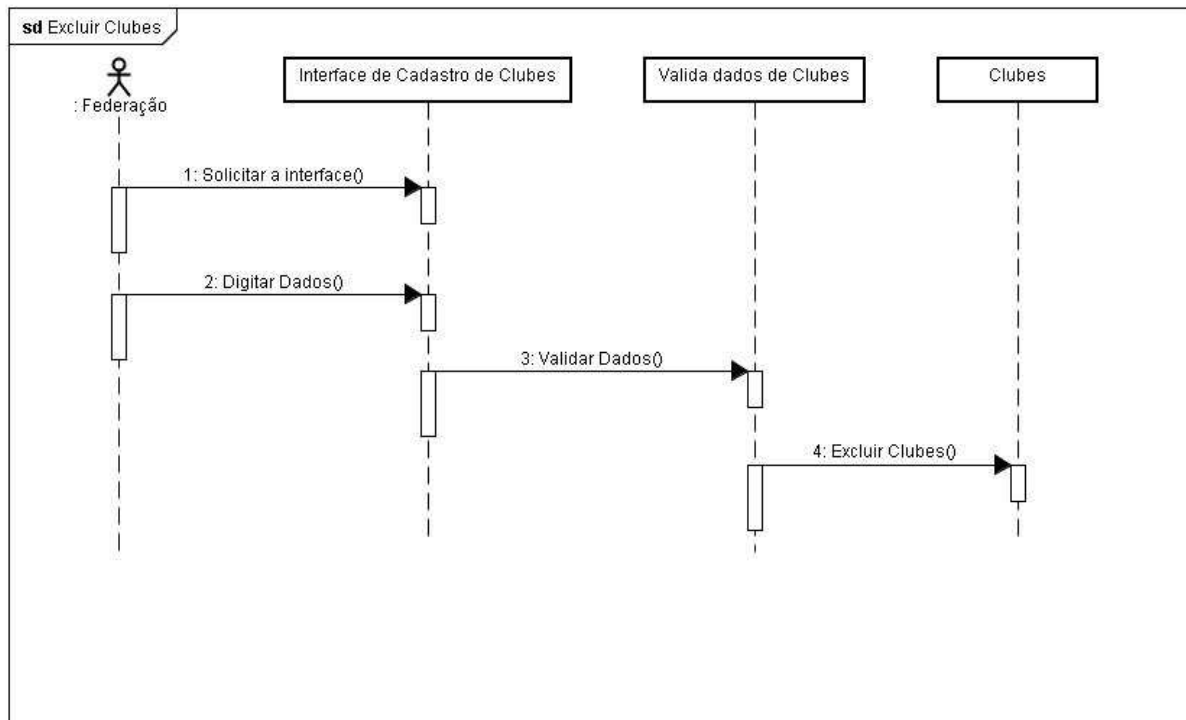
7.1. DIAGRAMA DE SEQUÊNCIA INCLUIR CLUBES



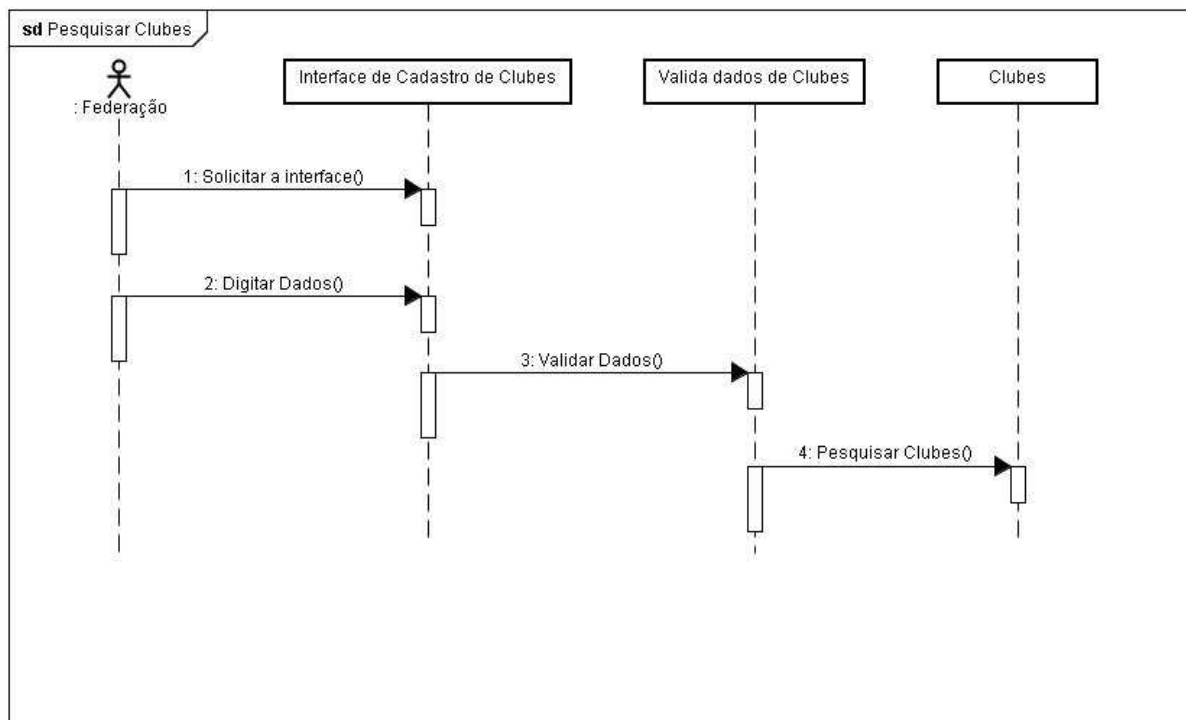
7.2. DIAGRAMA DE SEQUÊNCIA ALTERAR CLUBES



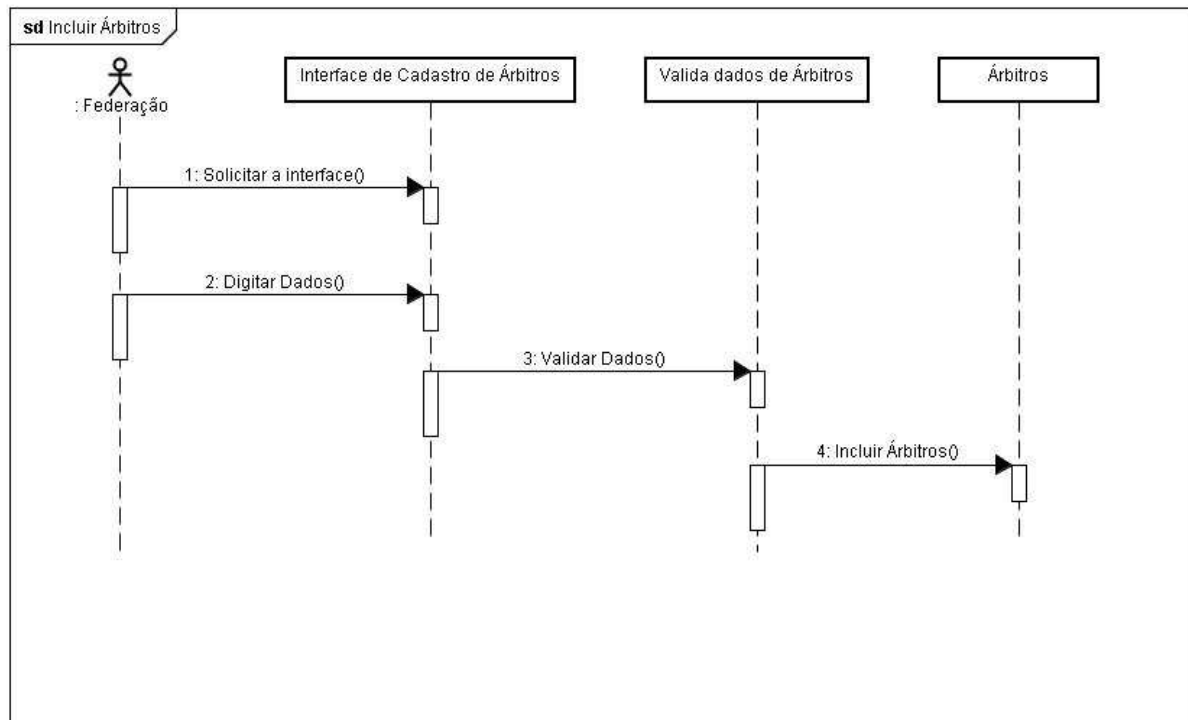
7.3. DIAGRAMA DE SEQUÊNCIA EXCLUIR CLUBES



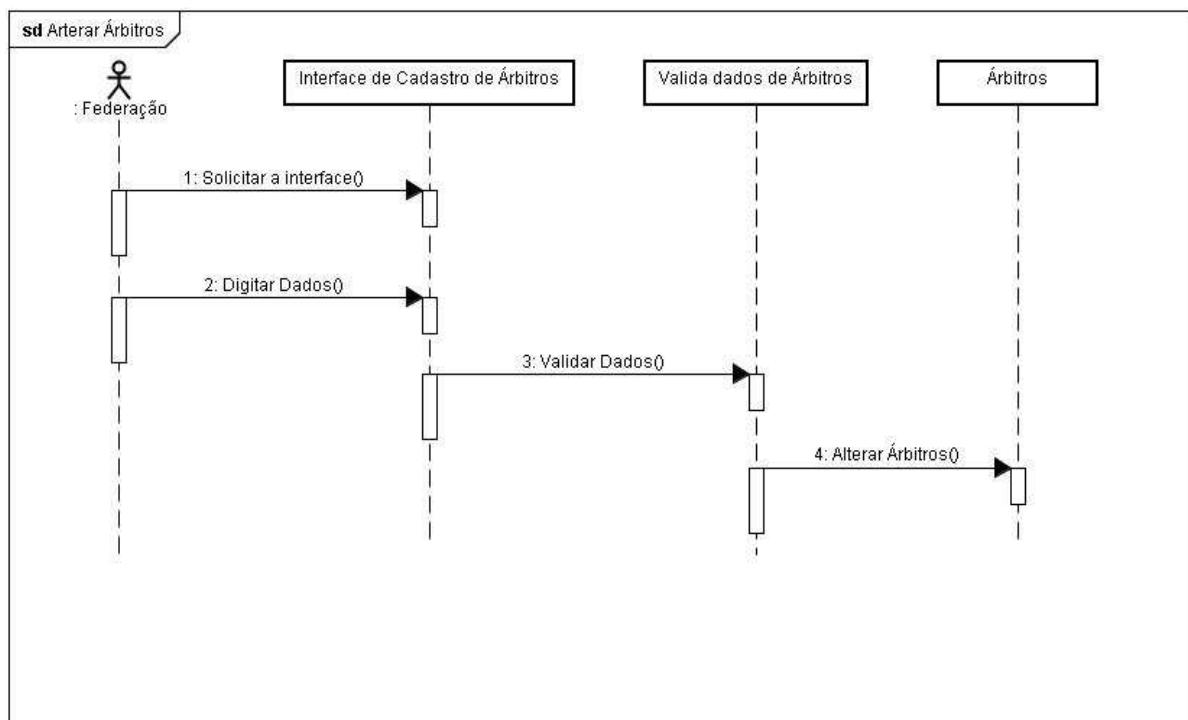
7.4. DIAGRAMA DE SEQUÊNCIA PESQUISAR CLUBES



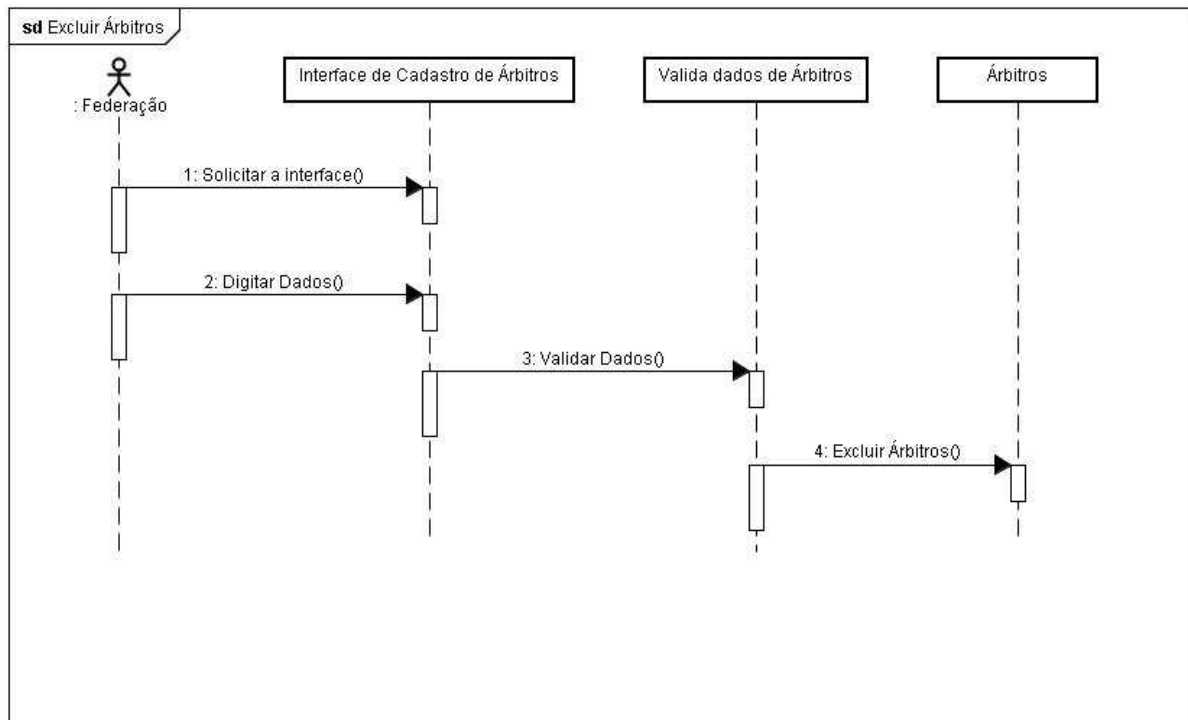
7.5. DIAGRAMA DE SEQUÊNCIA INCLUIR ÁRBITROS



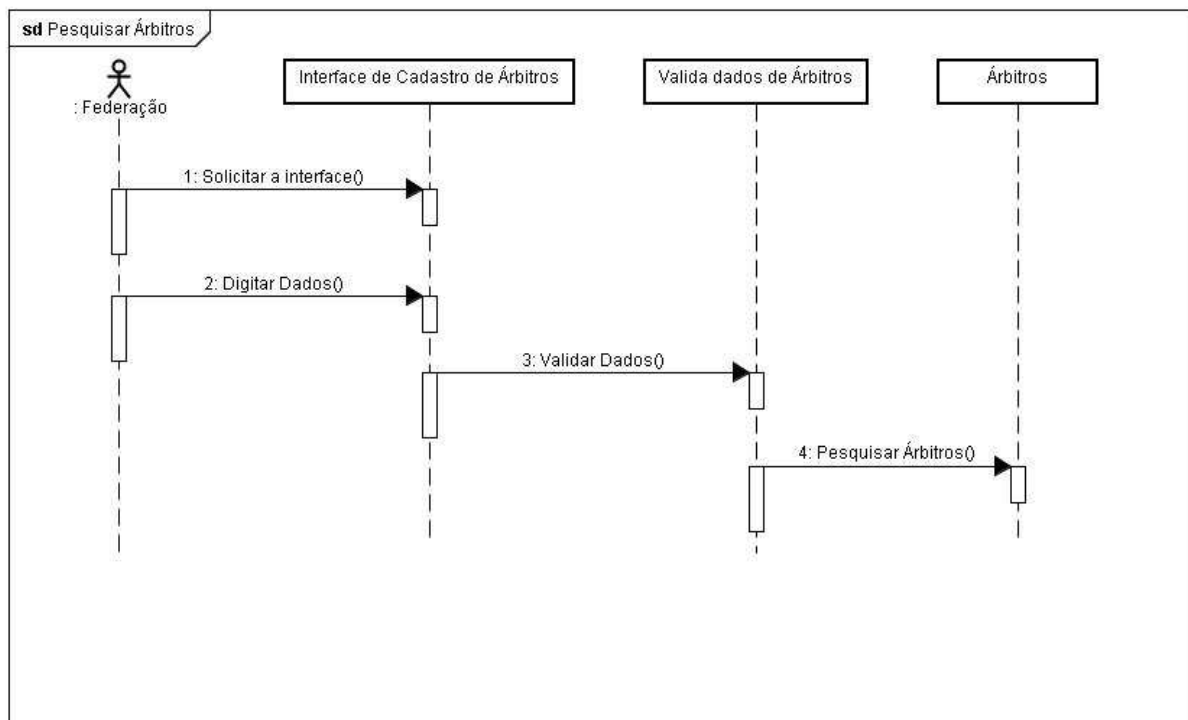
7.6. DIAGRAMA DE SEQUÊNCIA ALTERAR ÁRBITROS



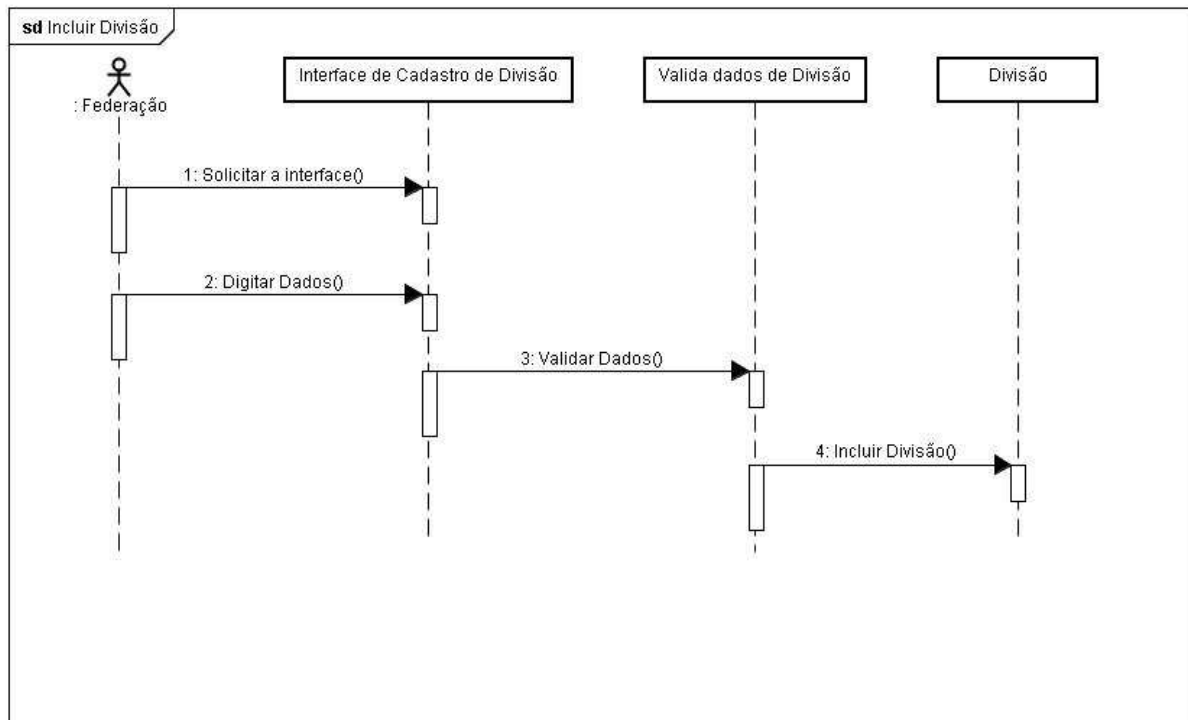
7.7. DIAGRAMA DE SEQUÊNCIA EXCLUIR ÁRBITROS



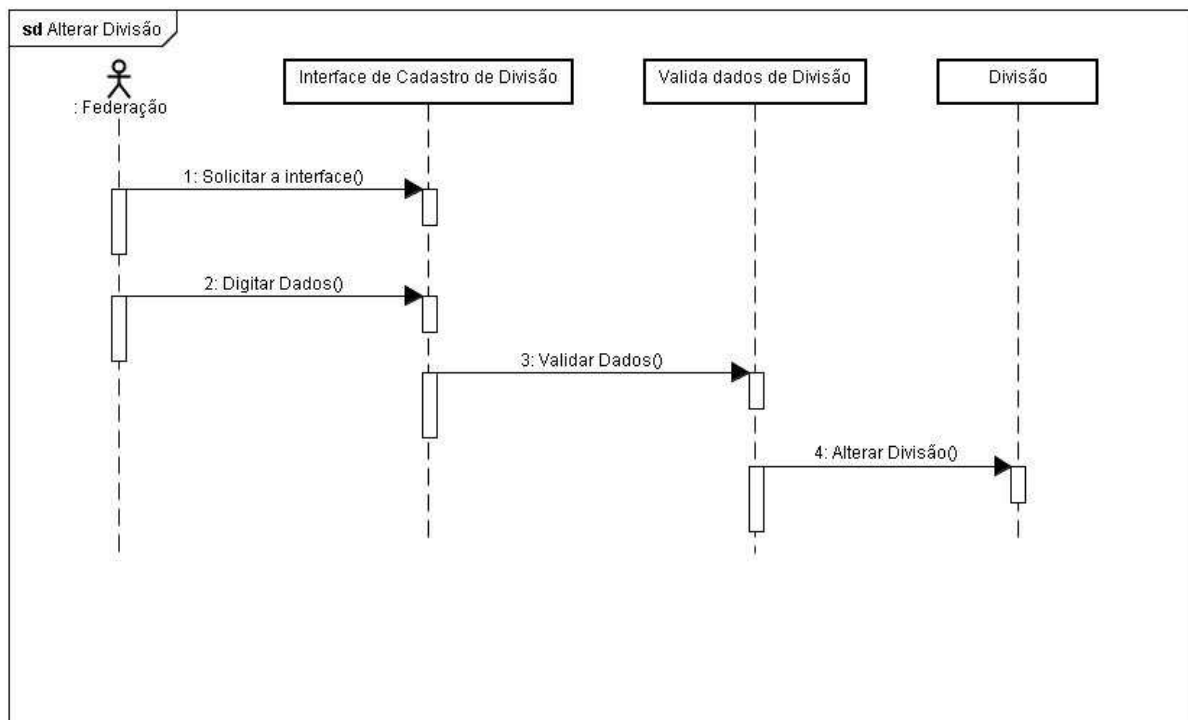
7.8. DIAGRAMA DE SEQUÊNCIA PESQUISAR ÁRBITROS



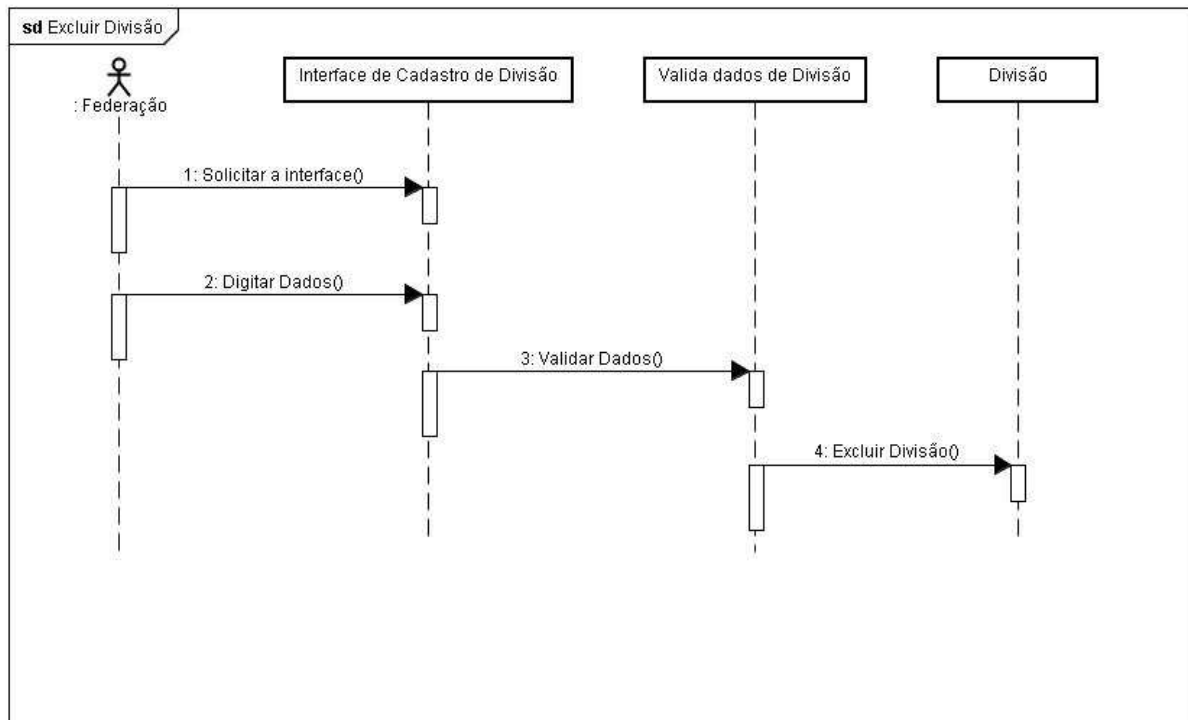
7.9. DIAGRAMA DE SEQUÊNCIA INCLUIR DIVISÃO



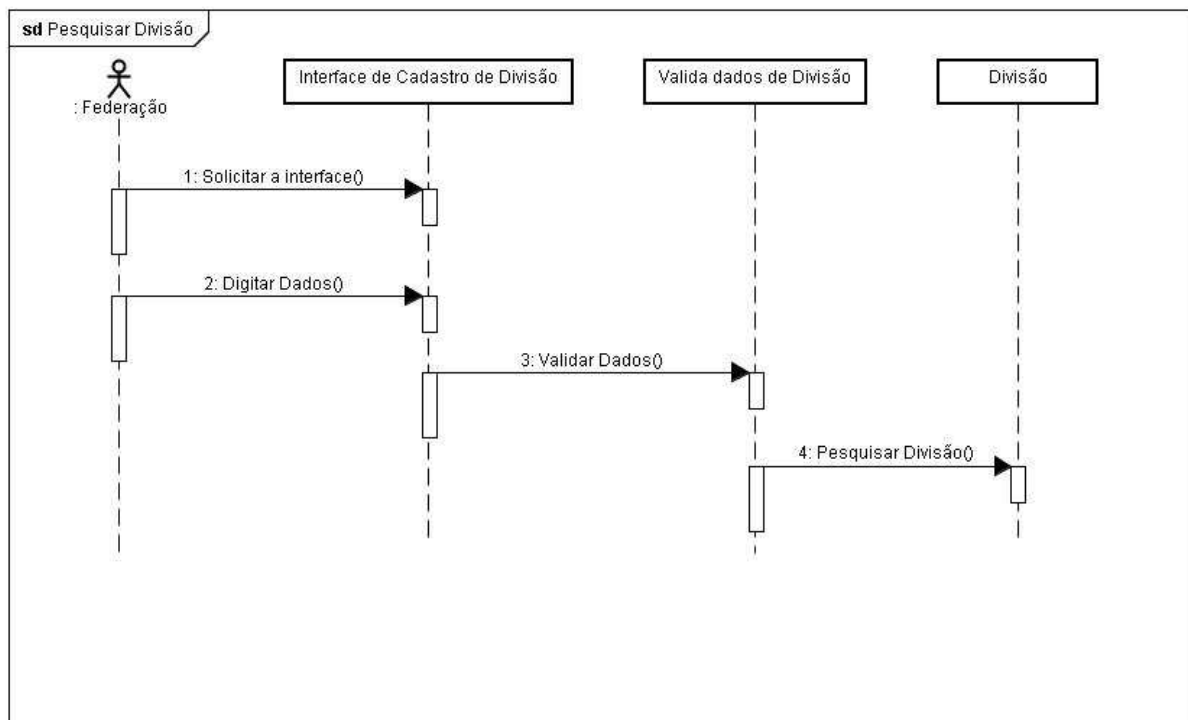
7.10. DIAGRAMA DE SEQUÊNCIA ALTERAR DIVISÃO



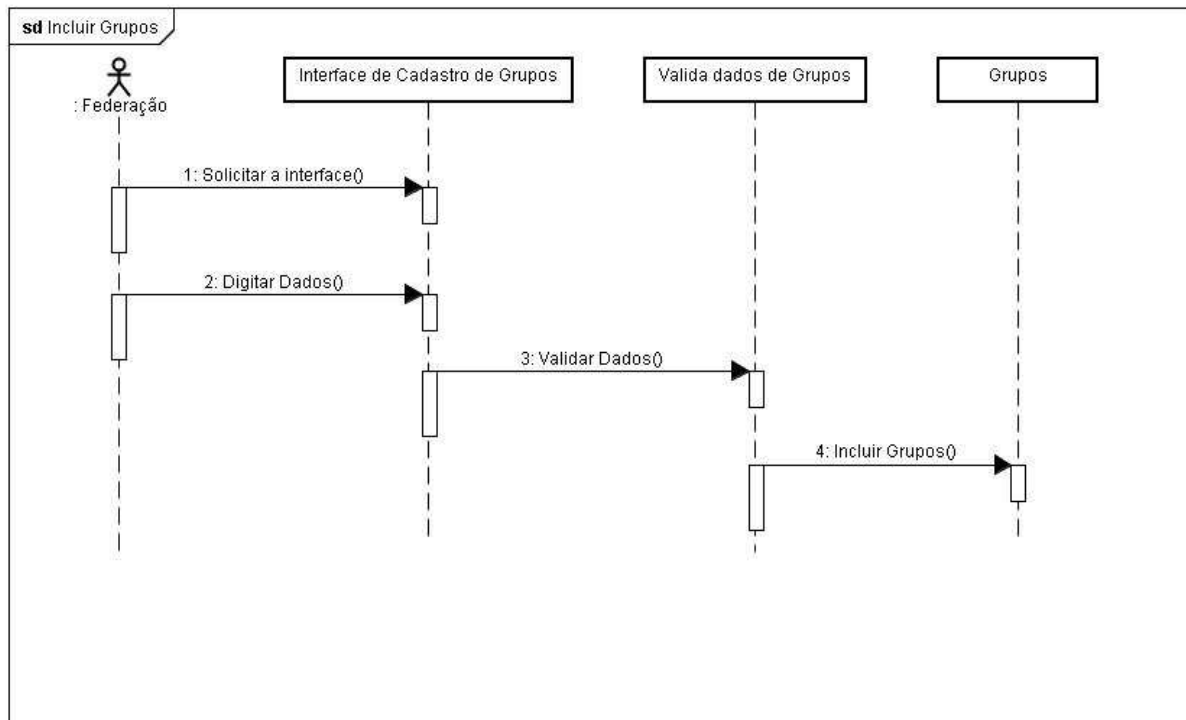
7.11. DIAGRAMA DE SEQUÊNCIA EXCLUIR DIVISÃO



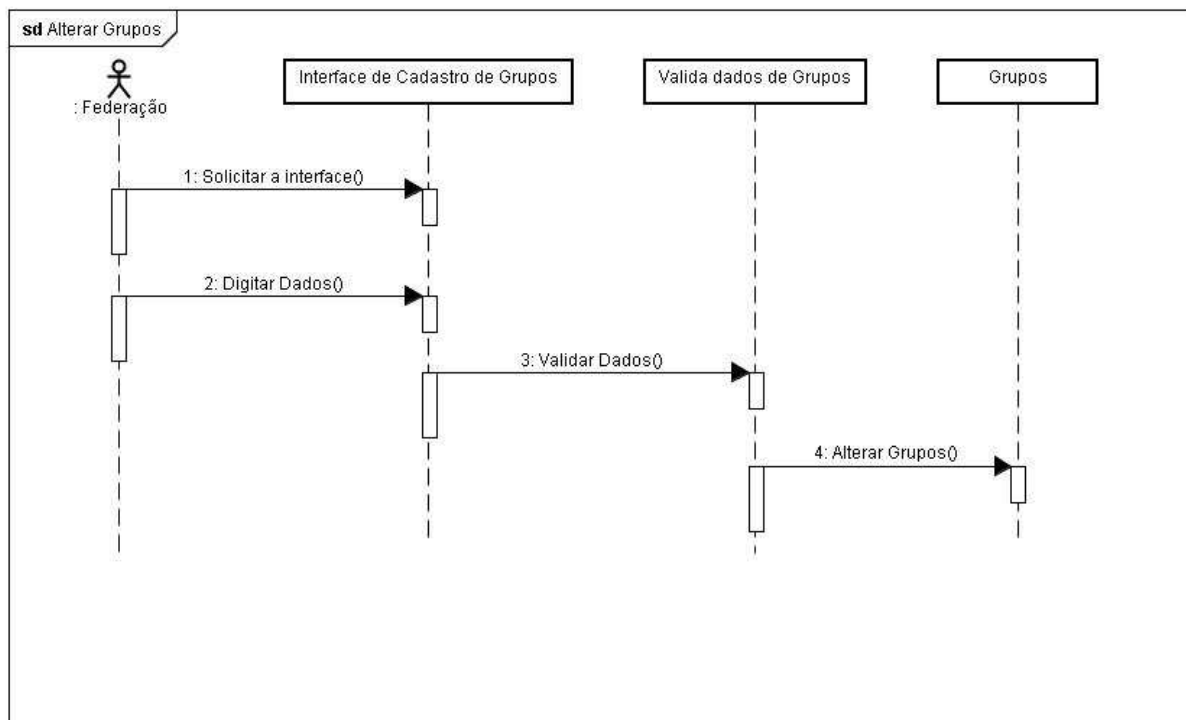
7.12. DIAGRAMA DE SEQUÊNCIA PESQUISAR DIVISÃO



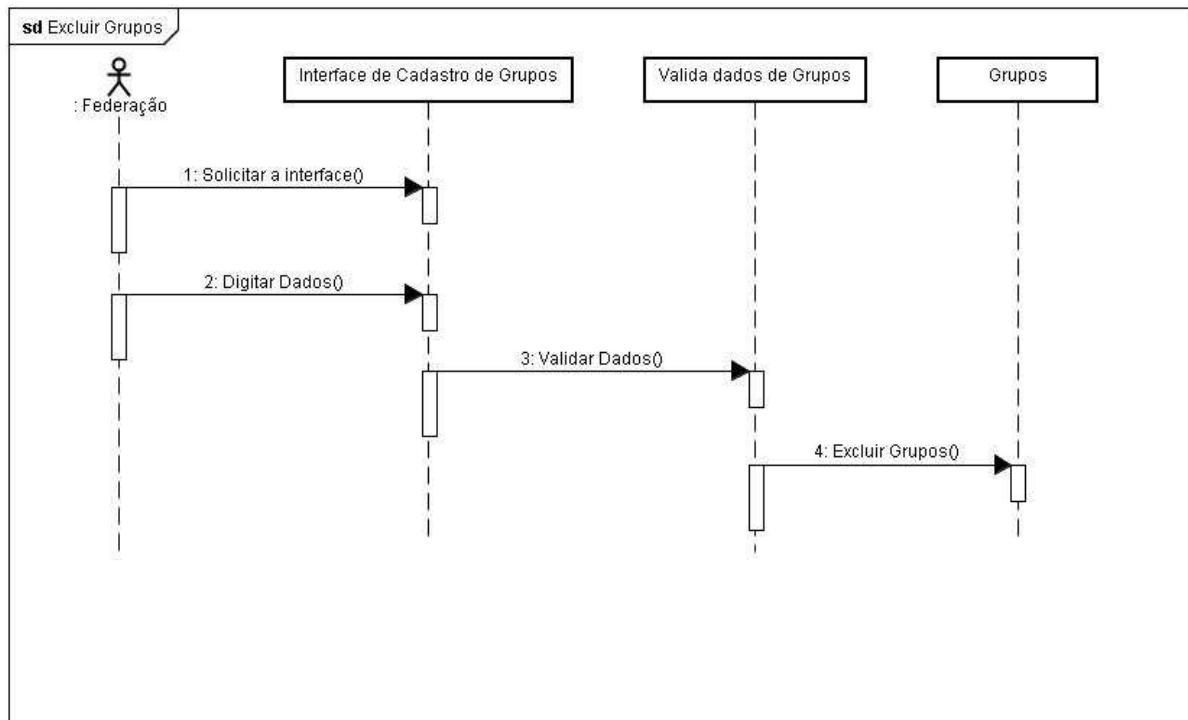
7.13. DIAGRAMA DE SEQUÊNCIA INCLUIR GRUPOS



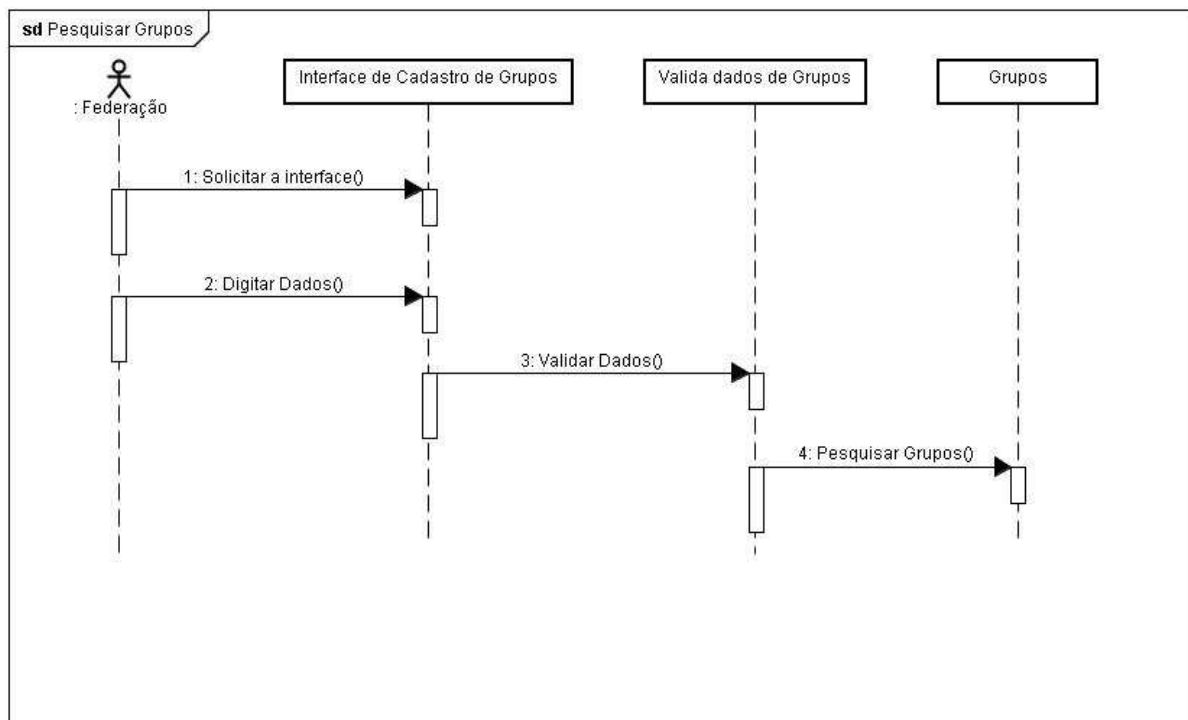
7.14. DIAGRAMA DE SEQUÊNCIA ALTERAR GRUPOS



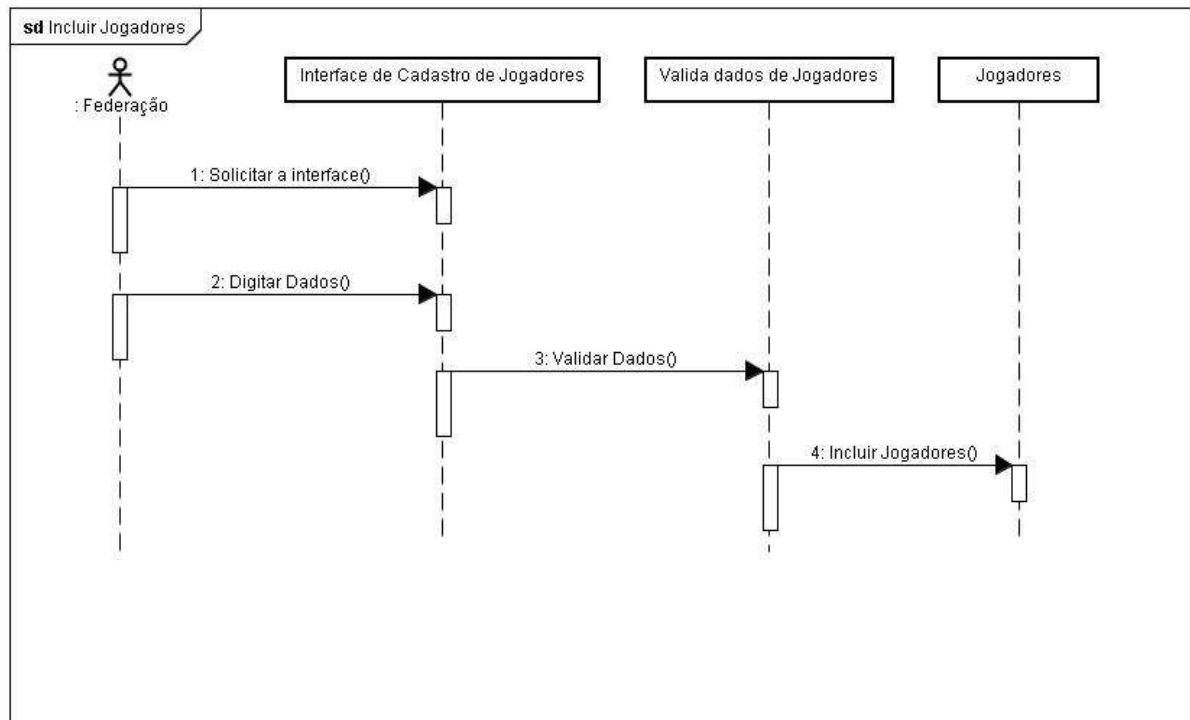
7.15. DIAGRAMA DE SEQUÊNCIA EXCLUIR GRUPOS



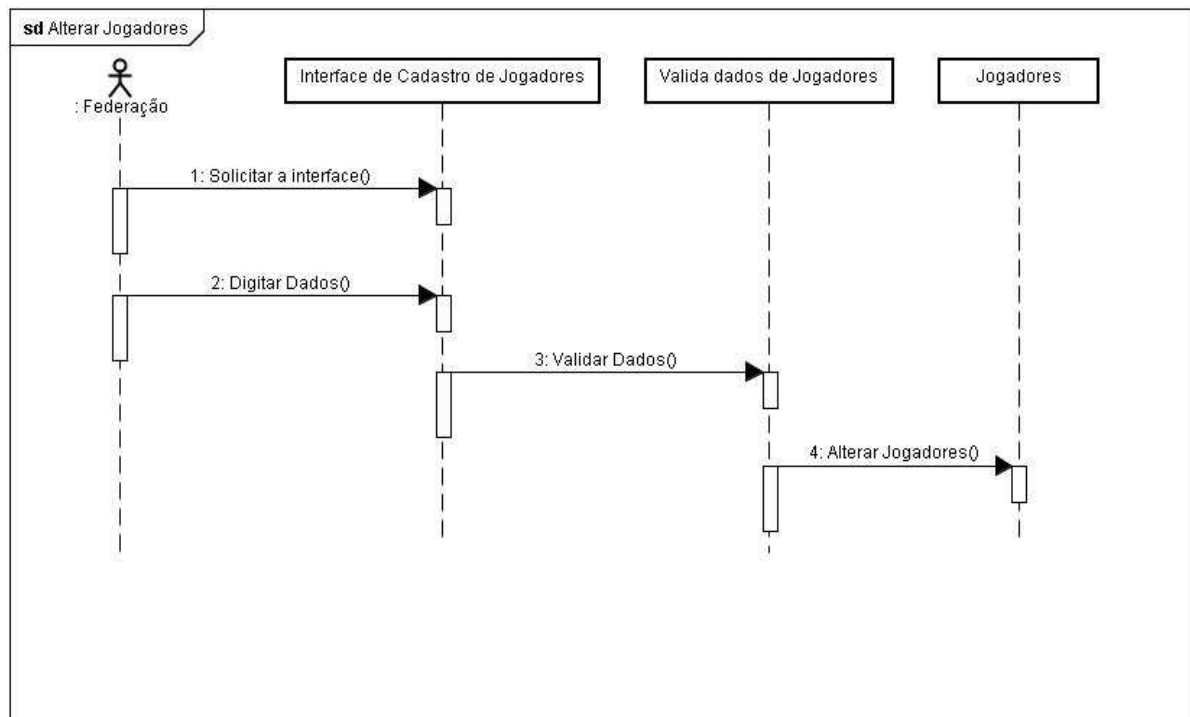
7.16. DIAGRAMA DE SEQUÊNCIA DE PESQUISAR GRUPOS



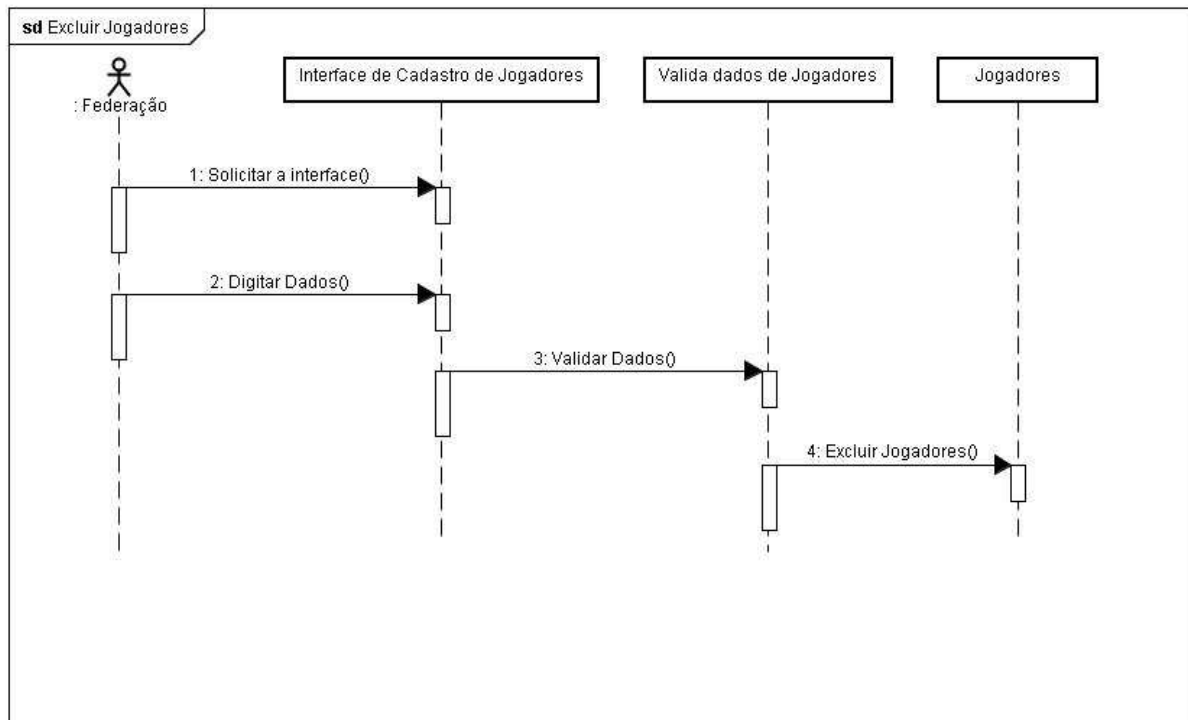
7.17. DIAGRAMA DE SEQUÊNCIA INCLUIR JOGADORES



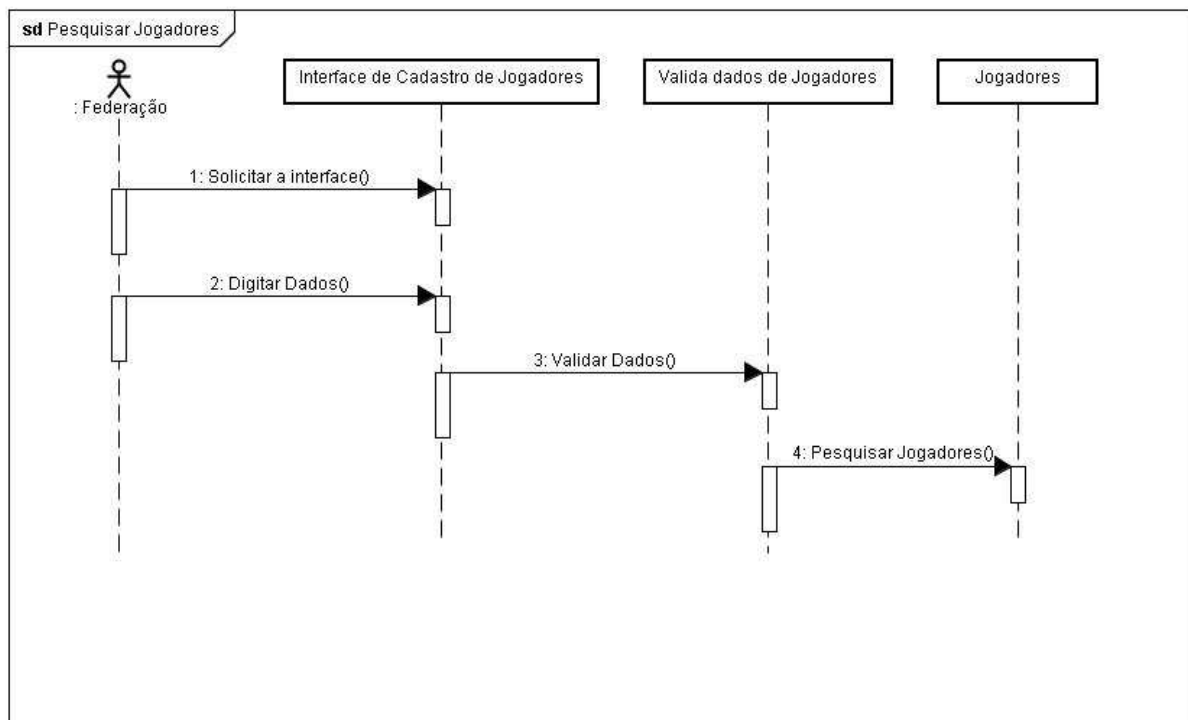
7.18. DIAGRAMA DE SEQUÊNCIA ALTERAR JOGADORES



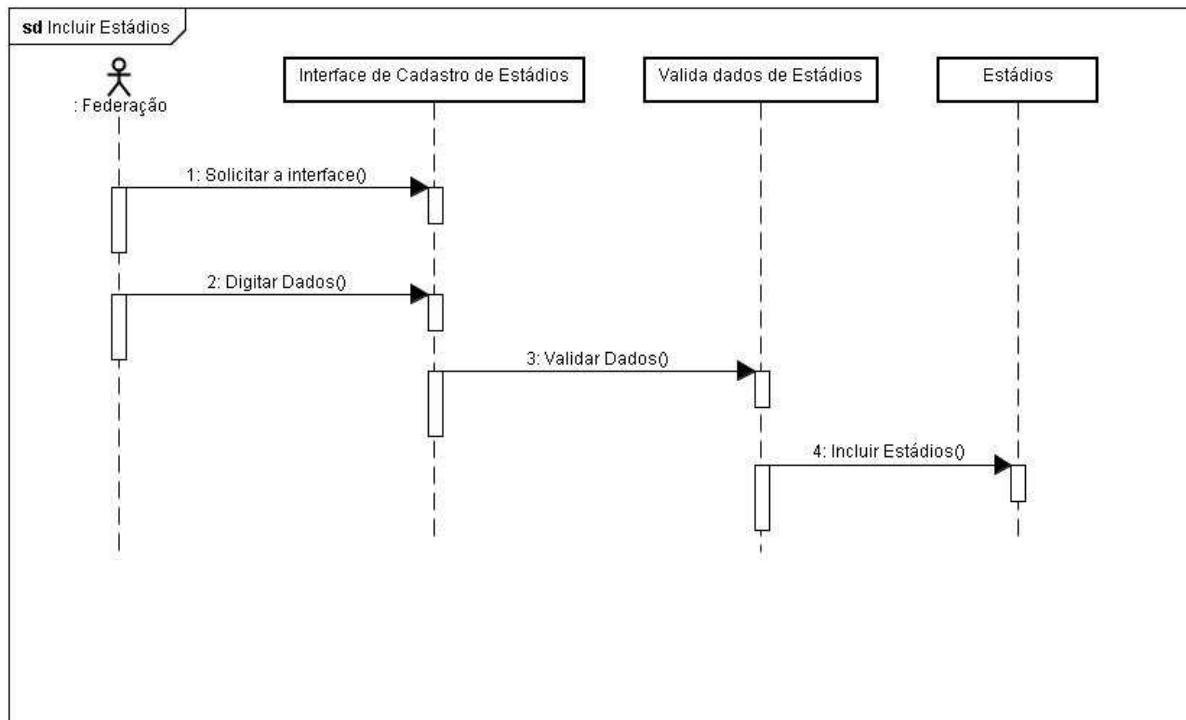
7.19. DIAGRAMA DE SEQUÊNCIA EXCLUIR JOGADORES



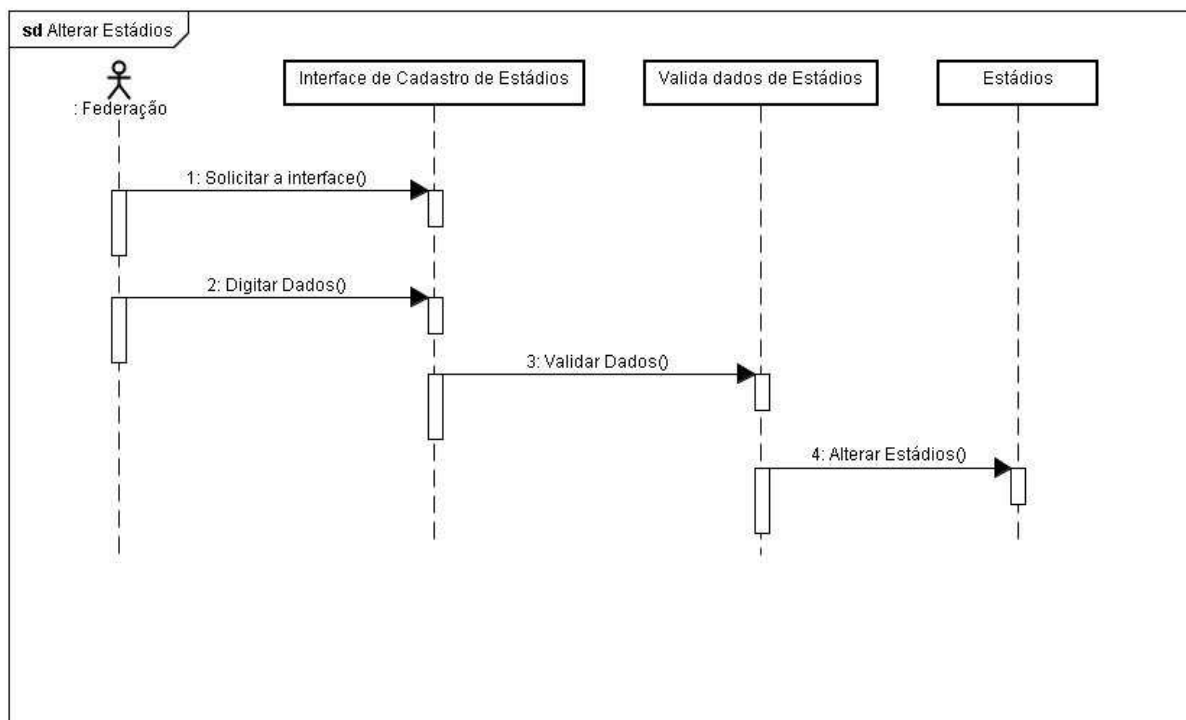
7.20. DIAGRAMA DE SEQUÊNCIA PESQUISAR JOGADORES



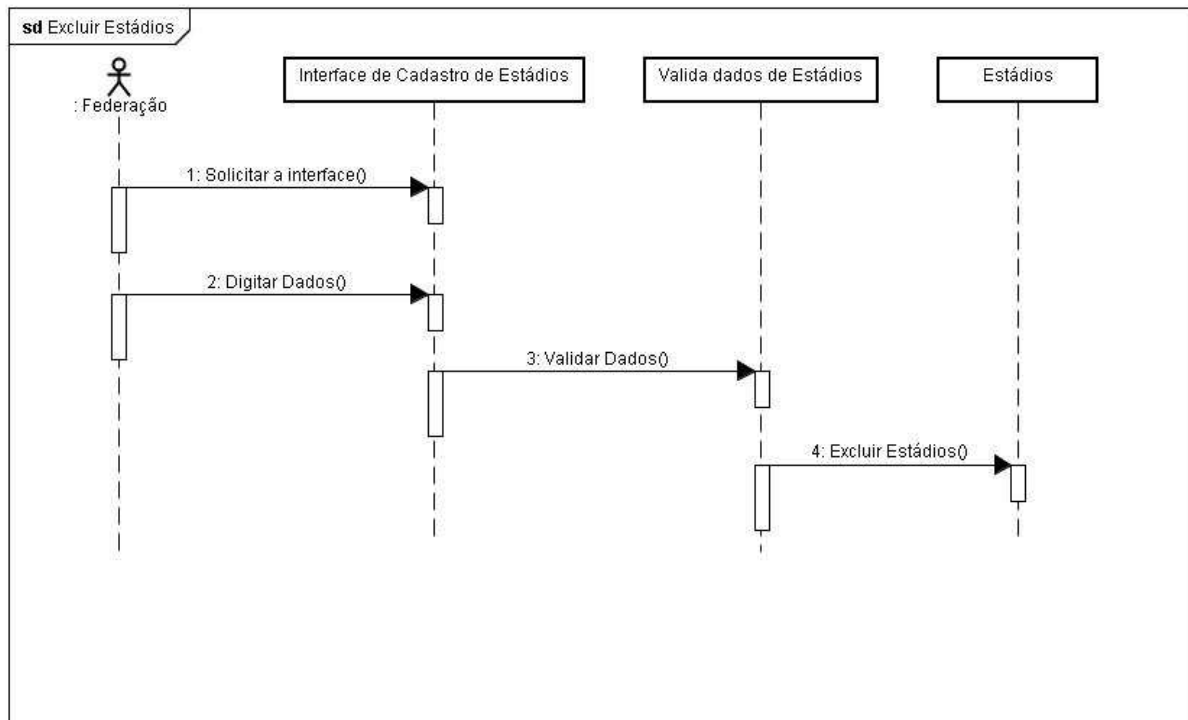
7.21. DIAGRAMA DE SEQUÊNCIA INCLUIR ESTÁDIOS



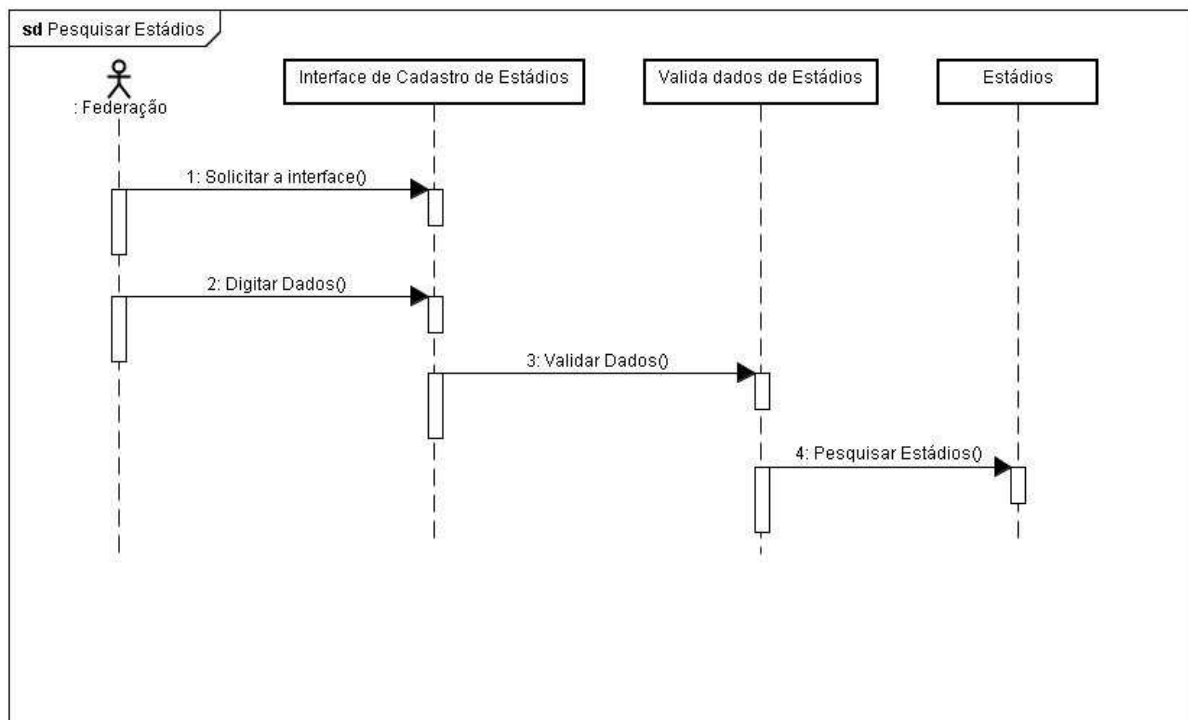
7.22. DIAGRAMA DE SEQUENCIA ALTERAR ESTÁDIOS



7.23. DIAGRAMA DE SEQUÊNCIA EXCLUIR ESTÁDIOS



7.24. DIAGRAMA DE SEQUÊNCIA PESQUISAR ESTÁDIOS



7.25. DIAGRAMA DE SEQUÊNCIA DE RELATÓRIOS

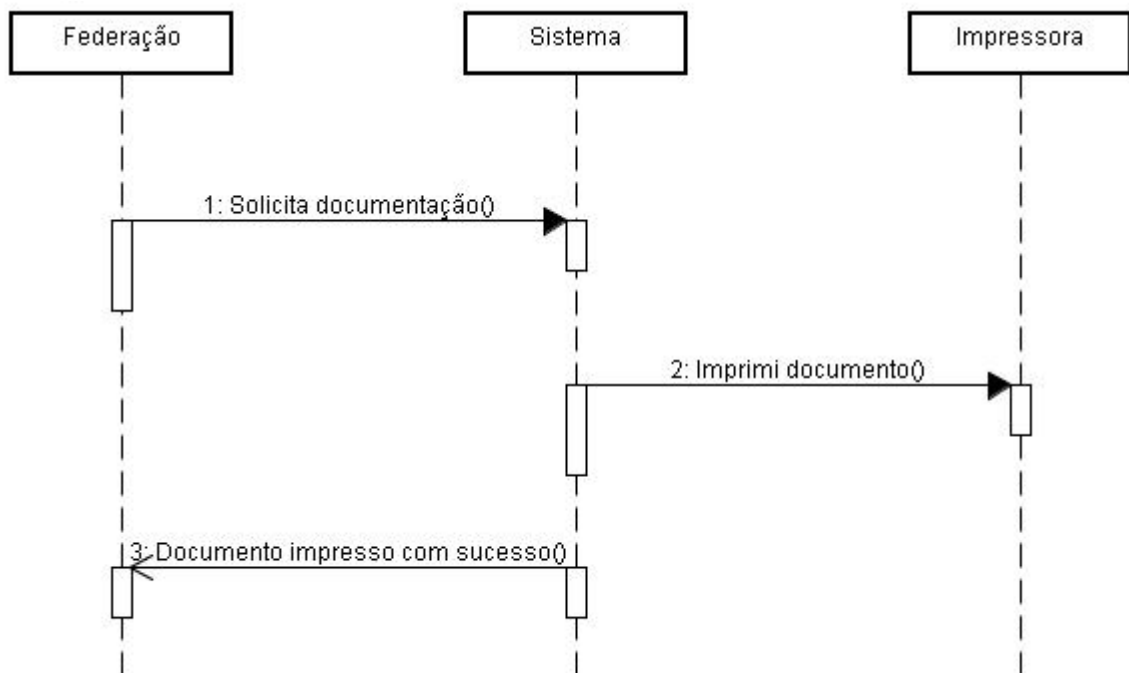


Diagrama de Sequência de Relatórios

7.26. DIAGRAMA DE SEQUENCIA JOGADOR SÚMULA

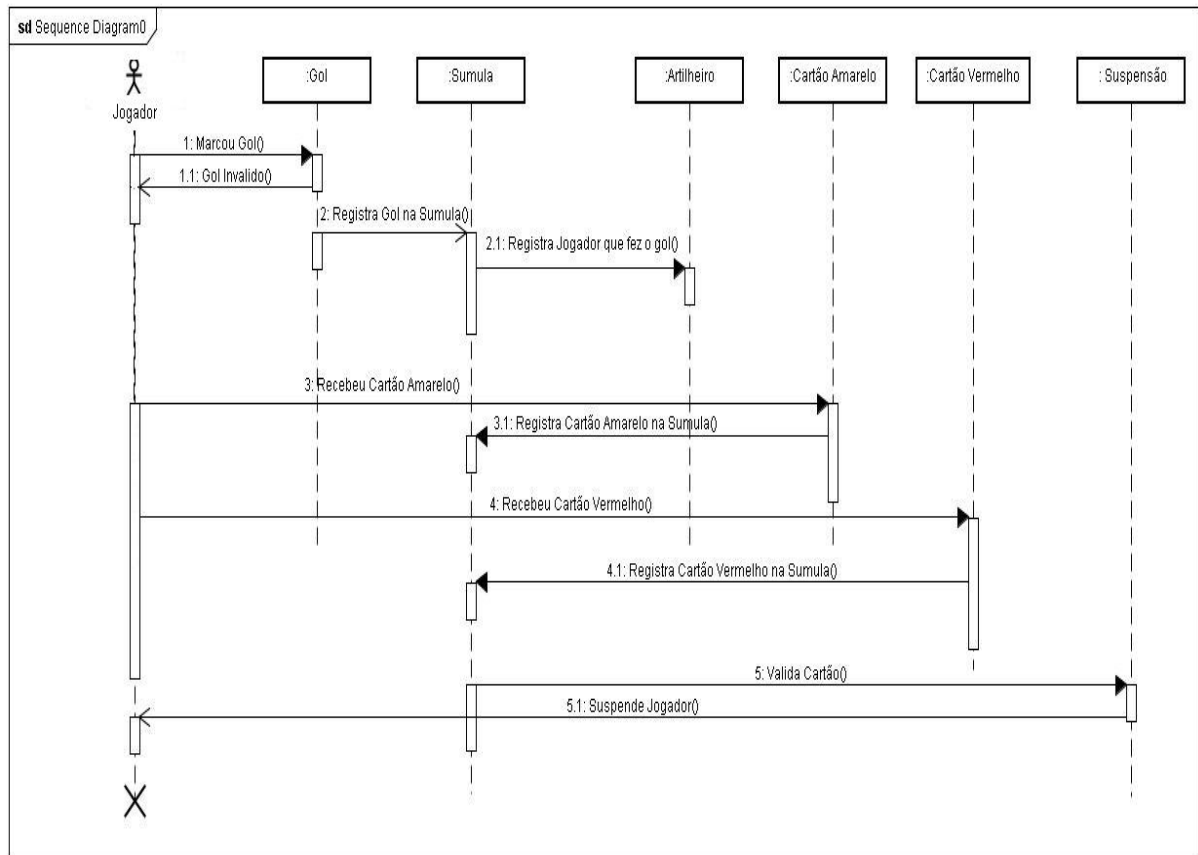


Diagrama de Sequência de Jogador SÚMULA

7.27. DIAGRAMA DE SEQUÊNCIA DE SÚMULA

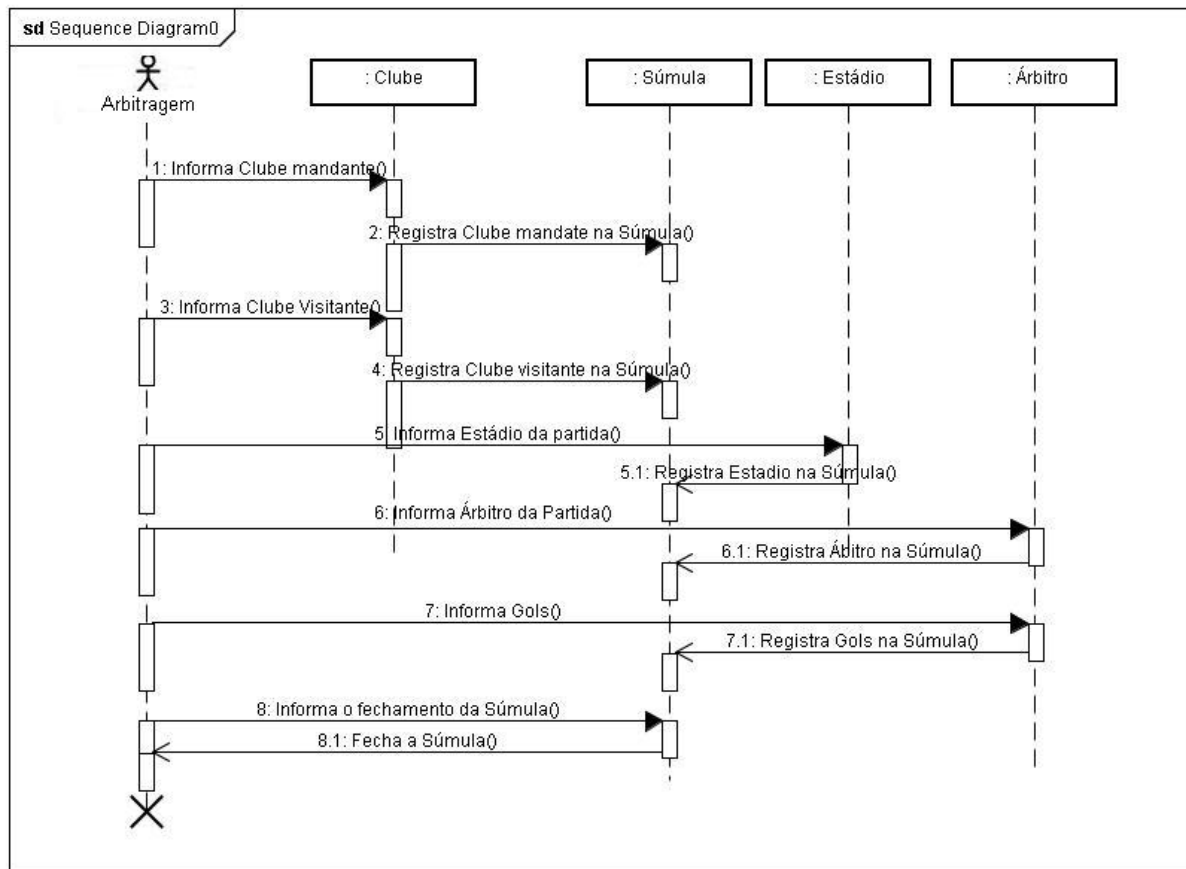


Diagrama de Sequência de Súmula

8. DIAGRAMA DE CLASSES

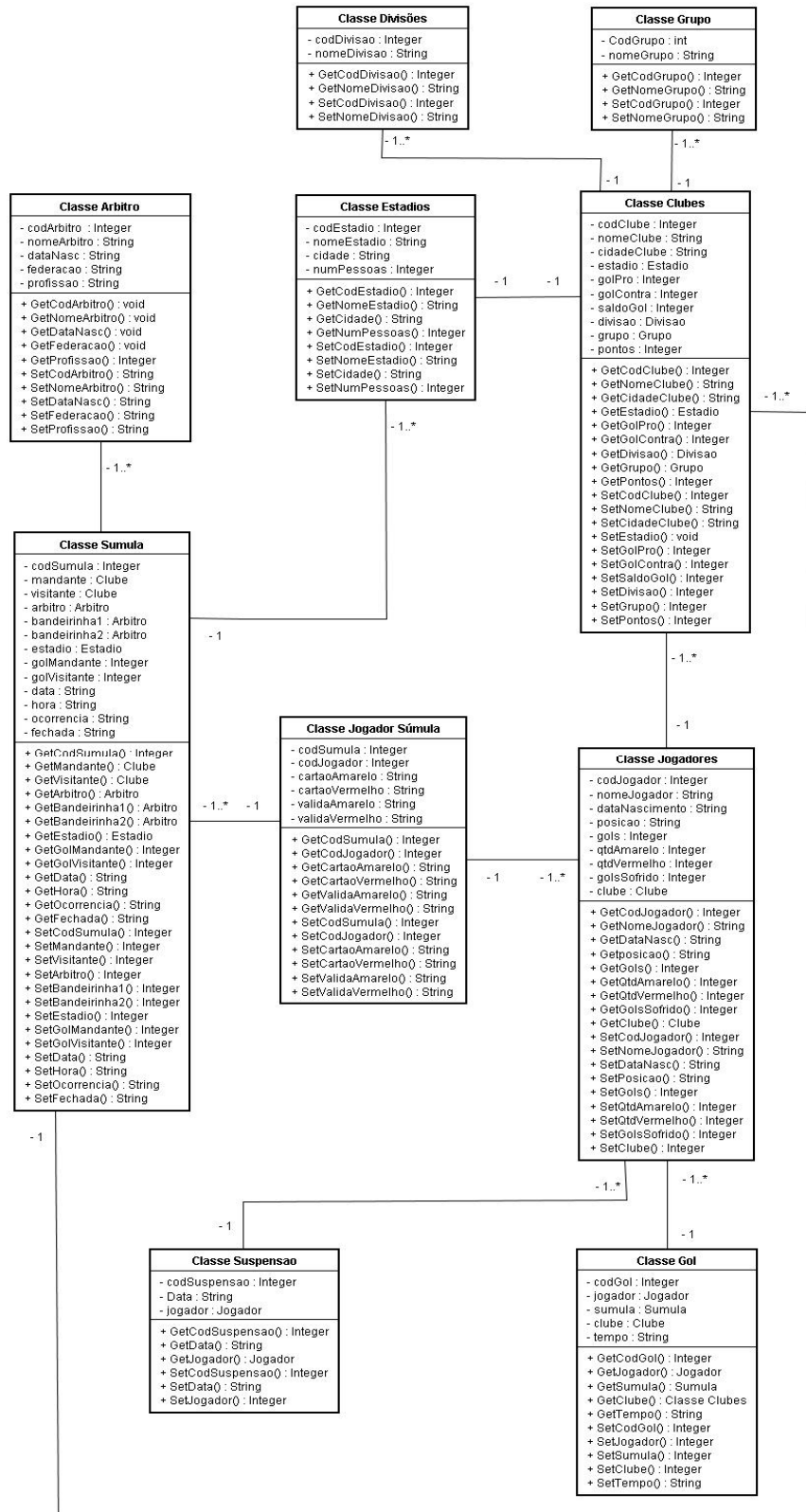


Diagrama de Classes

9. IMPLEMENTAÇÃO

Nesta seção serão apresentadas as telas do Software de Súmula Eletrônica, desenvolvido em Delphi com banco de dados MySql.

9.1. TELA INICIAL

Esta será a tela principal do software, onde o usuário terá acesso às opções de Cadastro, Súmula e Relatórios.

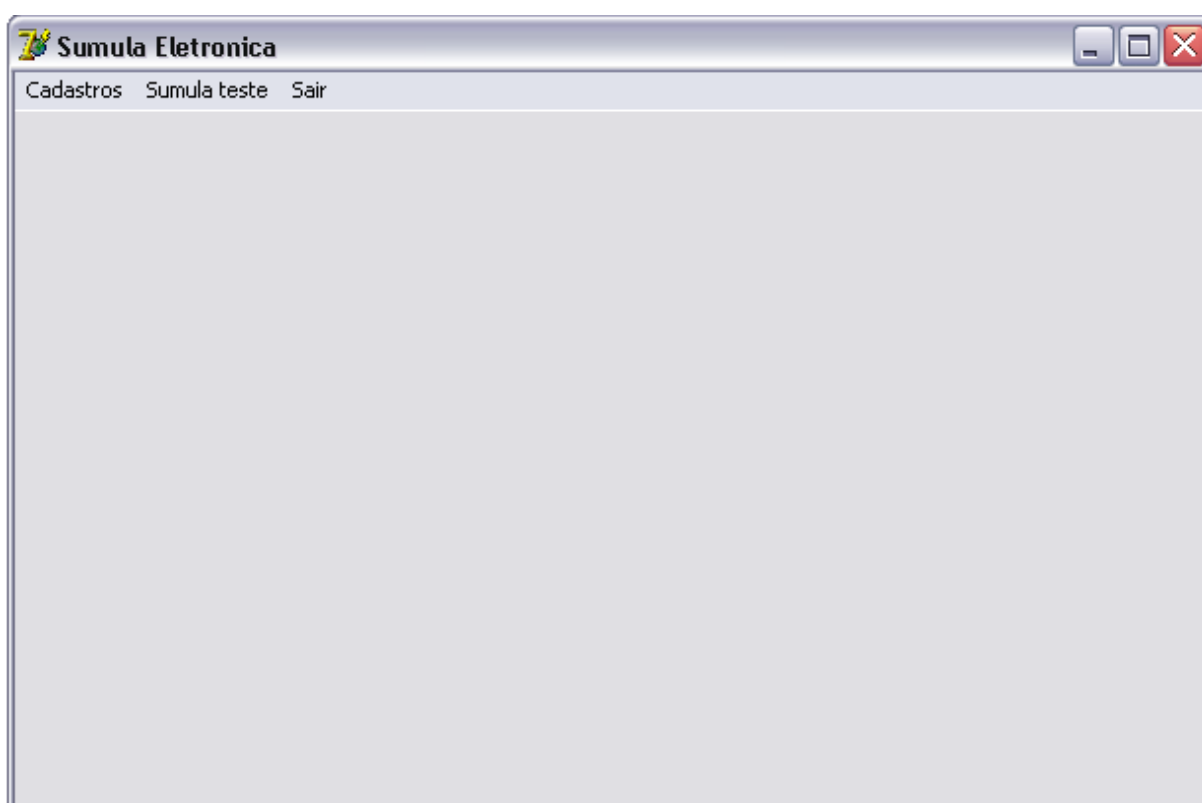


Figura 12: Tela Inicial

9.2. CADASTRO

Para chegar ao cadastro o usuário deverá acessar a guia cadastro na tela inicial que exibirá as opções de cadastro do software.

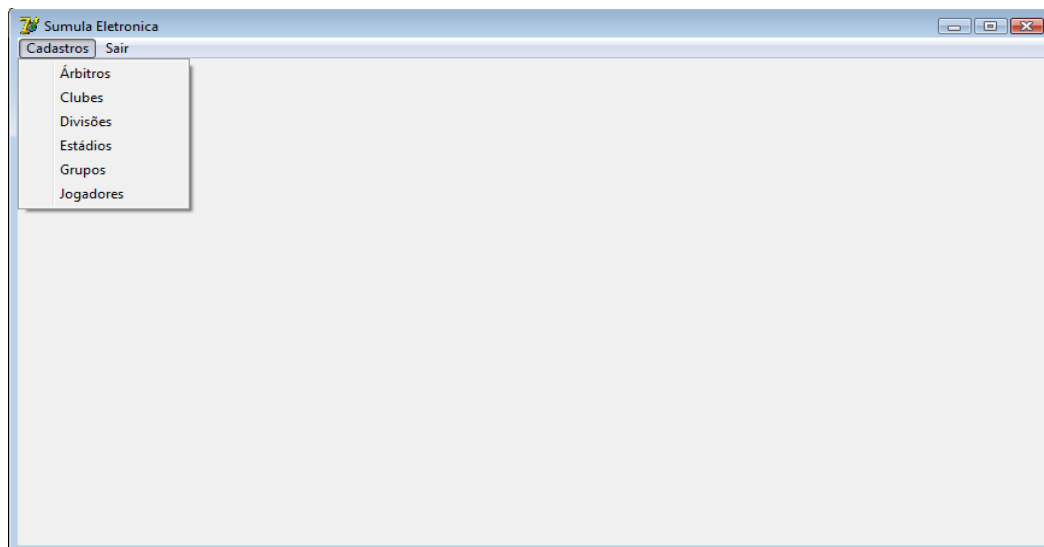


Figura 13: Tela Inicial – Cadastro

Ao clicar em uma das opções aparecerá o cadastro escolhido pelo cliente, vide (figura 14), que exibe o cadastro de jogadores do software, onde o usuário pode inserir, excluir, gravar, alterar e pesquisar os jogadores.

A imagem mostra uma janela de software intitulada 'Cadastro de Jogadores'. O título da janela é 'Cadastro de Jogadores'. O formulário contém os seguintes campos: 'Nome:' com um campo de texto; 'Nascimento:' com um campo de texto; 'Clube:' com um menu suspenso; 'Posição:' com um campo de texto. Abaixo, há duas seções: 'Gols' com subcampos 'Marcados:' e 'Sofridos:', cada um com um campo de texto contendo o valor '0'; e 'Cartões' com subcampos 'Amarelo:' e 'Vermelhos:', cada um com um campo de texto contendo o valor '0'. Na base da janela, há uma barra com quatro botões: 'Novo', 'Salvar', 'Pesquisar' e 'Sair'.

Figura 14: Cadastro de Jogadores.

9.3. SÚMULA

Acessando o guia Súmula, o usuário terá acesso ao formulário onde serão manipulados todos os dados das partidas do campeonato.

The screenshot shows a software window titled "Sumula Jogo". At the top right, there is a "Tempo" (Time) display showing "00:00" and control buttons for play, pause, and stop. Below this, a small box contains the number "0".

The main section is titled "Campeonato Amador de Cândido Mota". It contains a score display "0 X 0" between two dropdown menus for the teams. Below the score, there are fields for "Data" (Date) set to "28/10/2008", "Horário" (Time) set to "16:00:00", and "Estádio" (Stadium) with a dropdown menu.

The "Arbitragem" (Referee) section includes three rows for "Árbitro" (Referee), "Auxiliar 1" (Assistant 1), and "Auxiliar 2" (Assistant 2). Each row has a dropdown menu for the name and a text field for the "Federação" (Federation).

The "Relação dos Atletas Titulares" (List of Starting Players) section contains two large empty tables, one for each team, with columns for player information.

At the bottom, there are three buttons: "Anotar Gol" (Record Goal) with a soccer ball icon, "Anotar Cartão Amarelo" (Record Yellow Card) with a yellow card icon, and "Anotar Cartão Vermelho" (Record Red Card) with a red card icon.

The "Ocorrências" (Incidents) section at the very bottom has a text area labeled "MOcorrencias".

Figura 15: Súmula.

11. CONCLUSÃO

Como o futebol é uma paixão nacional, em inúmeras cidades do país existem ligas e federação que organizam campeonatos amadores e até profissionais. Más, até então todo controle do sistema de informações das ligas e federações não foi contemplado com uma tecnologia para auxiliar em todo o processo de armazenamento, controle e movimentação de informações.

Este trabalho buscou contribuir para a informatização de todo um sistema na execução de um jogo de futebol, que sempre foi controlado de forma manual sem a interferência da tecnologia.

Implementado em um campeonato, este software tem como objetivo melhorar o processo de trabalho da comissão de arbitragem e da federação, armazenando os dados das partidas em seu banco de dados.

E para finalizar, conclui-se, que o desenvolvimento deste software é um grande avanço em conhecimentos de programação e conhecimentos em UML, que serão de grande importância para estar apto a entrar no mercado de trabalho após a conclusão do curso.

Visando sua comercialização, existem diversas modificações e atualizações no sistema para torná-lo mais inteligente e seguro. Alterações podem ser feitas na súmula para tornar mais pratica a inclusão dos times e seus respectivos jogadores, e incluindo um cadastro de cidades, tornaria mais pratica a inclusão dos clubes apenas selecionando a cidade em uma *combobox*. Porém, uma mudança essencial seria investir em registros de segurança, tornando o acesso a base de dados restrita. Obtendo assim, um software completo para a comercialização.

12. REFERÊNCIAS BIBLIOGRÁFICAS

FOWLER, Martin e SCOTT. *UML essencial: um breve guia para a linguagem padrão de modelagem de objetos*. trad. Vera Pezerico e Christian ThomasPrice. 2 ed. Porto Alegre: Bookman, 2000.

LARMAN, Craig. *Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos*. trad, Luiz A. Meirelles Salgado. Porto Alegre: Bookman, 2000.

OLIVEIRA, J.F. *Metodologia para Desenvolvimento de Projetos de Sistemas: Guia Prático*. 3ª edição. São Paulo: Érica, 1999

YOURDON, Edward. *Análise e projeto orientados a objetos*. São Paulo: Makron Books. 1999

Professor Marcos Costa Sampaio. **UML**

Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/>

Acessado em 02/06/2008

Professor Marcelo Fontana. **UML Aula 1**

Disponível em: http://www.marcelofontana.com.br/OO/Aula_1_UML.pdf

Acessado em 02/06/2008

