

Carlos Alberto Ceolato Junior

CHUTEGOL - CHance de Um Time Em GOLs
Um sistema para prognósticos de jogos de futebol

Florianópolis – SC
2007

Carlos Alberto Ceolato Junior

CHUTEGOL - CHance de Um Time Em GOLs
Um sistema para prognósticos de jogos de futebol

Monografia apresentada ao programa de Bacharelado em Sistemas de Informação da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

Orientador: Dalton Francisco de Andrade

Florianópolis – SC
2007

Monografia de graduação sob o título CHUTEGOL - CHance de Um Time Em GOLs - Um sistema para prognósticos de jogos de futebol, defendida por Carlos Alberto Ceolato Junior em Florianópolis, Santa Catarina, banca examinadora constituída pelos professores:

Prof. PhD. Dalton Francisco de Andrade
Orientador
Universidade Federal de Santa Catarina

Prof. Dr. Ronaldo dos Santos Mello
Universidade Federal de Santa Catarina

Prof. Dr. João Cândido Dovicchi
Universidade Federal de Santa Catarina

Dedico este trabalho
à minha esposa Cristiane,
pelo carinho, companhia e
compreensão dedicados
durante esta etapa de
nossas vidas.

Índice

1.	Introdução	12
1.1.	Considerações Iniciais	12
1.2.	Objetivos do Trabalho	13
1.2.1.	Objetivo geral	13
1.2.2.	Objetivos específicos	13
2.	Futebol	14
3.	O Modelo Probabilístico	16
3.1.	Introdução	16
3.2.	Cálculo das probabilidades	16
3.3.	Considerações para o cálculo das probabilidades	18
3.4.	Estimação de parâmetros	19
3.5.	Limites no cálculo das probabilidades	20
4.	O Banco de Dados	25
4.1.	Estratégia Descendente	25
4.2.	Estratégia “Inside-out”	26
4.3.	Modelagem ER	26
4.4.	Esquema Relacional	28
4.5.	Escolha do SGBD	29
4.6.	Código SQL das tabelas	30
5.	Análise e Projeto do Sistema	32
5.1.	Análise do Sistema	32
5.1.1.	Requisitos Funcionais	32
5.1.2.	Requisitos não funcionais	34
5.1.3.	Diagrama de Casos de Uso	34
5.2.	Projeto do Sistema	34
5.2.1.	Linguagem de Programação	35
5.2.2.	Ferramentas de Desenvolvimento	35
5.2.3.	Definição dos elementos do sistema	35
5.2.4.	Diagrama de Classes do Sistema	36
6.	Chutegol	37
6.1.	Projeto Web	37
6.2.	Classes de Entidade do Banco de Dados	37
6.3.	Páginas JSF de Classes de Entidades	37
6.4.	Lógica do Negócio	38
6.4.1.	Criar parâmetros	38
6.4.2.	Atualizar Parâmetros	39
6.4.3.	Calcular Probabilidades da partida	40
6.4.4.	Calcular probabilidades dos possíveis placares	41
6.5.	Telas do Sistema - Usuário	42
6.5.1.	Tela inicial	42
6.5.2.	Clube – List.jsp	43
6.5.3.	Partida – List.jsp	44
6.5.4.	Partida – Detail.jsp	45
6.6.	Telas do sistema – Administrador	46
6.6.1.	Tela Inicial	46
6.6.2.	Campeonato – List.jsp	47
6.6.3.	Campeonato – New.jsp	48
6.6.4.	Campeonato – Edit.jsp	49

6.6.5.	Temporada – List.jsp	50
6.6.6.	Temporada – New.jsp.....	51
6.6.7.	Temporada – Edit.jsp.....	52
6.6.8.	Clube – List.jsp	53
6.6.9.	Clube – New.jsp.....	54
6.6.10.	Clube – Edit.jsp	55
6.6.11.	Partida – List.jsp.....	56
6.6.12.	Partida – New.jsp	57
6.6.13.	Partida – Edit.jsp	58
7.	Resultados	59
7.1.	Resumo da análise	73
8.	Considerações Finais e Trabalhos futuros	74
9.	Bibliografia	75
10.	Apêndice A - Artigo.....	77
11.	Apêndice B – Código Fonte.....	83
11.1.	index.jsp	83
11.2.	Admin – index.jsp	83
11.3.	Admin – Campeonato – New.jsp	84
11.4.	Admin – Campeonato – List.jsp.....	85
11.5.	Admin – Campeonato – Edit.jsp	86
11.6.	Admin – Campeonato – Detail.jsp	87
11.7.	Admin – Clube – New.jsp	88
11.8.	Admin – Clube – Edit.jsp	88
11.9.	Admin – Clube – List.jsp.....	89
11.10.	Admin – Clube – Detail.jsp	91
11.11.	Admin – Temporada – New.jsp	91
11.12.	Admin – Temporada – Edit.jsp	92
11.13.	Admin – Temporada – List.jsp.....	93
11.14.	Admin – Temporada – Detail.jsp	94
11.15.	Admin – Partida – New.jsp	95
11.16.	Admin – Partida – Edit.jsp	96
11.17.	Admin – Partida – List.jsp.....	98
11.18.	Admin – Partida – Detail.jsp	101
11.19.	User – Clube – List.jsp	105
11.20.	User – Partida – List.jsp	107
11.21.	User – Partida – Detail.jsp.....	110
11.22.	login.jsp	113
11.23.	error.jsp	114
11.24.	chutegol.css	114
11.25.	_cabecalho.jsfp.....	117
11.26.	_rodape.jsfp.....	117
11.27.	_menuAdmin.jsfp.....	117
11.28.	_menuUser.jsfp	117
11.29.	web.xml	117
11.30.	faces-config.xml.....	119
11.31.	context.xml	122
11.32.	persistence.xml.....	123
11.33.	Persistência - Campeonato.java.....	123
11.34.	Persistência - Clube.java	126
11.35.	Persistência - Partida.java.....	130

11.36.	Persistência - Parametro.java	135
11.37.	Persistência - ParametroPK.java.....	139
11.38.	Persistência - Temporada.java	141
11.39.	Admin - CampeonatoController.java.....	144
11.40.	Admin - CampeonatoConverter.java	148
11.41.	Admin - ClubeControler.java	149
11.42.	Admin - ClubeConverter.java	152
11.43.	Admin - ParametroController.java	153
11.44.	Admin - ParametroConverter.java	157
11.45.	Admin - ParametroPKConverter.java	159
11.46.	Admin - PartidaController.java.....	160
11.47.	Admin - PartidaConverter.java	170
11.48.	Admin - TemporadaController.java.....	171
11.49.	Admin - TemporadaConverter.java.....	175
11.50.	User - ClubeController.java	175
11.51.	User - ClubeConverter.java	178
11.52.	User - PartidaController.java	179
11.53.	User - PartidaConverter.java	185
11.54.	Útil - Probabilidade.java.....	185
11.55.	Útil - ProbComparator.java	187

Índice de Figuras

Figura 1 – Campeonato Brasileiro 2006.....	20
Figura 2 – Campeonato Catarinense 2007	21
Figura 3 – Médias para Cálculo das probabilidades.....	21
Figura 4 - Entidade Partida	26
Figura 5 - Atributos de "Partida"	27
Figura 6 - Identificando novas Entidades	27
Figura 7 - Modelo ER Completo	27
Figura 8 - Diagrama de Casos de Uso	34
Figura 9 - Diagrama de Classes.....	36
Figura 10 - Tela Inicial do Sistema – Usuário.....	42
Figura 11 - Lista de Clubes - Usuário.....	43
Figura 12 - Lista de Partidas – Usuário.....	44
Figura 13 - Detalhe de Partida - Usuário.....	45
Figura 14 - Tela inicial do Sistema – Administrador	46
Figura 15 - Lista de Campeonato - Administrador.....	47
Figura 16 - Novo Campeonato – Administrador	48
Figura 17 - Editar Campeonato - Administrador	49
Figura 18 - Lista de Temporadas - Administrador	50
Figura 19 - Administrador - Nova Temporada	51
Figura 20 - Editar Temporada - Administrador	52
Figura 21 - Lista de Clube - Administrador	53
Figura 22 - Novo Clube – Administrador	54
Figura 23 - Editar Clube - Administrador	55
Figura 24 - Lista de Partida - Administrador.....	56
Figura 25 - Nova Partida – Administrador	57
Figura 26 - Editar Partida - Administrador	58
Figura 27 - Rodada 1	60
Figura 28 - Rodada 2	60
Figura 29 - Rodada 3	60
Figura 30 - Rodada 4	60
Figura 31 - Rodada 5	61
Figura 32 - Rodada 6	61
Figura 33 - Rodada 7	61
Figura 34 - Rodada 8	61
Figura 35 - Rodada 9	62
Figura 36 - Rodada 10	62
Figura 37 - Rodada 11	62
Figura 38 - Rodada 12	62
Figura 39 - Rodada 13	63
Figura 40 - Rodada 14	63
Figura 41 - Rodada 15	63
Figura 42 - Rodada 16	63
Figura 43 - Rodada 17	64
Figura 44 - Rodada 18	64
Figura 45 - Rodada 19	64
Figura 46 - Rodada 20	64
Figura 47 - Rodada 21	65

Figura 48 - Rodada 22	65
Figura 49 - Rodada 23	65
Figura 50 - Rodada 24	65
Figura 51 - Rodada 25	66
Figura 52 - Rodada 26	66
Figura 53 - Rodada 27	66
Figura 54 - Estadísticas Rodada 5	67
Figura 55 - Estadísticas Rodada 6	67
Figura 56 - Estadísticas Rodada 7	68
Figura 57 - Estadísticas Rodada 8	68
Figura 58 - Estadísticas Rodada 9	68
Figura 59 - Estadísticas Rodada 10	68
Figura 60 - Estadísticas Rodada 11	69
Figura 61 - Estadísticas Rodada 12	69
Figura 62 - Estadísticas Rodada 13	69
Figura 63 - Estadísticas Rodada 14	69
Figura 64 - Estadísticas Rodada 15	70
Figura 65 - Estadísticas Rodada 16	70
Figura 66 - Estadísticas Rodada 17	70
Figura 67 - Estadísticas Rodada 18	70
Figura 68 - Estadísticas Rodada 19	71
Figura 69 - Estadísticas Rodada 20	71
Figura 70 - Estadísticas Rodada 21	71
Figura 71 - Estadísticas Rodada 22	71
Figura 72 - Estadísticas Rodada 23	72
Figura 73 - Estadísticas Rodada 24	72
Figura 74 - Estadísticas Rodada 25	72
Figura 75 - Estadísticas Rodada 26	72
Figura 76 - Estadísticas Rodada 27	73

Resumo

Futebol é, no mundo inteiro, um dos esportes que mais fascina as pessoas. Todos os anos, centenas de clubes entram em campo com um único objetivo, ser campeão.

Descobrir o resultado das partidas de futebol para fazer apostas entre amigos, concorrer a prêmios ou tentar ganhar na loteria é um ingrediente extra para os torcedores deste esporte.

O objetivo deste trabalho foi desenvolver um sistema de informação capaz de auxiliar o usuário no prognóstico das partidas de futebol, fornecendo as probabilidades de ocorrer vitória do time mandante, o empate ou vitória do time visitante. Para tanto, definimos um modelo probabilístico baseado em Poisson para efetuar os cálculos, construímos uma base de dados para armazenar as informações e desenvolvemos um sistema web capaz de exibir as informações para que o usuário possa através de sua análise, definir seus prognósticos.

Palavras-Chave: futebol, prognóstico, modelo probabilístico.

Abstract

Football is, worldwide, one of the sports that most fascinates people. Every year, hundreds of clubs come into the field with a single aim, to be champion.

Discovering the outcome of the football matches to make bets between friends, compete for prizes or try to win the lottery is an extra ingredient for the fans of this sport.

The objective of this work is to develop an information system capable to help the user in the prognosis of football matches, giving the probability to occur victory of the team's principal , the draw or victory the visiting team . For it, we defined a model based in Poisson to make the calculations, building a database to store the information and developed a web system able to display the information so the user can through its analysis, define their prognoses.

Keyword: football, prognoses, probabilistic model

1. Introdução

1.1. Considerações Iniciais

O Brasil é conhecido como o país do futebol, todos os anos, diversos clubes se preparam para disputar uma taça, seja em campeonatos estaduais, regionais, nacionais ou internacionais, o objetivo é único: “Ser Campeão”. De olho neste mercado, o governo brasileiro disponibiliza duas loterias nas quais o apostador deve acertar o resultado de partidas, a LOTECA, onde o objetivo é acertar entre: vitória do time mandante, empate ou vitória do time visitante em quatorze jogos escolhidos pela entidade organizadora da loteria, e a LOTOGOL cujo objetivo é acertar o placar em gols de cinco jogos com opção de zero, um, dois, três ou mais gols para cada time. Alguns sites na internet administram bolões onde os usuários devem acertar o resultado de partidas para conquistar pontos visando ser o ganhador de prêmios oferecidos, outros funcionam como casas de apostas on-line, onde os apostadores do mundo todo apostam no resultado de qualquer evento esportivo.

Nesse contexto, este trabalho se propõe a desenvolver um sistema de informação que auxilie o usuário no prognóstico de jogos de futebol através do cálculo das probabilidades de ocorrer vitória do time mandante, empate ou vitória do time visitante numa partida de futebol.

Para o cálculo das probabilidades, será usado como parâmetro, o desempenho dos clubes nas últimas partidas realizadas jogando como mandante ou como visitante

1.2. Objetivos do Trabalho

1.2.1. Objetivo geral

Desenvolver um sistema baseado em dados de jogos de futebol, que permita ao usuário fazer o prognóstico de uma partida.

1.2.2. Objetivos específicos

- 1) Definir um modelo probabilístico que permita calcular a probabilidade de vitória / empate / derrota de cada time em um confronto futebolístico;
- 2) Construir uma base de dados a ser alimentada por resultados de jogos de futebol;
- 3) Desenvolver um sistema que seja capaz de acessar esta base de dados e manipular as informações para fornecer ao usuário a probabilidade de cada time em uma partida, calculada pelo modelo apresentado em 1).

2. Futebol

A organização do futebol coube aos ingleses, porém a sua origem data de muito antes. [Duarte, 1997] relata que há cerca de 2600 anos a.C. na China, foi criado um jogo denominado KEMARI onde oito jogadores de cada lado tentavam passar uma bola entre duas estacas. Ainda na China, durante a dinastia do imperador Huang-ti, havia o costume de se chutar os crânios dos inimigos derrotados num esporte denominado TSU-CHU.

Na Grécia, há cerca de 3500 anos, um esporte disputado com os pés por duas equipes de nove a quinze jogadores era denominado EPYSKIROS.

No Império Romano, por volta de 200 anos a.C. disputava-se o HARPASTUM, que era um exercício militar jogado com uma bola feita de bexiga de boi.

Na França, jogava-se o SOULE, na Inglaterra, nesta mesma época, jogava-se um futebol selvagem, violento e sem regras. Na Itália, em 1529 criou-se o Calcio.

No Brasil, o futebol chegou em 1894, quando Charles Miller trouxe em sua bagagem, vindo da Inglaterra, duas bolas que possibilitaram aos Brasileiros praticarem um pouco do esporte que veio a se tornar uma paixão nacional.

Atualmente, o órgão que controla e organiza o futebol pelo mundo é a Fédération Internationale de Football Association, conhecida pelo acrônimo FIFA, fundada em 1904 com sede em Zurique e 217 países membros.

A FIFA (www.fifa.com) organiza entre outros campeonatos, a Copa do Mundo de futebol, cuja fase final atualmente é disputada a cada quatro anos por 32 países. O último campeonato foi disputado na Alemanha em 2006 e foi vencido pela seleção da Itália. Ela trabalha em conjunto com as confederações regionais de futebol, que são:

- OFC – Oceania;

- CAF – África;
- AFC – Ásia;
- UEFA – Europa;
- CONCACAF – Américas do Norte e Central;
- CONMEBOL – América do Sul.

Esta última é a confederação que organiza todos os campeonatos internacionais que envolvam países da América do Sul. Com sede em Assunção no Paraguai foi fundada em 1916 pelas Confederações da Argentina, Chile, Uruguai e Brasil.

A entidade máxima do futebol no Brasil é a Confederação Brasileira de Futebol, conhecida pelo acrônimo CBF (www.cbf.com.br), foi fundada em 1919 e é a organizadora dos campeonatos nacionais de futebol como o Campeonato Brasileiro que será objeto de análise deste trabalho.

A CBF tem como filiadas, as Federações Estaduais de Futebol que organizam os campeonatos estaduais de cada unidade da federação que serão também objeto de estudo deste trabalho.

3. O Modelo Probabilístico

3.1. Introdução

O objetivo deste capítulo é desenvolver um modelo probabilístico capaz de determinar a chance de vitória de cada equipe numa partida de futebol.

Segundo [Barbetta, 2004], “muitas vezes, as condições do experimento não permitem deduzir o resultado, mas somente a chance (ou a probabilidade) de possíveis resultados”. Há diversos fatores que podem interferir no resultado de uma partida de futebol, como por exemplo: fenômenos da natureza (chuva, direção do vento, temperatura), lesão ou expulsão de um jogador durante a partida, dentre muitos outros, sendo praticamente impossível prever o resultado exato de uma partida. Entretanto, com os dados estatísticos que podemos coletar a respeito de jogos passados das equipes, é possível determinar a chance de cada resultado acontecer e com isso, termos um indicativo do resultado provável para uma partida futura.

3.2. Cálculo das probabilidades

Para calcularmos o provável resultado de uma partida, precisamos ter um indicativo da probabilidade do time mandante fazer X gols em uma partida e do time visitante fazer Y gols em uma partida. Segundo [Arruda, 2000], a classe Poisson bivariada de Holgate é a mais indicada para a modelagem de resultados de partidas de futebol, pois atende aos três critérios seguintes:

1. As distribuições normais devem ser Poisson – No caso, o número de gols de cada equipe, pode ser considerado conforme cita [Barbetta, 2004], “situações

em que se avalia o número de ocorrências de um tipo de evento por unidade de tempo”.

2. A distribuição conjunta deve possuir suporte pleno, ao menos perto da origem.
3. A distribuição conjunta e as marginais devem ser infinitamente divisíveis. – No caso, uma partida de futebol pode ser dita como o resultado de um jogo de 90 minutos ou como o somatório de noventa jogos de 1 minuto cada.

Então, podemos usar a distribuição de Poisson que é dada pelas fórmulas abaixo:

$$P(X = x) = \frac{e^{-\lambda_x} (\lambda_x)^x}{x!}, \lambda_x > 0 \quad (1)$$

$$P(Y = y) = \frac{e^{-\lambda_y} (\lambda_y)^y}{y!}, \lambda_y > 0 \quad (2)$$

Onde, para o problema proposto sobre jogos de futebol, X é o número de gols do time mandante em uma partida, λ_x é a média de gols do time mandante por partida, Y é o número de gols do time visitante por partida e λ_y é a média de gols do time visitante por partida.

Conforme teorema mencionado por [Griffiths, 1979], “toda distribuição bivariada de Poisson infinitamente divisível possui correlação não-negativa”. Esse teorema apóia à noção de que, quaisquer que sejam duas equipes, X e Y, podemos ter apenas uma das seguintes situações:

1. X é melhor do que Y – Neste caso, teremos a vitória do time mandante e a probabilidade de sua vitória pode ser calculada por:

$$P(M) = \sum_{x>y} P(X = x, Y = y) \quad (3)$$

2. X e Y são equivalentes – Neste caso, ocorrerá um empate cuja probabilidade pode ser calculada por:

$$P(E) = \sum_{x=y} P(X = x, Y = y) \quad (4)$$

3. Y é melhor do que X – Neste caso, teremos a vitória do time visitante com probabilidade dada por:

$$P(V) = \sum_{x < y} P(X = x, Y = y) \quad (5)$$

Podemos dizer ainda que:

$$P(M) + P(E) + P(V) = 1 \quad (6)$$

3.3. Considerações para o cálculo das probabilidades

No presente trabalho, admitiremos que haja independência entre as quantidades de gols marcados pela equipe mandante e visitante em cada partida. Desta forma podemos admitir que:

$$P(X = x, Y = y) = P(X = x) \times P(Y = y) \quad (7)$$

Desta forma, substituindo em (3), (4) e (5) teremos:

- Para a vitória do time mandante

$$\sum_{x > y} (P(X = x) \times P(Y = y)) \quad (8)$$

- Para o empate

$$\sum_{x=y} (P(X = x) \times P(Y = y)) \quad (9)$$

- Para a vitória do visitante

$$\sum_{x < y} (P(X = x) \times P(Y = y)) \quad (10)$$

3.4. Estimação de parâmetros

O maior problema deste modelo é determinar os parâmetros, no caso, as médias a serem usadas.

Segundo [Witter, 1990], “no futebol, o objetivo a ser atingido é sempre o gol. Assim, durante uma partida disputada entre duas equipes, a defesa e o ataque são constantes”.

[Arruda, 2000] propõe dentre outros, métodos implícitos para a determinação dos parâmetros. Estes métodos são motivados pelo conceito de probabilidade subjetiva citado por [Barnett, 1982] no qual a probabilidade é uma medida de incerteza individual que todo o ser humano está capacitado a manifestar na avaliação de situações de incerteza.

Poucas coisas são tão incertas quanto o futebol, como diz a célebre frase atribuída a Vicente Matheus, “o jogo só termina quando acaba”, cuja veracidade pode ser comprovada pelas inúmeras situações nas quais o resultado de uma partida mudou quando muitos torcedores já haviam deixado o estádio rumo a suas casas.

Partindo deste pressuposto, podemos propor uma estimativa de λ_x como uma média entre a quantidade de gols que o time mandante faz quando joga em casa e a quantidade de gols que o time visitante sofre quando joga fora de casa, da mesma forma estimamos λ_y como uma média entre a quantidade de gols que o time visitante faz quando joga fora de casa e a quantidade de gols que o time mandante sofre quando joga em casa.

Podemos então calcular os parâmetros da seguinte forma:

$$\hat{\lambda}_x = \frac{X_{Favor} + Y_{Contra}}{2} \quad (11)$$

Onde X_{Favor} é a média de gols feitos pelo time mandante e Y_{Contra} é a média de gols sofridos pelo time visitante.

$$\hat{\lambda}_y = \frac{Y_{Favor} + X_{Contra}}{2} \quad (12)$$

Onde Y_{Favor} é a média de gols feitos pelo time visitante e X_{Contra} é a média de gols sofridos pelo time mandante.

3.5. Limites no cálculo das probabilidades

Analisando os dados do Campeonato Brasileiro de 2006 no qual foram disputados 380 jogos, tivemos a distribuição de gols / partida, conforme mostra a Figura 1, levando em consideração dois dados para cada jogo, sendo um referente ao time mandante e outro ao visitante, totalizando 760 registros.

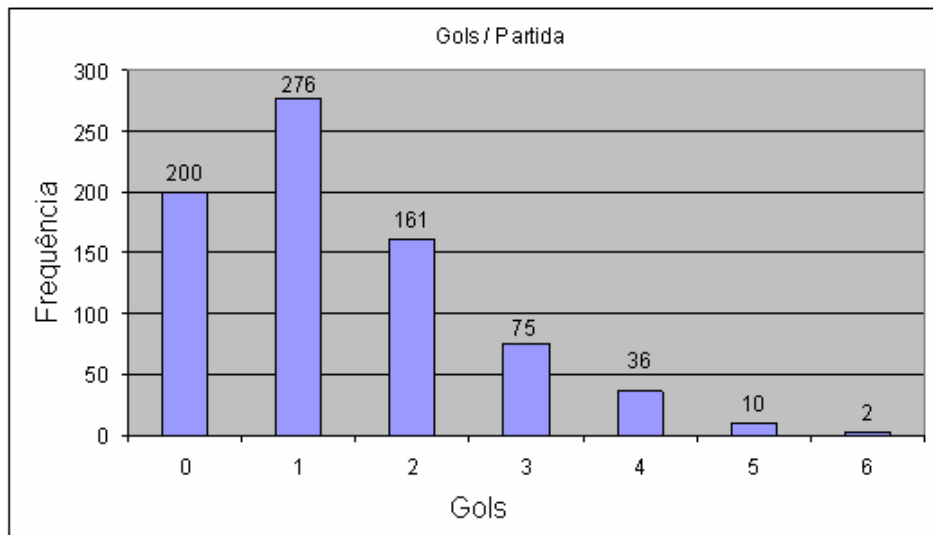


Figura 1 – Campeonato Brasileiro 2006

Podemos observar que em 98,42% dos dados, o número de gols do time foi no máximo quatro e em apenas 1,58% o número de gols foi superior a quatro.

Analisando o campeonato Catarinense do ano de 2007 temos dados semelhantes, conforme mostra a Figura 2, em 97,72% das partidas, o número máximo de quatro gols, ou seja, o time mandante ou o time visitante fizeram no máximo quatro gols.

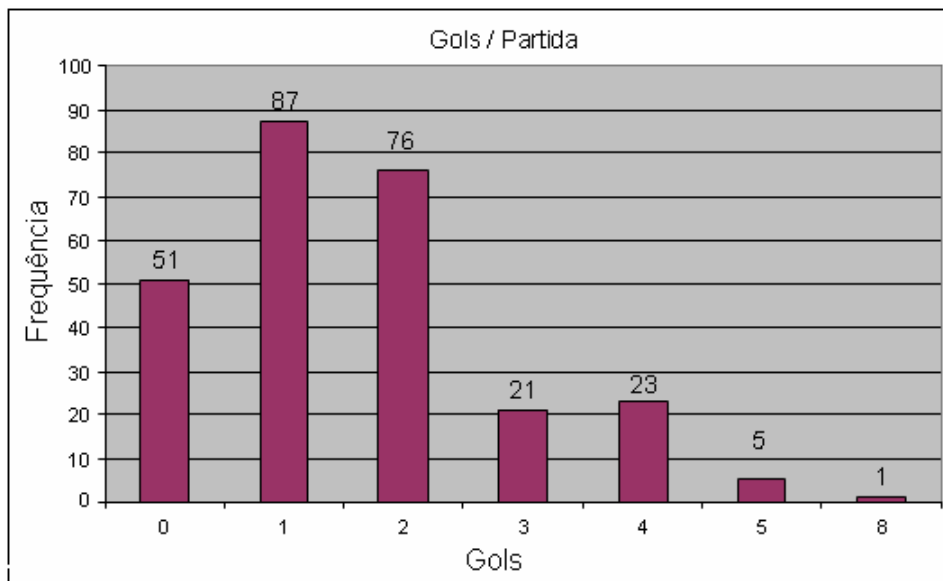


Figura 2 – Campeonato Catarinense 2007

Em face disto e pelo fato da loteria da Caixa, que será usada como um dos objetos deste trabalho, também considerar um número de quatro gols como limite nas apostas, usaremos para os cálculos das probabilidades os valores de 0 a 4 gols sendo o valor “+” como sendo a probabilidade de ocorrer quatro ou mais gols.

Para ilustrar, vamos simular um confronto da final do campeonato catarinense de 2007, disputada entre Chapecoense e Criciúma. Durante o campeonato, a média de gols a favor e contra dos dois times apresentou valores conforme ilustrados na Figura 3:

Média de Gols a Favor	Casa	Fora
Chapecoense	1,54	1,90
Criciúma	2,81	1,81

Média de Gols Sofridos	Casa	Fora
Chapecoense	0,54	1,54
Criciúma	1,09	0,90

Figura 3 – Médias para Cálculo das probabilidades

Para calcularmos a probabilidade de cada time fazer n gols, tomaremos como λ o valor médio entre os gols feitos quando o time joga em casa e os gols sofridos pelo time visitante quando joga fora de casa,

$$\text{Média de gols da Chapecoense: } \frac{1,54 + 0,90}{2} = 1,22$$

A probabilidade de este time fazer 0 (zero) gols numa partida jogando em casa fica:

$$P(x = 0) = \frac{1,22^0 e^{-1,22}}{0!} = 0,2953$$

Para 1 (um) gol, teremos:

$$P(x = 1) = \frac{1,22^1 e^{-1,22}}{1!} = 0,3602$$

Para 2 (dois) gols a probabilidade é:

$$P(x = 2) = \frac{1,22^2 e^{-1,22}}{2!} = 0,2197$$

Para 3 (três) gols teremos:

$$P(x = 3) = \frac{1,22^3 e^{-1,22}}{3!} = 0,0894$$

E por fim, a probabilidade de termos mais de três gols, é dada por:

$$P(x = +) = 1 - [P(0) + P(1) + P(2) + P(3)] =$$

$$P(x = +) = 1 - (0,2953 + 0,3602 + 0,2197 + 0,0894) = 0,0354$$

Fazendo os mesmos cálculos para o Criciúma, onde λ passa a ser um valor médio entre os gols feitos jogando fora de casa e os gols sofridos pelo time mandante quando joga em casa teremos os seguintes cálculos:

Média de gols do Criciúma: $\frac{1,81 + 0,54}{2} = 1,175$

Os cálculos das probabilidades de gols do Criciúma ficam:

$$P(y = 0) = \frac{1,175^0 e^{-1,175}}{0!} = 0,3089$$

$$P(y = 1) = \frac{1,175^1 e^{-1,175}}{1!} = 0,3629$$

$$P(y = 2) = \frac{1,175^2 e^{-1,175}}{2!} = 0,2132$$

$$P(y = 3) = \frac{1,175^3 e^{-1,175}}{3!} = 0,0835$$

$$P(y = +) = 1 - (0,3089 + 0,3629 + 0,2132 + 0,0835) = 0,0315$$

Para chegarmos às probabilidades de vitória, empate e derrota dos times, precisamos calcular a probabilidade de cada resultado ocorrer:

Para a vitória do mandante:

$$\begin{aligned} P(x > y) = & P(x = 1, y = 0) + P(x = 2, y = 0) + \\ & P(x = 3, y = 0) + P(x = +, y = 0) + P(x = 2, y = 1) + \\ & P(x = 3, y = 1) + P(x = +, y = 1) + P(x = 3, y = 2) + \\ & P(x = +, y = 2) + P(x = +, y = 3) \end{aligned}$$

Para o empate, vamos considerar para efeito de cálculo das probabilidades, que se ambas equipes fizerem mais de três gols na mesma partida, o resultado é empate:

$$\begin{aligned} P(x = y) = & P(x = 0, y = 0) + P(x = 1, y = 1) + \\ & P(x = 2, y = 2) + P(x = 3, y = 3) + P(x = +, y = +) \end{aligned}$$

E para a vitória do visitante:

$$\begin{aligned} P(x < y) = & P(x = 0, y = 1) + P(x = 0, y = 2) + \\ & P(x = 0, y = 3) + P(x = 0, y = +) + P(x = 1, y = 2) + \\ & P(x = 1, y = 3) + P(x = 1, y = +) + P(x = 2, y = 3) + \\ & P(x = 2, y = +) + P(x = 3, y = +) \end{aligned}$$

Onde, por exemplo: $P(x = 0, y = 0)$ é a probabilidade do placar do jogo terminar 0x0, ou seja, é a probabilidade de $P(x = 0) \times P(y = 0)$ onde o primeiro termo refere-se ao mandante e o segundo ao visitante.

Para este jogo, finalmente teremos:

Probabilidade de vitória do mandante:

$$P(M) = P(x > y) = 37,22\%$$

Probabilidade do Empate:

$$P(E) = P(x = y) = 27,74\%$$

E a probabilidade do Visitante:

$$P(V) = P(x < y) = 35,04\%$$

O resultado com maior probabilidade de ocorrer nesta partida foi o 1x1 com 13,07%. O resultado final da partida foi o de 1x0, ou seja, vitória do time mandante conforme apontado acima. Este placar foi o segundo com maior probabilidade de ocorrer com 11,13%.

Através deste modelo, é possível calcular a probabilidade de ocorrência dos resultados de uma partida de modo a prever a probabilidade de cada time com base na média de gols a favor e contra dos times em jogos passados.

4. O Banco de Dados

Segundo Heuser [2004], para a modelagem de novos sistemas, para os quais não existem descrições de dados, deve-se usar uma estratégia de modelagem partindo de pessoas. Neste caso, podem ser usadas duas estratégias, a descendente (“top-down”) ou a “inside-out”.

4.1. Estratégia Descendente

Nesta modelagem, o processo inicia com a identificação de entidades genéricas e a partir de então, são definidos os atributos, os relacionamentos e as especializações das entidades. Pode ser dividida em três etapas:

Modelagem Superficial

- Enumeração das Entidades.
- Identificação dos relacionamentos, generalizações e especializações das entidades.
- Determinação dos atributos.
- Determinação dos identificadores.
- Verificação do aspecto temporal.

Modelagem Detalhada

- Domínio dos atributos.
- Define-se cardinalidades.
- Define-se restrições de integridade.

Validação do Modelo

- Busca-se construções redundantes no modelo.

- Valida-se o modelo com o usuário.

4.2. Estratégia “Inside-out”

O processo inicia com a identificação de uma entidade importante no modelo e que supostamente está relacionada a várias outras entidades. A partir de então, procura-se atributos, entidades relacionadas, generalizações e especializações da entidade foco até ter-se o modelo completo.

4.3. Modelagem ER

Para a modelagem do banco de dados, optaremos pela estratégia “inside-out”. Primeiramente identificamos uma entidade chave no banco de dados, no caso, como ilustra a Figura 4, “Partida”.

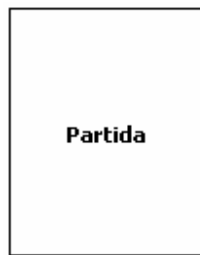


Figura 4 - Entidade Partida

Passamos então a identificar os seus atributos, tais como data e hora que a partida foi disputada, número de gols do mandante e do visitante, as médias utilizadas para o cálculo das probabilidades desta partida e as probabilidades de vitória, empate e derrota, ilustrados na Figura 5. Devido a grande quantidade de atributos, escolhemos um contador auto-incrementado para a chave-primária da tabela.

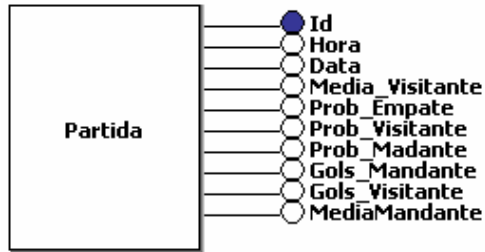


Figura 5 - Atributos de "Partida"

O passo seguinte consiste em identificar entidades relacionadas a entidade principal mostrados na Figura 6.

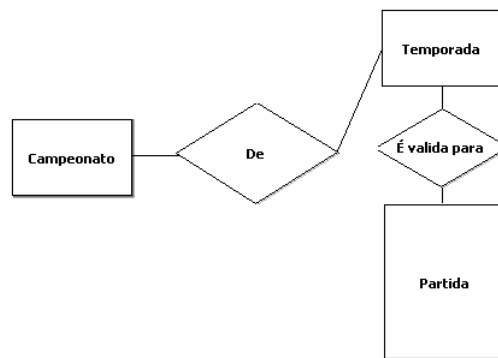


Figura 6 - Identificando novas Entidades

Para a finalização do modelo, identificamos atributos e entidades relacionadas de modo recursivo até definirmos o esquema completo ilustrado na Figura 7.

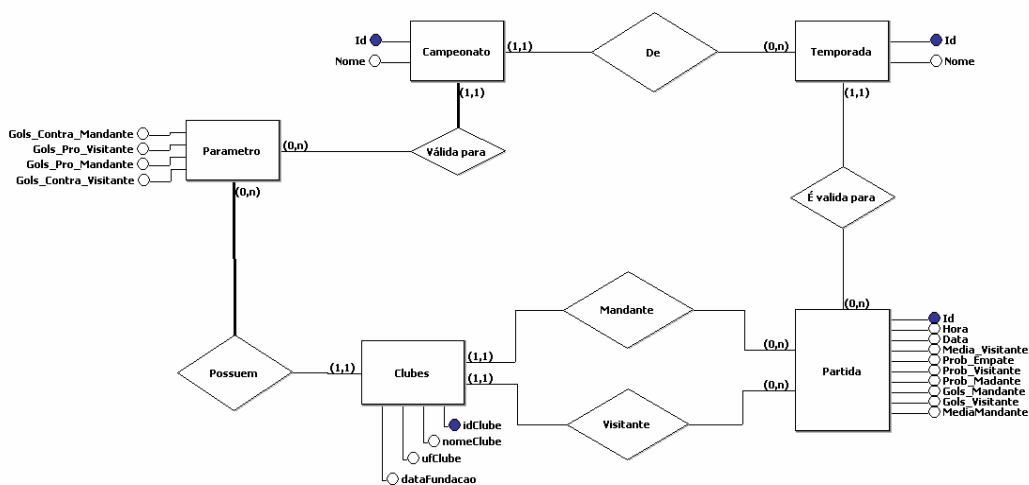


Figura 7 - Modelo ER Completo

4.4. Esquema Relacional

A abordagem relacional modela os dados no nível de SGBD, para tanto, fazemos a transformação do modelo ER para o modelo relacional.

Para o nosso banco de dados, usaremos como convenção o **negrito** como notação para chave primária e o sublinhado para chave estrangeira, tendo como resultado as seguintes tabelas:

- Clube (**idClube**, nomeClube, ufClube, dataFundacao)
- Partida (**idPartida**, clubeMandanteld, clubeVisitanteld, dataPartida, horaPartida, golsMandante, golsVisitante, mediaMandante, mediaVisitante, probMandante, probEmpate, probVisitante, temporadaId)

Restrições de Integridade:

- Nenhum clube pode ser mandante e visitante na mesma partida.
- Nenhum clube, pode disputar mais de uma partida na mesma data.

- Temporada (**idTemporada**, nomeTemporada, campeonatold)
- Campeonato (**idCampeonato**, nomeCampeonato)
- Parametro (campeonatold, clubeld, golsContraMandante, golsContraVisitante, golsFavorMandante, golsFavorVisitante)

4.5. Escolha do SGBD

Conforme [Elmars, Navathe, 2005], Um Sistema de Gerenciamento de Banco de Dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados.

Um dos objetivos do nosso trabalho é a criação de um banco de dados capaz de armazenar os dados referentes a partidas de futebol para que possam ser manipulados de forma a calcularmos as probabilidades dos times em uma disputa futebolística. A manutenção desta base de dados, permitirá que ao longo do tempo, resultados históricos dos jogos entre as equipes possam ser usados na análise dos dados e definição dos prognósticos.

Para o desenvolvimento do banco de dados deste trabalho, escolhemos o SGBD MySQL por possuir as seguintes características:

- Portabilidade, funciona em vários sistemas operacionais, tais como: Windows, Linux, FreeBSD, Solaris, Mac OS X, etc.
- Compatibilidade, o MySQL contém drivers ODBC, JDBC e .NET além de módulos de interface para linguagens de programação como Delphi, Java, C/C++, PHP, Ruby entre outras.
- Desempenho
- Estabilidade
- Facilidade de uso
- E principalmente por ser um software livre e de código aberto.

4.6. Código SQL das tabelas

O código SQL necessário para a criação das tabelas em banco de dados é o seguinte:

```
Create Table CLUBE ( idClube          INT          NOT NULL,
                    nomeClube       varchar(50)  NOT NULL,
                    ufClube         varchar(2)   NOT NULL,
                    dataFundacao    date,
                    primary key (idClube));
```

```
Create Table PARTIDA ( idPartida     INT          NOT NULL,
                      clubeMandanteld INT          NOT NULL,
                      clubeVisitanteld INT          NOT NULL,
                      dataPartida    date,
                      horaPartida    time,
                      golsMandante    INT,
                      golsVisitante  INT,
                      mediaMandante  DECIMAL(3,2),
                      mediaVisitante DECIMAL(3,2),
                      probMandante   DECIMAL(3,2),
                      probEmpate     DECIMAL(3,2),
                      probVisitante  DECIMAL(3,2),
                      temporadaId     INT          NOT NULL,
                      primary key (idPartida),
                      foreign key (temporadaId) references TEMPORADA(idTemporada),
                      foreign key (clubeMandanteld) references CLUBE (idClube),
                      foreign key (clubeVisitanteld) references CLUBE (idClube),
                      on delete no action);
```

```
Create Table TEMPORADA ( idTemporada INT          NOT NULL,
                        nomeTemporada varchar(50)  NOT NULL,
                        campeonatoId  INT          NOT NULL,
                        primary key (idTemporada),
                        foreign key (campeonatoId) references CAMPEONATO(idCampeonato)
                        on delete no action);
```

```
Create Table CAMPEONATO ( idCampeonato INT          NOT NULL,
                          nomeCampeonato varchar(50)  NOT NULL,
                          primary key (idCampeonato));
```

```

Create Table PARAMETRO ( campeonatoold      INT      NOT NULL,
                        clubeld             INT      NOT NULL,
                        golsContraMandante  DECIMAL(3,2),
                        golsContraVisitante DECIMAL(3,2),
                        golsFavorMandante   DECIMAL(3,2),
                        golsFavorVisitante   DECIMAL(3,2),
primary key (campeonatoold, clubeld),
foreign key (campeonatoold) references CAMPEONATO (idCampeonato),
foreign key (clubeld) references CLUBE (idClube)
on delete no action);

```

5. Análise e Projeto do Sistema

Diante do propósito de elaborar um sistema capaz de receber dados, processá-los e emitir informações que auxiliem o usuário na tomada de decisão quanto ao prognóstico de jogos de futebol, precisamos levantar uma série de informações para que seja possível conforme diz [Larman, 2004], fazer a coisa certa (análise) e fazer certo a coisa (projeto).

Durante a análise do sistema, o objetivo é definir os requisitos e descobrir os componentes do domínio do problema, descrevendo-os, já na fase de projeto. O objetivo é definir como os componentes colaboram para atender os requisitos.

5.1. Análise do Sistema

Conforme citado anteriormente, esta fase do desenvolvimento do sistema é responsável pelo levantamento de requisitos e pela definição dos componentes do domínio do problema.

Requisitos são capacidades e condições às quais um sistema deve atender [JBR99] e podem ser classificados em funcionais e não funcionais. Casos de uso são uma forma de descrever os requisitos do sistema, são narrativas que mostram como o sistema deve agir para alcançar um objetivo.

5.1.1. Requisitos Funcionais

- Cadastra – O sistema deve permitir ao administrador o cadastro de clubes, campeonatos, temporadas e partidas.
- Exclui – O sistema deve permitir a exclusão de partidas. Quanto a campeonatos, temporadas e clubes, não devem ser excluídos para não afetar os registros históricos do sistema.

- Edita – O sistema deve permitir a edição dos dados de clubes, temporadas, campeonatos e partidas de modo a manter a informação correta em virtude do registro de valores incorretos.
- Visualiza – O sistema deve permitir, em diferentes níveis, a visualização dos dados. Para o administrador, deve ser possível visualizar os dados de campeonato, temporada, clubes e partidas. Ainda no item partidas, deve ser possível visualizar no detalhe, onde será mostrado os valores das probabilidades para cada placar possível para a referida partida.
- Atualização automática dos parâmetros para o cálculo das probabilidades – Deve atualizar os parâmetros dos clubes que disputaram uma partida quando a mesma for editada de modo a ser informado o placar, assim passa a ser utilizado o novo parâmetro no cálculo das probabilidades para jogo futuro. Para a referida partida, os cálculos de probabilidades devem levar em consideração os parâmetros da própria;
- Calcular probabilidades – Deve, com base nos parâmetros para os clubes que disputarão uma partida, calcular a probabilidade de ocorrer vitória do time mandante, empate ou vitória do time visitante; Deve ainda, calcular probabilidades da ocorrência dos possíveis resultados com base nos parâmetros. Ex.: calcular a probabilidade de a partida terminar com o placar de 1x0, 3x1, etc., dentro as possibilidades tidas como referência para o sistema;
- Deve criar de forma automática, os parâmetros para novos clubes, caso não exista;

5.1.2. Requisitos não funcionais

- Deve ser desenvolvido em ambiente web de modo a facilitar o acesso dos usuários ao sistema;
- Deve ser compatível com os browsers mais utilizados do mercado;
- Deve ter todas as informações em Português do Brasil;
- Deve prover segurança para os dados – O acesso a cadastros edições e exclusões só poderá ser feito por pessoa autorizada. Usuários terão acesso apenas a consultas diversas.

5.1.3. Diagrama de Casos de Uso

Com base nos requisitos do sistema, é possível construirmos o diagrama de casos de uso, onde são ilustrados os requisitos funcionais e sua associação com os usuários do sistema (atores). Tal diagrama é ilustrado na Figura 8.

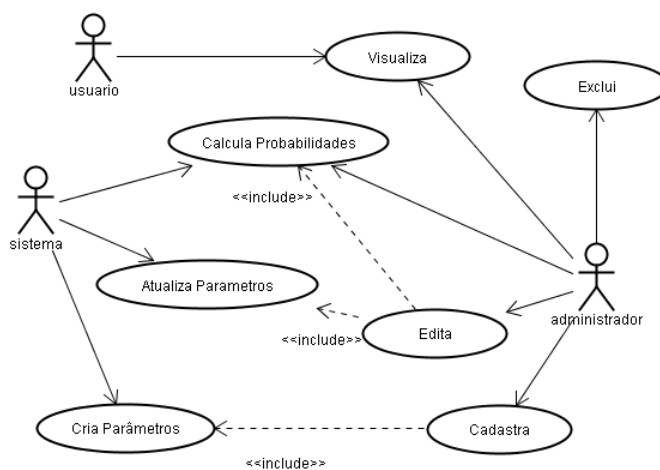


Figura 8 - Diagrama de Casos de Uso

5.2. Projeto do Sistema

Com base nos requisitos do sistema, é iniciado o processo de projeto e desenvolvimento, isto é, projetar de maneira competente uma solução para satisfazer os

requisitos do sistema [Larman, 1994]. Neste capítulo, descreveremos os passos para o desenvolvimento do projeto, como a escolha da linguagem de programação, ferramentas de desenvolvimento e definição dos elementos do software.

5.2.1. Linguagem de Programação

Para o desenvolvimento do sistema, optamos pela linguagem Java na sua versão 5 [JAVA01] pois apresenta um conjunto de frameworks que facilitam a implementação de sistemas web, como por exemplo, Java Server Faces [JSF01].

5.2.2. Ferramentas de Desenvolvimento

A escolha das ferramentas de desenvolvimento do sistema pode ser crucial para o projeto, uma escolha mal feita acarretará perda de tempo precioso além de impedir o correto andamento do processo de desenvolvimento.

Como opções, foram analisadas duas ferramentas bastante difundidas entre a comunidade de programadores, sendo elas: Eclipse [ECLI01] e Netbeans [NETB01].

Ambas possuem funcionalidades muito semelhantes e a escolha acabou sendo pelo Netbeans por ser desenvolvida pela SUN [SUN01] que é a mesma empresa desenvolvedora da linguagem escolhida além de uma maior familiaridade no uso desta ferramenta.

5.2.3. Definição dos elementos do sistema

Conforme elencado dentre os requisitos não funcionais, o sistema deverá ser distribuído para acesso público na internet.

Utilizaremos MVC [Deitel, 2003] no desenvolvimento do sistema. O MVC (Model-View-Controller) é um padrão de arquitetura de software. A aplicação é dividida em camadas, cada qual com sua responsabilidade.

O Model é a representação da informação da aplicação, no caso, seus objetos, as entidades relacionadas no capítulo sobre o banco de dados.

O View é a visão da aplicação, sua interface com o usuário e que será desenvolvida utilizando a tecnologia JSF citada anteriormente.

O Controller é o controlador da aplicação, é onde está a lógica do sistema. Ele processa e responde a eventos, invoca alterações no model.

5.2.4. Diagrama de Classes do Sistema

Baseado nas entidades criadas no banco de dados é possível fazermos o Diagrama de Classes para a representação do Model do sistema. A Figura 9, mostra as entidades, seus atributos e seus métodos.

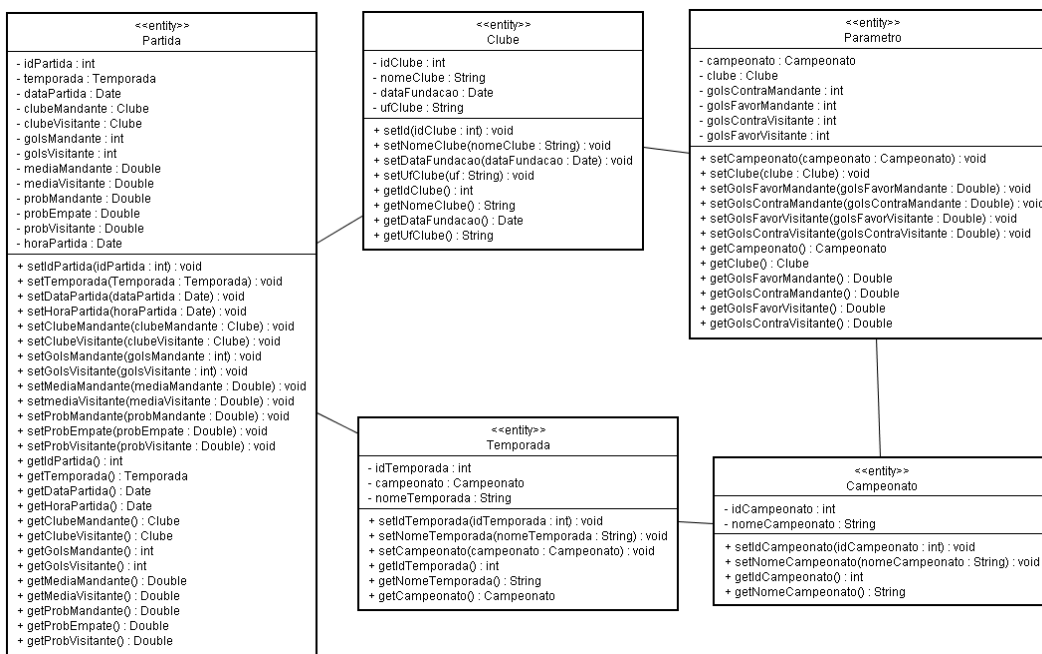


Figura 9 - Diagrama de Classes

6. Chutegol

Chutegol é um acrônimo para “Chance de Um Time Em GOLs”, além de fazer uma alusão ao fato de tentar descobrir o placar de uma partida de futebol, ou seja, chutar quantos gols cada time irá fazer numa partida. Por estes motivos, foi o nome escolhido para o sistema.

Este capítulo visa descrever como foi o desenvolvimento do sistema através do uso da ferramenta de desenvolvimento escolhida, além de apresentar as telas do software.

6.1. Projeto Web

O primeiro passo no desenvolvimento do sistema foi a criação de um novo projeto web no Netbeans, denominado “chutegol”.

6.2. Classes de Entidade do Banco de Dados

Com o projeto criado, o próximo passo foi utilizar a ferramenta de desenvolvimento para construir as classes das entidades mostradas no diagrama de classes através do banco de dados criado no Capítulo 4.

O Netbeans possui uma opção chamada Classes de Entidade do Banco de Dados que executa esta tarefa, bastando para isso fazermos sua conexão com a base de dados.

6.3. Páginas JSF de Classes de Entidades

A última função automatizada do Netbeans utilizada no desenvolvimento foi a criação das páginas em Java Server Faces. Utilizamos a opção do Netbeans “Páginas JSF de Classes de Entidades” para criar as páginas Detail.jsp, Edit.jsp, List.jsp e New.jsp. Neste momento, o Netbeans também criou de forma automática, os Controllers para cada entidade com todos os métodos necessários para que o sistema ficasse funcional a ponto

de podermos fazer as tarefas de CRUD (Create, Retrieve, Update e Destroy) que são as quatro operações básicas em banco de dados relacionais.

6.4. Lógica do Negócio

O próximo passo foi a implementação da lógica do negócio, ou seja, a implementação dos requisitos do sistema para que o mesmo possa, como citado anteriormente, fazer a coisa certa e fazer certo a coisa.

Os requisitos que precisam ser implementados para a lógica do sistema são:

- Criar parâmetros para o clube no campeonato (caso não exista);
- Atualizar os parâmetros após a inserção do resultado da partida;
- Calcular a probabilidade de vitória do mandante, empate e vitória do visitante;
- Calcular a probabilidade dos possíveis placares para a partida;

Vamos detalhar cada um desses casos de uso de forma a facilitar a sua implementação.

6.4.1. Criar parâmetros

Quando o administrador solicita que sejam calculadas as probabilidades da partida, o sistema deve verificar se existem no banco de dados, os parâmetros para os clubes envolvidos na disputa. Caso não exista, deve criar o parâmetro com valor inicial zero.

Caso de uso

- Sistema recebe solicitação para retornar parâmetro;
- Sistema recebe dados do campeonato e do clube;

- Sistema verifica no banco de dados se existe parâmetro para o clube no campeonato;
- Sistema retorna o parâmetro encontrado no banco ou cria um novo com valores zerados, armazenando o mesmo no banco;

6.4.2. Atualizar Parâmetros

Quando o administrador informa o resultado de uma partida, o sistema deve atualizar o valor dos parâmetros das equipes de acordo com o resultado da partida.

De modo a não viciarmos o parâmetro devido a valores discrepantes, e conforme mostrado no Capítulo 3, mais de 90% dos jogos apresentam resultados com até 4 gols, no cálculo da atualização dos parâmetros, caso o time tenha feito mais de quatro gols na partida, será usado o valor quatro como argumento do cálculo.

O método de cálculo da atualização do parâmetro foi arbitrado da seguinte forma:

$$Parâmetro = \frac{(Parâmetro \times 4) + gols}{5} \quad (13)$$

- Administrador informa resultado da partida;
- Sistema verifica se foram informados os valores de gols para o mandante e o visitante;
- Sistema atribui valor 4 caso o número de gols seja superior a 4;
- Sistema calcula o novo parâmetro conforme fórmula acima;
- Sistema armazena o novo valor no banco de dados para uso futuro;

6.4.3. Calcular Probabilidades da partida

Quando o administrador do sistema solicita o cálculo das probabilidades de vitória do mandante, empate e vitória do visitante, o sistema deve buscar os parâmetros das equipes e calcular o valor do parâmetro para a partida. Fica arbitrado que os parâmetros para a partida serão calculados conforme fórmulas (11) e (12) apresentadas no Capítulo 3:

$$P(M) = \frac{GolsFavorMand + GolsContraVis}{2}$$

$$P(V) = \frac{GolsFavorVis + GolsContraMand}{2}$$

Esses parâmetros calculados serão usados para o cálculo das probabilidades conforme fórmulas (8), (9) e (10), demonstradas no capítulo 3. Esses valores serão armazenados no banco de dados para uso futuro.

Conforme citado ainda no capítulo 3, foi arbitrado que o cálculo das probabilidades levará em consideração a quantidade de gols até quatro. Desta forma, é possível que o cálculo não atinja o percentual de 100%, sendo necessário o ajuste das probabilidades da seguinte forma:

$$P(M) = \frac{P(M)}{P(M) + P(E) + P(V)} \quad (14)$$

$$P(E) = \frac{P(E)}{P(M) + P(E) + P(V)} \quad (15)$$

$$P(V) = \frac{P(V)}{P(M) + P(E) + P(V)} \quad (16)$$

- Administrador solicita cálculo das Probabilidades;
- Sistema busca os parâmetros dos clubes envolvidos no banco de dados;

- Sistema calcula os parâmetros da partida conforme fórmulas (11) e (12) e armazena no banco de dados;
- Sistema calcula as probabilidades conforme fórmulas (8), (9) e (10);
- Sistema ajusta as probabilidades conforme fórmulas (14), (15) e (16);
- Sistema armazena probabilidades no banco de dados;

6.4.4. Calcular probabilidades dos possíveis placares

Quando o usuário ou o administrador solicita visualizar a página de detalhes de uma partida, o sistema exibirá as probabilidades de ocorrência dos possíveis placares dentre as vinte e cinco possibilidades de zero a quatro gols para o time mandante e visitante.

A exibição se dará em três tabelas, contendo uma as possibilidades de vitória do mandante, outra para o empate e a última com as possibilidades de vitória do time visitante, todas elas ordenadas de forma decrescente pela probabilidade.

Para efetuar o cálculo, o sistema utilizará os parâmetros da partida calculados no item anterior que estão armazenados no banco de dados e utilizá-los nas fórmulas (8), (9) e (10).

- Usuário ou Administrador solicita visualizar a página de detalhes de uma partida;
- Sistema busca os parâmetros da partida no banco de dados;
- Sistema calcula as probabilidades utilizando as fórmulas (8), (9) e (10);
- Sistema exibe a página de detalhes com as tabelas de vitória do time mandante, empate e vitória do time visitante, todas ordenadas de forma decrescente pela probabilidade;

6.5. Telas do Sistema - Usuário

6.5.1. Tela inicial

A Figura 10 mostra a tela inicial do sistema, ela é exibida aos usuários quando os mesmos acessam o sistema através do endereço web principal.

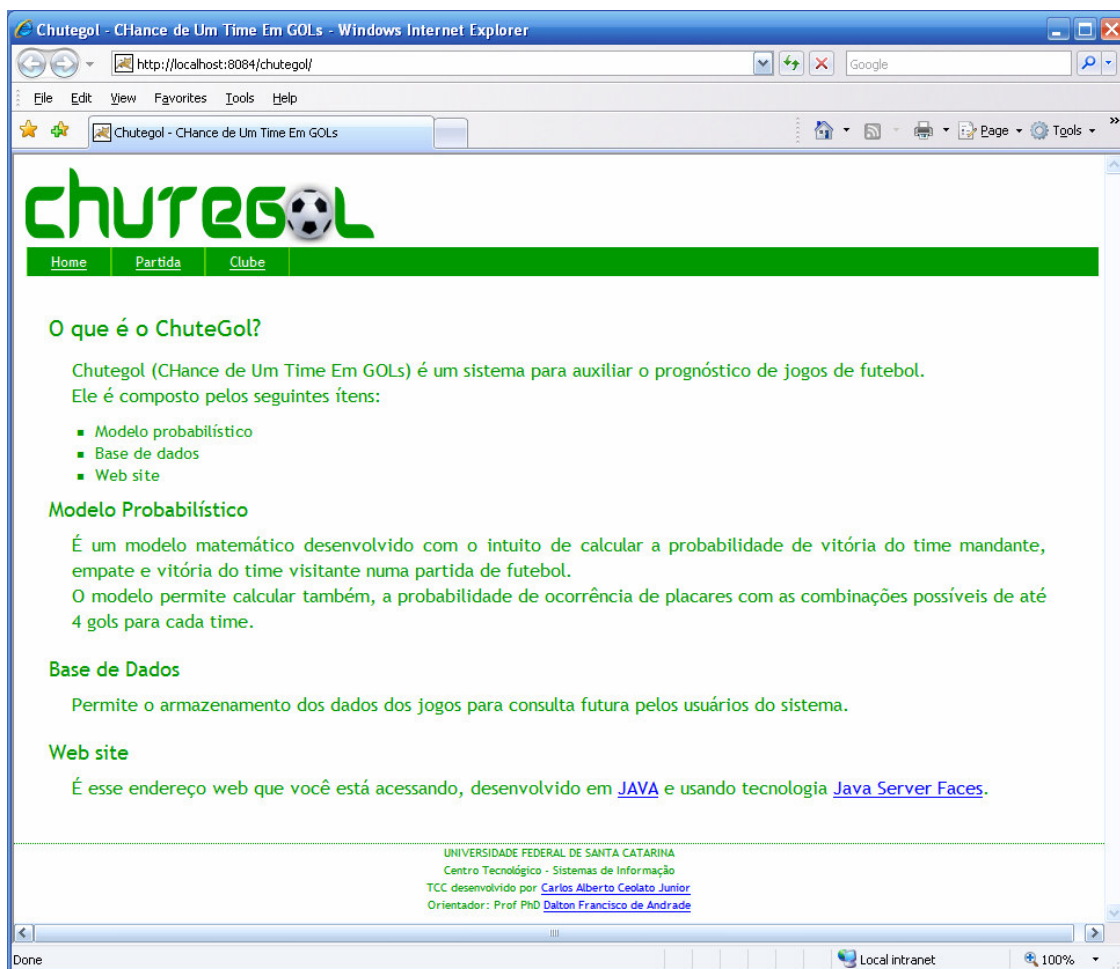


Figura 10 - Tela Inicial do Sistema – Usuário

6.5.2. Clube – List.jsp

A Figura 11 mostra uma listagem de clubes que estão cadastrados no sistema contendo o nome do clube, a Unidade da Federação a qual ele faz parte e sua data de fundação.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Centro Tecnológico - Sistemas de Informação
TCC desenvolvido por [Carlos Alberto Ceolato Junior](#)
Orientador: Prof PhD [Dalton Francisco de Andrade](#)

Clube	UF	Fundacao
Grêmio-RS	RS	15/09/1903
Inter-RS	RS	04/04/1909
Juventude-RS	RS	29/07/1913
América-RN	RN	14/07/1915
Atlético-MG	MG	25/03/1908
Atlético-PR	PR	26/03/1924
Botafogo-RJ	RJ	12/08/1904
Corinthians-SP	SP	01/09/1910
Cruzeiro-MG	MG	02/01/1921
Figueirense-SC	SC	12/06/1921

Figura 11 - Lista de Clubes - Usuário

6.5.3. Partida – List.jsp

A Figura 12 mostra a relação de partidas inseridas no sistema contendo a temporada, a data e a hora de realização da partida, os clubes mandante e visitante, o resultado da partida e as probabilidades de vitória do time mandante, de empate e de vitória do time visitante.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Centro Tecnológico - Sistemas de Informação
TCC desenvolvido por [Carlos Alberto Cealato Junior](#)
Orientador: Prof PhD [Dalton Francisco de Andrade](#)

Figura 12 - Lista de Partidas – Usuário

6.5.4. Partida – Detail.jsp

A Figura 13 mostra o detalhe da partida, nesta página, além da informação da partida, é possível visualizar as probabilidades de ocorrência dos placares com todas as combinações de até quatro gols para cada time em três tabelas, sendo uma com as situações que determinam a vitória do time mandante, outra com as situações de empate e a última com as que determinam a vitória do time visitante.

CHUTEGOL

Home Partida Clube

Detalhe da partida

Id	Temporada	Data	Hora	Mandante		X		Visitante	Prob M	Prob E	Prob V
259	Brasileirão 2007	16/09/2007	18:10	Grêmio-RS	1	x	0	Inter-RS	56,0%	24,0%	19,0%

Probabilidade Mandante			Probabilidade Empate			Probabilidade Visitante		
Mand	Vis	Prob	Mand	Vis	Prob	Mand	Vis	Prob
1	0	12,9%	1	1	11,2%	0	1	6,6%
2	0	10,9%	0	0	7,6%	1	2	4,9%
2	1	9,5%	2	2	4,1%	0	2	2,9%
3	0	6,2%	3	3	0,7%	1	3	1,4%
3	1	5,4%	4	4	0,1%	2	3	1,2%
4	0	2,6%				0	3	0,8%
3	2	2,3%				1	4	0,3%
4	1	2,3%				2	4	0,3%
4	2	1,0%				0	4	0,2%
4	3	0,3%				3	4	0,1%

UNIVERSIDADE FEDERAL DE SANTA CATARINA
 Centro Tecnológico - Sistemas de Informação
 TCC desenvolvido por [Carlos Alberto Ceolato Junior](#)
 Orientador: Prof PhD [Dalton Francisco de Andrade](#)

Figura 13 - Detalhe de Partida - Usuário

6.6. Telas do sistema – Administrador

6.6.1. Tela Inicial

A Figura 14 mostra a tela inicial do sistema, ela é exibida aos administradores quando os mesmos acessam o sistema através do endereço web de administração do sistema.

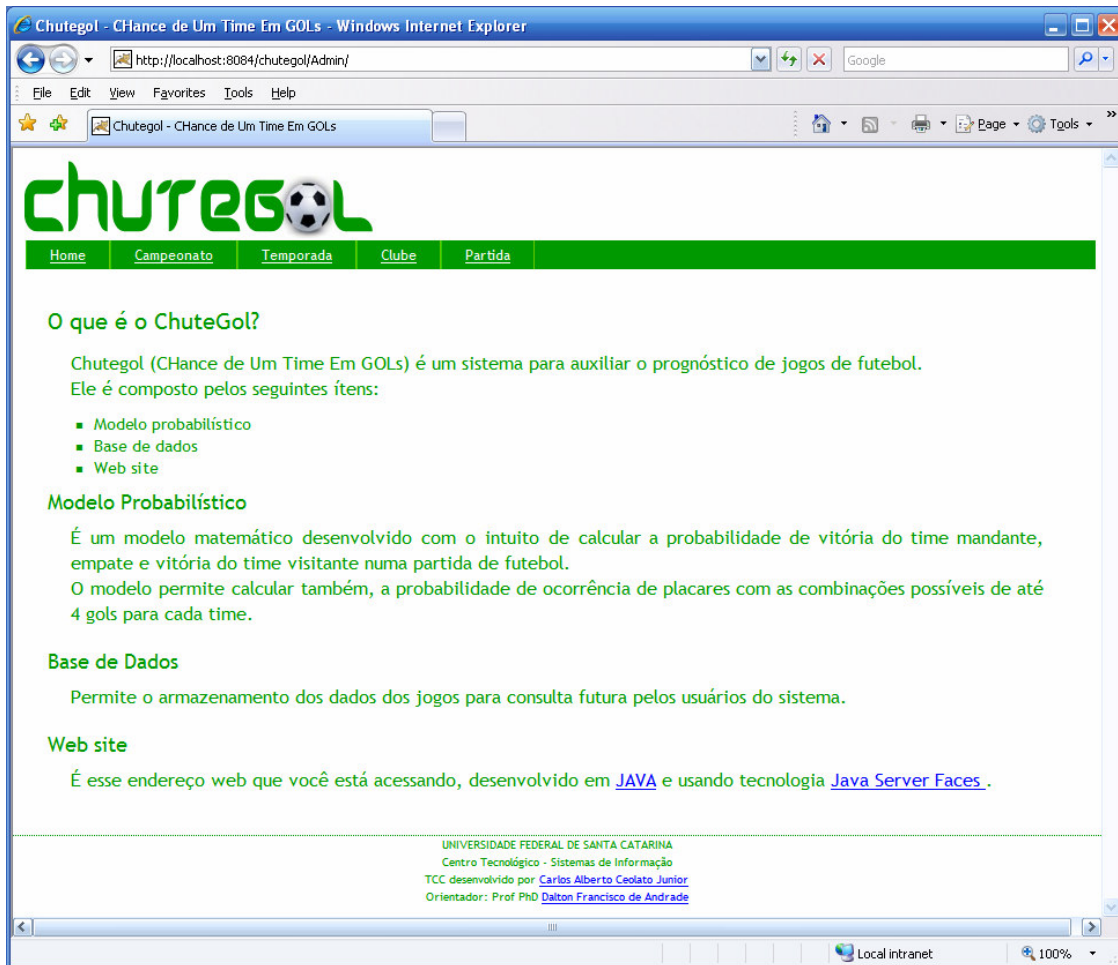


Figura 14 - Tela inicial do Sistema – Administrador

6.6.2. Campeonato – List.jsp

A Figura 15 mostra a listagem dos campeonatos cadastrados no sistema e apresenta um link que direciona para a página de cadastro de novos campeonatos.

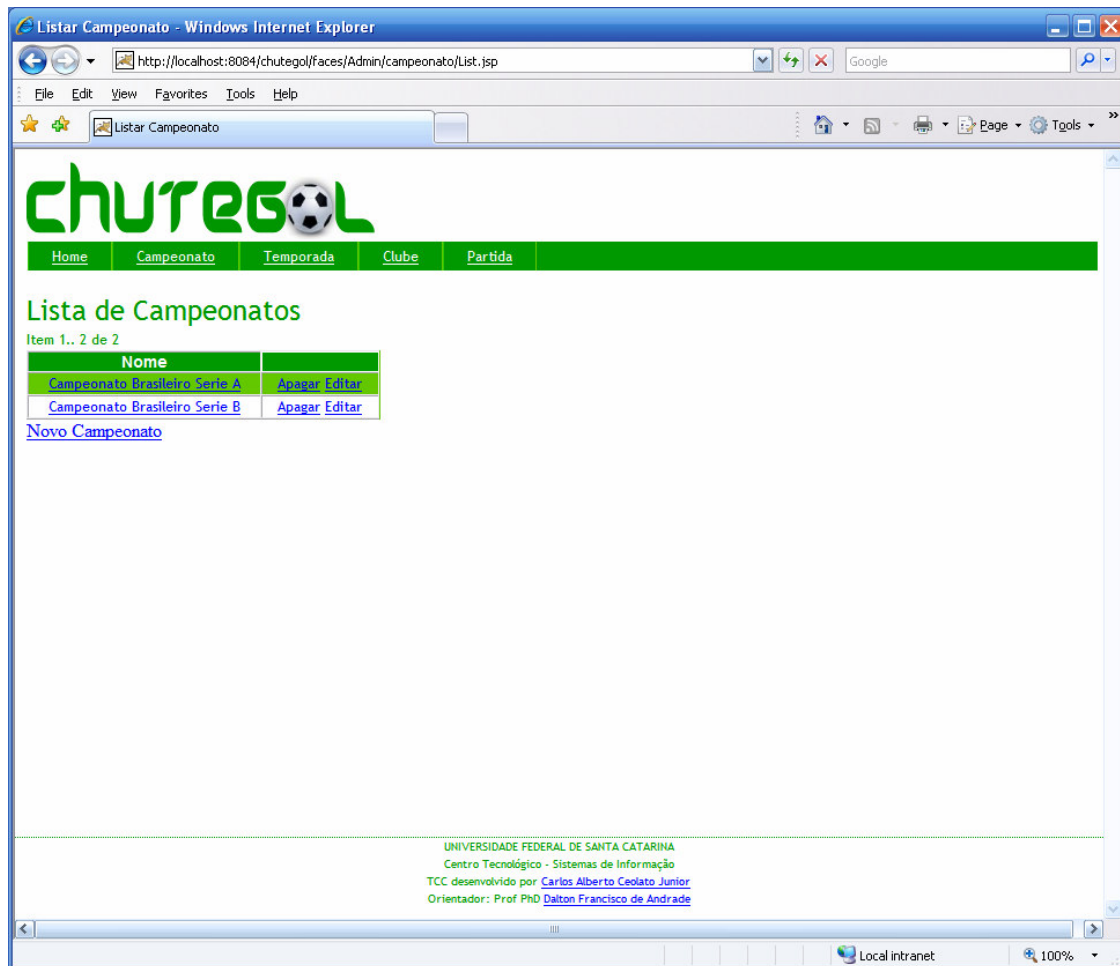


Figura 15 - Lista de Campeonato - Administrador

6.6.3. Campeonato – New.jsp

A Figura 16 é utilizada para o cadastro de novos campeonatos, é necessário apenas fornecer um nome para o campeonato.

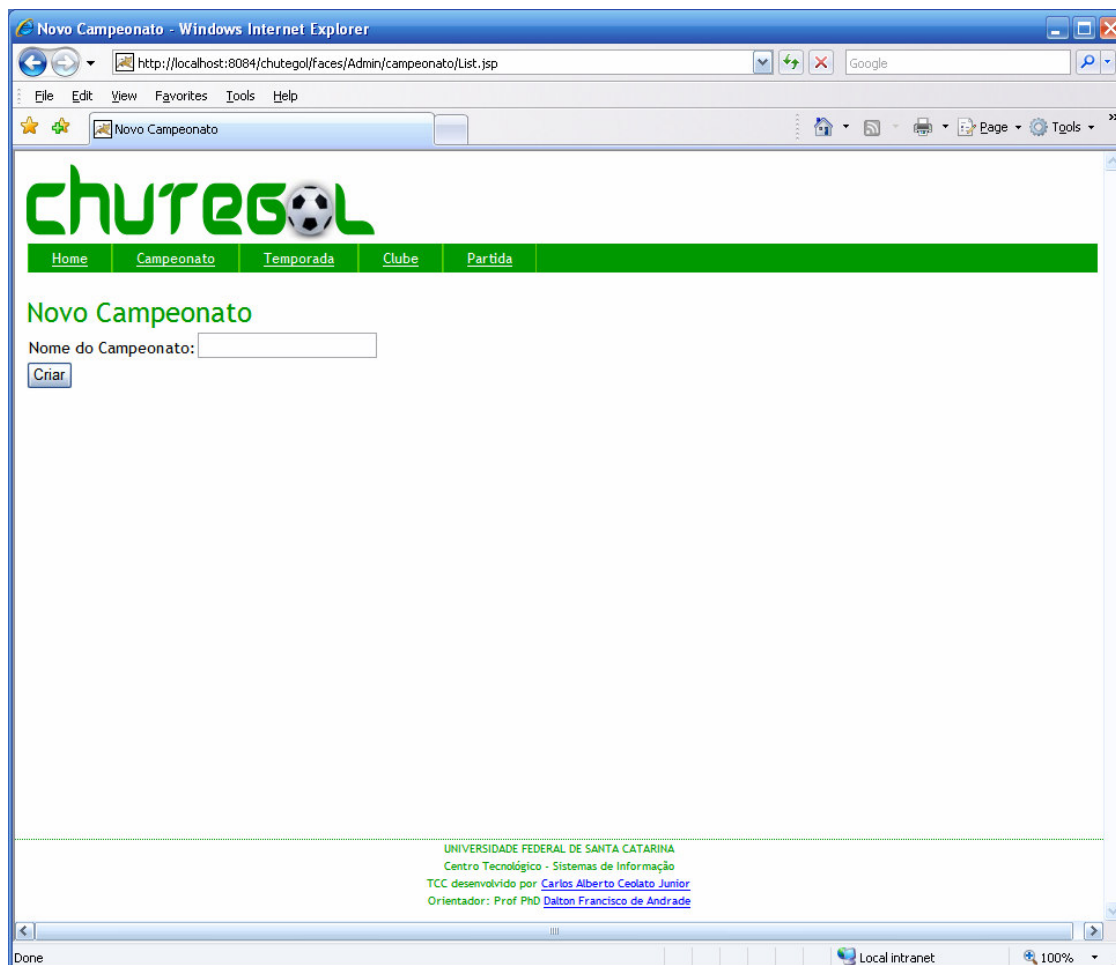


Figura 16 - Novo Campeonato – Administrador

6.6.4. Campeonato – Edit.jsp

A Figura 17 ilustra a tela de edição de campeonatos, nela é possível alterar o nome do campeonato.

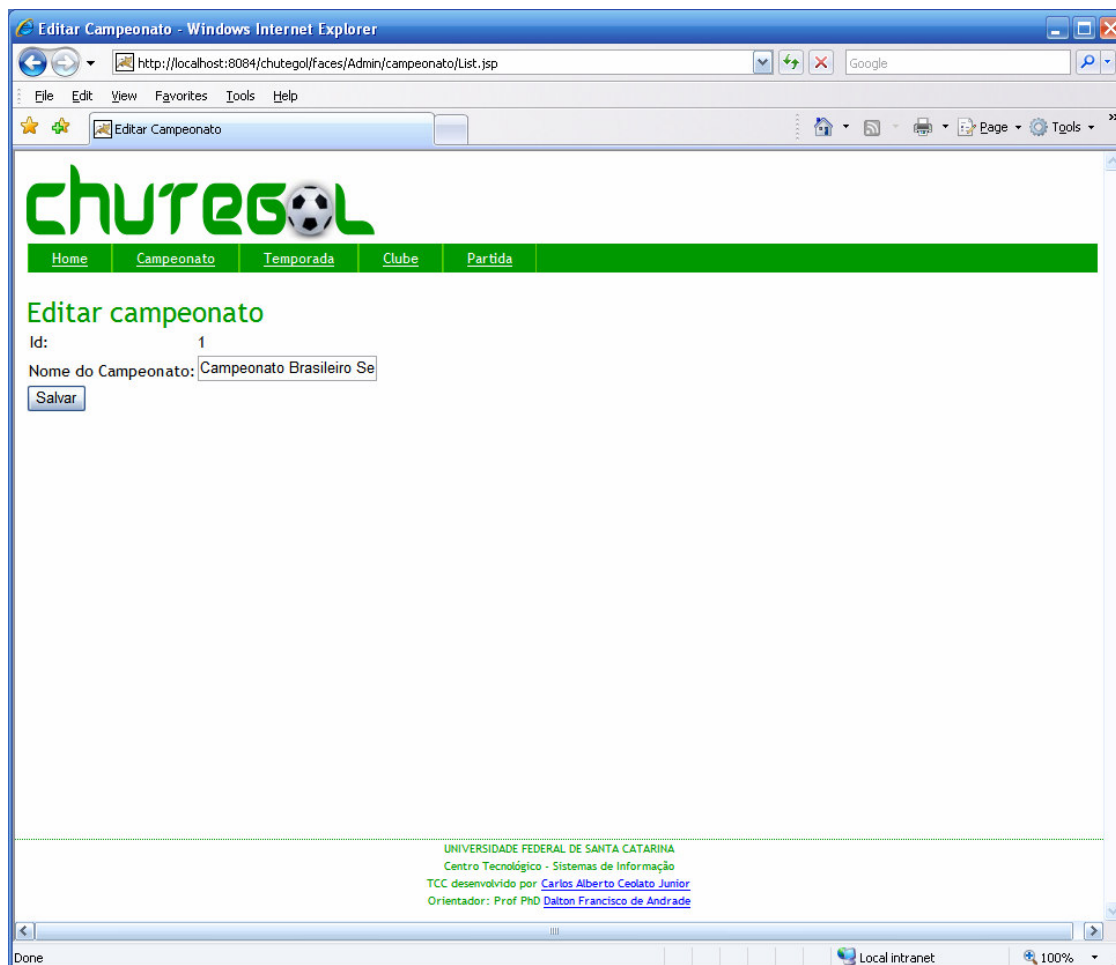


Figura 17 - Editar Campeonato - Administrador

6.6.5. Temporada – List.jsp

A Figura 18 apresenta uma listagem com as temporadas cadastradas no sistema, da mesma forma que o campeonato, é exibido um link para a criação de uma nova temporada.

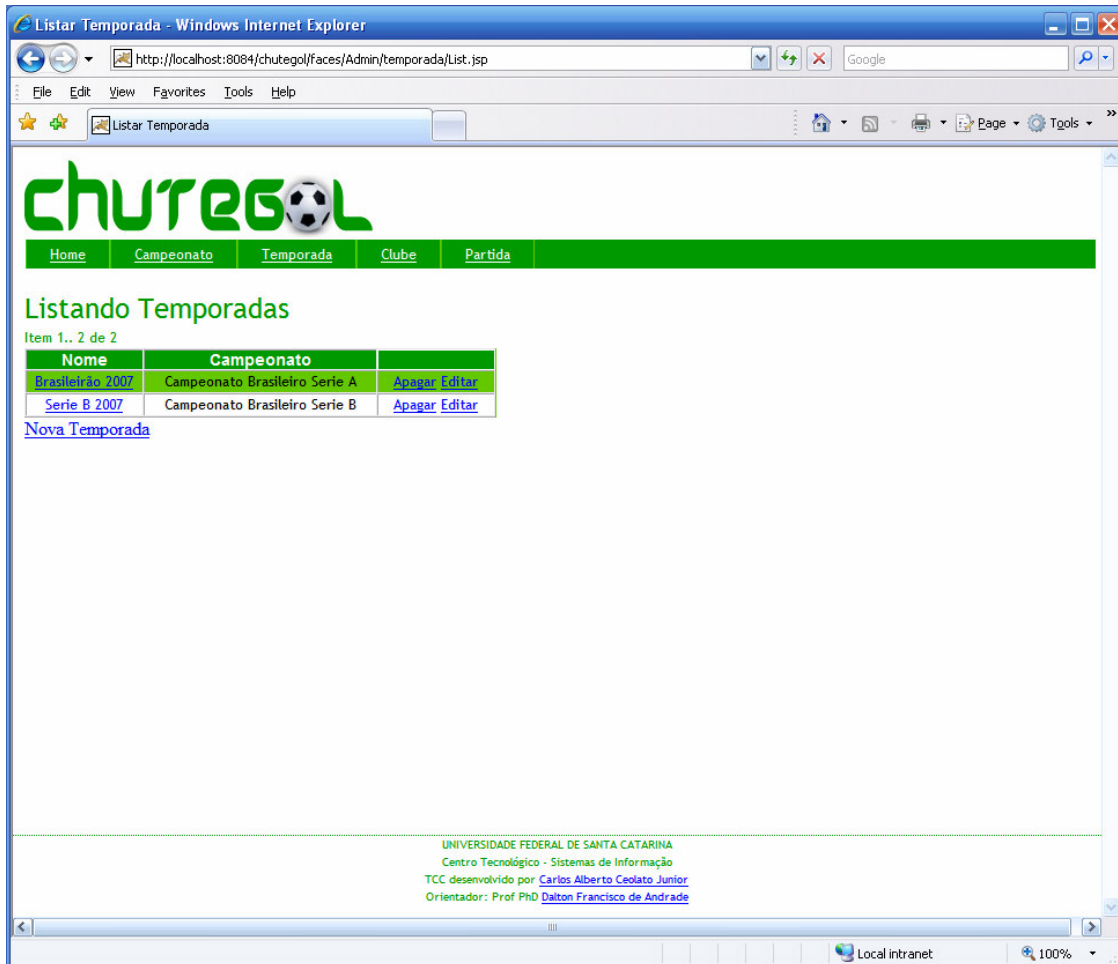


Figura 18 - Lista de Temporadas - Administrador

6.6.6. Temporada – New.jsp

A Figura 19 é utilizada para o cadastro de novas temporadas, basta fornecer um nome e escolher o campeonato ao qual ela pertence.

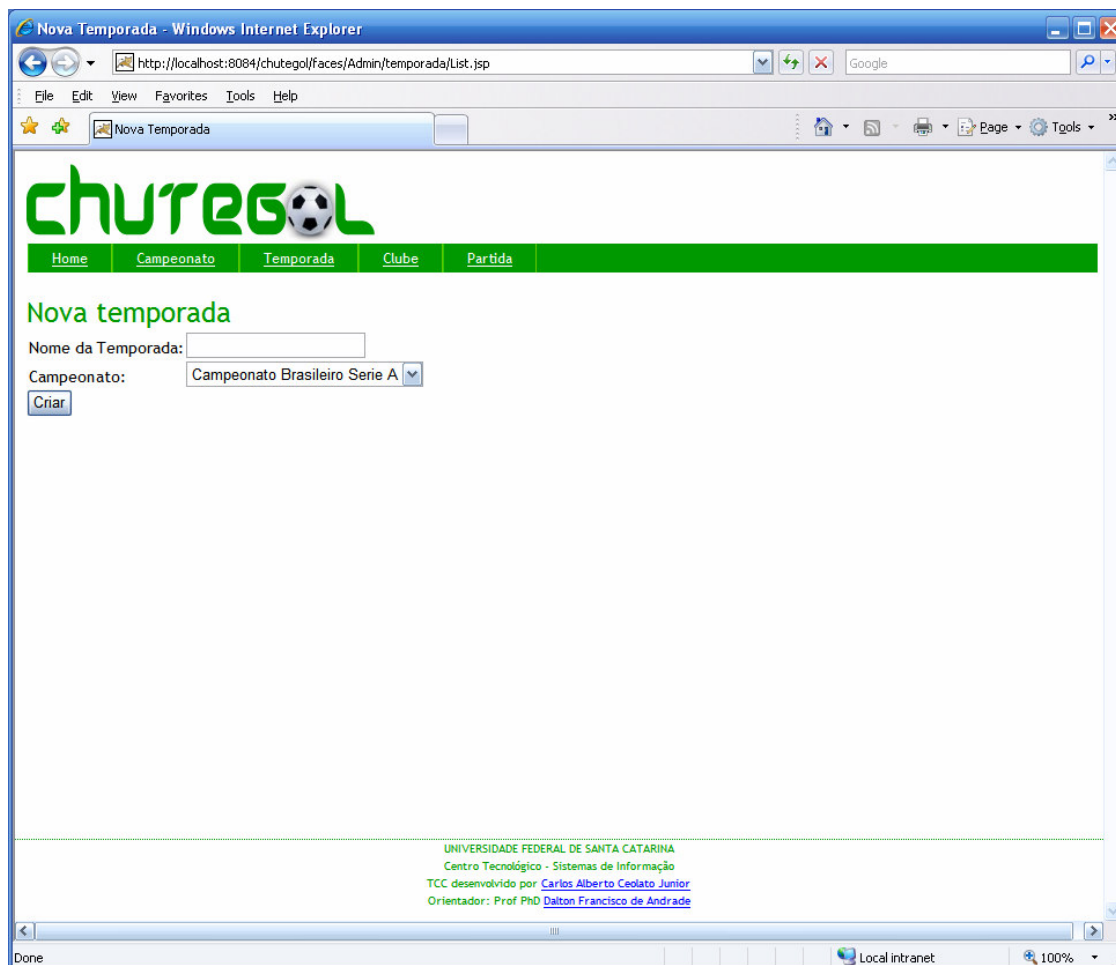


Figura 19 - Administrador - Nova Temporada

6.6.7. Temporada – Edit.jsp

A Figura 20 exibe a tela de edição de temporadas, nela é possível alterar o nome da temporada ou o campeonato a qual ela pertence.

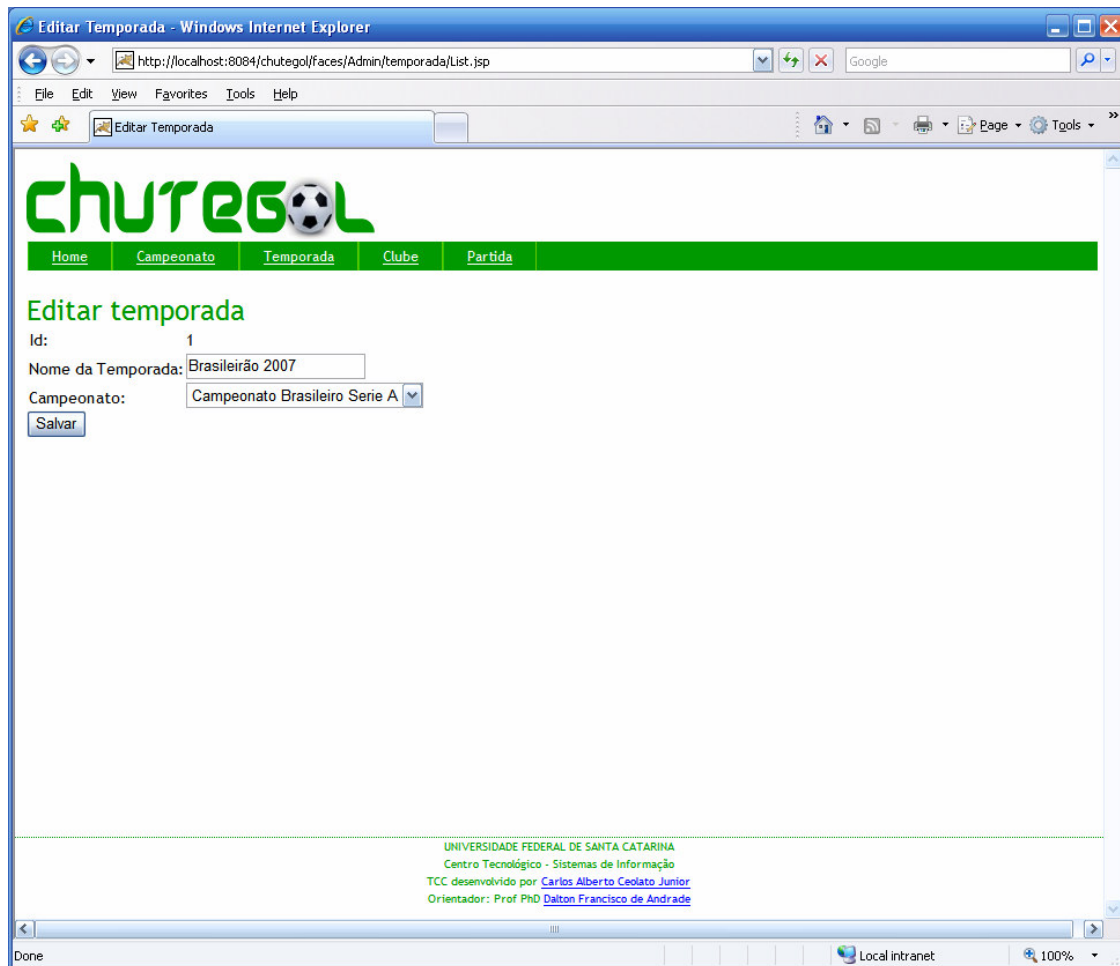


Figura 20 - Editar Temporada - Administrador

6.6.8. Clube – List.jsp

A Figura 21 apresenta a listagem de clubes na visão do administrador, nela, além dos dados exibidos aos usuários, é possível acessar as áreas de edição ou inclusão de registros ou excluir um registro.

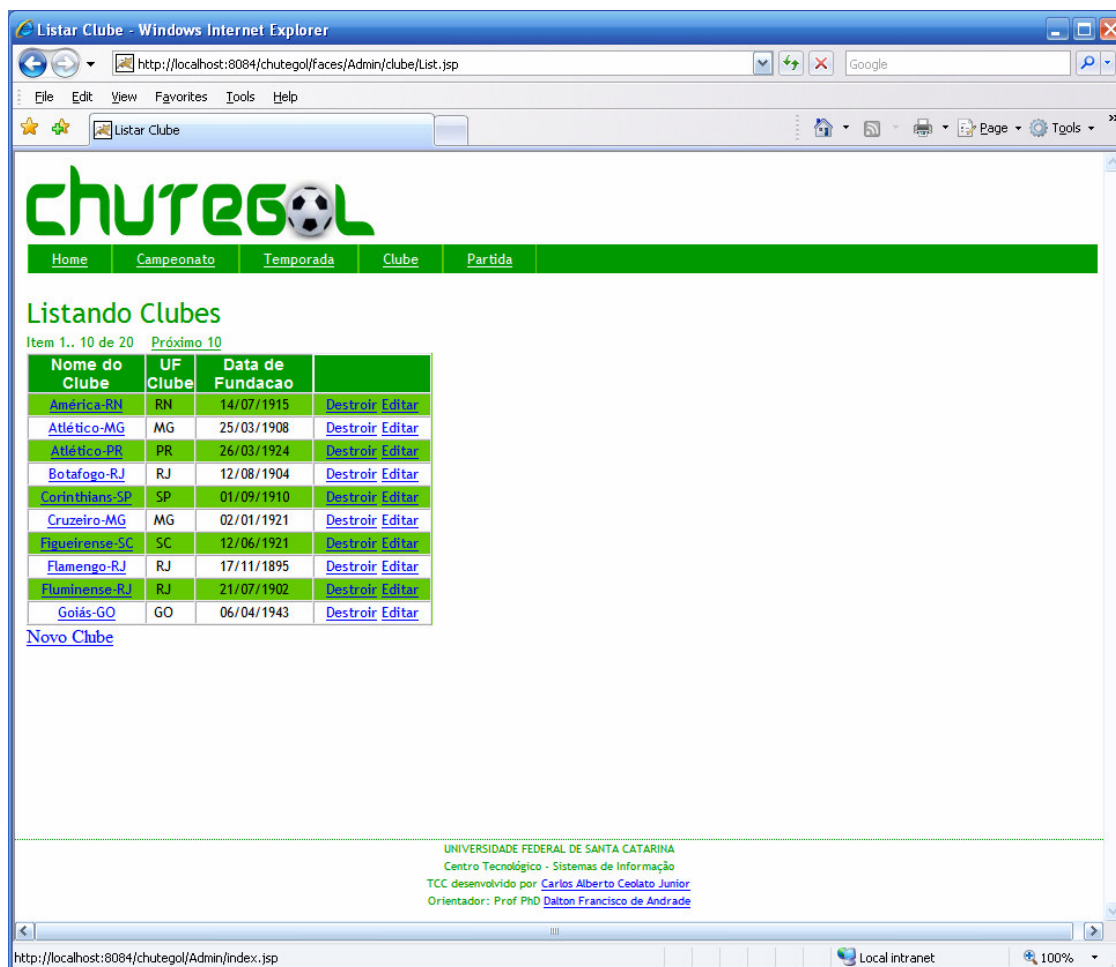
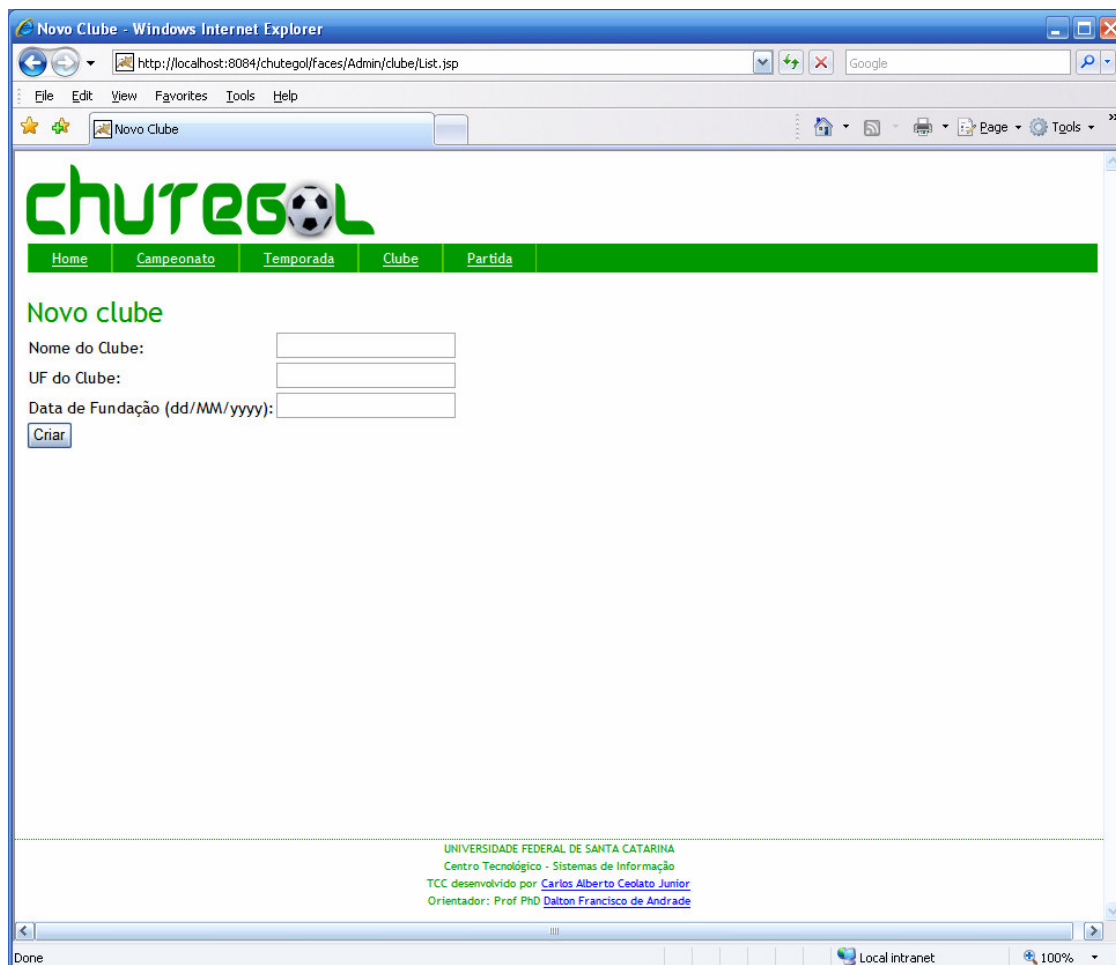


Figura 21 - Lista de Clube - Administrador

6.6.9. Clube – New.jsp

A Figura 22 é utilizada para o cadastro de novos clubes, sendo necessário fornecer um nome, a UF e a data de fundação.



The image shows a screenshot of a web browser window titled "Novo Clube - Windows Internet Explorer". The address bar displays "http://localhost:8084/chuteqol/faces/Admin/clube/List.jsp". The page features a green navigation bar with the "CHUTEQOL" logo and menu items: Home, Campeonato, Temporada, Clube, and Partida. The main content area is titled "Novo clube" and contains a registration form with the following fields:

- Nome do Clube:
- UF do Clube:
- Data de Fundação (dd/MM/yyyy):

Below the form is a "Criar" button. At the bottom of the page, there is a footer with the following text:

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Centro Tecnológico - Sistemas de Informação
TCC desenvolvido por [Carlos Alberto Cealato Junior](#)
Orientador: Prof PhD [Dalton Francisco de Andrade](#)

The browser's status bar at the bottom shows "Done", "Local intranet", and "100%" zoom.

Figura 22 - Novo Clube – Administrador

6.6.10. Clube – Edit.jsp

A Figura 23 exibe a tela de edição de clubes, sendo possível alterar o nome, a UF e a data de fundação.

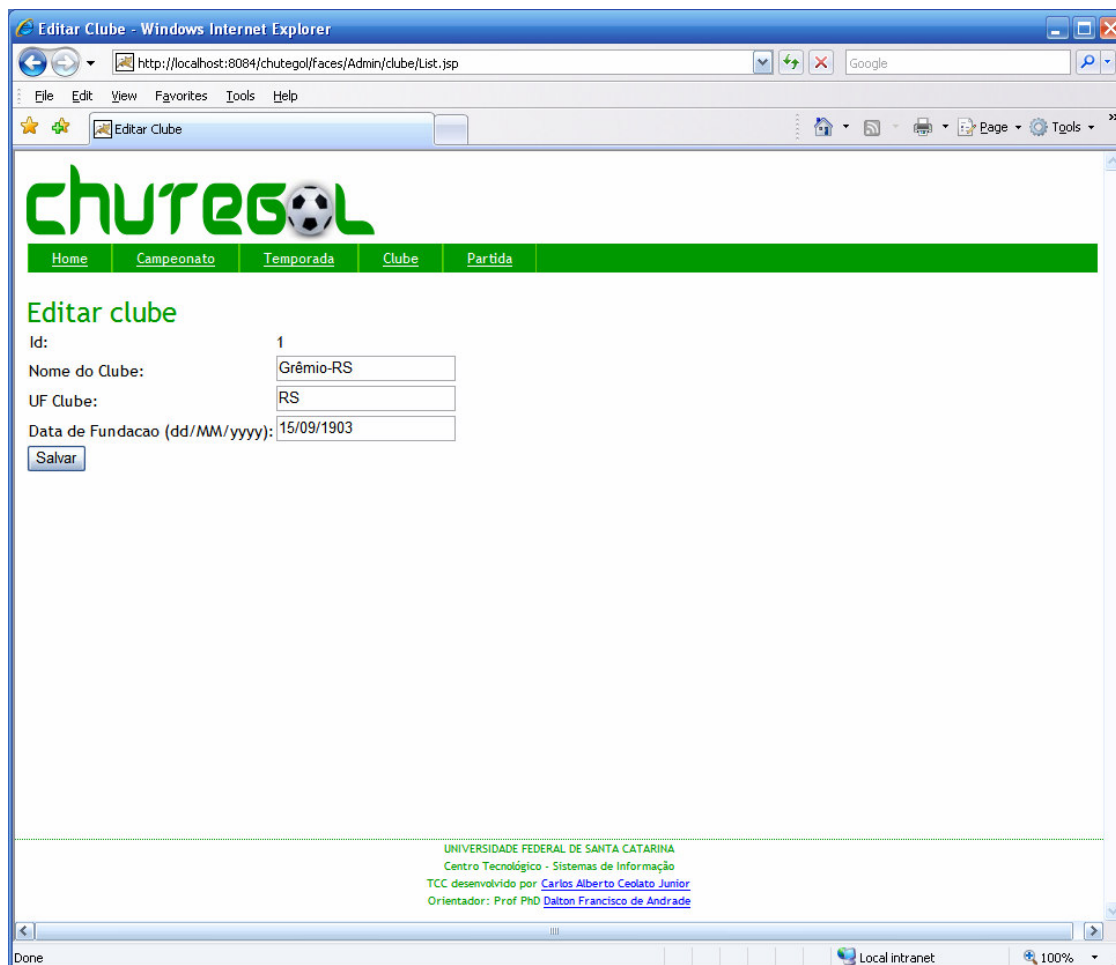


Figura 23 - Editar Clube - Administrador

6.6.11. Partida – List.jsp

A Figura 24 exibe a listagem de partidas na visão do administrador, sendo possível visualizar além das informações fornecidas aos usuários, os parâmetros válidos para cada partida e os links para exclusão, edição, inclusão e para solicitar o cálculo das probabilidades. Este último só é possível de ser executado caso não tenha sido informado o resultado da partida.

Id	Temporada	Data	Hora	Mandante	x	Visitante	P Man	P Vis	Prob M	Prob E	Prob V	Links
251	Brasileirão 2007	15/09/2007	18:10	Goiás-GO	0 x 3	Náutico-PE	1,88	1,55	44%	23%	32%	Apagar Editar Calc
252	Brasileirão 2007	15/09/2007	18:10	Fluminense-RJ	2 x 0	América-RN	1,77	0,58	65%	23%	13%	Apagar Editar Calc
253	Brasileirão 2007	15/09/2007	18:10	São Paulo-SP	2 x 1	Santos-SP	1,99	0,70	66%	22%	12%	Apagar Editar Calc
254	Brasileirão 2007	16/09/2007	16:00	Atlético-MG	3 x 4	Cruzeiro-MG	1,55	1,28	43%	26%	31%	Apagar Editar Calc
255	Brasileirão 2007	16/09/2007	16:00	Corinthians-SP	0 x 1	Botafogo-RJ	1,59	0,96	51%	26%	23%	Apagar Editar Calc
256	Brasileirão 2007	16/09/2007	16:00	Atlético-PR	2 x 1	Palmeiras-SP	1,35	0,88	47%	28%	24%	Apagar Editar Calc
257	Brasileirão 2007	16/09/2007	16:00	Sport-PE	3 x 1	Paraná-PR	1,72	0,80	58%	25%	18%	Apagar Editar Calc
258	Brasileirão 2007	16/09/2007	18:10	Figueirense-SC	4 x 1	Juventude-RS	1,76	0,97	55%	24%	21%	Apagar Editar Calc
259	Brasileirão 2007	16/09/2007	18:10	Grêmio-RS	1 x 0	Inter-RS	1,70	0,87	56%	24%	19%	Apagar Editar Calc
260	Brasileirão 2007	16/09/2007	18:10	Flamengo-RJ	1 x 1	Vasco-RJ	1,94	0,90	60%	23%	17%	Apagar Editar Calc

[Nova Partida](#)

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Centro Tecnológico - Sistemas de Informação
TCC desenvolvido por [Carlos Alberto Ceolato Junior](#)
Orientador: Prof PhD [Dalton Francisco de Andrade](#)

Figura 24 - Lista de Partida - Administrador

6.6.12. Partida – New.jsp

A Figura 25 mostra a tela utilizada para o cadastro de partidas, é necessário fornecer os clubes mandantes e visitantes, a data e hora da partida, resultado (opcional) e a temporada. Caso a partida não tenha sido disputada ou se queira efetuar o cálculo das probabilidades para a referida partida, omite-se a informação do resultado (Gols Mandante e Gols Visitante), sendo possível editar posteriormente.

The screenshot displays a web browser window titled "Nova Partida - Windows Internet Explorer". The address bar shows the URL "http://localhost:8084/chuteGol/faces/Admin/partida/List.jsp". The page features the "chuteGol" logo and a navigation menu with links for "Home", "Campeonato", "Temporada", "Clube", and "Partida". The main content area is titled "Nova partida" and contains the following form fields:

- Clube Mandante: Grêmio-RS (dropdown menu)
- Clube Visitante: Inter-RS (dropdown menu)
- Data (dd/MM/yyyy): [text input]
- Hora (HH:mm): [text input]
- Gols Mandante: [text input]
- Gols Visitante: [text input]
- Temporada: Brasileirão 2007 (dropdown menu)

A "Criar" button is located below the form fields. At the bottom of the page, the following text is displayed:

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Centro Tecnológico - Sistemas de Informação
TCC desenvolvido por [Carlos Alberto Cealato Junior](#)
Orientador: Prof PhD [Dalton Francisco de Andrade](#)

The browser's status bar at the bottom shows "Done" and "Local intranet" with a zoom level of 100%.

Figura 25 - Nova Partida – Administrador

6.6.13. Partida – Edit.jsp

A Figura 26 exibe a tela de edição de partidas, nela é possível alterar os dados da partida e informar o resultado da mesma. Os dados de médias e probabilidades são calculados pelo sistema, não sendo permitido ao administrador fazer qualquer edição.

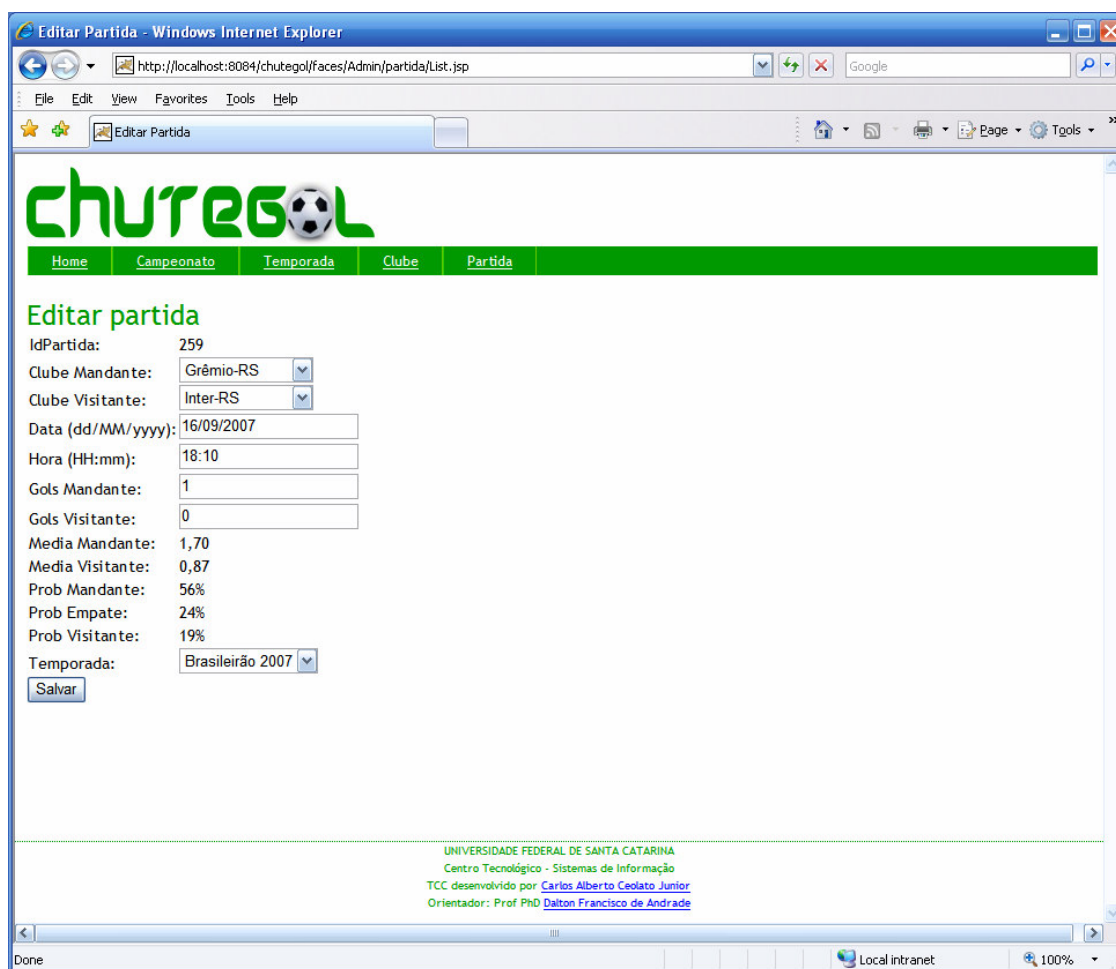


Figura 26 - Editar Partida - Administrador

7. Resultados

Para efeito de análise dos resultados obtidos com o uso do ChuteGol, iremos descartar os dados das primeiras quatro rodadas do Campeonato Brasileiro de 2007, visto que, como nenhum time tinha parâmetros iniciais, essas quatro rodadas servirão para inicializar os parâmetros dos clubes, no caso, duas partidas como mandante e duas como visitante.

Em cada rodada, analisaremos as partidas disputadas verificando a ocorrência da probabilidade maior, probabilidade média ou probabilidade menor. No caso de haver duas probabilidades iguais para a referida partida, a ocorrência será da probabilidade maior ou da probabilidade menor.

Atribuiremos uma pontuação para cada tipo de probabilidade (Maior, Média e Menor), sendo a pontuação igual ao percentual da probabilidade. Assim, teremos um valor mínimo (todos os jogos com resultado de menor probabilidade) e máximo (todos os jogos com resultado de maior probabilidade) de pontos que uma rodada pode atingir. Com isso, podemos normalizar o resultado da rodada e tentar estimar um intervalo sobre a pontuação que tem uma chance maior de ocorrer.

As Figuras 27 a 53 exibem os dados das vinte e sete rodadas do campeonato que foram utilizadas na análise dos resultados obtidos pelo ChuteGol. Vale ressaltar que as primeiras quatro rodadas foram descartadas para efeito de análise dos resultados.

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
1	Brasileirão 2007	12/05/2007	18:10	Figueirense-SC	3 x 6	Atlético-PR	0%	100%	0%
2	Brasileirão 2007	12/05/2007	18:10	Fluminense-RJ	2 x 2	Cruzeiro-MG	0%	100%	0%
3	Brasileirão 2007	12/05/2007	18:10	São Paulo-SP	2 x 0	Goiás-GO	0%	100%	0%
4	Brasileirão 2007	13/05/2007	16:00	Atlético-MG	2 x 1	Náutico-PE	0%	100%	0%
5	Brasileirão 2007	13/05/2007	16:00	Inter-RS	2 x 3	Botafogo-RJ	0%	100%	0%
6	Brasileirão 2007	13/05/2007	16:00	Flamengo-RJ	2 x 4	Palmeiras-SP	0%	100%	0%
7	Brasileirão 2007	13/05/2007	16:00	Paraná-PR	3 x 0	Grêmio-RS	0%	100%	0%
8	Brasileirão 2007	13/05/2007	18:10	Corinthians-SP	1 x 0	Juventude-RS	0%	100%	0%
9	Brasileirão 2007	13/05/2007	18:10	América-RN	0 x 1	Vasco-RJ	0%	100%	0%
10	Brasileirão 2007	13/05/2007	18:10	Sport-PE	4 x 1	Santos-SP	0%	100%	0%

Figura 27 - Rodada 1

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
11	Brasileirão 2007	19/05/2007	18:10	Juventude-RS	1 x 2	Paraná-PR	0%	100%	0%
12	Brasileirão 2007	19/05/2007	18:10	Atlético-PR	2 x 1	Inter-RS	0%	100%	0%
13	Brasileirão 2007	19/05/2007	18:10	Santos-SP	2 x 3	América-RN	0%	100%	0%
14	Brasileirão 2007	20/05/2007	16:00	Goiás-GO	1 x 3	Flamengo-RJ	0%	100%	0%
15	Brasileirão 2007	20/05/2007	16:00	Botafogo-RJ	2 x 1	Atlético-MG	0%	100%	0%
16	Brasileirão 2007	20/05/2007	16:00	Cruzeiro-MG	0 x 3	Corinthians-SP	0%	100%	0%
17	Brasileirão 2007	20/05/2007	16:00	Palmeiras-SP	2 x 1	Figueirense-SC	0%	100%	0%
18	Brasileirão 2007	20/05/2007	18:10	Vasco-RJ	3 x 1	Sport-PE	0%	100%	0%
19	Brasileirão 2007	20/05/2007	18:10	Náutico-PE	1 x 0	São Paulo-SP	0%	100%	0%
20	Brasileirão 2007	20/05/2007	18:10	Grêmio-RS	2 x 0	Fluminense-RJ	0%	100%	0%

Figura 28 - Rodada 2

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
21	Brasileirão 2007	26/05/2007	18:10	América-RN	0 x 1	Figueirense-SC	15%	70%	15%
22	Brasileirão 2007	26/05/2007	18:10	Corinthians-SP	0 x 0	Atlético-MG	24%	69%	7%
23	Brasileirão 2007	26/05/2007	18:10	Fluminense-RJ	3 x 0	Inter-RS	26%	56%	18%
24	Brasileirão 2007	27/05/2007	16:00	Atlético-PR	0 x 1	Santos-SP	39%	50%	11%
25	Brasileirão 2007	27/05/2007	16:00	Náutico-PE	2 x 2	Vasco-RJ	9%	83%	9%
26	Brasileirão 2007	27/05/2007	16:00	Grêmio-RS	1 x 0	Sport-PE	36%	58%	6%
27	Brasileirão 2007	27/05/2007	16:00	São Paulo-SP	0 x 0	Palmeiras-SP	24%	52%	24%
28	Brasileirão 2007	27/05/2007	18:10	Flamengo-RJ	2 x 2	Botafogo-RJ	19%	43%	38%
29	Brasileirão 2007	27/05/2007	18:10	Cruzeiro-MG	3 x 4	Paraná-PR	6%	58%	36%
30	Brasileirão 2007	27/05/2007	18:10	Goiás-GO	3 x 1	Juventude-RS	14%	64%	22%

Figura 29 - Rodada 3

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
31	Brasileirão 2007	02/06/2007	18:10	Figueirense-SC	2 x 1	Goiás-GO	29%	49%	22%
32	Brasileirão 2007	02/06/2007	18:10	Atlético-MG	1 x 1	Atlético-PR	27%	47%	27%
33	Brasileirão 2007	02/06/2007	18:10	Botafogo-RJ	3 x 0	Grêmio-RS	36%	58%	6%
34	Brasileirão 2007	03/06/2007	16:00	Palmeiras-SP	1 x 3	Cruzeiro-MG	26%	56%	18%
35	Brasileirão 2007	03/06/2007	16:00	Sport-PE	2 x 2	Flamengo-RJ	60%	29%	11%
36	Brasileirão 2007	03/06/2007	16:00	Santos-SP	1 x 1	Corinthians-SP	27%	27%	45%
37	Brasileirão 2007	03/06/2007	16:00	Juventude-RS	3 x 0	América-RN	59%	23%	18%
38	Brasileirão 2007	03/06/2007	18:10	Vasco-RJ	1 x 1	Fluminense-RJ	78%	20%	2%
39	Brasileirão 2007	03/06/2007	18:10	Paraná-PR	0 x 1	São Paulo-SP	57%	31%	12%
40	Brasileirão 2007	04/06/2007	20:30	Inter-RS	2 x 0	Náutico-PE	24%	52%	24%

Figura 30 - Rodada 4

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
41	Brasileirão 2007	09/06/2007	18:10	Náutico-PE	4 x 4	Paraná-PR	28%	38%	34%
42	Brasileirão 2007	09/06/2007	18:10	Palmeiras-SP	1 x 1	Botafogo-RJ	29%	44%	27%
43	Brasileirão 2007	09/06/2007	18:10	Vasco-RJ	4 x 0	Grêmio-RS	42%	53%	5%
44	Brasileirão 2007	10/06/2007	16:00	América-RN	1 x 2	Corinthians-SP	0%	62%	38%
45	Brasileirão 2007	10/06/2007	16:00	Fluminense-RJ	3 x 0	Sport-PE	47%	42%	10%
46	Brasileirão 2007	10/06/2007	16:00	Figueirense-SC	4 x 0	Flamengo-RJ	19%	43%	38%
47	Brasileirão 2007	10/06/2007	16:00	Atlético-PR	0 x 3	Goiás-GO	26%	62%	12%
48	Brasileirão 2007	10/06/2007	18:10	Cruzeiro-MG	2 x 3	Juventude-RS	29%	38%	33%
49	Brasileirão 2007	10/06/2007	18:10	São Paulo-SP	0 x 1	Atlético-MG	26%	68%	6%
50	Brasileirão 2007	10/06/2007	18:10	Inter-RS	1 x 0	Santos-SP	28%	47%	25%

Figura 31 - Rodada 5

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
51	Brasileirão 2007	16/06/2007	18:10	Grêmio-RS	0 x 2	Cruzeiro-MG	31%	57%	12%
52	Brasileirão 2007	16/06/2007	18:10	Corinthians-SP	0 x 0	Paraná-PR	31%	35%	34%
53	Brasileirão 2007	16/06/2007	18:10	Flamengo-RJ	2 x 2	Inter-RS	38%	38%	25%
54	Brasileirão 2007	17/06/2007	16:00	Goiás-GO	3 x 1	Palmeiras-SP	25%	42%	33%
55	Brasileirão 2007	17/06/2007	16:00	Atlético-MG	4 x 1	Figueirense-SC	24%	58%	18%
56	Brasileirão 2007	17/06/2007	16:00	São Paulo-SP	2 x 0	Vasco-RJ	20%	56%	24%
57	Brasileirão 2007	17/06/2007	16:00	Atlético-PR	1 x 1	Fluminense-RJ	19%	53%	28%
58	Brasileirão 2007	17/06/2007	18:10	Sport-PE	2 x 2	América-RN	34%	46%	20%
59	Brasileirão 2007	17/06/2007	18:10	Botafogo-RJ	3 x 1	Náutico-PE	28%	59%	13%
60	Brasileirão 2007	17/06/2007	18:10	Juventude-RS	0 x 2	Santos-SP	28%	52%	20%

Figura 32 - Rodada 6

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
61	Brasileirão 2007	23/06/2007	18:10	Paraná-PR	1 x 0	Sport-PE	56%	41%	3%
62	Brasileirão 2007	23/06/2007	18:10	Juventude-RS	1 x 1	Figueirense-SC	29%	42%	29%
63	Brasileirão 2007	24/06/2007	16:00	Palmeiras-SP	0 x 2	Atlético-PR	28%	43%	29%
64	Brasileirão 2007	24/06/2007	16:00	América-RN	0 x 1	Fluminense-RJ	21%	52%	27%
65	Brasileirão 2007	24/06/2007	16:00	Santos-SP	0 x 2	São Paulo-SP	20%	60%	20%
66	Brasileirão 2007	24/06/2007	16:00	Cruzeiro-MG	4 x 2	Atlético-MG	21%	35%	44%
67	Brasileirão 2007	24/06/2007	18:10	Náutico-PE	0 x 2	Goiás-GO	30%	35%	35%
68	Brasileirão 2007	24/06/2007	18:10	Inter-RS	0 x 2	Grêmio-RS	51%	39%	10%
69	Brasileirão 2007	15/08/2007	20:30	Botafogo-RJ	2 x 3	Corinthians-SP	27%	42%	31%
70	Brasileirão 2007	16/10/2007	20:30	Vasco-RJ	x	Flamengo-RJ	56%	33%	11%

Figura 33 - Rodada 7

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V
71	Brasileirão 2007	28/06/2007	20:30	Sport-PE	4 x 1	Náutico-PE	47%	36%	17%
72	Brasileirão 2007	28/06/2007	20:30	Figueirense-SC	0 x 0	São Paulo-SP	36%	41%	23%
73	Brasileirão 2007	30/06/2007	16:00	Cruzeiro-MG	3 x 1	Vasco-RJ	36%	29%	34%
74	Brasileirão 2007	30/06/2007	16:00	América-RN	0 x 3	Goiás-GO	9%	42%	49%
75	Brasileirão 2007	30/06/2007	16:00	Santos-SP	0 x 0	Grêmio-RS	31%	40%	29%
76	Brasileirão 2007	30/06/2007	18:10	Paraná-PR	2 x 2	Atlético-PR	30%	44%	26%
77	Brasileirão 2007	30/06/2007	18:10	Inter-RS	1 x 1	Atlético-MG	31%	38%	31%
78	Brasileirão 2007	30/06/2007	18:10	Corinthians-SP	0 x 1	Palmeiras-SP	29%	51%	20%
79	Brasileirão 2007	30/06/2007	18:10	Fluminense-RJ	1 x 2	Botafogo-RJ	46%	33%	20%
80	Brasileirão 2007	23/08/2007	20:30	Flamengo-RJ	4 x 0	Juventude-RS	34%	31%	34%

Figura 34 - Rodada 8

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
81	Brasileirão 2007	03/07/2007	19:30	Palmeiras-SP	2	x	0	América-RN	24%	39%	37%
82	Brasileirão 2007	03/07/2007	20:30	Figueirense-SC	2	x	1	Cruzeiro-MG	32%	40%	28%
83	Brasileirão 2007	03/07/2007	20:30	Atlético-PR	1	x	1	Náutico-PE	40%	34%	25%
84	Brasileirão 2007	03/07/2007	20:30	Grêmio-RS	3	x	1	Juventude-RS	47%	35%	18%
85	Brasileirão 2007	03/07/2007	20:30	Fluminense-RJ	0	x	0	Paraná-PR	40%	29%	30%
86	Brasileirão 2007	03/07/2007	20:30	São Paulo-SP	1	x	0	Inter-RS	47%	39%	14%
87	Brasileirão 2007	03/07/2007	21:45	Goiás-GO	1	x	1	Botafogo-RJ	36%	31%	32%
88	Brasileirão 2007	04/07/2007	16:00	Atlético-MG	1	x	1	Flamengo-RJ	50%	35%	15%
89	Brasileirão 2007	04/07/2007	16:00	Vasco-RJ	4	x	0	Santos-SP	47%	37%	15%
90	Brasileirão 2007	04/07/2007	16:00	Sport-PE	2	x	1	Corinthians-SP	37%	30%	32%

Figura 35 - Rodada 9

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
91	Brasileirão 2007	06/07/2007	20:30	Paraná-PR	0	x	1	América-RN	44%	35%	20%
92	Brasileirão 2007	07/07/2007	16:00	Corinthians-SP	1	x	1	Fluminense-RJ	18%	62%	20%
93	Brasileirão 2007	07/07/2007	16:00	Santos-SP	4	x	1	Cruzeiro-MG	20%	41%	38%
94	Brasileirão 2007	07/07/2007	16:00	Atlético-MG	0	x	1	Grêmio-RS	48%	37%	15%
95	Brasileirão 2007	07/07/2007	18:10	Botafogo-RJ	2	x	0	Atlético-PR	32%	31%	37%
96	Brasileirão 2007	07/07/2007	18:10	Goiás-GO	3	x	2	Sport-PE	53%	32%	15%
97	Brasileirão 2007	07/07/2007	18:10	Inter-RS	2	x	1	Figueirense-SC	34%	37%	29%
98	Brasileirão 2007	07/07/2007	18:10	Náutico-PE	0	x	1	Palmeiras-SP	29%	32%	39%
99	Brasileirão 2007	07/07/2007	18:10	Juventude-RS	2	x	0	Vasco-RJ	33%	38%	28%
100	Brasileirão 2007	07/07/2007	18:10	São Paulo-SP	0	x	0	Flamengo-RJ	44%	40%	15%

Figura 36 - Rodada 10

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
101	Brasileirão 2007	12/07/2007	20:30	Paraná-PR	1	x	2	Figueirense-SC	44%	34%	21%
102	Brasileirão 2007	12/07/2007	20:30	Juventude-RS	1	x	1	Náutico-PE	45%	33%	21%
103	Brasileirão 2007	12/07/2007	20:30	Vasco-RJ	1	x	0	Atlético-PR	58%	28%	14%
104	Brasileirão 2007	14/07/2007	16:00	Sport-PE	2	x	0	Atlético-MG	52%	28%	20%
105	Brasileirão 2007	14/07/2007	16:00	Santos-SP	3	x	0	Botafogo-RJ	34%	32%	34%
106	Brasileirão 2007	14/07/2007	16:00	Grêmio-RS	1	x	1	Palmeiras-SP	33%	38%	29%
107	Brasileirão 2007	14/07/2007	20:00	América-RN	1	x	2	Inter-RS	25%	38%	37%
108	Brasileirão 2007	14/07/2007	20:00	Cruzeiro-MG	2	x	1	Goiás-GO	27%	28%	46%
109	Brasileirão 2007	14/07/2007	20:30	Corinthians-SP	1	x	1	São Paulo-SP	14%	62%	24%
110	Brasileirão 2007	16/08/2007	20:30	Fluminense-RJ	0	x	1	Flamengo-RJ	44%	37%	19%

Figura 37 - Rodada 11

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
111	Brasileirão 2007	14/06/2007	20:30	Botafogo-RJ	4	x	0	Vasco-RJ	55%	29%	16%
112	Brasileirão 2007	18/07/2007	19:30	Figueirense-SC	0	x	1	Sport-PE	57%	28%	15%
113	Brasileirão 2007	18/07/2007	20:30	Atlético-PR	4	x	0	Juventude-RS	45%	30%	25%
114	Brasileirão 2007	18/07/2007	20:30	Atlético-MG	4	x	1	América-RN	36%	36%	27%
115	Brasileirão 2007	18/07/2007	20:30	Náutico-PE	1	x	4	Cruzeiro-MG	35%	30%	34%
116	Brasileirão 2007	18/07/2007	21:45	São Paulo-SP	0	x	1	Fluminense-RJ	33%	51%	16%
117	Brasileirão 2007	18/07/2007	21:45	Inter-RS	3	x	0	Corinthians-SP	28%	32%	39%
118	Brasileirão 2007	19/07/2007	20:30	Palmeiras-SP	2	x	2	Santos-SP	46%	35%	19%
119	Brasileirão 2007	19/07/2007	20:30	Goiás-GO	0	x	0	Grêmio-RS	43%	32%	25%
120	Brasileirão 2007	19/07/2007	20:30	Flamengo-RJ	1	x	2	Paraná-PR	41%	29%	31%

Figura 38 - Rodada 12

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
121	Brasileirão 2007	21/07/2007	18:10	América-RN	2	x	1	Atlético-PR	22%	33%	45%
122	Brasileirão 2007	21/07/2007	18:10	Juventude-RS	2	x	0	Inter-RS	38%	35%	27%
123	Brasileirão 2007	21/07/2007	18:10	Vasco-RJ	4	x	0	Atlético-MG	64%	27%	9%
124	Brasileirão 2007	22/07/2007	16:00	Corinthians-SP	0	x	3	Náutico-PE	40%	37%	23%
125	Brasileirão 2007	22/07/2007	16:00	Cruzeiro-MG	1	x	2	São Paulo-SP	37%	31%	31%
126	Brasileirão 2007	22/07/2007	16:00	Grêmio-RS	1	x	0	Flamengo-RJ	36%	38%	25%
127	Brasileirão 2007	22/07/2007	16:00	Fluminense-RJ	3	x	0	Goiás-GO	26%	35%	39%
128	Brasileirão 2007	22/07/2007	18:10	Santos-SP	3	x	1	Figueirense-SC	48%	29%	22%
129	Brasileirão 2007	22/07/2007	18:10	Paraná-PR	1	x	0	Palmeiras-SP	29%	36%	35%
130	Brasileirão 2007	22/07/2007	18:10	Sport-PE	3	x	3	Botafogo-RJ	55%	27%	18%

Figura 39 - Rodada 13

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
131	Brasileirão 2007	25/07/2007	19:30	Atlético-PR	2	x	2	Cruzeiro-MG	39%	28%	32%
132	Brasileirão 2007	25/07/2007	19:30	Goiás-GO	1	x	0	Santos-SP	48%	29%	23%
133	Brasileirão 2007	25/07/2007	20:30	Náutico-PE	0	x	2	Grêmio-RS	23%	32%	44%
134	Brasileirão 2007	25/07/2007	20:30	Flamengo-RJ	3	x	1	América-RN	46%	27%	26%
135	Brasileirão 2007	25/07/2007	20:30	Palmeiras-SP	3	x	2	Vasco-RJ	57%	27%	16%
136	Brasileirão 2007	25/07/2007	21:45	Figueirense-SC	2	x	2	Corinthians-SP	42%	31%	27%
137	Brasileirão 2007	25/07/2007	21:45	Inter-RS	1	x	0	Paraná-PR	37%	30%	32%
138	Brasileirão 2007	26/07/2007	15:00	Botafogo-RJ	3	x	1	Juventude-RS	71%	19%	10%
139	Brasileirão 2007	26/07/2007	20:30	Atlético-MG	3	x	0	Fluminense-RJ	42%	35%	23%
140	Brasileirão 2007	26/07/2007	20:30	São Paulo-SP	3	x	1	Sport-PE	42%	40%	18%

Figura 40 - Rodada 14

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
141	Brasileirão 2007	28/07/2007	18:10	Santos-SP	1	x	2	Náutico-PE	48%	28%	24%
142	Brasileirão 2007	28/07/2007	18:10	Grêmio-RS	1	x	1	Atlético-PR	45%	32%	22%
143	Brasileirão 2007	28/07/2007	18:10	Vasco-RJ	4	x	1	Goiás-GO	62%	26%	12%
144	Brasileirão 2007	29/07/2007	16:00	Corinthians-SP	2	x	2	Flamengo-RJ	23%	41%	36%
145	Brasileirão 2007	29/07/2007	16:00	Sport-PE	1	x	5	Inter-RS	55%	26%	19%
146	Brasileirão 2007	29/07/2007	16:00	Juventude-RS	1	x	1	Palmeiras-SP	38%	37%	25%
147	Brasileirão 2007	29/07/2007	16:00	Paraná-PR	1	x	3	Atlético-MG	55%	29%	16%
148	Brasileirão 2007	29/07/2007	18:10	Fluminense-RJ	1	x	1	Figueirense-SC	53%	28%	19%
149	Brasileirão 2007	29/07/2007	18:10	América-RN	0	x	1	São Paulo-SP	19%	34%	46%
150	Brasileirão 2007	29/07/2007	18:10	Cruzeiro-MG	3	x	2	Botafogo-RJ	42%	25%	33%

Figura 41 - Rodada 15

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
151	Brasileirão 2007	01/08/2007	19:30	Goiás-GO	2	x	0	Paraná-PR	40%	32%	28%
152	Brasileirão 2007	01/08/2007	19:30	Náutico-PE	0	x	0	Fluminense-RJ	25%	31%	43%
153	Brasileirão 2007	01/08/2007	20:30	Palmeiras-SP	1	x	2	Sport-PE	51%	27%	22%
154	Brasileirão 2007	01/08/2007	20:30	Botafogo-RJ	4	x	2	América-RN	63%	22%	15%
155	Brasileirão 2007	01/08/2007	21:45	Atlético-PR	2	x	2	Corinthians-SP	43%	28%	30%
156	Brasileirão 2007	01/08/2007	21:45	Atlético-MG	1	x	2	Santos-SP	60%	26%	14%
157	Brasileirão 2007	01/08/2007	21:45	Inter-RS	0	x	2	Vasco-RJ	61%	24%	14%
158	Brasileirão 2007	02/08/2007	20:30	Figueirense-SC	1	x	0	Grêmio-RS	32%	35%	33%
159	Brasileirão 2007	02/08/2007	20:30	São Paulo-SP	3	x	1	Juventude-RS	65%	24%	11%
160	Brasileirão 2007	12/09/2007	20:30	Flamengo-RJ	3	x	1	Cruzeiro-MG	46%	25%	29%

Figura 42 - Rodada 16

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
161	Brasileirão 2007	04/08/2007	18:10	Corinthians-SP	1	x	0	Goiás-GO	37%	29%	34%
162	Brasileirão 2007	04/08/2007	18:10	América-RN	1	x	5	Náutico-PE	23%	29%	47%
163	Brasileirão 2007	04/08/2007	18:10	Sport-PE	3	x	2	Atlético-PR	42%	26%	32%
164	Brasileirão 2007	05/08/2007	16:00	Fluminense-RJ	0	x	1	Palmeiras-SP	42%	34%	24%
165	Brasileirão 2007	05/08/2007	16:00	Cruzeiro-MG	3	x	2	Inter-RS	40%	25%	35%
166	Brasileirão 2007	05/08/2007	16:00	Santos-SP	3	x	0	Flamengo-RJ	45%	29%	25%
167	Brasileirão 2007	05/08/2007	16:00	Juventude-RS	1	x	2	Atlético-MG	47%	29%	23%
168	Brasileirão 2007	05/08/2007	18:10	Vasco-RJ	2	x	2	Figueirense-SC	68%	21%	11%
169	Brasileirão 2007	05/08/2007	18:10	Paraná-PR	0	x	0	Botafogo-RJ	39%	27%	35%
170	Brasileirão 2007	05/08/2007	18:10	Grêmio-RS	0	x	2	São Paulo-SP	29%	39%	32%

Figura 43 - Rodada 17

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
171	Brasileirão 2007	08/08/2007	19:30	Cruzeiro-MG	2	x	0	Sport-PE	47%	24%	28%
172	Brasileirão 2007	08/08/2007	19:30	Juventude-RS	0	x	0	Fluminense-RJ	38%	36%	25%
173	Brasileirão 2007	08/08/2007	20:30	Náutico-PE	4	x	2	Figueirense-SC	29%	28%	42%
174	Brasileirão 2007	08/08/2007	21:45	Botafogo-RJ	0	x	2	São Paulo-SP	45%	28%	28%
175	Brasileirão 2007	08/08/2007	21:45	Atlético-PR	2	x	0	Flamengo-RJ	48%	27%	24%
176	Brasileirão 2007	08/08/2007	21:45	Goiás-GO	3	x	2	Atlético-MG	46%	27%	26%
177	Brasileirão 2007	09/08/2007	20:30	Vasco-RJ	2	x	0	Corinthians-SP	59%	22%	19%
178	Brasileirão 2007	09/08/2007	20:30	Grêmio-RS	3	x	0	América-RN	48%	27%	26%
179	Brasileirão 2007	09/08/2007	20:30	Palmeiras-SP	1	x	1	Inter-RS	39%	26%	35%
180	Brasileirão 2007	09/08/2007	20:30	Santos-SP	2	x	0	Paraná-PR	52%	27%	21%

Figura 44 - Rodada 18

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
181	Brasileirão 2007	11/08/2007	18:10	Sport-PE	3	x	0	Juventude-RS	59%	21%	20%
182	Brasileirão 2007	11/08/2007	18:10	São Paulo-SP	2	x	0	Atlético-PR	51%	28%	21%
183	Brasileirão 2007	11/08/2007	18:10	Flamengo-RJ	2	x	1	Náutico-PE	35%	26%	39%
184	Brasileirão 2007	12/08/2007	16:00	Corinthians-SP	2	x	1	Grêmio-RS	27%	38%	34%
185	Brasileirão 2007	12/08/2007	16:00	Atlético-MG	1	x	2	Palmeiras-SP	42%	31%	26%
186	Brasileirão 2007	12/08/2007	16:00	Figueirense-SC	1	x	1	Botafogo-RJ	44%	28%	27%
187	Brasileirão 2007	12/08/2007	16:00	Inter-RS	1	x	0	Goiás-GO	46%	30%	24%
188	Brasileirão 2007	12/08/2007	18:10	América-RN	1	x	2	Cruzeiro-MG	23%	25%	52%
189	Brasileirão 2007	12/08/2007	18:10	Paraná-PR	0	x	0	Vasco-RJ	42%	30%	29%
190	Brasileirão 2007	12/08/2007	18:10	Fluminense-RJ	3	x	0	Santos-SP	42%	31%	26%

Figura 45 - Rodada 19

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
191	Brasileirão 2007	18/08/2007	18:10	Atlético-PR	1	x	1	Figueirense-SC	52%	24%	24%
192	Brasileirão 2007	18/08/2007	18:10	Santos-SP	2	x	0	Sport-PE	61%	24%	15%
193	Brasileirão 2007	18/08/2007	18:10	Grêmio-RS	2	x	0	Paraná-PR	49%	30%	21%
194	Brasileirão 2007	19/08/2007	16:00	Palmeiras-SP	2	x	1	Flamengo-RJ	48%	28%	23%
195	Brasileirão 2007	19/08/2007	16:00	Náutico-PE	0	x	1	Atlético-MG	39%	25%	36%
196	Brasileirão 2007	19/08/2007	16:00	Vasco-RJ	2	x	0	América-RN	73%	18%	10%
197	Brasileirão 2007	19/08/2007	16:00	Juventude-RS	2	x	2	Corinthians-SP	43%	30%	28%
198	Brasileirão 2007	19/08/2007	18:10	Botafogo-RJ	1	x	1	Inter-RS	47%	25%	28%
199	Brasileirão 2007	19/08/2007	18:10	Goiás-GO	0	x	0	São Paulo-SP	30%	30%	39%
200	Brasileirão 2007	19/08/2007	18:10	Cruzeiro-MG	4	x	2	Fluminense-RJ	48%	28%	23%

Figura 46 - Rodada 20

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
201	Brasileirão 2007	25/08/2007	18:10	Corinthians-SP	0	x	3	Cruzeiro-MG	33%	28%	40%
202	Brasileirão 2007	25/08/2007	18:10	Fluminense-RJ	1	x	1	Grêmio-RS	47%	33%	19%
203	Brasileirão 2007	25/08/2007	18:10	Inter-RS	1	x	0	Atlético-PR	48%	29%	22%
204	Brasileirão 2007	26/08/2007	16:00	Paraná-PR	3	x	1	Juventude-RS	59%	26%	15%
205	Brasileirão 2007	26/08/2007	16:00	Atlético-MG	1	x	2	Botafogo-RJ	45%	27%	29%
206	Brasileirão 2007	26/08/2007	16:00	São Paulo-SP	5	x	0	Náutico-PE	43%	27%	29%
207	Brasileirão 2007	26/08/2007	16:00	Figueirense-SC	1	x	2	Palmeiras-SP	33%	33%	34%
208	Brasileirão 2007	26/08/2007	18:10	América-RN	1	x	4	Santos-SP	33%	28%	39%
209	Brasileirão 2007	26/08/2007	18:10	Sport-PE	0	x	0	Vasco-RJ	53%	24%	23%
210	Brasileirão 2007	26/08/2007	18:10	Flamengo-RJ	3	x	1	Goiás-GO	54%	26%	21%

Figura 47 - Rodada 21

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
211	Brasileirão 2007	29/08/2007	19:30	Paraná-PR	2	x	2	Cruzeiro-MG	30%	28%	43%
212	Brasileirão 2007	29/08/2007	19:30	Juventude-RS	2	x	0	Goiás-GO	48%	28%	23%
213	Brasileirão 2007	29/08/2007	20:30	Figueirense-SC	3	x	1	América-RN	56%	25%	20%
214	Brasileirão 2007	29/08/2007	20:30	Botafogo-RJ	1	x	1	Flamengo-RJ	56%	25%	20%
215	Brasileirão 2007	29/08/2007	21:30	Palmeiras-SP	0	x	1	São Paulo-SP	26%	32%	41%
216	Brasileirão 2007	29/08/2007	21:45	Sport-PE	2	x	0	Grêmio-RS	46%	29%	26%
217	Brasileirão 2007	29/08/2007	21:45	Atlético-MG	5	x	2	Corinthians-SP	43%	26%	31%
218	Brasileirão 2007	29/08/2007	21:45	Inter-RS	1	x	4	Fluminense-RJ	48%	32%	19%
219	Brasileirão 2007	30/08/2007	20:30	Santos-SP	3	x	1	Atlético-PR	64%	24%	12%
220	Brasileirão 2007	30/08/2007	20:30	Vasco-RJ	4	x	1	Náutico-PE	63%	21%	16%

Figura 48 - Rodada 22

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
221	Brasileirão 2007	01/09/2007	18:10	São Paulo-SP	6	x	0	Paraná-PR	67%	23%	10%
222	Brasileirão 2007	01/09/2007	18:10	América-RN	0	x	3	Juventude-RS	43%	24%	33%
223	Brasileirão 2007	01/09/2007	18:10	Flamengo-RJ	1	x	1	Sport-PE	59%	23%	18%
224	Brasileirão 2007	01/09/2007	16:00	Corinthians-SP	2	x	0	Santos-SP	31%	28%	42%
225	Brasileirão 2007	02/09/2007	16:00	Atlético-PR	1	x	0	Atlético-MG	44%	27%	30%
226	Brasileirão 2007	02/09/2007	16:00	Cruzeiro-MG	5	x	0	Palmeiras-SP	43%	25%	32%
227	Brasileirão 2007	02/09/2007	16:00	Grêmio-RS	3	x	0	Botafogo-RJ	46%	29%	26%
228	Brasileirão 2007	02/09/2007	16:00	Goiás-GO	2	x	1	Figueirense-SC	49%	26%	25%
229	Brasileirão 2007	02/09/2007	18:10	Fluminense-RJ	1	x	1	Vasco-RJ	49%	31%	19%
230	Brasileirão 2007	02/09/2007	18:10	Náutico-PE	1	x	1	Inter-RS	32%	27%	40%

Figura 49 - Rodada 23

Id	Temporada	Data	Hora	Mandante		x		Visitante	Prob M	Prob E	Prob V
231	Brasileirão 2007	05/09/2007	19:30	Corinthians-SP	1	x	0	América-RN	55%	25%	21%
232	Brasileirão 2007	05/09/2007	19:30	Goiás-GO	2	x	3	Atlético-PR	56%	26%	18%
233	Brasileirão 2007	05/09/2007	20:30	Juventude-RS	1	x	0	Cruzeiro-MG	34%	27%	40%
234	Brasileirão 2007	05/09/2007	20:30	Flamengo-RJ	4	x	1	Figueirense-SC	52%	24%	24%
235	Brasileirão 2007	05/09/2007	21:45	Atlético-MG	0	x	0	São Paulo-SP	31%	29%	39%
236	Brasileirão 2007	05/09/2007	21:45	Santos-SP	2	x	1	Inter-RS	55%	25%	20%
237	Brasileirão 2007	05/09/2007	21:45	Grêmio-RS	3	x	1	Vasco-RJ	57%	28%	15%
238	Brasileirão 2007	06/09/2007	20:30	Paraná-PR	2	x	4	Náutico-PE	49%	25%	26%
239	Brasileirão 2007	06/09/2007	20:30	Botafogo-RJ	1	x	1	Palmeiras-SP	47%	27%	27%
240	Brasileirão 2007	06/09/2007	20:30	Sport-PE	0	x	2	Fluminense-RJ	46%	27%	28%

Figura 50 - Rodada 24

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V	
241	Brasileirão 2007	08/09/2007	16:00	Figueirense-SC	2	x 1	Atlético-MG	45%	28%	28%
242	Brasileirão 2007	08/09/2007	16:00	Vasco-RJ	0	x 2	São Paulo-SP	54%	28%	18%
243	Brasileirão 2007	08/09/2007	16:00	Inter-RS	3	x 0	Flamengo-RJ	42%	29%	28%
244	Brasileirão 2007	08/09/2007	18:10	Cruzeiro-MG	2	x 0	Grêmio-RS	61%	22%	17%
245	Brasileirão 2007	08/09/2007	18:10	Santos-SP	1	x 0	Juventude-RS	63%	22%	15%
246	Brasileirão 2007	09/09/2007	19:00	Fluminense-RJ	2	x 0	Atlético-PR	50%	27%	23%
247	Brasileirão 2007	09/09/2007	19:00	América-RN	1	x 1	Sport-PE	27%	26%	47%
248	Brasileirão 2007	09/09/2007	19:00	Náutico-PE	4	x 1	Botafogo-RJ	39%	28%	34%
249	Brasileirão 2007	09/09/2007	19:00	Palmeiras-SP	2	x 0	Goiás-GO	51%	27%	22%
250	Brasileirão 2007	09/09/2007	19:00	Paraná-PR	1	x 0	Corinthians-SP	43%	24%	33%

Figura 51 - Rodada 25

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V	
251	Brasileirão 2007	15/09/2007	18:10	Goiás-GO	0	x 3	Náutico-PE	44%	23%	32%
252	Brasileirão 2007	15/09/2007	18:10	Fluminense-RJ	2	x 0	América-RN	65%	23%	13%
253	Brasileirão 2007	15/09/2007	18:10	São Paulo-SP	2	x 1	Santos-SP	66%	22%	12%
254	Brasileirão 2007	16/09/2007	16:00	Atlético-MG	3	x 4	Cruzeiro-MG	43%	26%	31%
255	Brasileirão 2007	16/09/2007	16:00	Corinthians-SP	0	x 1	Botafogo-RJ	51%	26%	23%
256	Brasileirão 2007	16/09/2007	16:00	Atlético-PR	2	x 1	Palmeiras-SP	47%	28%	24%
257	Brasileirão 2007	16/09/2007	16:00	Sport-PE	3	x 1	Paraná-PR	58%	25%	18%
258	Brasileirão 2007	16/09/2007	18:10	Figueirense-SC	4	x 1	Juventude-RS	55%	24%	21%
259	Brasileirão 2007	16/09/2007	18:10	Grêmio-RS	1	x 0	Inter-RS	56%	24%	19%
260	Brasileirão 2007	16/09/2007	18:10	Flamengo-RJ	1	x 1	Vasco-RJ	60%	23%	17%

Figura 52 - Rodada 26

Id	Temporada	Data	Hora	Mandante	x	Visitante	Prob M	Prob E	Prob V	
261	Brasileirão 2007	22/09/2007	18:10	Goiás-GO	1	x 1	América-RN	51%	25%	24%
262	Brasileirão 2007	22/09/2007	18:10	São Paulo-SP	2	x 0	Figueirense-SC	69%	19%	12%
263	Brasileirão 2007	22/09/2007	18:10	Grêmio-RS	1	x 0	Santos-SP	59%	24%	17%
264	Brasileirão 2007	23/09/2007	16:00	Palmeiras-SP	1	x 0	Corinthians-SP	50%	26%	24%
265	Brasileirão 2007	23/09/2007	16:00	Náutico-PE	2	x 0	Sport-PE	48%	27%	26%
266	Brasileirão 2007	23/09/2007	16:00	Atlético-MG	2	x 2	Inter-RS	43%	25%	32%
267	Brasileirão 2007	23/09/2007	16:00	Juventude-RS	2	x 2	Flamengo-RJ	58%	27%	15%
268	Brasileirão 2007	23/09/2007	18:10	Atlético-PR	2	x 1	Paraná-PR	63%	23%	14%
269	Brasileirão 2007	23/09/2007	18:10	Botafogo-RJ	0	x 2	Fluminense-RJ	36%	28%	37%
270	Brasileirão 2007	23/09/2007	18:10	Vasco-RJ	0	x 2	Cruzeiro-MG	49%	23%	27%

Figura 53 - Rodada 27

As Figuras 54 a 76 exibem as estatísticas das vinte e três rodadas analisadas. Nelas são exibidas informações como:

- Probabilidade Maior – a quantidade de partidas na qual ocorreu o resultado de maior probabilidade;
- Probabilidade Menor – a quantidade de partidas na qual ocorreu o resultado de menor probabilidade;

- Probabilidade Média – a quantidade de partidas na qual não ocorreu nem o resultado menos provável, nem o mais provável;
- Pontuação Máxima – A pontuação da rodada caso ocorram todos os resultados de maior probabilidade;
- Pontuação Mínima – A pontuação da rodada caso ocorram todos os resultados de menor probabilidade;
- Pontuação da Rodada – A pontuação da rodada levando em consideração os resultados ocorridos;
- Pontuação Normalizada da Rodada – Valor normalizado da pontuação da rodada, para obter-se este valor, é utilizado a seguinte fórmula;

$$P_{Normalizada} = \frac{P_{Rodada} - P_{Minima}}{P_{Maxima} - P_{Minima}}$$

- Placares exatos – A quantidade de partidas na qual ocorreu o placar de maior probabilidade;

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
3	4	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
502	161	307
Pontuação Normalizada da Rodada:		0,43
Placares Exatos		0

Figura 54 - Estatísticas Rodada 5

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
4	2	4
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
496	203	301
Pontuação Normalizada da Rodada:		0,33
Placares Exatos		1

Figura 55 - Estatísticas Rodada 6

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
3	3	4
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
425	189	271
Pontuação Normalizada da Rodada:		0,35
Placares Exatos		0

Figura 56 - Estatísticas Rodada 7

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
8	0	2
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
426	235	369
Pontuação Normalizada da Rodada:		0,70
Placares Exatos		2

Figura 57 - Estatísticas Rodada 8

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
4	3	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
423	229	363
Pontuação Normalizada da Rodada:		0,69
Placares Exatos		0

Figura 58 - Estatísticas Rodada 9

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
3	4	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
443	220	348
Pontuação Normalizada da Rodada:		0,57
Placares Exatos		2

Figura 59 - Estatísticas Rodada 10

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
5	2	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
461	222	381
Pontuação Normalizada da Rodada:		0,67
Placares Exatos		0

Figura 60 - Estatísticas Rodada 11

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
3	4	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
448	230	327
Pontuação Normalizada da Rodada:		0,44
Placares Exatos		0

Figura 61 - Estatísticas Rodada 12

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
3	2	5
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
440	232	344
Pontuação Normalizada da Rodada:		0,54
Placares Exatos		0

Figura 62 - Estatísticas Rodada 13

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
8	1	1
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
468	224	446
Pontuação Normalizada da Rodada:		0,91
Placares Exatos		2

Figura 63 - Estatísticas Rodada 14

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
4	3	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
485	204	347
Pontuação Normalizada da Rodada:		0,51
Placares Exatos		1

Figura 64 - Estatísticas Rodada 15

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
4	1	5
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
507	214	355
Pontuação Normalizada da Rodada:		0,48
Placares Exatos		0

Figura 65 - Estatísticas Rodada 16

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
5	2	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
446	242	338
Pontuação Normalizada da Rodada:		0,47
Placares Exatos		0

Figura 66 - Estatísticas Rodada 17

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
6	2	2
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
464	247	419
Pontuação Normalizada da Rodada:		0,79
Placares Exatos		2

Figura 67 - Estatísticas Rodada 18

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
5	3	2
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
455	249	396
Pontuação Normalizada da Rodada:		0,71
Placares Exatos		1

Figura 68 - Estatísticas Rodada 19

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
5	2	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
499	224	424
Pontuação Normalizada da Rodada:		0,73
Placares Exatos		3

Figura 69 - Estatísticas Rodada 20

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
7	3	0
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
462	243	403
Pontuação Normalizada da Rodada:		0,73
Placares Exatos		1

Figura 70 - Estatísticas Rodada 21

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
7	1	2
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
508	216	433
Pontuação Normalizada da Rodada:		0,74
Placares Exatos		1

Figura 71 - Estatísticas Rodada 22

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
5	4	1
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
482	229	394
Pontuação Normalizada da Rodada:		0,65
Placares Exatos		1

Figura 72 - Estatísticas Rodada 23

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
4	3	3
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
496	233	381
Pontuação Normalizada da Rodada:		0,56
Placares Exatos		2

Figura 73 - Estatísticas Rodada 24

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
8	0	2
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
495	229	438
Pontuação Normalizada da Rodada:		0,79
Placares Exatos		1

Figura 74 - Estatísticas Rodada 25

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
6	3	1
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
545	196	456
Pontuação Normalizada da Rodada:		0,74
Placares Exatos		1

Figura 75 - Estatísticas Rodada 26

Probabilidade Maior	Probabilidade Media	Probabilidade Menor
6	3	1
Pontuação Máxima	Pontuação Mínima	Pontuação da Rodada
527	208	430
Pontuação Normalizada da Rodada:		0,70
Placares Exatos		3

Figura 76 - Estatísticas Rodada 27

7.1. Resumo da análise

Foram analisados 229 jogos, desconsiderando os primeiros 40 que foram usados para inicializar os parâmetros dos clubes. Em 116 jogos ocorreu o resultado mais provável (50,65%), em 55 o resultado foi o de probabilidade média (24,02%) e em 58, o de menor probabilidade (25,33%).

Se considerarmos que a chance de acertamos um resultado ao acaso é de 33,3%, pois temos três possibilidades, o Chutegol conseguiu um ganho de aproximadamente 52%.

Analisando o placar mais provável de ocorrer, o Chutegol conseguiu um acerto de 24 resultados (10,48%). Se considerarmos probabilidades iguais para cada um dos 25 possíveis resultados analisados, teríamos uma chance de 4% de acertar o resultado ao acaso. Com isso, tivemos um ganho de 162%.

A análise da pontuação das rodadas nos mostra valores desde 0,33 até 0,91, o valor médio obtido nesta análise foi de 0,62 com um desvio padrão de 0,15, se utilizamos como parâmetro, um desvio padrão em torno da média, podemos dizer que, estatisticamente, em 68% dos casos teremos uma rodada apresentando pontuação entre 0,47 e 0,77.

Estes dados podem ser utilizados para filtro de apostas num jogo da loteria, pelo qual o apostador pode excluir as combinações de resultados que estejam fora do intervalo escolhido.

8. Considerações Finais e Trabalhos futuros

Com este trabalho, foi possível mostrar a utilidade de um sistema de auxílio ao prognóstico de jogos de futebol. Vale ressaltar que não é possível prever com exatidão o resultado de todas as partidas. Como foi citado no início do trabalho, existem diversos fatores, intra e extra campo que fazem com que uma partida de futebol tenha um clima envolvente e fascinante para todas as torcidas, o que transforma um jogo num espetáculo.

Um dos desafios deste trabalho foi a estimação dos parâmetros. A solução adotada foi arbitrarmos um método de cálculo para estimarmos os parâmetros. Os resultados, embora satisfatórios, deixam um questionamento quanto a existência de métodos mais eficazes se considerarmos que haja dependência na quantidade de gols de cada equipe.

A população da base de dados com resultados passados é um trabalho que pode trazer um enriquecimento a análise dos jogos, tornando possível verificarmos confrontos históricos entre duas equipes e usando estas informações na escolha do prognóstico mais adequado para a disputa.

Durante o desenvolvimento do projeto, foi possível colocar em prática grande parte do conhecimento adquirido no decorrer do curso, disciplinas como Banco de Dados, Engenharia de software, Análise e Projeto de Sistemas Orientados a Objetos e Programação Web, foram de grande valia na conclusão deste trabalho.

Por fim, convido todos a acessarem o projeto no endereço <http://java.inf.ufsc.br/chutegol>, onde poderão fazer uso do mesmo para definirem seus prognósticos para as próximas partidas.

9. Bibliografia

ARRUDA, M.L., (2000), **Poisson, Bayes, Futebol e DeFinetti**, São Paulo, Dissertação de Mestrado IME/USP.

BARBETTA, P.A., (2004), **Estatística para cursos de engenharia e informática**, São Paulo, Atlas.

BARNETT, V. (1982), *Comparative Statistical Inference*, New York: Wiley.

DEITEL, H. M., DEITEL. P. J. (2003), *JAVA – Como Programar*, São Paulo, Bookman.

DUARTE, O., (1997), **Futebol, histórias e regras**, São Paulo, Makron.

ELMASRI, R., NAVATHE, S.B., (2005), **Sistemas de banco de dados**, São Paulo, Addison Wesley.

GRIFFITHS, R.C., MILNE, R.K & WOOD, R. (1979), **Aspects of Correlation in Bivariate Poisson Distributions and Processes**, *Australian Journal of Statistics* 21.

HEUSER, C.A., (2004), **Projeto de banco de dados**, Porto Alegre, Ed Sagra Luzzatto.

HOLGATE, P. (1964), **Estimation for the Bivariate Poisson Distribution**, *Biometrika* 51, 241-245.

LARMAN, Craig, (2004), **Utilizando UML e Padrões**: uma introdução à análise e ao projeto orientado a objetos e ao Processo Unificado / Craig Larman; trad. Luiz Augusto Meirelles Salgado e João Tortello. – 2.ed. – Porto Alegre: Bookman.

WITTER, J.S., (1990), **O que é futebol**, São Paulo, Ed. Brasiliense.

ECLI01, IDE Eclipse, disponível em <http://www.eclipse.org/>, acesso em junho de 2007.

JAVA01, Linguagem Java, Disponível em <http://java.sun.com/>, acesso em novembro de 2006.

JSF01, Java Server Faces, Disponível em <http://java.sun.com/javaee/jaserverfaces/>, acesso em junho de 2007.

NETB01, IDE Netbeans, disponível em <http://www.netbeans.org/>, acesso em junho de 2007.

SUN01, Sun Microsystems, disponível em <http://www.sun.org/>, acesso em agosto de 2007.

10. Apêndice A - Artigo

ChuteGol – CHance de Um Time Em GOLs

Carlos Alberto Ceolato Junior
UFSC – Universidade Federal de Santa Catarina
cceolato@inf.ufsc.br

Abstract

Football is, worldwide, one of the sports that most fascinates people. Every year, hundreds of clubs come into the field with a single aim, to be champion.

Discovering the outcome of the football matches to make bets between friends, compete for prizes or try to win the lottery is an extra ingredient for the fans of this sport.

The objective of this work is to develop an information system capable to help the user in the prognosis of football matches, giving the probability to occur victory of the team's principal, the draw or victory the visiting team. For it, we defined a model based in Poisson to make the calculations, building a database to store the information and developed a web system able to display the information so the user can through its analysis, define their prognoses.

Keyword: football, prognoses, probabilistic model

1. Introdução

O Brasil é conhecido como o país do futebol, todos os anos, centenas de clubes entram em campo buscando títulos.

O governo Brasileiro disponibiliza duas loterias nas quais o objetivo é acertar o resultado de partidas de futebol, diversos sites na internet oferecem bolões onde os usuários cadastrados acumulam pontos acertando o resultado de jogos em busca de prêmios.

Neste contexto, este trabalho se propõe a desenvolver um sistema de informação capaz de auxiliar os usuários na escolha do prognóstico das partidas de futebol. Para tanto, usaremos dados estatísticos dos clubes das últimas partidas disputadas.

2. Futebol

A organização do futebol coube aos ingleses, porém a sua origem data de muito antes. [1] Há relatos de vários jogos com características que

lembram o futebol; Kemari na China, Epyskiros na Grécia, Harpastum no império Romano, Soule na França e Calcio na Itália são exemplos.

No Brasil a entidade máxima do futebol é a CBF, fundada em 1919 é a organizadora do campeonato Brasileiro que será objeto de estudo deste trabalho.

3. O Modelo Probabilístico

Segundo Barbeta [2], muitas vezes, as condições do experimento não permitem deduzir o resultado, mas somente a chance (ou a probabilidade) dos possíveis resultados. Há diversos fatores que podem interferir no resultado de uma partida de futebol, como por exemplo: fenômenos da natureza (chuva, direção do vento, temperatura), lesão ou expulsão de um jogador durante a partida, dentre muitos outros, sendo praticamente impossível prever o resultado exato de uma partida. Entretanto, com os dados estatísticos que podemos coletar a respeito de jogos passados das equipes, é possível determinar a chance de cada resultado acontecer e com isso, termos um indicativo do resultado provável para uma partida futura.

3.1. Cálculo das Probabilidades

Para calcularmos o provável resultado de uma partida, precisamos ter um indicativo da probabilidade do time mandante fazer X gols em uma partida e do time visitante fazer Y gols em uma partida. Segundo [3], a classe Poisson bivariada de Holgate é a mais indicada para a modelagem de resultados de partidas de futebol, pois atende aos três critérios seguintes:

- As distribuições normais devem ser Poisson – No caso, o número de gols de cada equipe, pode ser considerado conforme cita [2], “situações em que se avalia o número de ocorrências de um tipo de evento por unidade de tempo”.
- A distribuição conjunta deve possuir suporte pleno, ao menos perto da origem.
- A distribuição conjunta e as marginais devem ser infinitamente divisíveis. – No caso, uma partida

de futebol pode ser dita como o resultado de um jogo de 90 minutos ou como o somatório de noventa jogos de 1 minuto cada.

Então, podemos usar a distribuição de Poisson que é dada pelas fórmulas abaixo:

$$P(X = x) = \frac{e^{-\lambda_x} (\lambda_x)^x}{x!}, \lambda_x > 0$$

Onde, para o problema proposto sobre jogos de futebol, X é o número de gols de um time numa partida e λ_x é a média de gols do time por partida.

Conforme teorema mencionado por [4], “toda distribuição bivariada de Poisson infinitamente divisível possui correlação não-negativa”. Esse teorema apóia à noção de que, quaisquer que sejam duas equipes M e V, podemos ter apenas uma das seguintes situações:

1. M é melhor do que V – Neste caso, teremos a vitória do time mandante e a probabilidade de sua vitória pode ser calculada por:

$$P(M) = \sum_{v>m} P(M = m, V = v)$$

2. M e V são equivalentes – Neste caso, ocorrerá um empate cuja probabilidade pode ser calculada por:

$$P(E) = \sum_{m=v} P(M = m, V = v)$$

3. V é melhor do que M – Neste caso, teremos a vitória do time visitante com probabilidade dada por:

$$P(V) = \sum_{m<v} P(M = m, V = v)$$

Podemos dizer ainda que:

$$P(M) + P(E) + P(V) = 1$$

3.2. Considerações para o Cálculo das Probabilidades

No presente trabalho, admitiremos que haja independência entre as quantidades de gols marcados pela equipe mandante e visitante em cada partida. Desta forma podemos admitir que:

$$P(M = m, V = v) = P(M = m) \times P(V = v)$$

3.3. Estimação de Parâmetros

Segundo [5], “no futebol, o objetivo a ser atingido é sempre o gol. Assim, durante uma partida disputada entre duas equipes, a defesa e o ataque são constantes”.

[3] Propõe dentre outros, métodos implícitos para a determinação dos parâmetros. Estes métodos são motivados pelo conceito de probabilidade subjetiva citado por [6] no qual a probabilidade é uma medida de incerteza individual que todo o ser humano está capacitado a manifestar na avaliação de situações de incerteza.

Poucas coisas são tão incertas quanto o futebol, como diz a célebre frase atribuída a Vicente Matheus, “o jogo só termina quando acaba”, cuja veracidade pode ser comprovada pelas inúmeras situações nas quais o resultado de uma partida mudou quando muitos torcedores já haviam deixado o estádio rumo a suas casas.

Partindo deste pressuposto, podemos propor uma estimativa de λ_m como uma média entre a quantidade de gols que o time mandante faz quando joga em casa e a quantidade de gols que o time visitante sofre quando joga fora de casa, da mesma forma estimamos λ_v como uma média entre a quantidade de gols que o time visitante faz quando joga fora de casa e a quantidade de gols que o time mandante sofre quando joga em casa.

Podemos então calcular os parâmetros da seguinte forma:

$$\hat{\lambda}_x = \frac{M_{Favor} + V_{Contra}}{2} \text{ e } \hat{\lambda}_y = \frac{V_{Favor} + M_{Contra}}{2}$$

Considerando o fato de que em mais de 95% das partidas, nenhum dos times faz mais de 4 gols e que uma das loterias da Caixa considerar o número de 4 ou mais gols como uma única opção de aposta, utilizaremos este valor como limite no cálculo das probabilidades.

4. O Banco de Dados

Segundo [7], para a modelagem de novos sistemas, para os quais não existem descrições de dados, deve-se usar uma estratégia de modelagem partindo de pessoas. Neste caso, podem ser usadas duas estratégias, a descendente (“top-down”) ou a “inside-out”.

Para a modelagem do banco de dados, optaremos pela estratégia “inside-out”, na qual, o processo inicia com a identificação de uma entidade importante no modelo e que supostamente está relacionada a várias outras entidades. A partir de então, procura-se atributos, entidades relacionadas, generalizações e especializações da entidade foco até ter-se o modelo completo.

No nosso banco de dados, a entidade chave identificada é a Partida, a partir da qual construímos o modelo completo apresentado na Figura 1.

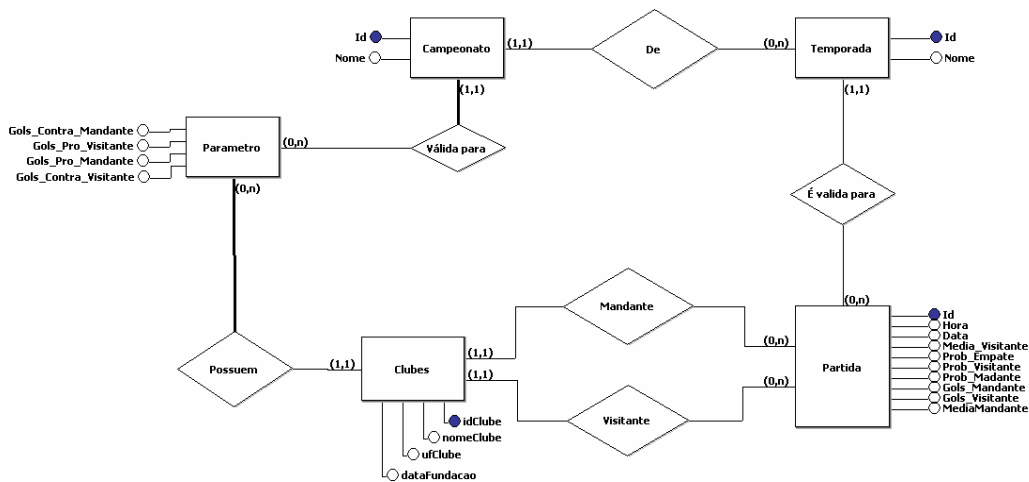


Figura 1 - Modelo ER completo

4.1. Escolha do SGBD

Conforme [8], Um Sistema de Gerenciamento de Banco de Dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados.

Para o desenvolvimento do banco de dados deste trabalho, escolhemos o SGBD MySQL por possuir as seguintes características:

- Portabilidade, funciona em vários sistemas operacionais, tais como: Windows, Linux, FreeBSD, Solaris, Mac OS X, etc.
- Compatibilidade, o MySQL contém drivers ODBC, JDBC e .NET além de módulos de interface para linguagens de programação como Delphi, Java, C/C++, PHP, Ruby entre outras.
- Desempenho
- Estabilidade
- Facilidade de uso
- Por ser um software livre e de código aberto.

5. Análise e Projeto do Sistema

Durante a análise do sistema, o objetivo é definir os requisitos e descobrir os componentes do domínio do problema, descrevendo-os, já na fase de projeto. O objetivo é definir como os componentes colaboram para atender os requisitos.

5.1. Requisitos Funcionais

- Cadastra – O sistema deve permitir ao administrador o cadastro de clubes, campeonatos, temporadas e partidas.

- Exclui – O sistema deve permitir a exclusão de partidas. Quanto a campeonatos, temporadas e clubes, não devem ser excluídos para não afetar os registros históricos do sistema.

- Edita – O sistema deve permitir a edição dos dados de clubes, temporadas, campeonatos e partidas de modo a manter a informação correta em virtude do registro de valores incorretos.

- Visualiza – O sistema deve permitir, em diferentes níveis, a visualização dos dados. Para o administrador, deve ser possível visualizar os dados de campeonato, temporada, clubes e partidas. Ainda no item partidas, deve ser possível visualizar no detalhe, onde será mostrado os valores das probabilidades para cada placar possível para a referida partida.

- Atualização automática dos parâmetros para o cálculo das probabilidades – Deve atualizar os parâmetros dos clubes que disputaram uma partida quando a mesma for editada de modo a ser informado o placar, assim passa a ser utilizado o novo parâmetro no cálculo das probabilidades para jogo futuro;

- Calcular probabilidades – Deve, com base nos parâmetros para os clubes que disputarão uma partida, calcular a probabilidade de ocorrer vitória do mandante, empate ou vitória do visitante; Deve ainda, calcular probabilidades da ocorrência dos possíveis resultados com base nos parâmetros;

- Deve criar de forma automática, os parâmetros para novos clubes, caso não exista;

5.2. Requisitos Não Funcionais

- Deve ser desenvolvido em ambiente web de modo a facilitar o acesso dos usuários ao sistema;

- Deve ser compatível com os browsers mais utilizados do mercado;
- Deve ter todas as informações em Português do Brasil;
- Deve prover segurança para os dados – O acesso a cadastros edições e exclusões só poderá ser feito por pessoa autorizada. Usuários terão acesso apenas a consultas diversas.

5.3. Diagrama de Casos de Uso

Com base nos requisitos do sistema, é possível construirmos o diagrama de casos de uso, onde são ilustrados os requisitos funcionais e sua associação com os usuários do sistema (atores). Tal diagrama é ilustrado na Figura 2.

5.4. Projeto do Sistema

Para o desenvolvimento do sistema, optamos pela linguagem Java na sua versão 5 [9] pois apresenta um conjunto de frameworks que facilitam a implementação de sistemas web, como por exemplo, Java Server Faces [10]. Como IDE de desenvolvimento optamos pelo NetBeans [11] pela familiaridade com o uso da ferramenta e por ser da mesma empresa desenvolvedora da linguagem de programação escolhida.

5.5. Diagrama de Classes do Sistema

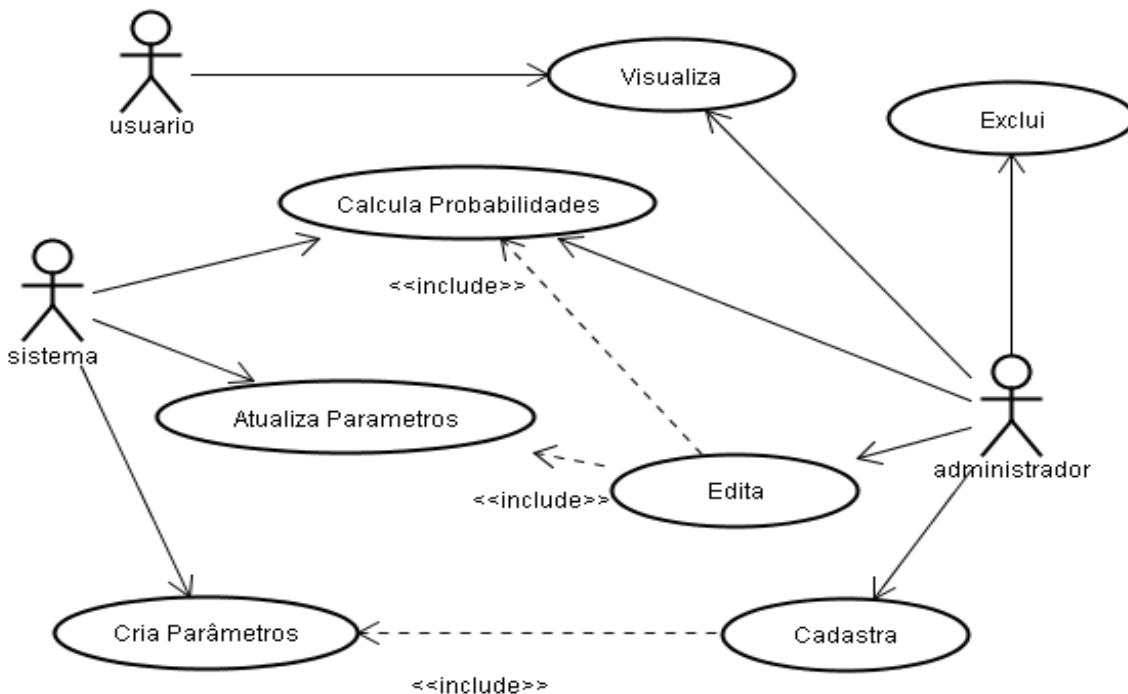


Figura 2 - Diagrama de Casos de Uso

Baseado nas entidades criadas no banco de dados é possível fazermos o Diagrama de Classes para a representação do Model do sistema. A Figura 3 mostra as entidades, seus atributos e seus métodos.

6. Chutegol

Chutegol é um acrônimo para “Chance de Um Time Em GOLs”, além de fazer uma alusão ao fato de tentar descobrir o placar de uma partida de futebol, ou seja, chutar quantos gols cada time irá fazer numa partida. Por estes motivos, foi o nome escolhido para o sistema.

6.1. Projeto Web

O primeiro passo no desenvolvimento do sistema foi a criação de um novo projeto web no Netbeans, denominado “chutegol”.

Com o projeto criado, o próximo passo foi utilizar a ferramenta de desenvolvimento para construir as classes das entidades mostradas no diagrama de classes através do banco de dados criado no Capítulo 4.

O Netbeans possui uma opção chamada Classes de Entidade do Banco de Dados que executa esta tarefa, bastando para isso fazermos sua conexão com a base de dados

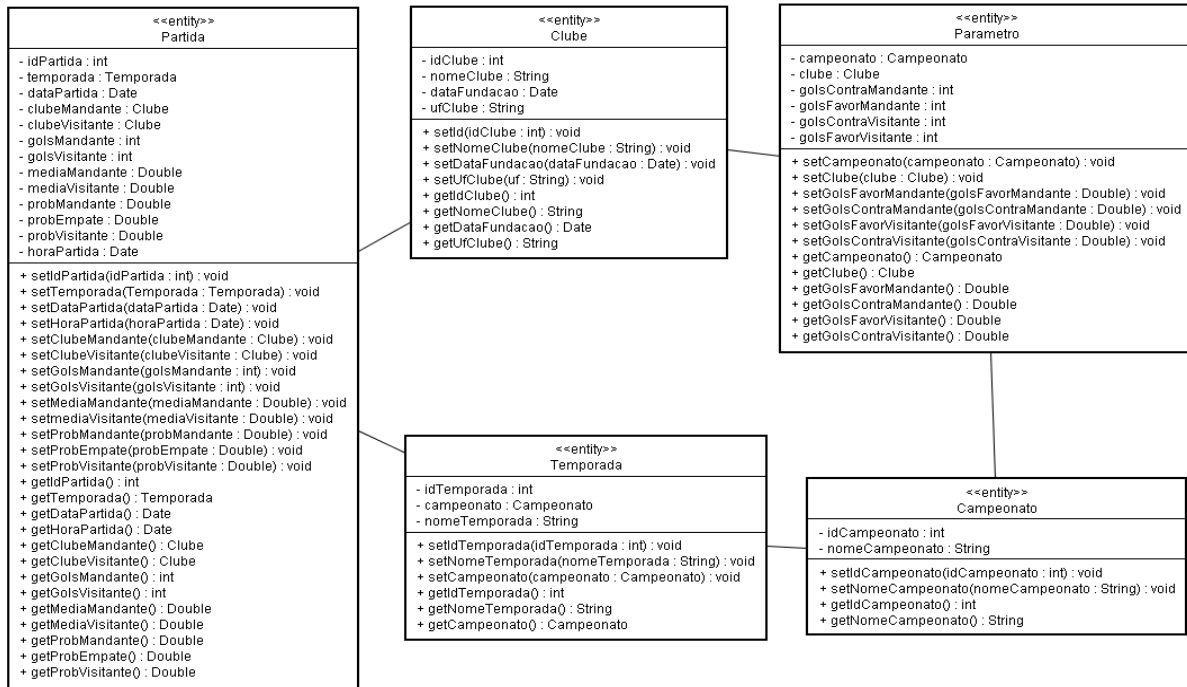


Figura 3 - Diagrama de Classes

A última função automatizada do Netbeans utilizada no desenvolvimento foi a criação das páginas em Java Server Faces. Utilizamos a opção do Netbeans “Páginas JSF de Classes de Entidades” para criar as páginas Detail.jsp, Edit.jsp, List.jsp e New.jsp.

Neste momento, o Netbeans também criou de forma automática, os Controllers para cada entidade com todos os métodos necessários para que o sistema ficasse funcional a ponto de podermos fazer as tarefas de CRUD (Create, Retrieve, Update e Destroy) que são as quatro operações básicas em banco de dados relacionais.

O próximo passo foi a implementação da lógica do negócio, ou seja, a implementação dos requisitos do sistema para que o mesmo possa, como citado anteriormente, fazer a coisa certa e fazer certo a coisa.

Os requisitos que precisam ser implementados para a lógica do sistema são:

- Criar parâmetros para o clube no campeonato (caso não exista);
 - Atualizar os parâmetros após a inserção do resultado da partida;
 - Calcular a probabilidade de vitória do mandante, empate e vitória do visitante;
 - Calcular a probabilidade dos possíveis placares para a partida;

7. Resultados

Para efeito de análise dos resultados obtidos com o uso do ChuteGol, iremos descartar os dados das primeiras quatro rodadas do Campeonato Brasileiro de 2007, visto que, como nenhum time tinha parâmetros iniciais, essas quatro rodadas servirão para inicializar os parâmetros dos clubes, no caso, duas partidas como mandante e duas como visitante.

Em cada rodada, analisaremos as partidas disputadas verificando a ocorrência da probabilidade maior, probabilidade média ou probabilidade menor. No caso de haver duas probabilidades iguais para a referida partida, a ocorrência será da probabilidade maior ou da probabilidade menor.

Atribuiremos uma pontuação para cada tipo de probabilidade (Maior, Média e Menor), sendo a pontuação igual ao percentual da probabilidade. Assim, teremos um valor mínimo (todos os jogos com resultado de menor probabilidade) e máximo (todos os jogos com resultado de maior probabilidade) de pontos que uma rodada pode atingir. Com isso, podemos normalizar o resultado da rodada e tentar estimar um intervalo sobre a pontuação que tem uma chance maior de ocorrer.

Foram analisados 229 jogos, desconsiderando os primeiros 40 que foram usados para inicializar os parâmetros dos clubes. Em 116 jogos ocorreu o resultado mais provável (50,65%), em 55 o

resultado foi o de probabilidade média (24,02%) e em 58, o de menor probabilidade (25,33%).

Se considerarmos que a chance de acertarmos um resultado ao acaso é de 33,3%, pois temos três possibilidades, o Chutegol conseguiu um ganho de aproximadamente 52%.

Analisando o placar mais provável de ocorrer, o Chutegol conseguiu um acerto de 21 resultados (10,46%). Se considerarmos probabilidades iguais para cada um dos 25 possíveis resultados analisados, teríamos uma chance de 4% de acertar o resultado ao acaso. Com isso, tivemos um ganho de 161,5%.

A análise da pontuação das rodadas nos mostra valores desde 0,33 até 0,91, o valor médio obtido nesta análise foi de 0,62 com um desvio padrão de 0,15, se utilizamos como parâmetro, um desvio padrão em torno da média, podemos dizer que aproximadamente 68% dos casos teremos uma rodada apresentando pontuação entre 0,47 e 0,77.

Estes dados podem ser utilizados para filtro de apostas num jogo da loteria, pelo qual o apostador pode excluir as combinações de resultados que estejam fora do intervalo escolhido.

8. Considerações Finais e Trabalhos Futuros

Com este trabalho, foi possível mostrar a utilidade de um sistema de auxílio ao prognóstico de jogos de futebol. Vale ressaltar que não é possível prever com exatidão o resultado de todas as partidas. Como foi citado no início do trabalho, existem diversos fatores, intra e extra campo que fazem com que uma partida de futebol tenha um clima envolvente e fascinante para todas as torcidas, o que transforma um jogo num espetáculo.

Um dos desafios deste trabalho foi a estimação dos parâmetros. A solução adotada foi arbitramos um método de cálculo para estimarmos os parâmetros. Os resultados, embora satisfatórios, deixam um questionamento quanto a existência de métodos mais eficazes se considerarmos que haja dependência na quantidade de gols de cada equipe.

A população da base de dados com resultados passados é um trabalho que pode trazer um enriquecimento a análise dos jogos, tornando possível verificarmos confrontos históricos entre duas equipes e usando estas informações na escolha do prognóstico mais adequado para a disputa.

Durante o desenvolvimento do projeto, foi possível colocar em prática grande parte do conhecimento adquirido no decorrer do curso, disciplinas como Banco de Dados, Engenharia de

software, Análise e Projeto de Sistemas Orientados a Objetos e Programação Web, foram de grande valia na conclusão deste trabalho.

Por fim, convido todos a acessarem o projeto no endereço <http://java.inf.ufsc.br/chutegol>, onde poderão fazer uso do mesmo para definirem seus prognósticos para as próximas partidas.

9. Bibliografia

- [1] DUARTE, O., *Futebol, histórias e regras*, Makron, São Paulo, 1997.
- [2] BARBETTA, P.A., *Estatística para cursos de engenharia e informática*, Atlas, São Paulo, 2004.
- [3] ARRUDA, M.L., (2000), Poisson, Bayes, Futebol e DeFinetti, São Paulo, Dissertação de Mestrado IME/USP.
- [4] GRIFFITHS, R.C., MILNE, R.K & WOOD, R. (1979), Aspects of Correlation in Bivariate Poisson Distributions and Processes, *Australian Journal of Statistics* 21.
- [5] WITTER, J.S., (1990), O que é futebol, São Paulo, Ed. Brasiliense.
- [6] BARNETT, V. (1982), *Comparative Statistical Inference*, New York: Wiley.
- [7] HEUSER, C.A., (2004), Projeto de banco de dados, Porto Alegre, Ed Sagra Luzzatto.
- [8] ELMASRI, R., NAVATHE, S.B., (2005), *Sistemas de banco de dados*, São Paulo, Addison Wesley.
- [9] JAVA – Linguagem de Programação, Disponível em <http://java.sun.com/>, acesso em novembro de 2006.
- [10] JSF - Java Server Faces, Disponível em <http://java.sun.com/javaee/javaserverfaces/>, acesso em junho de 2007.
- [11] IDE Netbeans, disponível em <http://www.netbeans.org/>, acesso em junho de 2007.

11. Apêndice B – Código Fonte

11.1. index.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Chutegol - CHance de Um Time Em GOLs</title>
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuUser.jspf" %>
    <div id="corpo">
      <h2>O que é o ChuteGol?</h2>
      <p>Chutegol (CHance de Um Time Em GOLs) é um sistema para auxiliar
o prognóstico de jogos de futebol.
      <br>Ele é composto pelos seguintes itens:</p>
      <ul>
        <li>Modelo probabilístico</li>
        <li>Base de dados</li>
        <li>Web site</li>
      </ul>
      <h3>Modelo Probabilístico</h3>
      <p>É um modelo matemático desenvolvido com o intuito de calcular a
probabilidade de vitória do time mandante, empate e vitória do time visitante
numa partida de futebol.
      <br>O modelo permite calcular também, a probabilidade de
ocorrência de placares com as combinações possíveis de até 4 gols para cada
time.</p>
      <h3>Base de Dados</h3>
      <p>Permite o armazenamento dos dados dos jogos para consulta
futura pelos usuários do sistema.</p>
      <h3>Web site</h3>
      <p>É esse endereço web que você está acessando, desenvolvido em <a
href="http://java.sun.com">JAVA</a> e usando tecnologia
      <a href="http://java.sun.com/javaee/javaserverfaces/">Java Server
Faces</a>.</p>
    </div>
    <%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
  </body>
</html>
```

11.2. Admin – index.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Chutegol - CHance de Um Time Em GOLs</title>
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
  </head>
  <body>
```

```

<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
<div id="corpo">
  <h2>O que é o ChuteGol?</h2>
  <p>Chutegol (CHance de Um Time Em GOLs) é um sistema para auxiliar
    o prognóstico de jogos de futebol.
  <br>Ele é composto pelos seguintes itens:</p>
  <ul>
    <li>Modelo probabilístico</li>
    <li>Base de dados</li>
    <li>Web site</li>
  </ul>
  <h3>Modelo Probabilístico</h3>
  <p>É um modelo matemático desenvolvido com o intuito de calcular a
    probabilidade de vitória do time mandante, empate e vitória do time visitante
    numa partida de futebol.
  <br>O modelo permite calcular também, a probabilidade de
    ocorrência de placares com as combinações possíveis de até 4 gols para cada
    time.</p>
  <h3>Base de Dados</h3>
  <p>Permite o armazenamento dos dados dos jogos para consulta
    futura
    pelos usuários do sistema.</p>
  <h3>Web site</h3>
  <p>É esse endereço web que você está acessando, desenvolvido em
    <a href="http://java.sun.com">JAVA</a> e usando tecnologia
    <a href="http://java.sun.com/javaee/javaserverfaces/">
      Java Server Faces
    </a>.</p>
</div>

<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.3. Admin – Campeonato – New.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Novo Campeonato</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Novo Campeonato</h1>
        <h:form>
          <h:panelGrid columns="2" styleClass="tabelaCadastro">
            <h:outputText value="Nome do Campeonato:"/>
            <h:inputText
              id="nomeCampeonato"
              value="#{campeonatoAdmin.campeonato.nomeCampeonato}"
              title="Nome do Campeonato" />
          </h:panelGrid>
          <h:commandButton action="#{campeonatoAdmin.create}"
            value="Criar"/>
        </h:form>
      </f:view>
    </div>
  </body>
</html>

```

```

        </f:view>
    </div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.4. Admin – Campeonato – List.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
<title>Listar Campeonato</title>
</head>
<body>
<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
<div id="corpo">
<f:view>
<h:messages errorStyle="color: red" infoStyle="color: green"
    layout="table"/>
<h1>Lista de Campeonatos</h1>
<h:form>
<h:outputText
    value="Item #{campeonatoAdmin.firstItem + 1}..
        #{campeonatoAdmin.lastItem} de
        #{campeonatoAdmin.itemCount}"
    styleClass="item"/>&nbsp;
<h:commandLink
    action="#{campeonatoAdmin.prev}"
    value="Anterior #{campeonatoAdmin.batchSize}"
    rendered="#{campeonatoAdmin.firstItem >=
        campeonatoAdmin.batchSize}"
    styleClass="item"/>&nbsp;
<h:commandLink
    action="#{campeonatoAdmin.next}"
    value="Próximo #{campeonatoAdmin.batchSize}"
    rendered="#{campeonatoAdmin.lastItem +
        campeonatoAdmin.batchSize <=
        campeonatoAdmin.itemCount}"
    styleClass="item"/>&nbsp;
<h:commandLink
    action="#{campeonatoAdmin.next}"
    value="Restando #{campeonatoAdmin.itemCount -
        campeonatoAdmin.lastItem}"
    rendered="#{campeonatoAdmin.lastItem <
        campeonatoAdmin.itemCount &&
        campeonatoAdmin.lastItem +
        campeonatoAdmin.batchSize >
        campeonatoAdmin.itemCount}"
    styleClass="item"/>
<h:dataTable value=&apos;#{campeonatoAdmin.campeonatos}&apos;
    var=&apos;item&apos;
    border="1" cellpadding="2" cellspacing="0"
    styleClass="tabela" headerClass="tabela_header"
    rowClasses="tabela_row1,tabela_row2"
    columnClasses="t_campeonato,t_links" >
<h:column>
<f:facet name="header">
<h:outputText value="Nome"/>
</f:facet>

```

```

        <h:commandLink action="#{campeonatoAdmin.detailSetup}"
                    value="#{item.nomeCampeonato}"/>
    </h:column>
    <h:column>
        <h:commandLink value="Apagar"
                    action="#{campeonatoAdmin.destroy}">
            <f:param name="idCampeonato"
                    value="#{item.idCampeonato}"/>
        </h:commandLink>
        <h:outputText value=" " />
        <h:commandLink value="Editar"
                    action="#{campeonatoAdmin.editSetup}">
            <f:param name="idCampeonato"
                    value="#{item.idCampeonato}"/>
        </h:commandLink>
    </h:column>
</h:dataTable>
<h:commandLink action="#{campeonatoAdmin.createSetup}"
                value="Novo Campeonato"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.5. Admin – Campeonato – Edit.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
<title>Editar Campeonato</title>
</head>
<body>
<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
<div id="corpo">
<f:view>
<h:messages errorStyle="color: red" infoStyle="color: green"
            layout="table"/>
<h1>Editar campeonato</h1>
<h:form>
<h:inputHidden value="#{campeonatoAdmin.campeonato}"
                immediate="true"/>
<h:panelGrid columns="2" styleClass="tabelaCadastro">
<h:outputText value="Id:" />
<h:outputText
                value="#{campeonatoAdmin.campeonato.idCampeonato}"
                title="IdCampeonato" />
<h:outputText value="Nome do Campeonato:" />
<h:inputText
                id="nomeCampeonato"
                value="#{campeonatoAdmin.campeonato.nomeCampeonato}"
                title="Nome do Campeonato" />
</h:panelGrid>
<h:commandButton action="#{campeonatoAdmin.edit}"
                 value="Salvar"/>
</h:form>

```

```

        </f:view>
    </div>
    <%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.6. Admin – Campeonato – Detail.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Detalhe de Campeonato</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red"
          infoStyle="color: green" layout="table"/>
        <h1>Detalhe de Campeonato</h1>
        <h:form>
          <h:panelGrid columns="2" styleClass="tabelaCadastro">
            <h:outputText value="Nome do Campeonato:"/>
            <h:outputText
              value="#{campeonatoAdmin.campeonato.nomeCampeonato}"
              title="NomeCampeonato" />
          </h:panelGrid>
          <h:commandButton action="campeonatoAdmin_edit" value="Editar" />
        </h:form>
      </f:view>
    </div>
    <%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
  </body>
</html>

```

11.7. Admin – Clube – New.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Novo Clube</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Novo clube</h1>
        <h:form enctype="multipart/form-data">
          <h:panelGrid columns="2" styleClass="tabelaCadastro">
            <h:outputText value="Nome do Clube:"/>
            <h:inputText id="nomeClube"
              value="#{clubeAdmin.clube.nomeClube}"
              title="NomeClube" />
            <h:outputText value="UF do Clube:"/>
            <h:inputText
              id="ufClube"
              value="#{clubeAdmin.clube.ufClube}"
              title="UfClube" />
            <h:outputText value="Data de Fundação (dd/MM/yyyy):"/>
            <h:inputText id="dataFundacao"
              value="#{clubeAdmin.clube.dataFundacao}"
              title="DataFundacao" />
            <f:convertDateTime type="DATE" pattern="dd/MM/yyyy" />
          </h:inputText>
          </h:panelGrid>
          <h:commandButton action="#{clubeAdmin.create}" value="Criar"/>
        </h:form>
      </f:view>
    </div>
    <%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
  </body>
</html>
```

11.8. Admin – Clube – Edit.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Editar Clube</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
```



```

<f:view>
  <h:messages errorStyle="color: red" infoStyle="color: green"
    layout="table"/>
  <h1>Editar clube</h1>
  <h:form enctype="multipart/form-data">
    <h:inputHidden value="#{clubeAdmin.clube}" immediate="true"/>
    <h:panelGrid columns="2" styleClass="tabelaCadastro">
      <h:outputText value="Id:"/>
      <h:outputText value="#{clubeAdmin.clube.idClube}"
        title="IdClube" />
      <h:outputText value="Nome do Clube:"/>
      <h:inputText id="nomeClube"
        value="#{clubeAdmin.clube.nomeClube}"
        title="NomeClube" />
      <h:outputText value="UF Clube:"/>
      <h:inputText id="ufClube"
        value="#{clubeAdmin.clube.ufClube}"
        title="UfClube" />
      <h:outputText value="Data de Fundacao (dd/MM/yyyy):"/>
      <h:inputText id="dataFundacao"
        value="#{clubeAdmin.clube.dataFundacao}"
        title="DataFundacao" >
        <f:convertDateTime type="DATE" pattern="dd/MM/yyyy" />
      </h:inputText>
    </h:panelGrid>
    <h:commandButton action="#{clubeAdmin.edit}" value="Salvar"/>
  </h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.9. Admin – Clube – List.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Listar Clube</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Listando Clubes</h1>
        <h:form>
          <h:outputText value="Item #{clubeAdmin.firstItem + 1}..
            #{clubeAdmin.lastItem} de
            #{clubeAdmin.itemCount}"
            styleClass="item"/>&nbsp;
          <h:commandLink action="#{clubeAdmin.prev}"
            value="Anterior #{clubeAdmin.batchSize}"
            rendered="#{clubeAdmin.firstItem >=
              clubeAdmin.batchSize}"

```

```

        styleClass="item"/>&nbsp;
<h:commandLink action="#{clubeAdmin.next}"
value="Próximo #{clubeAdmin.batchSize}"
rendered="#{clubeAdmin.lastItem +
clubeAdmin.batchSize <=
clubeAdmin.itemCount}"
styleClass="item"/>&nbsp;
<h:commandLink action="#{clubeAdmin.next}"
value="Restando #{clubeAdmin.itemCount -
clubeAdmin.lastItem}"
rendered="#{clubeAdmin.lastItem <
clubeAdmin.itemCount &&
clubeAdmin.lastItem +
clubeAdmin.batchSize >
clubeAdmin.itemCount}"
styleClass="item"/>
<h:dataTable value="&apos;#{clubeAdmin.clubes}&apos;
var=&apos;item&apos;
border="1" cellpadding="2" cellspacing="0"
styleClass="tabela" headerClass="tabela_header"
rowClasses="tabela_row1,tabela_row2"
columnClasses="t_clube,t_uf,t_dataFundacao,t_links">
<h:column>
<f:facet name="header">
<h:outputText value="Nome do Clube"/>
</f:facet>
<h:commandLink action="#{clubeAdmin.detailSetup}"
value="#{item.nomeClube}"/>
</h:column>
<h:column>
<f:facet name="header">
<h:outputText value="UF Clube"/>
</f:facet>
<h:outputText value="#{item.ufClube}"/>
</h:column>
<h:column>
<f:facet name="header">
<h:outputText value="Data de Fundacao"/>
</f:facet>
<h:outputText value="#{item.dataFundacao}">
<f:convertDateTime type="DATE"
pattern="dd/MM/yyyy" />
</h:outputText>
</h:column>
<h:column>
<h:commandLink value="Destroir"
action="#{clubeAdmin.destroy}">
<f:param name="idClube" value="#{item.idClube}"/>
</h:commandLink>
<h:outputText value=" "/>
<h:commandLink value="Editar"
action="#{clubeAdmin.editSetup}">
<f:param name="idClube" value="#{item.idClube}"/>
</h:commandLink>
</h:column>
</h:dataTable>
<h:commandLink action="#{clubeAdmin.createSetup}"
value="Novo Clube"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.10. Admin – Clube – Detail.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Detalhe de Clube</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Detalhe de clube</h1>
        <h:form enctype="multipart/form-data">
          <h:panelGrid columns="2">
            <h:outputText value="Id:" />
            <h:outputText value="#{clubeAdmin.clube.idClube}"
              title="IdClube" />
            <h:outputText value="Nome do Clube:" />
            <h:outputText value="#{clubeAdmin.clube.nomeClube}"
              title="NomeClube" />
            <h:outputText value="UF Clube:" />
            <h:outputText value="#{clubeAdmin.clube.ufClube}"
              title="UfClube" />
            <h:outputText value="Data de Fundacao:" />
            <h:outputText value="#{clubeAdmin.clube.dataFundacao}"
              title="DataFundacao" />
          </h:panelGrid>
          <h:commandButton action="clubeAdmin_edit" value="Editar" />
        </h:form>
      </f:view>
    </div>
    <%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
  </body>
</html>
```

11.11. Admin – Temporada – New.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Nova Temporada</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Nova temporada</h1>
      </f:view>
    </div>
  </body>
</html>
```

```

<h:form>
  <h:panelGrid columns="2" styleClass="tabelaCadastro">
    <h:outputText value="Nome da Temporada:"/>
    <h:inputText id="nomeTemporada"
      value="#{temporadaAdmin.temporada.nomeTemporada}"
      title="Nome da Temporada" />
    <h:outputText value="Campeonato:"
      rendered="#{temporadaAdmin.temporada.campeonatoId == null}"/>
    <h:selectOneMenu id="campeonatoId"
      value="#{temporadaAdmin.temporada.campeonatoId}"
      title="Campeonato"
      rendered="#{temporadaAdmin.temporada.campeonatoId == null}"/>
      <f:selectItems value="#{temporadaAdmin.campeonatoIds}"/>
    </h:selectOneMenu>
  </h:panelGrid>
  <h:commandButton action="#{temporadaAdmin.create}"
    value="Criar"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.12. Admin – Temporada – Edit.jsp

```

<%@ page contentType="text/html" %>
<%@ page pageEncoding="UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Editar Temporada</title>
  </head>
  <body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>

    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Editar temporada</h1>
        <h:form>
          <h:inputHidden value="#{temporadaAdmin.temporada}"
            immediate="true"/>
          <h:panelGrid columns="2" styleClass="tabelaCadastro">
            <h:outputText value="Id:"/>
            <h:outputText
              value="#{temporadaAdmin.temporada.idTemporada}"
              title="IdTemporada" />
            <h:outputText value="Nome da Temporada:"/>
            <h:inputText
              id="nomeTemporada"
              value="#{temporadaAdmin.temporada.nomeTemporada}"
              title="Nome da Temporada" />
            <h:outputText value="Campeonato:"/>
            <h:selectOneMenu
              id="campeonatoId"
              value="#{temporadaAdmin.temporada.campeonatoId}"
              title="Campeonato">

```

```

        <f:selectItems value="#{temporadaAdmin.campeonatoIds}"/>
    </h:selectOneMenu>
</h:panelGrid>
<h:commandButton action="#{temporadaAdmin.edit}"
    value="Salvar"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.13. Admin – Temporada – List.jsp

```

<%@ page contentType="text/html" %>
<%@ page pageEncoding="UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Listar Temporada</title>
</head>
<body>
    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
    <div id="corpo">
        <f:view>
            <h:messages errorStyle="color: red" infoStyle="color: green"
                layout="table"/>
            <h1>Listando Temporadas</h1>
            <h:form>
                <h:outputText value="Item #{temporadaAdmin.firstItem + 1}..
                    #{temporadaAdmin.lastItem} de
                    #{temporadaAdmin.itemCount}"
                    styleClass="item"/>&nbsp;
                <h:commandLink action="#{temporadaAdmin.prev}"
                    value="Anterior #{temporadaAdmin.batchSize}"
                    rendered="#{temporadaAdmin.firstItem >=
                        temporadaAdmin.batchSize}"
                    styleClass="item"/>&nbsp;
                <h:commandLink action="#{temporadaAdmin.next}"
                    value="Próximo #{temporadaAdmin.batchSize}"
                    rendered="#{temporadaAdmin.lastItem +
                        temporadaAdmin.batchSize <=
                        temporadaAdmin.itemCount}"
                    styleClass="item"/>&nbsp;
                <h:commandLink action="#{temporadaAdmin.next}"
                    value="Restando #{temporadaAdmin.itemCount -
                        temporadaAdmin.lastItem}"
                    rendered="#{temporadaAdmin.lastItem <
                        temporadaAdmin.itemCount &&
                        temporadaAdmin.lastItem +
                        temporadaAdmin.batchSize >
                        temporadaAdmin.itemCount}"
                    styleClass="item"/>
                <h:dataTable value='&apos;#{temporadaAdmin.temporadas}&apos;
                    var='&apos;item&apos; border="1" cellpadding="2"
                    cellspacing="0" styleClass="tabela"
                    headerClass="tabela_header"

```

```

        rowClasses="tabela_row1,tabela_row2"
        columnClasses="t_temporada,t_campeonato,t_links">
</h:column>
<f:facet name="header">
    <h:outputText value="Nome"/>
</f:facet>
<h:commandLink action="#{temporadaAdmin.detailSetup}"
    value="#{item.nomeTemporada}"/>
</h:column>
<h:column>
<f:facet name="header">
    <h:outputText value="Campeonato"/>
</f:facet>
<h:outputText
    value="#{item.campeonatoId.nomeCampeonato}"/>
</h:column>
<h:column>
<h:commandLink value="Apagar"
    action="#{temporadaAdmin.destroy}">
    <f:param name="idTemporada"
        value="#{item.idTemporada}"/>
</h:commandLink>
<h:outputText value=" "/>
<h:commandLink value="Editar"
    action="#{temporadaAdmin.editSetup}">
    <f:param name="idTemporada"
        value="#{item.idTemporada}"/>
</h:commandLink>
</h:column>
</h:dataTable>
<h:commandLink action="#{temporadaAdmin.createSetup}"
    value="Nova Temporada"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.14. Admin – Temporada – Detail.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Detalhe de Temporada</title>
</head>
<body>
<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
<div id="corpo">
<f:view>
    <h:messages errorStyle="color: red" infoStyle="color: green"
        layout="table"/>
    <h1>Detalhe de temporada</h1>
    <h:form>
        <h:panelGrid columns="2" styleClass="tabelaCadastro">
            <h:outputText value="Id:" />
            <h:outputText

```

```

        value="#{temporadaAdmin.temporada.idTemporada}"
        title="IdTemporada" />
</h:outputText>
<h:outputText value="Nome da Temporada:" />
</h:outputText>
        value="#{temporadaAdmin.temporada.nomeTemporada}"
        title="NomeTemporada" />
<h:outputText value="Campeonato:" />
</h:outputText>
value="#{temporadaAdmin.temporada.campeonatoId.nomeCampeonato}"
        title="Campeonato" />
</h:panelGrid>
<h:commandButton action="temporadaAdmin_edit" value="Editar" />
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.15. Admin – Partida – New.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
<title>Nova Partida</title>
</head>
<body>
<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
<div id="corpo">
<f:view>
<h:messages errorStyle="color: red" infoStyle="color: green"
        layout="table"/>
<h1>Nova partida</h1>
<h:form>
<h:panelGrid columns="2" styleClass="tabelaCadastro">
<h:outputText value="Clube Mandante:"
        rendered="#{partidaAdmin.partida.clubeMandanteId == null}"/>
<h:selectOneMenu id="clubeMandanteId"
        value="#{partidaAdmin.partida.clubeMandanteId}"
        title="Clube Mandante"
        rendered="#{partidaAdmin.partida.clubeMandanteId == null}">
<f:selectItems
        value="#{partidaAdmin.clubeMandanteIds}"/>
</h:selectOneMenu>
<h:outputText value="Clube Visitante:"
        rendered="#{partidaAdmin.partida.clubeVisitanteId == null}"/>
<h:selectOneMenu id="clubeVisitanteId"
        value="#{partidaAdmin.partida.clubeVisitanteId}"
        title="Clube Visitante"
        rendered="#{partidaAdmin.partida.clubeVisitanteId == null}">
<f:selectItems
        value="#{partidaAdmin.clubeVisitanteIds}"/>
</h:selectOneMenu>
<h:outputText value="Data (dd/MM/yyyy):"/>
<h:inputText id="dataPartida"
        value="#{partidaAdmin.partida.dataPartida}"

```

```

        title="Data da Partida" >
            <f:convertDateTime type="DATE" pattern="dd/MM/yyyy" />
        </h:inputText>
        <h:outputText value="Hora (HH:mm) :"/>
        <h:inputText id="horaPartida"
            value="#{partidaAdmin.partida.horaPartida}"
            title="Hora da Partida" >
            <f:convertDateTime type="TIME" pattern="HH:mm" />
        </h:inputText>
        <h:outputText value="Gols Mandante :"/>
        <h:inputText id="golsMandante"
            value="#{partidaAdmin.partida.golsMandante}"
            title="GolsMandante" />
        <h:outputText value="Gols Visitante :"/>
        <h:inputText id="golsVisitante"
            value="#{partidaAdmin.partida.golsVisitante}"
            title="GolsVisitante" />
        <h:outputText value="Temporada : "
            rendered="#{partidaAdmin.partida.temporadaId == null}"/>
        <h:selectOneMenu id="temporadaId"
            value="#{partidaAdmin.partida.temporadaId}"
            title="Temporada"
            rendered="#{partidaAdmin.partida.temporadaId == null}"/>
            <f:selectItems value="#{partidaAdmin.temporadaIds}"/>
        </h:selectOneMenu>
    </h:panelGrid>
    <h:commandButton action="#{partidaAdmin.create}" value="Criar"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.16. Admin – Partida – Edit.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
        <title>Editar Partida</title>
    </head>
    <body>
        <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
        <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>
        <div id="corpo">
            <f:view>
                <h:messages errorStyle="color: red" infoStyle="color: green"
                    layout="table"/>
                <h1>Editar partida</h1>
                <h:form>
                    <h:inputHidden value="#{partidaAdmin.partida}" immediate="true"/>
                    <h:panelGrid columns="2" styleClass="tabelaCadastro">
                        <h:outputText value="IdPartida :"/>
                        <h:outputText value="#{partidaAdmin.partida.idPartida}"
                            title="IdPartida" />
                        <h:outputText value="Clube Mandante :"/>
                        <h:selectOneMenu id="clubeMandanteId"

```



```

value="#{partidaAdmin.partida.clubeMandanteId}"
title="ClubeMandanteId">
    <f:selectItems value="#{partidaAdmin.clubeMandanteIds}"/>
</h:selectOneMenu>
<h:outputText value="Clube Visitante:"/>
<h:selectOneMenu id="clubeVisitanteId"
value="#{partidaAdmin.partida.clubeVisitanteId}"
title="ClubeVisitanteId">
    <f:selectItems value="#{partidaAdmin.clubeVisitanteIds}"/>
</h:selectOneMenu>
<h:outputText value="Data (dd/MM/yyyy) :"/>
<h:inputText id="dataPartida"
value="#{partidaAdmin.partida.dataPartida}"
title="DataPartida" >
    <f:convertDateTime type="DATE" pattern="dd/MM/yyyy" />
</h:inputText>
<h:outputText value="Hora (HH:mm) :"/>
<h:inputText id="horaPartida"
value="#{partidaAdmin.partida.horaPartida}"
title="HoraPartida" >
    <f:convertDateTime type="TIME" pattern="HH:mm" />
</h:inputText>
<h:outputText value="Gols Mandante:"/>
<h:inputText id="golsMandante"
value="#{partidaAdmin.partida.golsMandante}"
title="GolsMandante" />
<h:outputText value="Gols Visitante:"/>
<h:inputText id="golsVisitante"
value="#{partidaAdmin.partida.golsVisitante}"
title="GolsVisitante" />
<h:outputText value="Media Mandante:"/>
<h:outputText id="mediaMandante"
value="#{partidaAdmin.partida.mediaMandante}"
title="MediaMandante" >
    <f:convertNumber pattern="0.00"/>
</h:outputText>
<h:outputText value="Media Visitante:"/>
<h:outputText id="mediaVisitante"
value="#{partidaAdmin.partida.mediaVisitante}"
title="MediaVisitante" >
    <f:convertNumber pattern="0.00"/>
</h:outputText>
<h:outputText value="Prob Mandante:"/>
<h:outputText id="probMandante"
value="#{partidaAdmin.partida.probMandante}"
title="ProbMandante" >
    <f:convertNumber pattern="#0%"/>
</h:outputText>
<h:outputText value="Prob Empate:"/>
<h:outputText id="probEmpate"
value="#{partidaAdmin.partida.probEmpate}"
title="ProbEmpate" >
    <f:convertNumber pattern="#0%"/>
</h:outputText>
<h:outputText value="Prob Visitante:"/>
<h:outputText id="probVisitante"
value="#{partidaAdmin.partida.probVisitante}"
title="ProbVisitante" >
    <f:convertNumber pattern="#0%"/>
</h:outputText>
<h:outputText value="Temporada:"/>
<h:selectOneMenu id="temporadaId"
value="#{partidaAdmin.partida.temporadaId}"

```

```

        title="TemporadaId">
            <f:selectItems value="#{partidaAdmin.temporadaIds}"/>
        </h:selectOneMenu>
    </h:panelGrid>
    <h:commandButton action="#{partidaAdmin.edit}" value="Salvar"/>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.17. Admin – Partida – List.jsp

```

<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Listar Partida</title>
        <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    </head>
    <body>

        <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
        <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>

        <div id="corpo">
            <f:view>
                <h:messages errorStyle="color: red" infoStyle="color: green"
                    layout="table"/>
                <h1>Listando Partidas</h1>

                <h:form>
                    <h:selectOneMenu id="temporadaId"
                        value="#{partidaAdmin.temporada}" title="Temporada" >
                        <f:selectItems value="#{partidaAdmin.temporadaIds}"/>
                    </h:selectOneMenu>
                    <h:commandButton action="#{partidaAdmin.partidasTemp}"
                        value="Selecione"/>
                </h:form>

                <h:form>
                    <h:outputText value="Item #{partidaAdmin.firstItem + 1}..
                        #{partidaAdmin.lastItem} de
                        #{partidaAdmin.itemCount}"
                        styleClass="item"/>&nbsp;
                    <h:commandLink action="#{partidaAdmin.first}" value="Primeiro"
                        rendered="#{partidaAdmin.lastItem >
                            partidaAdmin.batchSize}"
                        styleClass="item"/>
                    <h:commandLink action="#{partidaAdmin.prev}" value="Anterior"
                        rendered="#{partidaAdmin.firstItem >= partidaAdmin.batchSize}"
                        styleClass="item"/>&nbsp;
                    <h:commandLink action="#{partidaAdmin.next}" value="Próximo"
                        rendered="#{partidaAdmin.lastItem +
                            partidaAdmin.batchSize <=
                            partidaAdmin.itemCount}"
                        styleClass="item"/>&nbsp;
                    <h:commandLink action="#{partidaAdmin.next}"
                        value="Restando #{partidaAdmin.itemCount -

```

```

        partidaAdmin.lastItem}"
        rendered="#{partidaAdmin.lastItem <
            partidaAdmin.itemCount &&
            partidaAdmin.lastItem +
            partidaAdmin.batchSize >
            partidaAdmin.itemCount}"
        styleClass="item"/>
<h:commandLink action="#{partidaAdmin.last}" value="Ultimo"
    rendered="#{partidaAdmin.lastItem <
        partidaAdmin.itemCount -
        partidaAdmin.batchSize + 1}"
        styleClass="item"/>
<h:commandLink action="#{partidaAdmin. hoje}" value="Hoje"
    styleClass="item"/>

<h:dataTable value=&apos;#{partidaAdmin.partidasTemporada}&apos;
var=&apos;item&apos; border="1" cellpadding="2"
    cellspacing="0"
    styleClass="tabela" headerClass="tabela_header"
    rowClasses="tabela_row1,tabela_row2"
    columnClasses="t_id,t_temporada,t_data,t_hora,t_mandante,
        t_golsMandante,t_x,t_golsVisitante,t_visitante,
        t_pMandante,t_pVisitante,t_prob,t_prob,
        t_prob,t_links" >
    <h:column>
        <f:facet name="header">
            <h:outputText value="Id" title="ID da partida"/>
        </f:facet>
        <h:commandLink action="#{partidaAdmin.detailSetup}"
            value="#{item.idPartida}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Temporada"
                title="Temporada do Campeonato"/>
        </f:facet>
        <h:outputText value="#{item.temporadaId.nomeTemporada}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Data" title="Data da partida"/>
        </f:facet>
        <h:outputText value="#{item.dataPartida}">
            <f:convertDateTime type="DATE" pattern="dd/MM/yyyy"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Hora"
                title="Hora da partida"/>
        </f:facet>
        <h:outputText value="#{item.horaPartida}">
            <f:convertDateTime type="TIME" pattern="HH:mm" />
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Mandante"
                title="Clube Mandante"/>
        </f:facet>
        <h:outputText value="#{item.clubeMandanteId.nomeClube}"
            styleClass="mandante"/>
    </h:column>
</h:column>
</h:column>

```

```

        </f:facet name="header">
            <h:outputText value=""
                title="Gols do Clube Mandante"/>
        </f:facet>
        <h:outputText value="#{item.golsMandante}"/>
    </h:column>

    <h:column>
        <f:facet name="header">
            <h:outputText value="x"/>
        </f:facet>
        <h:outputText value="x"/>
    </h:column>

    <h:column>
        <f:facet name="header">
            <h:outputText value=""
                title="Gols do Clube Visitante"/>
        </f:facet>
        <h:outputText value="#{item.golsVisitante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Visitante"
                title="Clube Visitante"/>
        </f:facet>
        <h:outputText value="#{item.clubeVisitanteId.nomeClube}"
            styleClass="visitante"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="P Man"
                title="Parâmetro do Clube Mandante para a
partida"/>
        </f:facet>
        <h:outputText value="#{item.mediaMandante}"
            <f:convertNumber pattern="0.00"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="P Vis"
                title="Parâmetro do Clube Visitante para a
partida"/>
        </f:facet>
        <h:outputText value="#{item.mediaVisitante}"
            <f:convertNumber pattern="0.00"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Prob M"
                title="Probabilidade de Vitória do Clube
Mandante"/>
        </f:facet>
        <h:outputText value="#{item.probMandante}"
            <f:convertNumber pattern="#0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Prob E"
                title="Probabilidade de Empate"/>
        </f:facet>

```

```

        <h:outputText value="#{item.probEmpate}">
            <f:convertNumber pattern="#0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Prob V"
                title="Probabilidade de Vitória do Clube
Visitante"/>
        </f:facet>
        <h:outputText value="#{item.probVisitante}">
            <f:convertNumber pattern="#0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Links"/>
        </f:facet>
        <h:commandLink value="Apagar"
            action="#{partidaAdmin.destroy}">
            <f:param name="idPartida"
                value="#{item.idPartida}"/>
        </h:commandLink>
        <h:outputText value=" "/>
        <h:commandLink value="Editar"
            action="#{partidaAdmin.editSetup}">
            <f:param name="idPartida"
                value="#{item.idPartida}"/>
        </h:commandLink>
        <h:outputText value=" "/>
        <h:commandLink value="Calc"
            action="#{partidaAdmin.buscaParametros}">
            <f:param name="idPartida"
                value="#{item.idPartida}"/>
        </h:commandLink>
    </h:column>
</h:dataTable>
<h:commandLink action="#{partidaAdmin.createSetup}"
    value="Nova Partida"/>
</h:form>
</f:view>

</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.18. Admin – Partida – Detail.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
        <title>Detalhe da Partida</title>
    </head>
    <body>
        <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
        <%@ include file="/WEB-INF/jspf/_menuAdmin.jspf" %>

```

```

<div id="corpo">
  <f:view>
    <h:messages errorStyle="color: red" infoStyle="color: green"
      layout="table"/>
    <h1>Detalhe da partida</h1>
    <h:form>

      <h:dataTable value=&apos;#{partidaAdmin.partida}&apos;
        var=&apos;item&apos; border="1" cellpadding="2"
        cellspacing="0"
        styleClass="tabela" headerClass="tabela_header"
        rowClasses="tabela_row1,tabela_row2"
        columnClasses="t_id,t_temporada,t_data,t_hora,t_mandante,
t_golsMandante,t_x,t_golsVisitante,t_visitante,t_pMandante,
t_pVisitante,t_prob,t_prob,t_prob,t_links" >
      <h:column>
        <f:facet name="header">
          <h:outputText value="Id" title="ID da partida"/>
        </f:facet>
        <h:outputText value="#{item.idPartida}"/>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Temporada"
            title="Temporada do Campeonato"/>
        </f:facet>
        <h:outputText
          value="#{item.temporadaId.nomeTemporada}"/>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Data"
            title="Data da partida"/>
        </f:facet>
        <h:outputText value="#{item.dataPartida}"
          <f:convertDateTime type="DATE"
            pattern="dd/MM/yyyy" />
        </h:outputText>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Hora"
            title="Hora da partida"/>
        </f:facet>
        <h:outputText value="#{item.horaPartida}"
          <f:convertDateTime type="TIME"
            pattern="HH:mm" />
        </h:outputText>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Mandante"
            title="Clube Mandante"/>
        </f:facet>
        <h:outputText
          value="#{item.clubeMandanteId.nomeClube}"/>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value=""
            title="Gols do Clube Mandante"/>
        </f:facet>
        <h:outputText value="#{item.golsMandante}"/>
      </h:column>
    </h:dataTable>
    </h:form>
  </f:view>
</div>

```

```

</h:column>
  <f:facet name="header">
    <h:outputText value="x"/>
  </f:facet>
  <h:outputText value="x"/>
</h:column>

<h:column>
  <f:facet name="header">
    <h:outputText value=""
      title="Gols do Clube Visitante"/>
  </f:facet>
  <h:outputText value="#{item.golsVisitante}"/>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Visitante"
      title="Clube Visitante"/>
  </f:facet>
  <h:outputText
    value="#{item.clubeVisitanteId.nomeClube}"/>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="P Man"
      title="Parâmetro do Clube Mandante para a
partida"/>
  </f:facet>
  <h:outputText value="#{item.mediaMandante}"
    <f:convertNumber pattern="0.00"/>
  </h:outputText>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="P Vis"
      title="Parâmetro do Clube Visitante para a
partida"/>
  </f:facet>
  <h:outputText value="#{item.mediaVisitante}"
    <f:convertNumber pattern="0.00"/>
  </h:outputText>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Prob M"
      title="Probabilidade de Vitória do Clube
Mandante"/>
  </f:facet>
  <h:outputText value="#{item.probMandante}"
    <f:convertNumber pattern="#0.0%"/>
  </h:outputText>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Prob E"
      title="Probabilidade de Empate"/>
  </f:facet>
  <h:outputText value="#{item.probEmpate}"
    <f:convertNumber pattern="#0.0%"/>
  </h:outputText>
</h:column>
<h:column>
  <f:facet name="header">
    <h:outputText value="Prob V"

```

```

        title="Probabilidade de Vitória do Clube
Visitante"/>
        </f:facet>
        <h:outputText value="#{item.probVisitante}">
            <f:convertNumber pattern="#0.0%"/>
        </h:outputText>
    </h:column>
</h:dataTable>

<div id="mandante">
    <h:dataTable
        value="#{partidaAdmin.probabilidadesMandante}"
        var=&apos;item&apos; border="1" cellpadding="2"
cellspacing="0"
        styleClass="tabela" headerClass="tabela_header"
        rowClasses="tabela_row1,tabela_row2">
        <f:facet name="header">
            <h:outputText value="Probabilidade Mandante" />
        </f:facet>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Mand"
                    styleClass="t_pMandante"/>
            </f:facet>
            <h:outputText value="#{item.placarMandante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Vis"
                    styleClass="t_pVisitante"/>
            </f:facet>
            <h:outputText value="#{item.placarVisitante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Prob"
                    styleClass="t_prob"/>
            </f:facet>
            <h:outputText value="#{item.prob}" >
                <f:convertNumber pattern="##0.0%" />
            </h:outputText>
        </h:column>
    </h:dataTable>
</div>
<div id="empate">
    <h:dataTable
        value="#{partidaAdmin.probabilidadesEmpate}"
        var=&apos;item&apos; border="1" cellpadding="2"
cellspacing="0"
        styleClass="tabela" headerClass="tabela_header"
        rowClasses="tabela_row1,tabela_row2">
        <f:facet name="header">
            <h:outputText value="Probabilidade Empate" />
        </f:facet>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Mand"
                    styleClass="t_pMandante"/>
            </f:facet>
            <h:outputText value="#{item.placarMandante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Vis"

```



```

        styleClass="t_pVisitante"/>
    </f:facet>
    <h:outputText value="#{item.placarVisitante}"/>
</h:column>
<h:column>
    <f:facet name="header">
        <h:outputText value="Prob"
            styleClass="t_prob"/>
    </f:facet>
    <h:outputText value="#{item.prob}" >
        <f:convertNumber pattern="##0.0%" />
    </h:outputText>
</h:column>
</h:dataTable>
</div>
<div id="visitante">
    <h:dataTable
        value="#{partidaAdmin.probabilidadesVisitante}"
        var=&apos;item&apos; border="1" cellpadding="2"
        cellspacing="0"
        styleClass="tabela" headerClass="tabela_header"
        rowClasses="tabela_row1,tabela_row2">
        <f:facet name="header">
            <h:outputText value="Probabilidade Visitante" />
        </f:facet>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Mand"
                    styleClass="t_pMandante"/>
            </f:facet>
            <h:outputText value="#{item.placarMandante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Vis"
                    styleClass="t_pVisitante"/>
            </f:facet>
            <h:outputText value="#{item.placarVisitante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Prob"
                    styleClass="t_prob"/>
            </f:facet>
            <h:outputText value="#{item.prob}" >
                <f:convertNumber pattern="##0.0%" />
            </h:outputText>
        </h:column>
    </h:dataTable>
</div>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.19. User – Clube – List.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jspf/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jspf/html" prefix="h" %>

```

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
  <title>Listar Clube</title>
</head>
<body>
<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuUser.jspf" %>
<div id="corpo">
  <f:view>
    <h:messages errorStyle="color: red" infoStyle="color: green"
      layout="table"/>
    <h1>Lista de Clubes</h1>
    <h:form>
      <h:outputText value="Item #{clube.firstItem + 1}..
        #{clube.lastItem} de
        #{clube.itemCount}"
        styleClass="item"/>&nbsp;
      <h:commandLink action="#{clube.prev}"
        value="Anterior #{clube.batchSize}"
        rendered="#{clube.firstItem >= clube.batchSize}"
        styleClass="item"/>&nbsp;
      <h:commandLink action="#{clube.next}"
        value="Próximo #{clube.batchSize}"
        rendered="#{clube.lastItem +
          clube.batchSize <=
          clube.itemCount}"
        styleClass="item"/>&nbsp;
      <h:commandLink action="#{clube.next}"
        value="Restando #{clube.itemCount - clube.lastItem}"
        rendered="#{clube.lastItem <
          clube.itemCount &&
          clube.lastItem +
          clube.batchSize >
          clube.itemCount}"
        styleClass="item"/>
      <h:dataTable value="&apos;#{clube.clubes}&apos;
        var=&apos;item&apos; border="1" cellpadding="2"
        cellspacing="0"
        styleClass="tabela" headerClass="tabela_header"
        rowClasses="tabela_row1,tabela_row2"
        columnClasses="t_clube,t_uf,t_dataFundacao" >
        <h:column>
          <f:facet name="header">
            <h:outputText value="Clube" title="Nome do Clube"/>
          </f:facet>
          <h:outputText value="#{item.nomeClube}" />
        </h:column>
        <h:column>
          <f:facet name="header">
            <h:outputText value="UF" title="UD do Clube"/>
          </f:facet>
          <h:outputText value="#{item.ufClube}" />
        </h:column>
        <h:column>
          <f:facet name="header">
            <h:outputText value="Fundacao"
              title="Data de Fundação"/>
          </f:facet>
          <h:outputText value="#{item.dataFundacao}" >
            <f:convertDateTime type="DATE"
              pattern="dd/MM/yyyy" />
          </h:outputText>
      </h:dataTable>
    </h:form>
  </f:view>
</div>

```

```

        </h:column>
    </h:dataTable>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.20. User – Partida – List.jsp

```

<%@page contentType="text/html" %>
<%@page pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Listar Partida</title>
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
</head>
<body>

<%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
<%@ include file="/WEB-INF/jspf/_menuUser.jspf" %>

    <div id="corpo">
        <f:view>
            <h:messages errorStyle="color: red" infoStyle="color: green"
                layout="table"/>
            <h1>Lista de Partidas</h1>

            <h:form>
                <h:selectOneMenu id="temporadaId" value="#{partida.temporada}"
                    title="Temporada" >
                    <f:selectItems value="#{partida.temporadaIds}"/>
                </h:selectOneMenu>
                <h:commandButton action="#{partida.partidasTemp}"
                    value="Selecione"/>
            </h:form>

            <h:form>
                <h:outputText value="Item #{partida.firstItem + 1}..
                    #{partida.lastItem} of #{partida.itemCount}"
                    styleClass="item"/>&nbsp;
                <h:commandLink action="#{partida.first}" value="Primeiro"
                    rendered="#{partida.lastItem >
                        partida.batchSize}" styleClass="item"/>
                <h:commandLink action="#{partida.prev}" value="Anterior"
                    rendered="#{partida.firstItem >= partida.batchSize}"
                    styleClass="item"/>&nbsp;
                <h:commandLink action="#{partida.next}" value="Próximo"
                    rendered="#{partida.lastItem +
                        partida.batchSize <=
                            partida.itemCount}"
                    styleClass="item"/>&nbsp;
                <h:commandLink action="#{partida.next}"
                    value="Restando #{partida.itemCount - partida.lastItem}"
                    rendered="#{partida.lastItem <
                        partida.itemCount &&
                            partida.lastItem +
                                partida.batchSize >
                                    partida.itemCount}"

```

```

        styleClass="item"/>
<h:commandLink action="#{partida.last}" value="Último"
        rendered="#{partida.lastItem <
        partida.itemCount - partida.batchSize + 1}"
        styleClass="item"/>
<h:commandLink action="#{partida. hoje}" value="Hoje"
        styleClass="item"/>
<h:dataTable value=&apos;#{partida.partidasTemporada}&apos;
var=&apos;item&apos;
        border="1" cellpadding="2" cellspacing="0" styleClass="tabela"
        headerClass="tabela_header"
rowClasses="tabela_row1,tabela_row2"
        columnClasses="t_id,t_temporada,t_data,t_hora,t_mandante,
        t_golsMandante,t_x,t_golsVisitante,t_visitante,
        t_probMandante,t_probEmpate,t_ProbVisitante" >
    <h:column>
        <f:facet name="header">
            <h:outputText value="Id" title="ID da partida"/>
        </f:facet>
        <h:commandLink action="#{partida.detailSetup}"
            value="#{item.idPartida}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Temporada" title="Temporada do Campeonato"/>
        </f:facet>
        <h:outputText
            value="#{item.temporadaId.nomeTemporada}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Data" title="Data da partida"/>
        </f:facet>
        <h:outputText value="#{item.dataPartida}">
            <f:convertDateTime
                type="DATE" pattern="dd/MM/yyyy" />
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Hora" title="Hora da partida"/>
        </f:facet>
        <h:outputText value="#{item.horaPartida}">
            <f:convertDateTime type="TIME" pattern="HH:mm" />
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Mandante" title="Clube Mandante"/>
        </f:facet>
        <h:outputText
            value="#{item.clubeMandanteId.nomeClube}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="" title="Gols do Clube Mandante"/>
        </f:facet>
        <h:outputText value="#{item.golsMandante}"/>
    </h:column>
</h:column>

```

```

        <f:facet name="header">
            <h:outputText value="x"/>
        </f:facet>
        <h:outputText value="x"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="" title="Gols do Clube Visitante"/>
        </f:facet>
        <h:outputText value="#{item.golsVisitante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Visitante" title="Clube Visitante"/>
        </f:facet>
        <h:outputText
            value="#{item.clubeVisitanteId.nomeClube}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Prob M"
                title="Probabilidade de Vitória do Clube
Mandante"/>
        </f:facet>
        <h:outputText value="#{item.probMandante}">
            <f:convertNumber pattern="#0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Prob E"
                title="Probabilidade de Empate"/>
        </f:facet>
        <h:outputText value="#{item.probEmpate}">
            <f:convertNumber pattern="#0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Prob V"
                title="Probabilidade de Vitória do Clube
Visitante"/>
        </f:facet>
        <h:outputText value="#{item.probVisitante}">
            <f:convertNumber pattern="#0%"/>
        </h:outputText>
    </h:column>
</h:dataTable>
</h:form>
</f:view>
</div>
<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.21. User – Partida – Detail.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<%@taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="/chutegol/chutegol.css" rel="stylesheet" type="text/css">
    <title>Detalhe da Partida</title>
  </head>
  <body>

    <%@ include file="/WEB-INF/jspf/_cabecalho.jspf" %>
    <%@ include file="/WEB-INF/jspf/_menuUser.jspf" %>

    <div id="corpo">
      <f:view>
        <h:messages errorStyle="color: red" infoStyle="color: green"
          layout="table"/>
        <h1>Detalhe da partida</h1>
        <h:form>

          <h:dataTable value=&apos;#{partida.partida}&apos;
var=&apos;item&apos;
border="1" cellpadding="2" cellspacing="0"
styleClass="tabela" headerClass="tabela_header"
rowClasses="tabela_row1,tabela_row2"
columnClasses="t_id,t_temporada,t_data,t_hora,
t_mandante,t_golsMandante,t_x,t_golsVisitante,
t_visitante,t_prob,t_prob,t_prob" >
      <h:column>
        <f:facet name="header">
          <h:outputText value="Id" title="ID da partida"/>
        </f:facet>
        <h:outputText value="#{item.idPartida}"/>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Temporada"
            title="Temporada do Campeonato"/>
        </f:facet>
        <h:outputText
          value="#{item.temporadaId.nomeTemporada}"/>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Data"
            title="Data da partida"/>
        </f:facet>
        <h:outputText value="#{item.dataPartida}">
          <f:convertDateTime
            type="DATE" pattern="dd/MM/yyyy" />
        </h:outputText>
      </h:column>
      <h:column>
        <f:facet name="header">
          <h:outputText value="Hora"
            title="Hora da partida"/>
        </f:facet>
        <h:outputText value="#{item.horaPartida}">
          <f:convertDateTime
            type="TIME" pattern="HH:mm" />
        </h:outputText>
      </h:column>
    </h:form>
    </div>
  </body>
</html>
```

```

        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Mandante" title="Clube Mandante"/>
        </f:facet>
        <h:outputText
            value="#{item.clubeMandanteId.nomeClube}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value=""
                title="Gols do Clube Mandante"/>
        </f:facet>
        <h:outputText value="#{item.golsMandante}"/>
    </h:column>

    <h:column>
        <f:facet name="header">
            <h:outputText value="x"/>
        </f:facet>
        <h:outputText value="x"/>
    </h:column>

    <h:column>
        <f:facet name="header">
            <h:outputText value=""
                title="Gols do Clube Visitante"/>
        </f:facet>
        <h:outputText value="#{item.golsVisitante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Visitante"
                title="Clube Visitante"/>
        </f:facet>
        <h:outputText
            value="#{item.clubeVisitanteId.nomeClube}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Prob M"
                title="Probabilidade de Vitória do Clube
Mandante"/>
        </f:facet>
        <h:outputText value="#{item.probMandante}">
            <f:convertNumber pattern="#0.0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Prob E"
                title="Probabilidade de Empate"/>
        </f:facet>
        <h:outputText value="#{item.probEmpate}">
            <f:convertNumber pattern="#0.0%"/>
        </h:outputText>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText
                value="Prob V"

```

```

        title="Probabilidade de Vitória do Clube
Visitante"/>
        </f:facet>
        <h:outputText value="#{item.probVisitante}">
            <f:convertNumber pattern="#0.0%"/>
        </h:outputText>
    </h:column>
</h:dataTable>

<div id="mandante">
<h:dataTable value="#{partida.proBABILIDADESmandante}"
var=&apos;item&apos; border="1" cellpadding="2"
cellspacing="0"
styleClass="tabela" headerClass="tabela_header"
rowClasses="tabela_row1,tabela_row2">
    <f:facet name="header">
        <h:outputText value="Probabilidade Mandante" />
    </f:facet>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Mand"
styleClass="t_pMandante"/>
        </f:facet>
        <h:outputText value="#{item.placarMandante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Vis"
styleClass="t_pVisitante"/>
        </f:facet>
        <h:outputText value="#{item.placarVisitante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Prob"
styleClass="t_prob"/>
        </f:facet>
        <h:outputText value="#{item.prob}" >
            <f:convertNumber pattern="##0.0%" />
        </h:outputText>
    </h:column>
</h:dataTable>
</div>
<div id="empate">
<h:dataTable value="#{partida.proBABILIDADESempate}"
var=&apos;item&apos; border="1" cellpadding="2"
cellspacing="0"
styleClass="tabela" headerClass="tabela_header"
rowClasses="tabela_row1,tabela_row2">
    <f:facet name="header">
        <h:outputText value="Probabilidade Empate" />
    </f:facet>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Mand"
styleClass="t_pMandante"/>
        </f:facet>
        <h:outputText value="#{item.placarMandante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Vis"
styleClass="t_pVisitante"/>

```



```

        </f:facet>
        <h:outputText value="#{item.placarVisitante}"/>
    </h:column>
    <h:column>
        <f:facet name="header">
            <h:outputText value="Prob" styleClass="t_prob"/>
        </f:facet>
        <h:outputText value="#{item.prob}" >
            <f:convertNumber pattern="##0.0%" />
        </h:outputText>
    </h:column>
</h:dataTable>
</div>
<div id="visitante">
    <h:dataTable value="#{partida.probabilidadesVisitante}"
        var=&apos;item&apos; border="1" cellpadding="2"
        cellspacing="0"
        styleClass="tabela" headerClass="tabela_header"
        rowClasses="tabela_row1,tabela_row2">
        <f:facet name="header">
            <h:outputText value="Probabilidade Visitante" />
        </f:facet>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Mand"
                    styleClass="t_pMandante"/>
            </f:facet>
            <h:outputText value="#{item.placarMandante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Vis"
                    styleClass="t_pVisitante"/>
            </f:facet>
            <h:outputText value="#{item.placarVisitante}"/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Prob" styleClass="t_prob"/>
            </f:facet>
            <h:outputText value="#{item.prob}" >
                <f:convertNumber pattern="##0.0%" />
            </h:outputText>
        </h:column>
    </h:dataTable>
</div>
</h:form>
</f:view>
</div>

<%@ include file="/WEB-INF/jspf/_rodape.jspf" %>
</body>
</html>

```

11.22. login.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="pt-br">
<head>
<meta http-equiv="Content-Type" content="text/html" charset=iso-8859-1 />
<title>:: Autenticação ::</title>
</head>
<body>

```

```

<form method="POST" action="<%= response.encodeURL("j_security_check") %>" >
<fieldset title="Informe login e senha">
<legend>Login</legend>
<label for="j_username">Login:</label>
<input type="text" name="j_username" class="textBox"/><br/>
<label for="j_password">Senha:</label>
<input type="password" name="j_password" class="textBox"/><br/>
</fieldset>
<p align="center">
<input type="submit" value="Log In"/>
</p>
</form>
</body>
</html>

```

11.23. error.jsp

```

<html>
<head>
<title>Erro ao tentar autenticar usuário</title>
</head>
<body>
<p>Usuário ou senha inválidos.<br/>
Verifique o nome de usuário e a senha e
<a href="<%= response.encodeURL("login.jsp") %>">tente novamente</a>.</p>
</body>
</html>

```

11.24. chuteGol.css

```

* {
margin:0;
padding:0;
list-style:none;
vertical-align:baseline;
}

#cabecalho{
display:block;
text-indent:-9000px;
overflow:hidden;
height: 70px;
margin:0 auto;
background:url(/chuteGol/imagens/chuteGol.png) no-repeat; margin: 10px 0px 0px
10px;
}

#menu {
background: #090;
margin: 0 14px 0 11px;
height: 25px;
width: 925px;
}

#menu ul li {
display :inline ;
}

#menu a{
font: 12px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color: white;
float: left;
padding: 0 20px;
height: 25px;
line-height: 25px;
border-right: 1px solid #64c800;
}

```

```

border-left: 1px solid #30c801;
text-transform :capitalize ;
}

#menu ul li a:hover {background: #64c800;}

#corpo{
margin: 20px 10px;
height: 450px ;
}

#corpo h1{
font: 24px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color : #090;
}

#corpo h2{
font: 20px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color : #090;
line-height :50px ;
padding-left :20px ;
}

#corpo h3{
font: 18px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color : #090;
line-height :40px ;
padding-left :20px ;
}

#corpo p{
line-height: 18px;
padding: 0 40px 10px;
font: 16px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
text-align : justify;
color : #090;
}

#corpo ul li{
margin-left :60px ;
list-style :square ;
font: 14px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color : #090;
}

#rodape{
border-top : 1px dotted #090;
}

#rodape p{
font: 10px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
text-align : center;
color : #090;
}

#mandante{
position : absolute;
margin-top :20px ;
}

#empate{
position : absolute;
margin-left : 180px ;
margin-top :20px ;
}

#visitante{
font-family: serif;

```

```

margin-left: 343px;
margin-top: 20px;
position: absolute
}

.tabela{
font: 12px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
padding: 2px 2px 2px 2px;
border-right : solid 1px #64c800;
border-left : absolut;
}

.tabela td{text-align:center;}

.tabela_header{
background-color : #090;
font: 14px Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
font-weight : bold;
color: #fff;
}

.tabela_row1{
background-color : #64c800;
height :20 ;
}

.tabela_row2{
background-color : #fff;
height :20 ;
}

.tabelaCadastro{
font: 14px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
padding: 2px 2px 2px 2px;
}

.tabelaCadastro td{text-align:left;}

.item{
font: 12px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color : #090;
}

.criar{
font: 20px Trebuchet Ms, Arial, Tahoma, Verdana, Helvetica, Sans-Serif;
color : #090;
}

.t_id{width :40 ;}
.t_temporada{width :100 ;}
.t_campeonato{width :200 ;}
.t_data{width :70 ;}
.t_hora{width :50 ;}
.t_mandante{width :100 ;}
.t_golsMandante{width :30 ;}
.t_x{width :20 ;}
.t_golsVisitante{width :30 ;}
.t_visitante{width :100 ;}
.t_pMandante{width :50 ;}
.t_pVisitante{width :50 ;}
.t_prob{width :50 ;}
.t_links{width :100 ;}
.t_clube{width :100 ;}
.t_uf{width :30 ;}
.t_dataFundacao{width :100 ;}

```

11.25. _cabecalho.jsfp

```
<div id="cabecalho">
  <h1>Chutegol - Chance de Um Time Em GOLs</h1>
</div>
```

11.26. _rodape.jsfp

```
<div id="rodape">
  <p>UNIVERSIDADE FEDERAL DE SANTA CATARINA
  <br>Centro Tecnológico - Sistemas de Informação
  <br>TCC desenvolvido por
  <a href="mailto:cceolato@inf.ufsc.br">Carlos Alberto Ceolato Junior</a>
  <br>Orientador: Prof PhD
  <a href="mailto:dandrade@inf.ufsc.br">Dalton Francisco de Andrade</p>
</div>
```

11.27. _menuAdmin.jsfp

```
<div id="menu">
  <ul>
    <li>
      <a href="/chutegol/Admin/index.jsp">Home</a>
    </li>
    <li>
      <a href="/chutegol/faces/Admin/campeonato/List.jsp">Campeonato</a>
    </li>
    <li>
      <a href="/chutegol/faces/Admin/temporada/List.jsp">Temporada</a>
    </li>
    <li>
      <a href="/chutegol/faces/Admin/clube/List.jsp">Clube</a>
    </li>
    <li>
      <a href="/chutegol/faces/Admin/partida/List.jsp">Partida</a>
    </li>
  </ul>
</div>
```

11.28. _menuUser.jsfp

```
<div id="menu">
  <ul>
    <li><a href="/chutegol/index.jsp">Home</a> </li>
    <li><a href="/chutegol/faces/User/partida/List.jsp">Partida</a></li>
    <li><a href="/chutegol/faces/User/clube/List.jsp">Clube</a></li>
  </ul>
</div>
```

11.29. web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <context-param>
    <param-name>com.sun.faces.verifyObjects</param-name>
    <param-value>>false</param-value>
  </context-param>
  <context-param>
    <param-name>com.sun.faces.validateXml</param-name>
```

```

    <param-value>true</param-value>
</context-param>
<context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>client</param-value>
</context-param>
<servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<session-config>
    <session-timeout>
        30
    </session-timeout>
</session-config>
<welcome-file-list>
    <welcome-file>
        index.jsp
    </welcome-file>
</welcome-file-list>

<resource-ref>
    <description>Web Database</description>
    <res-ref-name>jdbc/stripesec</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
</resource-ref>

<security-constraint>
    <display-name>Área Restrita</display-name>
    <web-resource-collection>
        <web-resource-name>Arquivos protegidos por login
        </web-resource-name>
        <url-pattern>/Admin/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
        <http-method>HEAD</http-method>
        <http-method>PUT</http-method>
        <http-method>OPTIONS</http-method>
        <http-method>TRACE</http-method>
        <http-method>DELETE</http-method>
    </web-resource-collection>
    <web-resource-collection>
        <web-resource-name>Area Faces do Admin</web-resource-name>
        <description/>
        <url-pattern>/faces/Admin/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
        <http-method>HEAD</http-method>
        <http-method>PUT</http-method>
        <http-method>OPTIONS</http-method>
        <http-method>TRACE</http-method>
        <http-method>DELETE</http-method>
    </web-resource-collection>
    <auth-constraint>
        <role-name>admin</role-name>
    </auth-constraint>
</security-constraint>

<login-config>
    <auth-method>FORM</auth-method>
    <realm-name>StripesSecRealm</realm-name>
</form-login-config>

```

```

        <form-login-page>/login.jsp</form-login-page>
        <form-error-page>/error.jsp</form-error-page>
    </form-login-config>
</login-config>

<security-role>
    <role-name>admin</role-name>
</security-role>

</web-app>

```

11.30. faces-config.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces
Config 1.1//EN" "http://java.sun.com/dtd/web-facesconfig_1_1.dtd">
<!-- ===== FULL CONFIGURATION FILE ===== -->
<faces-config xmlns="http://java.sun.com/JSF/Configuration">
    <converter>
        <converter-for-class>persistencia.Campeonato</converter-for-class>
        <converter-class>chutegol.user.CampeonatoConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Clube</converter-for-class>
        <converter-class>chutegol.user.ClubeConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Parametro</converter-for-class>
        <converter-class>chutegol.user.ParametroConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.ParametroPK</converter-for-class>
        <converter-class>chutegol.user.ParametroPKConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Partida</converter-for-class>
        <converter-class>chutegol.user.PartidaConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Temporada</converter-for-class>
        <converter-class>chutegol.user.TemporadaConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Campeonato</converter-for-class>
        <converter-class>chutegol.admin.CampeonatoConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Clube</converter-for-class>
        <converter-class>chutegol.admin.ClubeConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Parametro</converter-for-class>
        <converter-class>chutegol.admin.ParametroConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.ParametroPK</converter-for-class>
        <converter-class>chutegol.admin.ParametroPKConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Partida</converter-for-class>
        <converter-class>chutegol.admin.PartidaConverter</converter-class>
    </converter>
    <converter>
        <converter-for-class>persistencia.Temporada</converter-for-class>
        <converter-class>chutegol.admin.TemporadaConverter</converter-class>
    </converter>

```

```

<managed-bean>
  <managed-bean-name>campeonato</managed-bean-name>
  <managed-bean-class>chutegol.user.CampeonatoController</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>clube</managed-bean-name>
  <managed-bean-class>chutegol.user.ClubeController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>parametro</managed-bean-name>
  <managed-bean-class>chutegol.user.ParametroController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>partida</managed-bean-name>
  <managed-bean-class>chutegol.user.PartidaController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>temporada</managed-bean-name>
  <managed-bean-class>chutegol.user.TemporadaController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>campeonatoAdmin</managed-bean-name>
  <managed-bean-class>chutegol.admin.CampeonatoController</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>clubeAdmin</managed-bean-name>
  <managed-bean-class>chutegol.admin.ClubeController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>parametroAdmin</managed-bean-name>
  <managed-bean-class>chutegol.admin.ParametroController</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>partidaAdmin</managed-bean-name>
  <managed-bean-class>chutegol.admin.PartidaController</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>temporadaAdmin</managed-bean-name>
  <managed-bean-class>chutegol.admin.TemporadaController</managed-bean-
class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<navigation-rule>
  <navigation-case>
    <from-outcome>clube_list</from-outcome>
    <to-view-id>/User/clube/List.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>clube_detail</from-outcome>
    <to-view-id>/User/clube/Detail.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>

```



```

        <from-outcome>partida_list</from-outcome>
        <to-view-id>/User/partida/List.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>partida_detail</from-outcome>
        <to-view-id>/User/partida/Detail.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>campeonatoAdmin_create</from-outcome>
        <to-view-id>/Admin/campeonato/New.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>campeonatoAdmin_list</from-outcome>
        <to-view-id>/Admin/campeonato/List.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>campeonatoAdmin_edit</from-outcome>
        <to-view-id>/Admin/campeonato/Edit.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>campeonatoAdmin_detail</from-outcome>
        <to-view-id>/Admin/campeonato/Detail.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>clubeAdmin_create</from-outcome>
        <to-view-id>/Admin/clube/New.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>clubeAdmin_list</from-outcome>
        <to-view-id>/Admin/clube/List.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>clubeAdmin_edit</from-outcome>
        <to-view-id>/Admin/clube/Edit.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>clubeAdmin_detail</from-outcome>
        <to-view-id>/Admin/clube/Detail.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>partidaAdmin_create</from-outcome>
        <to-view-id>/Admin/partida/New.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
<navigation-rule>
    <navigation-case>
        <from-outcome>partidaAdmin_list</from-outcome>
        <to-view-id>/Admin/partida/List.jsp</to-view-id>
    </navigation-case>

```

```

    </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>partidaAdmin_edit</from-outcome>
    <to-view-id>/Admin/partida/Edit.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>partidaAdmin_detail</from-outcome>
    <to-view-id>/Admin/partida/Detail.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>temporadaAdmin_create</from-outcome>
    <to-view-id>/Admin/temporada/New.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>temporadaAdmin_list</from-outcome>
    <to-view-id>/Admin/temporada/List.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>temporadaAdmin_edit</from-outcome>
    <to-view-id>/Admin/temporada/Edit.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <navigation-case>
    <from-outcome>temporadaAdmin_detail</from-outcome>
    <to-view-id>/Admin/temporada/Detail.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
</faces-config>

```

11.31. context.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<Context path="/chutegol" displayName="chutegol" reloadable="true">

  <Resource name="jdbc/stripesec"
    auth="Container"
    type="javax.sql.DataSource"
    maxActive="10"
    maxIdle="3"
    username="usuario"
    password="senha"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/chutegol" />

  <Realm className="org.apache.catalina.realm.DataSourceRealm" debug="99"
    dataSourceName="jdbc/stripesec" localDataSource="true"
    userTable="usuario"
    userNameCol="login"
    userCredCol="senha"
    userRoleTable="roles"
    roleNameCol="role" />

</Context>

```

11.32. persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
  <persistence-unit name="chutegolPU" transaction-type="RESOURCE_LOCAL">

<provider>oracle.toplink.essentials.ejb.cmp3.EntityManagerFactoryProvider</pro
vider>
  <class>persistencia.Parametro</class>
  <class>persistencia.Campeonato</class>
  <class>persistencia.Partida</class>
  <class>persistencia.Temporada</class>
  <class>persistencia.Clube</class>
  <properties>
    <property name="toplink.jdbc.url"
value="jdbc:mysql://localhost:3306/chutegol"/>
    <property name="toplink.jdbc.user" value="usuario"/>
    <property name="toplink.jdbc.driver" value="com.mysql.jdbc.Driver"/>
    <property name="toplink.jdbc.password" value="senha"/>
  </properties>
</persistence-unit>
</persistence>
```

11.33. Persistência - Campeonato.java

```
/*
 * Campeonato.java
 *
 * Created on 21 de Agosto de 2007, 18:26
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package persistencia;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;

/**
 * Classe de entidade Campeonato
 *
 * @author Carlos
 */
@Entity
@Table(name = "campeonato")
@NamedQueries( {
    @NamedQuery(name = "Campeonato.findByIdCampeonato", query = "SELECT c
FROM Campeonato c WHERE c.idCampeonato = :idCampeonato"),
    @NamedQuery(name = "Campeonato.findByNameCampeonato", query = "SELECT
c FROM Campeonato c WHERE c.nomeCampeonato = :nomeCampeonato")
})

public class Campeonato implements Serializable {

    @Id
```

```

@Column(name = "idCampeonato", nullable = false)
private Integer idCampeonato;

@Column(name = "nomeCampeonato", nullable = false)
private String nomeCampeonato;

@OneToMany(cascade = CascadeType.ALL, mappedBy = "campeonato")
private Collection<Parametro> parametroCollection;

@OneToMany(cascade = CascadeType.ALL, mappedBy = "campeonatoId")
private Collection<Temporada> temporadaCollection;

/** Creates a new instance of Campeonato */
public Campeonato() {
}

/**
 * Cria uma nova instância de Campeonato com os valores especificados.
 * @param idCampeonato o idCampeonato do Campeonato
 */
public Campeonato(Integer idCampeonato) {
    this.idCampeonato = idCampeonato;
}

/**
 * Cria uma nova instância de Campeonato com os valores especificados.
 * @param idCampeonato o idCampeonato do Campeonato
 * @param nomeCampeonato o nomeCampeonato do Campeonato
 */
public Campeonato(Integer idCampeonato, String nomeCampeonato) {
    this.idCampeonato = idCampeonato;
    this.nomeCampeonato = nomeCampeonato;
}

/**
 * Define o idCampeonato deste Campeonato.
 * @return o idCampeonato
 */
public Integer getIdCampeonato() {
    return this.idCampeonato;
}

/**
 * Define o idCampeonato deste Campeonato para o valor especificado.
 * @param idCampeonato o novo idCampeonato
 */
public void setIdCampeonato(Integer idCampeonato) {
    this.idCampeonato = idCampeonato;
}

/**
 * Define o nomeCampeonato deste Campeonato.
 * @return o nomeCampeonato
 */
public String getNomeCampeonato() {
    return this.nomeCampeonato;
}

/**
 * Define o nomeCampeonato deste Campeonato para o valor especificado.
 * @param nomeCampeonato o novo nomeCampeonato
 */
public void setNomeCampeonato(String nomeCampeonato) {
    this.nomeCampeonato = nomeCampeonato;
}

```

```

/**
 * Define o parametroCollection deste Campeonato.
 * @return o parametroCollection
 */
public Collection<Parametro> getParametroCollection() {
    return this.parametroCollection;
}

/**
 * Define o parametroCollection deste Campeonato para o valor
 especificado.
 * @param parametroCollection o novo parametroCollection
 */
public void setParametroCollection(Collection<Parametro> parametroCollection) {
    this.parametroCollection = parametroCollection;
}

/**
 * Define o temporadaCollection deste Campeonato.
 * @return o temporadaCollection
 */
public Collection<Temporada> getTemporadaCollection() {
    return this.temporadaCollection;
}

/**
 * Define o temporadaCollection deste Campeonato para o valor
 especificado.
 * @param temporadaCollection o novo temporadaCollection
 */
public void setTemporadaCollection(Collection<Temporada> temporadaCollection) {
    this.temporadaCollection = temporadaCollection;
}

/**
 * Retorna um valor de código hash para o objeto. Esta implementação
 computa
 * um valor de código hash baseado nos campos id deste objeto.
 * @return um valor de código hash para este objeto.
 */
@Override
public int hashCode() {
    int hash = 0;
    hash += (this.idCampeonato != null ? this.idCampeonato.hashCode() : 0);
    return hash;
}

/**
 * Determina se outro objeto é igual a este Campeonato. O resultado é
 * <code>>true</code> se e somente se o argumento não for nulo e for um
 objeto Campeonato o qual
 * tem o mesmo valor para o campo id como este objeto.
 * @param object o objeto de referência com o qual comparar
 * @return <code>>true</code> se este objeto é o mesmo como o argumento;
 * <code>false</code> caso contrário.
 */
@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id
 fields are not set
    if (!(object instanceof Campeonato)) {
        return false;
    }
    Campeonato other = (Campeonato)object;
    if (this.idCampeonato != other.idCampeonato && (this.idCampeonato == null
 || !this.idCampeonato.equals(other.idCampeonato))) return false;
    return true;
}

```

```

    }

    /**
     * Retorna uma representação literal deste objeto. Esta implementação
    cria
     * uma representação baseada nos campos id.
     * @return uma representação literal deste objeto.
     */
    @Override
    public String toString() {
        return this.nomeCampeonato;
    }
}

```

11.34. Persistência - Clube.java

```

/*
 * Clube.java
 *
 * Created on 21 de Agosto de 2007, 18:26
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package persistencia;

import java.io.Serializable;
import java.util.Collection;
import java.util.Date;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * Classe de entidade Clube
 *
 * @author Carlos
 */
@Entity
@Table(name = "clube")
@NamedQueries( {
    @NamedQuery(name = "Clube.findByIdClube", query = "SELECT c FROM Clube
c WHERE c.idClube = :idClube"),
    @NamedQuery(name = "Clube.findByNomeClube", query = "SELECT c FROM
Clube c WHERE c.nomeClube = :nomeClube"),
    @NamedQuery(name = "Clube.findByUfClube", query = "SELECT c FROM Clube
c WHERE c.ufClube = :ufClube"),
    @NamedQuery(name = "Clube.findByDataFundacao", query = "SELECT c FROM
Clube c WHERE c.dataFundacao = :dataFundacao")
})

public class Clube implements Serializable {

    @Id

```

```

@Column(name = "idClube", nullable = false)
private Integer idClube;

@Column(name = "nomeClube", nullable = false)
private String nomeClube;

@Column(name = "ufClube", nullable = false)
private String ufClube;

@Column(name = "dataFundacao")
@Temporal(TemporalType.DATE)
private Date dataFundacao;

@OneToMany(cascade = CascadeType.ALL, mappedBy = "clube")
private Collection<Parametro> parametroCollection;

@OneToMany(cascade = CascadeType.ALL, mappedBy = "clubeMandanteId")
private Collection<Partida> partidaCollectionMandante;

@OneToMany(cascade = CascadeType.ALL, mappedBy = "clubeVisitanteId")
private Collection<Partida> partidaCollectionVisitante;

/** Creates a new instance of Clube */
public Clube() {
}

/**
 * Cria uma nova instância de Clube com os valores especificados.
 * @param idClube o idClube do Clube
 */
public Clube(Integer idClube) {
    this.idClube = idClube;
}

/**
 * Cria uma nova instância de Clube com os valores especificados.
 * @param idClube o idClube do Clube
 * @param nomeClube o nomeClube do Clube
 * @param ufClube o ufClube do Clube
 */
public Clube(Integer idClube, String nomeClube, String ufClube) {
    this.idClube = idClube;
    this.nomeClube = nomeClube;
    this.ufClube = ufClube;
}

/**
 * Define o idClube deste Clube.
 * @return o idClube
 */
public Integer getIdClube() {
    return this.idClube;
}

/**
 * Define o idClube deste Clube para o valor especificado.
 * @param idClube o novo idClube
 */
public void setIdClube(Integer idClube) {
    this.idClube = idClube;
}

/**
 * Define o nomeClube deste Clube.
 * @return o nomeClube
 */
public String getNomeClube() {

```

```

        return this.nomeClube;
    }

    /**
     * Define o nomeClube deste Clube para o valor especificado.
     * @param nomeClube o novo nomeClube
     */
    public void setNomeClube(String nomeClube) {
        this.nomeClube = nomeClube;
    }

    /**
     * Define o ufClube deste Clube.
     * @return o ufClube
     */
    public String getUfClube() {
        return this.ufClube;
    }

    /**
     * Define o ufClube deste Clube para o valor especificado.
     * @param ufClube o novo ufClube
     */
    public void setUfClube(String ufClube) {
        this.ufClube = ufClube;
    }

    /**
     * Define o dataFundacao deste Clube.
     * @return o dataFundacao
     */
    public Date getDataFundacao() {
        return this.dataFundacao;
    }

    /**
     * Define o dataFundacao deste Clube para o valor especificado.
     * @param dataFundacao o novo dataFundacao
     */
    public void setDataFundacao(Date dataFundacao) {
        this.dataFundacao = dataFundacao;
    }

    /**
     * Define o parametroCollection deste Clube.
     * @return o parametroCollection
     */
    public Collection<Parametro> getParametroCollection() {
        return this.parametroCollection;
    }

    /**
     * Define o parametroCollection deste Clube para o valor especificado.
     * @param parametroCollection o novo parametroCollection
     */
    public void setParametroCollection(Collection<Parametro> parametroCollection) {
        this.parametroCollection = parametroCollection;
    }

    /**
     * Define o partidaCollection deste Clube.
     * @return o partidaCollection
     */
    public Collection<Partida> getPartidaCollectionMandante() {
        return this.partidaCollectionMandante;
    }
}

```



```

/**
 * Define o partidaCollection deste Clube para o valor especificado.
 * @param partidaCollection o novo partidaCollection
 */
public void setPartidaCollectionMandante(Collection<Partida>
partidaCollectionMandante) {
    this.partidaCollectionMandante = partidaCollectionMandante;
}

/**
 * Define o partidaCollection1 deste Clube.
 * @return o partidaCollection1
 */
public Collection<Partida> getPartidaCollectionVisitante() {
    return this.partidaCollectionVisitante;
}

/**
 * Define o partidaCollection1 deste Clube para o valor especificado.
 * @param partidaCollection1 o novo partidaCollection1
 */
public void setPartidaCollectionVisitante(Collection<Partida>
partidaCollectionVisitante) {
    this.partidaCollectionVisitante = partidaCollectionVisitante;
}

/**
 * Retorna um valor de código hash para o objeto. Esta implementação
computa
 * um valor de código hash baseado nos campos id deste objeto.
 * @return um valor de código hash para este objeto.
 */
@Override
public int hashCode() {
    int hash = 0;
    hash += (this.idClube != null ? this.idClube.hashCode() : 0);
    return hash;
}

/**
 * Determina se outro objeto é igual a este Clube. O resultado é
 * <code>>true</code> se e somente se o argumento não for nulo e for um
objeto Clube o qual
 * tem o mesmo valor para o campo id como este objeto.
 * @param object o objeto de referência com o qual comparar
 * @return <code>true</code> se este objeto é o mesmo como o argumento;
 * <code>false</code> caso contrário.
 */
@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id
fields are not set
    if (!(object instanceof Clube)) {
        return false;
    }
    Clube other = (Clube)object;
    if (this.idClube != other.idClube && (this.idClube == null ||
!this.idClube.equals(other.idClube))) return false;
    return true;
}

/**
 * Retorna uma representação literal deste objeto. Esta implementação
cria
 * uma representação baseada nos campos id.
 * @return uma representação literal deste objeto.

```

```

    */
    @Override
    public String toString() {
        //return "persistencia.Clube[idClube=" + idClube + "]";
        return this.nomeClube;
    }
}

```

11.35. Persistência - Partida.java

```

*
* Partida.java
*
* Created on 21 de Agosto de 2007, 18:26
*
* To change this template, choose Tools | Template Manager
* and open the template in the editor.
*/

package persistencia;

import java.io.Serializable;
import java.math.BigDecimal;
import java.util.Collection;
import java.util.Date;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * Classe de entidade Partida
 *
 * @author Carlos
 */
@Entity
@Table(name = "partida")
@NamedQueries( {
    @NamedQuery(name = "Partida.findByIdPartida", query = "SELECT p FROM
Partida p WHERE p.idPartida = :idPartida"),
    @NamedQuery(name = "Partida.findByDataPartida", query = "SELECT p FROM
Partida p WHERE p.dataPartida = :dataPartida"),
    @NamedQuery(name = "Partida.countByDataPartida", query = "SELECT
COUNT(p) FROM Partida p WHERE p.dataPartida < :dataPartida"),
    @NamedQuery(name = "Partida.countByDataTemporada", query = "SELECT
COUNT(p) FROM Partida p WHERE p.dataPartida < :dataPartida AND p.temporadaId =
:temporada"),
    @NamedQuery(name = "Partida.findByTemporada", query = "SELECT p FROM
Partida p WHERE p.temporadaId = :temporada"),
    @NamedQuery(name = "Partida.countByTemporada", query = "SELECT
COUNT(p) FROM Partida p WHERE p.temporadaId = :temporada")
})
public class Partida implements Serializable {

    @Id

```

```

@Column(name = "idPartida", nullable = false)
private Integer idPartida;

@Column(name = "dataPartida")
@Temporal(TemporalType.DATE)
private Date dataPartida;

@Column(name = "horaPartida")
@Temporal(TemporalType.TIME)
private Date horaPartida;

@Column(name = "golsMandante")
private Integer golsMandante;

@Column(name = "golsVisitante")
private Integer golsVisitante;

@Column(name = "mediaMandante")
private BigDecimal mediaMandante;

@Column(name = "mediaVisitante")
private BigDecimal mediaVisitante;

@Column(name = "probMandante")
private BigDecimal probMandante;

@Column(name = "probEmpate")
private BigDecimal probEmpate;

@Column(name = "probVisitante")
private BigDecimal probVisitante;

@JoinColumn(name = "clubeMandanteId", referencedColumnName = "idClube")
@ManyToOne
private Clube clubeMandanteId;

@JoinColumn(name = "clubeVisitanteId", referencedColumnName = "idClube")
@ManyToOne
private Clube clubeVisitanteId;

@JoinColumn(name = "temporadaId", referencedColumnName = "idTemporada")
@ManyToOne
private Temporada temporadaId;

/** Creates a new instance of Partida */
public Partida() {
}

/**
 * Cria uma nova instância de Partida com os valores especificados.
 * @param idPartida o idPartida do Partida
 */
public Partida(Integer idPartida) {
    this.idPartida = idPartida;
}

/**
 * Define o idPartida deste Partida.
 * @return o idPartida
 */
public Integer getIdPartida() {
    return this.idPartida;
}

/**
 * Define o idPartida deste Partida para o valor especificado.

```

```

    * @param idPartida o novo idPartida
    */
    public void setIdPartida(Integer idPartida) {
        this.idPartida = idPartida;
    }

    /**
     * Define o dataPartida deste Partida.
     * @return o dataPartida
     */
    public Date getDataPartida() {
        return this.dataPartida;
    }

    /**
     * Define o dataPartida deste Partida para o valor especificado.
     * @param dataPartida o novo dataPartida
     */
    public void setDataPartida(Date dataPartida) {
        this.dataPartida = dataPartida;
    }

    /**
     * Define o horaPartida deste Partida.
     * @return o horaPartida
     */
    public Date getHoraPartida() {
        return this.horaPartida;
    }

    /**
     * Define o horaPartida deste Partida para o valor especificado.
     * @param horaPartida o novo horaPartida
     */
    public void setHoraPartida(Date horaPartida) {
        this.horaPartida = horaPartida;
    }

    /**
     * Define o golsMandante deste Partida.
     * @return o golsMandante
     */
    public Integer getGolsMandante() {
        return this.golsMandante;
    }

    /**
     * Define o golsMandante deste Partida para o valor especificado.
     * @param golsMandante o novo golsMandante
     */
    public void setGolsMandante(Integer golsMandante) {
        this.golsMandante = golsMandante;
    }

    /**
     * Define o golsVisitante deste Partida.
     * @return o golsVisitante
     */
    public Integer getGolsVisitante() {
        return this.golsVisitante;
    }

    /**
     * Define o golsVisitante deste Partida para o valor especificado.
     * @param golsVisitante o novo golsVisitante
     */
    public void setGolsVisitante(Integer golsVisitante) {

```

```

        this.golsVisitante = golsVisitante;
    }

    /**
     * Define o mediaMandante deste Partida.
     * @return o mediaMandante
     */
    public BigDecimal getMediaMandante() {
        return this.mediaMandante;
    }

    /**
     * Define o mediaMandante deste Partida para o valor especificado.
     * @param mediaMandante o novo mediaMandante
     */
    public void setMediaMandante(BigDecimal mediaMandante) {
        this.mediaMandante = mediaMandante;
    }

    /**
     * Define o mediaVisitante deste Partida.
     * @return o mediaVisitante
     */
    public BigDecimal getMediaVisitante() {
        return this.mediaVisitante;
    }

    /**
     * Define o mediaVisitante deste Partida para o valor especificado.
     * @param mediaVisitante o novo mediaVisitante
     */
    public void setMediaVisitante(BigDecimal mediaVisitante) {
        this.mediaVisitante = mediaVisitante;
    }

    /**
     * Define o probMandante deste Partida.
     * @return o probMandante
     */
    public BigDecimal getProbMandante() {
        return this.probMandante;
    }

    /**
     * Define o probMandante deste Partida para o valor especificado.
     * @param probMandante o novo probMandante
     */
    public void setProbMandante(BigDecimal probMandante) {
        this.probMandante = probMandante;
    }

    /**
     * Define o probEmpate deste Partida.
     * @return o probEmpate
     */
    public BigDecimal getProbEmpate() {
        return this.probEmpate;
    }

    /**
     * Define o probEmpate deste Partida para o valor especificado.
     * @param probEmpate o novo probEmpate
     */
    public void setProbEmpate(BigDecimal probEmpate) {
        this.probEmpate = probEmpate;
    }
}

```

```

/**
 * Define o probVisitante deste Partida.
 * @return o probVisitante
 */
public BigDecimal getProbVisitante() {
    return this.probVisitante;
}

/**
 * Define o probVisitante deste Partida para o valor especificado.
 * @param probVisitante o novo probVisitante
 */
public void setProbVisitante(BigDecimal probVisitante) {
    this.probVisitante = probVisitante;
}

/**
 * Define o clubeMandanteId deste Partida.
 * @return o clubeMandanteId
 */
public Clube getClubeMandanteId() {
    return this.clubeMandanteId;
}

/**
 * Define o clubeMandanteId deste Partida para o valor especificado.
 * @param clubeMandanteId o novo clubeMandanteId
 */
public void setClubeMandanteId(Clube clubeMandanteId) {
    this.clubeMandanteId = clubeMandanteId;
}

/**
 * Define o clubeVisitanteId deste Partida.
 * @return o clubeVisitanteId
 */
public Clube getClubeVisitanteId() {
    return this.clubeVisitanteId;
}

/**
 * Define o clubeVisitanteId deste Partida para o valor especificado.
 * @param clubeVisitanteId o novo clubeVisitanteId
 */
public void setClubeVisitanteId(Clube clubeVisitanteId) {
    this.clubeVisitanteId = clubeVisitanteId;
}

/**
 * Define o temporadaId deste Partida.
 * @return o temporadaId
 */
public Temporada getTemporadaId() {
    return this.temporadaId;
}

/**
 * Define o temporadaId deste Partida para o valor especificado.
 * @param temporadaId o novo temporadaId
 */
public void setTemporadaId(Temporada temporadaId) {
    this.temporadaId = temporadaId;
}

/**
 * Retorna um valor de código hash para o objeto. Esta implementação
computa

```

```

    * um valor de código hash baseado nos campos id deste objeto.
    * @return um valor de código hash para este objeto.
    */
    @Override
    public int hashCode() {
        int hash = 0;
        hash += (this.idPartida != null ? this.idPartida.hashCode() : 0);
        return hash;
    }

    /**
     * Determina se outro objeto é igual a este Partida. O resultado é
     * <code>true</code> se e somente se o argumento não for nulo e for um
objeto Partida o qual
     * tem o mesmo valor para o campo id como este objeto.
     * @param object o objeto de referência com o qual comparar
     * @return <code>true</code> se este objeto é o mesmo como o argumento;
     * <code>false</code> caso contrário.
     */
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof Partida)) {
            return false;
        }
        Partida other = (Partida)object;
        if (this.idPartida != other.idPartida && (this.idPartida == null ||
!this.idPartida.equals(other.idPartida))) return false;
        return true;
    }

    /**
     * Retorna uma representação literal deste objeto. Esta implementação
cria
     * uma representação baseada nos campos id.
     * @return uma representação literal deste objeto.
     */
    @Override
    public String toString() {
        return idPartida.toString();
    }
}

```

11.36. Persistência - Parametro.java

```

/*
 * Parametro.java
 *
 * Created on 21 de Agosto de 2007, 18:26
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package persistencia;

import java.io.Serializable;
import java.math.BigDecimal;
import javax.persistence.Column;
import javax.persistence.EmbeddedId;
import javax.persistence.Entity;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

```

```

import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

/**
 * Classe de entidade Parametro
 *
 * @author Carlos
 */
@Entity
@Table(name = "parametro")
@NamedQueries( {
    @NamedQuery(name = "Parametro.findByCampeonatoId", query = "SELECT p
FROM Parametro p WHERE p.parametroPK.campeonatoId = :campeonatoId"),
    @NamedQuery(name = "Parametro.findByClubeId", query = "SELECT p FROM
Parametro p WHERE p.parametroPK.clubeId = :clubeId"),
    @NamedQuery(name = "Parametro.findByGolsContraMandante", query =
"SELECT p FROM Parametro p WHERE p.golsContraMandante = :golsContraMandante"),
    @NamedQuery(name = "Parametro.findByGolsContraVisitante", query =
"SELECT p FROM Parametro p WHERE p.golsContraVisitante =
:golsContraVisitante"),
    @NamedQuery(name = "Parametro.findByGolsFavorMandante", query =
"SELECT p FROM Parametro p WHERE p.golsFavorMandante = :golsFavorMandante"),
    @NamedQuery(name = "Parametro.findByGolsFavorVisitante", query =
"SELECT p FROM Parametro p WHERE p.golsFavorVisitante = :golsFavorVisitante"),
    @NamedQuery(name = "Parametro.findByCampeonatoIdAndClubeId", query =
"SELECT p FROM Parametro p WHERE p.parametroPK.campeonatoId = :campeonatoId
AND p.parametroPK.clubeId = :clubeId")
})
public class Parametro implements Serializable {

    /**
     * Campo de chave primária embutido
     */
    @EmbeddedId
    protected ParametroPK parametroPK;

    @Column(name = "golsContraMandante")
    private BigDecimal golsContraMandante;

    @Column(name = "golsContraVisitante")
    private BigDecimal golsContraVisitante;

    @Column(name = "golsFavorMandante")
    private BigDecimal golsFavorMandante;

    @Column(name = "golsFavorVisitante")
    private BigDecimal golsFavorVisitante;

    @JoinColumn(name = "campeonatoId", referencedColumnName = "idCampeonato",
insertable = false, updatable = false)
    @ManyToOne
    private Campeonato campeonato;

    @JoinColumn(name = "clubeId", referencedColumnName = "idClube", insertable
= false, updatable = false)
    @ManyToOne
    private Clube clube;

    /** Creates a new instance of Parametro */
    public Parametro() {
    }

    /**
     * Cria uma nova instância de Parametro com os valores especificados.
     * @param parametroPK o parametroPK do Parametro
     */
}

```



```

public Parametro(ParametroPK parametroPK) {
    this.parametroPK = parametroPK;
}

/**
 * Cria uma nova instância de ParametroPK com os valores especificados.
 * @param clubeId o clubeId do ParametroPK
 * @param campeonatoId o campeonatoId do ParametroPK
 */
public Parametro(int clubeId, int campeonatoId) {
    this.parametroPK = new ParametroPK(clubeId, campeonatoId);
}

/**
 * Define o parametroPK deste Parametro.
 * @return o parametroPK
 */
public ParametroPK getParametroPK() {
    return this.parametroPK;
}

/**
 * Define o parametroPK deste Parametro para o valor especificado.
 * @param parametroPK o novo parametroPK
 */
public void setParametroPK(ParametroPK parametroPK) {
    this.parametroPK = parametroPK;
}

/**
 * Define o golsContraMandante deste Parametro.
 * @return o golsContraMandante
 */
public BigDecimal getGolsContraMandante() {
    return this.golsContraMandante;
}

/**
 * Define o golsContraMandante deste Parametro para o valor especificado.
 * @param golsContraMandante o novo golsContraMandante
 */
public void setGolsContraMandante(BigDecimal golsContraMandante) {
    this.golsContraMandante = golsContraMandante;
}

/**
 * Define o golsContraVisitante deste Parametro.
 * @return o golsContraVisitante
 */
public BigDecimal getGolsContraVisitante() {
    return this.golsContraVisitante;
}

/**
 * Define o golsContraVisitante deste Parametro para o valor especificado.
 * @param golsContraVisitante o novo golsContraVisitante
 */
public void setGolsContraVisitante(BigDecimal golsContraVisitante) {
    this.golsContraVisitante = golsContraVisitante;
}

/**
 * Define o golsFavorMandante deste Parametro.
 * @return o golsFavorMandante
 */
public BigDecimal getGolsFavorMandante() {
    return this.golsFavorMandante;
}

```

```

}

/**
 * Define o golsFavorMandante deste Parametro para o valor especificado.
 * @param golsFavorMandante o novo golsFavorMandante
 */
public void setGolsFavorMandante(BigDecimal golsFavorMandante) {
    this.golsFavorMandante = golsFavorMandante;
}

/**
 * Define o golsFavorVisitante deste Parametro.
 * @return o golsFavorVisitante
 */
public BigDecimal getGolsFavorVisitante() {
    return this.golsFavorVisitante;
}

/**
 * Define o golsFavorVisitante deste Parametro para o valor especificado.
 * @param golsFavorVisitante o novo golsFavorVisitante
 */
public void setGolsFavorVisitante(BigDecimal golsFavorVisitante) {
    this.golsFavorVisitante = golsFavorVisitante;
}

/**
 * Define o campeonato deste Parametro.
 * @return o campeonato
 */
public Campeonato getCampeonato() {
    return this.campeonato;
}

/**
 * Define o campeonato deste Parametro para o valor especificado.
 * @param campeonato o novo campeonato
 */
public void setCampeonato(Campeonato campeonato) {
    this.campeonato = campeonato;
}

/**
 * Define o clube deste Parametro.
 * @return o clube
 */
public Clube getClube() {
    return this.clube;
}

/**
 * Define o clube deste Parametro para o valor especificado.
 * @param clube o novo clube
 */
public void setClube(Clube clube) {
    this.clube = clube;
}

/**
 * Retorna um valor de código hash para o objeto. Esta implementação
computa
 * um valor de código hash baseado nos campos id deste objeto.
 * @return um valor de código hash para este objeto.
 */
@Override
public int hashCode() {
    int hash = 0;

```

```

        hash += (this.parametroPK != null ? this.parametroPK.hashCode() : 0);
        return hash;
    }

    /**
     * Determina se outro objeto é igual a este Parametro. O resultado é
     * <code>>true</code> se e somente se o argumento não for nulo e for um
     objeto Parametro o qual
     * tem o mesmo valor para o campo id como este objeto.
     * @param object o objeto de referência com o qual comparar
     * @return <code>true</code> se este objeto é o mesmo como o argumento;
     * <code>false</code> caso contrário.
     */
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
        fields are not set
        if (!(object instanceof Parametro)) {
            return false;
        }
        Parametro other = (Parametro)object;
        if (this.parametroPK != other.parametroPK && (this.parametroPK == null ||
!this.parametroPK.equals(other.parametroPK))) return false;
        return true;
    }

    /**
     * Retorna uma representação literal deste objeto. Esta implementação
     cria
     * uma representação baseada nos campos id.
     * @return uma representação literal deste objeto.
     */
    @Override
    public String toString() {
        return this.campeonato.getIdCampeonato().toString() + "-" +
this.clube.getIdClube().toString();
    }
}

```

11.37. Persistência - ParametroPK.java

```

/*
 * ParametroPK.java
 *
 * Created on 21 de Agosto de 2007, 18:26
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */
package persistencia;

import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Embeddable;

/**
 * Classe de Chave Primária ParametroPK para classe de entidade Parametro
 *
 * @author Carlos
 */
@Embeddable
public class ParametroPK implements Serializable {

```

```

@Column(name = "campeonatoId", nullable = false)
private int campeonatoId;

@Column(name = "clubeId", nullable = false)
private int clubeId;

/** Creates a new instance of ParametroPK */
public ParametroPK() {
}

/**
 * Cria uma nova instância de ParametroPK com os valores especificados.
 * @param clubeId o clubeId do ParametroPK
 * @param campeonatoId o campeonatoId do ParametroPK
 */
public ParametroPK(int clubeId, int campeonatoId) {
    this.clubeId = clubeId;
    this.campeonatoId = campeonatoId;
}

/**
 * Define o campeonatoId deste ParametroPK.
 * @return o campeonatoId
 */
public int getCampeonatoId() {
    return this.campeonatoId;
}

/**
 * Define o campeonatoId deste ParametroPK para o valor especificado.
 * @param campeonatoId o novo campeonatoId
 */
public void setCampeonatoId(int campeonatoId) {
    this.campeonatoId = campeonatoId;
}

/**
 * Define o clubeId deste ParametroPK.
 * @return o clubeId
 */
public int getClubeId() {
    return this.clubeId;
}

/**
 * Define o clubeId deste ParametroPK para o valor especificado.
 * @param clubeId o novo clubeId
 */
public void setClubeId(int clubeId) {
    this.clubeId = clubeId;
}

/**
 * Retorna um valor de código hash para o objeto. Esta implementação
computa
 * um valor de código hash baseado nos campos id deste objeto.
 * @return um valor de código hash para este objeto.
 */
@Override
public int hashCode() {
    int hash = 0;
    hash += (int)clubeId;
    hash += (int)campeonatoId;
    return hash;
}

```

```

    }

    /**
     * Determina se outro objeto é igual a este ParametroPK. O resultado é
     * <code>>true</code> se e somente se o argumento não for nulo e for um
objeto ParametroPK o qual
     * tem o mesmo valor para o campo id como este objeto.
     * @param object o objeto de referência com o qual comparar
     * @return <code>true</code> se este objeto é o mesmo como o argumento;
     * <code>false</code> caso contrário.
     */
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof ParametroPK)) {
            return false;
        }
        ParametroPK other = (ParametroPK)object;
        if (this.clubeId != other.clubeId) return false;
        if (this.campeonatoId != other.campeonatoId) return false;
        return true;
    }

    /**
     * Retorna uma representação literal deste objeto. Esta implementação
cria
     * uma representação baseada nos campos id.
     * @return uma representação literal deste objeto.
     */
    @Override
    public String toString() {
        return "" + campeonatoId + "-" + clubeId;
    }
}

```

11.38. Persistência - Temporada.java

```

/*
 * Temporada.java
 *
 * Created on 21 de Agosto de 2007, 18:26
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package persistencia;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;

```

```

import javax.persistence.Table;

/**
 * Classe de entidade Temporada
 *
 * @author Carlos
 */
@Entity
@Table(name = "temporada")
@NamedQueries( {
    @NamedQuery(name = "Temporada.findByIdTemporada", query = "SELECT t
FROM Temporada t WHERE t.idTemporada = :idTemporada"),
    @NamedQuery(name = "Temporada.findByNameTemporada", query = "SELECT t
FROM Temporada t WHERE t.nomeTemporada = :nomeTemporada")
})
public class Temporada implements Serializable {

    @Id
    @Column(name = "idTemporada", nullable = false)
    private Integer idTemporada;

    @Column(name = "nomeTemporada", nullable = false)
    private String nomeTemporada;

    @OneToMany(cascade = CascadeType.ALL, mappedBy = "temporadaId")
    private Collection<Partida> partidaCollection;

    @JoinColumn(name = "campeonatoId", referencedColumnName = "idCampeonato")
    @ManyToOne
    private Campeonato campeonatoId;

    /** Creates a new instance of Temporada */
    public Temporada() {
    }

    /**
     * Cria uma nova instância de Temporada com os valores especificados.
     * @param idTemporada o idTemporada do Temporada
     */
    public Temporada(Integer idTemporada) {
        this.idTemporada = idTemporada;
    }

    /**
     * Cria uma nova instância de Temporada com os valores especificados.
     * @param idTemporada o idTemporada do Temporada
     * @param nomeTemporada o nomeTemporada do Temporada
     */
    public Temporada(Integer idTemporada, String nomeTemporada) {
        this.idTemporada = idTemporada;
        this.nomeTemporada = nomeTemporada;
    }

    /**
     * Define o idTemporada deste Temporada.
     * @return o idTemporada
     */
    public Integer getIdTemporada() {
        return this.idTemporada;
    }

    /**
     * Define o idTemporada deste Temporada para o valor especificado.
     * @param idTemporada o novo idTemporada
     */
    public void setIdTemporada(Integer idTemporada) {
        this.idTemporada = idTemporada;
    }
}

```

```

}

/**
 * Define o nomeTemporada deste Temporada.
 * @return o nomeTemporada
 */
public String getNomeTemporada() {
    return this.nomeTemporada;
}

/**
 * Define o nomeTemporada deste Temporada para o valor especificado.
 * @param nomeTemporada o novo nomeTemporada
 */
public void setNomeTemporada(String nomeTemporada) {
    this.nomeTemporada = nomeTemporada;
}

/**
 * Define o partidaCollection deste Temporada.
 * @return o partidaCollection
 */
public Collection<Partida> getPartidaCollection() {
    return this.partidaCollection;
}

/**
 * Define o partidaCollection deste Temporada para o valor especificado.
 * @param partidaCollection o novo partidaCollection
 */
public void setPartidaCollection(Collection<Partida> partidaCollection) {
    this.partidaCollection = partidaCollection;
}

/**
 * Define o campeonatoId deste Temporada.
 * @return o campeonatoId
 */
public Campeonato getCampeonatoId() {
    return this.campeonatoId;
}

/**
 * Define o campeonatoId deste Temporada para o valor especificado.
 * @param campeonatoId o novo campeonatoId
 */
public void setCampeonatoId(Campeonato campeonatoId) {
    this.campeonatoId = campeonatoId;
}

/**
 * Retorna um valor de código hash para o objeto. Esta implementação
computa
 * um valor de código hash baseado nos campos id deste objeto.
 * @return um valor de código hash para este objeto.
 */
@Override
public int hashCode() {
    int hash = 0;
    hash += (this.idTemporada != null ? this.idTemporada.hashCode() : 0);
    return hash;
}

/**
 * Determina se outro objeto é igual a este Temporada. O resultado é
 * <code>true</code> se e somente se o argumento não for nulo e for um
objeto Temporada o qual

```

```

    * tem o mesmo valor para o campo id como este objeto.
    * @param object o objeto de referência com o qual comparar
    * @return <code>true</code> se este objeto é o mesmo como o argumento;
    * <code>false</code> caso contrário.
    */
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id
fields are not set
        if (!(object instanceof Temporada)) {
            return false;
        }
        Temporada other = (Temporada)object;
        if (this.idTemporada != other.idTemporada && (this.idTemporada == null ||
!this.idTemporada.equals(other.idTemporada))) return false;
        return true;
    }

    /**
    * Retorna uma representação literal deste objeto. Esta implementação
cria
    * uma representação baseada nos campos id.
    * @return uma representação literal deste objeto.
    */
    @Override
    public String toString() {
        //return "persistencia.Temporada[idTemporada=" + idTemporada + "];
        return this.nomeTemporada;
    }
}

```

11.39. Admin - CampeonatoController.java

```

/*
 * CampeonatoController.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import java.util.ArrayList;
import java.util.Collection;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.Query;
import persistencia.Campeonato;

/**
 *
 * @author Carlos
 */
public class CampeonatoController {

    /** Creates a new instance of CampeonatoController */
    public CampeonatoController() {
        emf = Persistence.createEntityManagerFactory("chutegolPU");
    }
}

```



```

}

private Campeonato campeonato;

private DataModel model;

private EntityManagerFactory emf;

private EntityManager getEntityManager() {
    return emf.createEntityManager();
}

private int batchSize = 20;

private int firstItem = 0;

public Campeonato getCampeonato() {
    return campeonato;
}

public void setCampeonato(Campeonato campeonato) {
    this.campeonato = campeonato;
}

public DataModel getDetailCampeonatos() {
    return model;
}

public void setDetailCampeonatos(Collection<Campeonato> m) {
    model = new ListDataModel(new ArrayList(m));
}

public String createSetup() {
    this.campeonato = new Campeonato();
    return "campeonatoAdmin_create";
}

public String create() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        em.persist(campeonato);
        em.getTransaction().commit();
        addSuccessMessage("Campeonato was successfully created.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "campeonatoAdmin_list";
}

public String detailSetup() {
    setCampeonatoFromRequestParam();
    return "campeonatoAdmin_detail";
}

public String editSetup() {
    setCampeonatoFromRequestParam();
    return "campeonatoAdmin_edit";
}

```

```

}

public String edit() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        campeonato = em.merge(campeonato);
        em.getTransaction().commit();
        addSuccessMessage("Campeonato was successfully updated.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "campeonatoAdmin_list";
}

public String destroy() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        Campeonato campeonato = getCampeonatoFromRequestParam();
        campeonato = em.merge(campeonato);
        em.remove(campeonato);
        em.getTransaction().commit();
        addSuccessMessage("Campeonato was successfully deleted.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "campeonatoAdmin_list";
}

public Campeonato getCampeonatoFromRequestParam() {
    EntityManager em = getEntityManager();
    try {
        Campeonato o = (Campeonato) model.getRowData();
        o = em.merge(o);
        return o;
    } finally {
        em.close();
    }
}

public void setCampeonatoFromRequestParam() {
    Campeonato campeonato = getCampeonatoFromRequestParam();
    setCampeonato(campeonato);
}

public DataModel getCampeonatos() {
    EntityManager em = getEntityManager();
    try {
        Query q = em.createQuery("select object(o) from Campeonato as o");
    }
}

```

```

        q.setMaxResults(batchSize);
        q.setFirstResult(firstItem);
        model = new ListDataModel(q.getResultList());
        return model;
    } finally {
        em.close();
    }
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Campeonato findCampeonato(Integer id) {
    EntityManager em = getEntityManager();
    try{
        Campeonato o = (Campeonato) em.find(Campeonato.class, id);
        return o;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    try{
        int count = ((Long) em.createQuery("select count(o) from Campeonato
as o").getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

public int getBatchSize() {
    return batchSize;
}

public String next() {
    if (firstItem + batchSize < getItemCount()) {
        firstItem += batchSize;
    }
    return "campeonatoAdmin_list";
}

public String prev() {

```

```

        firstItem -= batchSize;
        if (firstItem < 0) {
            firstItem = 0;
        }
        return "campeonatoAdmin_list";
    }
}

```

11.40. Admin - CampeonatoConverter.java

```

/*
 * CampeonatoConverter.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Campeonato;

/**
 *
 * @author Carlos
 */
public class CampeonatoConverter implements Converter {

    /** Creates a new instance of CampeonatoConverter */
    public CampeonatoConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uComponent,
String string) {
        if (string == null) {
            return null;
        }
        Integer id = new Integer(string);
        CampeonatoController controller = (CampeonatoController)
facesContext.getApplication().getVariableResolver().resolveVariable(
            facesContext, "campeonatoAdmin");

        return controller.findCampeonato(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if (object instanceof Campeonato) {
            Campeonato o = (Campeonato) object;
            return "" + o.getIdCampeonato();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Campeonato");
        }
    }
}

```

11.41. Admin - ClubeControler.java

```
/*
 * ClubeController.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import java.util.ArrayList;
import java.util.Collection;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.Query;
import persistencia.Clube;

/**
 *
 * @author Carlos
 */
public class ClubeController {

    /** Creates a new instance of ClubeController */
    public ClubeController() {
        emf = Persistence.createEntityManagerFactory("chutegolPU");
    }

    private Clube clube;

    private DataModel model;

    private EntityManagerFactory emf;

    private EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    private int batchSize = 10;

    private int firstItem = 0;

    public Clube getClube() {
        return clube;
    }

    public void setClube(Clube clube) {
        this.clube = clube;
    }

    public DataModel getDetailClubes() {
        return model;
    }
}
```

```

public void setDetailClubes(Collection<Clube> m) {
    model = new ListDataModel(new ArrayList(m));
}

public String createSetup() {
    this.clube = new Clube();
    return "clubeAdmin_create";
}

public String create() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        em.persist(clube);
        em.getTransaction().commit();
        addSuccessMessage("Clube was successfully created.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "clubeAdmin_list";
}

public String detailSetup() {
    setClubeFromRequestParam();
    return "clubeAdmin_detail";
}

public String editSetup() {
    setClubeFromRequestParam();
    return "clubeAdmin_edit";
}

public String edit() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        clube = em.merge(clube);
        em.getTransaction().commit();
        addSuccessMessage("Clube was successfully updated.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "clubeAdmin_list";
}

public String destroy() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        Clube clube = getClubeFromRequestParam();

```

```

        clube = em.merge(clube);
        em.remove(clube);
        em.getTransaction().commit();
        addSuccessMessage("Clube was successfully deleted.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "clubeAdmin_list";
}

public Clube getClubeFromRequestParam() {
    EntityManager em = getEntityManager();
    try{
        Clube o = (Clube) model.getRowData();
        o = em.merge(o);
        return o;
    } finally {
        em.close();
    }
}

public void setClubeFromRequestParam() {
    Clube clube = getClubeFromRequestParam();
    setClube(clube);
}

public DataModel getClubes() {
    EntityManager em = getEntityManager();
    try{
        Query q = em.createQuery("select object(o) from Clube as o order by
o.nomeClube");
        q.setMaxResults(batchSize);
        q.setFirstResult(firstItem);
        model = new ListDataModel(q.getResultList());
        return model;
    } finally {
        em.close();
    }
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Clube findClube(Integer id) {
    EntityManager em = getEntityManager();
    try{

```

```

        Clube o = (Clube) em.find(Clube.class, id);
        return o;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    try{
        int count = ((Long) em.createQuery("select count(o) from Clube as
o").getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

public int getBatchSize() {
    return batchSize;
}

public String next() {
    if (firstItem + batchSize < getItemCount()) {
        firstItem += batchSize;
    }
    return "clubeAdmin_list";
}

public String prev() {
    firstItem -= batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "clubeAdmin_list";
}
}
}

```

11.42. Admin - ClubeConverter.java

```

/*
 * ClubeConverter.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Clube;

```



```

/**
 *
 * @author Carlos
 */
public class ClubeConverter implements Converter {

    /** Creates a new instance of ClubeConverter */
    public ClubeConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uComponent,
String string) {
        if (string == null) {
            return null;
        }
        Integer id = new Integer(string);
        ClubeController controller = (ClubeController)
facesContext.getApplication().getVariableResolver().resolveVariable(
        facesContext, "clubeAdmin");

        return controller.findClube(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if(object instanceof Clube) {
            Clube o = (Clube) object;
            return "" + o.getIdClube();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Clube");
        }
    }
}

```

11.43. Admin - ParametroController.java

```

/*
 * ParametroController.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;
import javax.faces.model.SelectItem;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.Query;

```

```

import persistencia.Campeonato;
import persistencia.Clube;
import persistencia.Parametro;
import persistencia.ParametroPK;

/**
 *
 * @author Carlos
 */
public class ParametroController {

    /** Creates a new instance of ParametroController */
    public ParametroController() {
        emf = Persistence.createEntityManagerFactory("chutegolPU");
    }

    private Parametro parametro;

    private DataModel model;

    private EntityManagerFactory emf;

    private EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    private int batchSize = 20;

    private int firstItem = 0;

    public Parametro getParametro() {
        return parametro;
    }

    public void setParametro(Parametro parametro) {
        this.parametro = parametro;
    }

    public DataModel getDetailParams() {
        return model;
    }

    public void setDetailParams(Collection<Parametro> m) {
        model = new ListDataModel(new ArrayList(m));
    }

    public String createSetup() {
        this.parametro = new Parametro();
        return "parametroAdmin_create";
    }

    public String create() {
        EntityManager em = getEntityManager();
        try {
            em.getTransaction().begin();
            em.persist(parametro);
            em.getTransaction().commit();
            addSuccessMessage("Parametro was successfully created.");
        } catch (Exception ex) {
            try {
                addErrorMessage(ex.getLocalizedMessage());
                em.getTransaction().rollback();
            } catch (Exception e) {
                addErrorMessage(e.getLocalizedMessage());
            }
        }
    }
}

```

```

    } finally {
        em.close();
    }
    return "parametroAdmin_list";
}

public String detailSetup() {
    setParametroFromRequestParam();
    return "parametroAdmin_detail";
}

public String editSetup() {
    setParametroFromRequestParam();
    return "parametro_edit";
}

public String edit() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        parametro = em.merge(parametro);
        em.getTransaction().commit();
        addSuccessMessage("Parametro was successfully updated.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "parametroAdmin_list";
}

public String destroy() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        Parametro parametro = getParametroFromRequestParam();
        parametro = em.merge(parametro);
        em.remove(parametro);
        em.getTransaction().commit();
        addSuccessMessage("Parametro was successfully deleted.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "parametroAdmin_list";
}

public Parametro getParametroFromRequestParam() {
    EntityManager em = getEntityManager();
    try{
        Parametro o = (Parametro) model.getRowData();
        o = em.merge(o);
        return o;
    }
}

```

```

    } finally {
        em.close();
    }
}

public void setParametroFromRequestParam() {
    Parametro parametro = getParametroFromRequestParam();
    setParametro(parametro);
}

public DataModel getParametros() {
    EntityManager em = getEntityManager();
    try{
        Query q = em.createQuery("select object(o) from Parametro as o");
        q.setMaxResults(batchSize);
        q.setFirstResult(firstItem);
        model = new ListDataModel(q.getResultList());
        return model;
    } finally {
        em.close();
    }
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Parametro findParametro(ParametroPK id) {
    EntityManager em = getEntityManager();
    try{
        Parametro o = (Parametro) em.find(Parametro.class, id);
        return o;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getCampeonatos() {
    EntityManager em = getEntityManager();
    try{
        List <Campeonato> l = (List <Campeonato>) em.createQuery("select o
from Campeonato as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Campeonato x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getClubes() {

```

```

    EntityManager em = getEntityManager();
    try{
        List <Clube> l = (List <Clube>) em.createQuery("select o from Clube
as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Clube x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    try{
        int count = ((Long) em.createQuery("select count(o) from Parametro as
o").getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

public int getBatchSize() {
    return batchSize;
}

public String next() {
    if (firstItem + batchSize < getItemCount()) {
        firstItem += batchSize;
    }
    return "parametroAdmin_list";
}

public String prev() {
    firstItem -= batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "parametroAdmin_list";
}
}

```

11.44. Admin - ParametroConverter.java

```

/*
 * ParametroConverter.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.

```

```

*/

package chutegol.admin;

import java.util.StringTokenizer;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Parametro;

/**
 *
 * @author Carlos
 */
public class ParametroConverter implements Converter {

    /** Creates a new instance of ParametroConverter */
    public ParametroConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uIComponent,
String string) {
        if (string == null) {
            return null;
        }
        persistencia.ParametroPK id = new persistencia.ParametroPK();
        StringTokenizer idTokens = new StringTokenizer(string, ";");
        String params[] = new String[2];
        int i = 0;
        while(idTokens.hasMoreTokens()) {
            params[i++] = idTokens.nextToken();
        }
        if (i != 2) {
            throw new IllegalArgumentException("Expected format of parameter string
is a set of 2 IDs delimited by ;");
        }
        id.setCampeonatoId(Integer.parseInt(params[0]));
        id.setClubeId(Integer.parseInt(params[1]));
        ParametroController controller = (ParametroController)
facesContext.getApplication().getVariableResolver().resolveVariable(
            facesContext, "parametroAdmin");

        return controller.findParametro(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uIComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if(object instanceof Parametro) {
            Parametro o = (Parametro) object;
            return o.getParametroPK().getCampeonatoId() + ";" +
o.getParametroPK().getClubeId();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Parametro");
        }
    }
}

```

11.45. Admin - ParametroPKConverter.java

```
/*
 * ParametroPKConverter.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import java.util.StringTokenizer;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.ParametroPK;

/**
 *
 * @author Carlos
 */
public class ParametroPKConverter implements Converter {

    /** Creates a new instance of ParametroPKConverter */
    public ParametroPKConverter() {

    }

    public Object getAsObject(FacesContext facesContext, UIComponent uIComponent,
String string) {
        if (string == null) {
            return null;
        }
        persistencia.ParametroPK id = new persistencia.ParametroPK();
        StringTokenizer idTokens = new StringTokenizer(string, ";");
        String params[] = new String[2];
        int i = 0;
        while(idTokens.hasMoreTokens()) {
            params[i++] = idTokens.nextToken();
        }
        if (i != 2) {
            throw new IllegalArgumentException("Expected format of parameter string
is a set of 2 IDs delimited by ;");
        }
        id.setCampeonatoId(Integer.parseInt(params[0]));
        id.setClubeId(Integer.parseInt(params[1]));
        return id;
    }

    public String getAsString(FacesContext facesContext, UIComponent uIComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if(object instanceof ParametroPK) {
            ParametroPK o = (ParametroPK) object;
            return o.getCampeonatoId() + ";" + o.getClubeId();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: ParametroPK");
        }
    }
}
}
```

11.46. Admin - PartidaController.java

```
/*
 * PartidaController.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import chutegol.util.ProbComparator;
import chutegol.util.Probabilidad;
import java.math.BigDecimal;
import java.math.MathContext;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;
import javax.faces.model.SelectItem;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.NoResultException;
import javax.persistence.Persistence;
import javax.persistence.Query;
import persistencia.Clube;
import persistencia.Parametro;
import persistencia.ParametroPK;
import persistencia.Partida;
import persistencia.Temporada;

/**
 *
 * @author Carlos
 */
public class PartidaController {

    /** Creates a new instance of PartidaController */
    public PartidaController() {
        emf = Persistence.createEntityManagerFactory("chutegolPU");
    }

    private Partida partida;
    private Temporada temporada;

    private DataModel model;

    private EntityManagerFactory emf;

    private EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    private int batchSize = 10;

    private int firstItem = 0;
}
```



```

public Partida getPartida() {
    return partida;
}

public void setPartida(Partida partida) {
    this.partida = partida;
}

public Temporada getTemporada() {
    return temporada;
}

public void setTemporada(Temporada temporada) {
    this.temporada = temporada;
}

public DataModel getDetailPartidas() {
    return model;
}

public void setDetailPartidas(Collection<Partida> m) {
    model = new ListDataModel(new ArrayList(m));
}

public String createSetup() {
    this.partida = new Partida();
    return "partidaAdmin_create";
}

public String create() {
    EntityManager em = getEntityManager();
    try {
        em.getTransaction().begin();
        em.persist(partida);
        em.getTransaction().commit();
        addSuccessMessage("Partida criada com sucesso.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "partidaAdmin_list";
}

public String detailSetup() {
    setPartidaFromRequestParam();
    if (partida.getMediaMandante() == null) {
        addErrorMessage("Parâmetros ainda não calculados para a partida");
        return "partidaAdmin_list";
    } else return "partidaAdmin_detail";
}

public String editSetup() {
    setPartidaFromRequestParam();
    return "partidaAdmin_edit";
}

public String edit() {
    this.atualizaParametros();
}

```

```

EntityManager em = getEntityManager();
try {
    em.getTransaction() .begin();
    partida = em.merge(partida);
    em.getTransaction() .commit();
    addSuccessMessage("Partida atualizada com sucesso.");
} catch (Exception ex) {
    try {
        addErrorMessage(ex.getLocalizedMessage());
        em.getTransaction() .rollback();
    } catch (Exception e) {
        addErrorMessage(e.getLocalizedMessage());
    }
} finally {
    em.close();
}
return "partidaAdmin_list";
}

public String destroy() {
EntityManager em = getEntityManager();
try {
    em.getTransaction() .begin();
    Partida partida = getPartidaFromRequestParam();
    partida = em.merge(partida);
    em.remove(partida);
    em.getTransaction() .commit();
    addSuccessMessage("Partida deletada com sucesso.");
} catch (Exception ex) {
    try {
        addErrorMessage(ex.getLocalizedMessage());
        em.getTransaction() .rollback();
    } catch (Exception e) {
        addErrorMessage(e.getLocalizedMessage());
    }
} finally {
    em.close();
}
return "partidaAdmin_list";
}

public Partida getPartidaFromRequestParam() {
EntityManager em = getEntityManager();
try{
    Partida o = (Partida) model.getRowData();
    o = em.merge(o);
    return o;
} finally {
    em.close();
}
}

public void setPartidaFromRequestParam() {
Partida partida = getPartidaFromRequestParam();
setPartida(partida);
}

public DataModel getPartidas() {
EntityManager em = getEntityManager();
try{
    Query q = em.createQuery("select object(o) from Partida as o");
    q.setMaxResults(batchSize);
    q.setFirstResult(firstItem);
    model = new ListDataModel(q.getResultList());
}
}

```

```

        return model;
    } finally {
        em.close();
    }
}

public DataModel getPartidasTemporada() {
    EntityManager em = getEntityManager();
    if (temporada == null) return this.getPartidas();
    else{
        try{
            int temporadaId = this.temporada.getIdTemporada();
            Query q = em.createNamedQuery("Partida.findByTemporada");
            q.setParameter("temporada", temporada);
            q.setMaxResults(batchSize);
            q.setFirstResult(firstItem);
            model = new ListDataModel(q.getResultList());
            return model;
        } finally {
            em.close();
        }
    }
}

public String partidasTemp(){
    return "partidaAdmin_list";
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Partida findPartida(Integer id) {
    EntityManager em = getEntityManager();
    try{
        Partida o = (Partida) em.find(Partida.class, id);
        return o;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getClubeMandanteIds() {
    EntityManager em = getEntityManager();
    try{
        List <Clube> l = (List <Clube>) em.createQuery("select o from Clube
as o order by o.nomeClube").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Clube x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {

```

```

        em.close();
    }
}

public javax.faces.model.SelectItem[] getClubeVisitanteIds() {
    EntityManager em = getEntityManager();
    try{
        List <Clube> l = (List <Clube>) em.createQuery("select o from Clube
as o order by o.nomeClube").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Clube x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getTemporadaIds() {
    EntityManager em = getEntityManager();
    try{
        List <Temporada> l = (List <Temporada>) em.createQuery("select o
from Temporada as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Temporada x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    if (temporada == null){
        try{
            int count = ((Long) em.createQuery("select count(o) from Partida
as o").getSingleResult()).intValue();
            return count;
        } finally {
            em.close();
        }
    } else{
        try{
            Query q = em.createNamedQuery("Partida.countByTemporada");
            q.setParameter("temporada", temporada);
            int count = ((Long) q.getSingleResult()).intValue();
            return count;
        } finally{
            em.close();
        }
    }
}

public int getItemCountDate() {
    EntityManager em = getEntityManager();
    Date hoje = new Date();
    if (temporada == null){
        try{
            Query q = em.createNamedQuery("Partida.countByDataPartida");

```

```

        q.setParameter("dataPartida", hoje);
        int count = ((Long) q.getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
} else {
    try {
        Query q = em.createNamedQuery("Partida.countByDataTemporada");
        q.setParameter("dataPartida", hoje);
        q.setParameter("temporada", temporada);
        int count = ((Long) q.getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
}
}

public String hoje() {
    firstItem = getItemCountDate() + 1;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "partidaAdmin_list";
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

public int getBatchSize() {
    return batchSize;
}

public String next() {
    if (firstItem + batchSize < getItemCount()) {
        firstItem += batchSize;
    }
    return "partidaAdmin_list";
}

public String prev() {
    firstItem -= batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "partidaAdmin_list";
}

public String first() {
    firstItem = 0;
    return "partidaAdmin_list";
}

public String last() {
    firstItem = getItemCount() - batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
}

```

```

        return "partidaAdmin_list";
    }

    public String buscaParametros() {
        Partida partida = this.getPartidaFromRequestParam();
        if (partida.getGolsMandante() == null && partida.getGolsVisitante() == null) {
            Parametro parametroMandante =
                this.buscaParametro(partida.getTemporadaId().getCampeonatoId().getIdCampeonato(),
                    partida.getClubeMandanteId().getIdClube());
            Parametro parametroVisitante =
                this.buscaParametro(partida.getTemporadaId().getCampeonatoId().getIdCampeonato(),
                    partida.getClubeVisitanteId().getIdClube());

            partida.setMediaMandante(partimeetroMandante.getGolsFavorMandante().add(partimeetroVisitante.getGolsContraVisitante()).divide(new BigDecimal(2)));

            partida.setMediaVisitante(partimeetroVisitante.getGolsFavorVisitante().add(partimeetroMandante.getGolsContraMandante()).divide(new BigDecimal(2)));
            double[] probMandante =
                this.geraProbabilidades(partida.getMediaMandante());
            double[] probVisitante =
                this.geraProbabilidades(partida.getMediaVisitante());
            partida.setProbMandante(this.calculaProbabilidadeMandante(probMandante, probVisitante));
            partida.setProbEmpate(this.calculaProbabilidadeEmpate(probMandante, probVisitante));
            partida.setProbVisitante(this.calculaProbabilidadeVisitante(probMandante, probVisitante));
            this.salvaPartida(this.ajustaProbabilidades(partida));
        }
        return "partidaAdmin_list";
    }

    private Parametro buscaParametro(Integer idCampeonato, Integer idClube) {
        EntityManager em = getEntityManager();
        Parametro parametro = null;
        Clube clube = (Clube)
            em.createNamedQuery("Clube.findByIdClube").setParameter("idClube", idClube).getSingleResult();
        try {
            Query q =
                em.createNamedQuery("Parametro.findByIdCampeonatoIdAndClubeId");
            q.setParameter("campeonatoId", idCampeonato);
            q.setParameter("clubeId", idClube);
            parametro = (Parametro) q.getSingleResult();
        } catch (NoResultException nre) {
            Partida partida = this.getPartidaFromRequestParam();
            int clubeId = idClube;
            int campeonato = idCampeonato;
            ParametroPK pk = new ParametroPK(clubeId, campeonato);
            parametro = new Parametro(pk);
            parametro.setCampeonato(partida.getTemporadaId().getCampeonatoId());
            parametro.setClube(clube);
            parametro.setGolsContraMandante(new BigDecimal(0d));
            parametro.setGolsContraVisitante(new BigDecimal(0d));
            parametro.setGolsFavorMandante(new BigDecimal(0d));
            parametro.setGolsFavorVisitante(new BigDecimal(0d));
            this.persistParametro(parametro);
        } finally {
            return parametro;
        }
    }

    private void persistParametro(Parametro parametro) {

```

```

EntityManager em = getEntityManager() ;
try {
    em.getTransaction() .begin() ;
    em.persist(parametro) ;
    em.getTransaction() .commit() ;
    addSuccessMessage("Parametro was successfully created.");
} catch (Exception ex) {
    try {
        addErrorMessage(ex.getLocalizedMessage()) ;
        em.getTransaction() .rollback() ;
    } catch (Exception e) {
        addErrorMessage(e.getLocalizedMessage()) ;
    }
} finally {
    em.close() ;
}
}

private String salvaParametro(Parametro parametro){
EntityManager em = getEntityManager() ;
try {
    em.getTransaction() .begin() ;
    em.merge(parametro) ;
    em.getTransaction() .commit() ;
    addSuccessMessage("Parametro was successfully updated.");
} catch (Exception ex) {
    try {
        addErrorMessage(ex.getLocalizedMessage()) ;
        em.getTransaction() .rollback() ;
    } catch (Exception e) {
        addErrorMessage(e.getLocalizedMessage()) ;
    }
} finally {
    em.close() ;
}
return "partidaAdmin_list";
}

private String salvaPartida(Partida partidaEditada) {
EntityManager em = getEntityManager() ;
try {
    em.getTransaction() .begin() ;
    partida = em.merge(partidaEditada) ;
    em.getTransaction() .commit() ;
    addSuccessMessage("Partida was successfully updated.");
} catch (Exception ex) {
    try {
        addErrorMessage(ex.getLocalizedMessage()) ;
        em.getTransaction() .rollback() ;
    } catch (Exception e) {
        addErrorMessage(e.getLocalizedMessage()) ;
    }
} finally {
    em.close() ;
}
return "partidaAdmin_list";
}

private BigDecimal calculaProbabilidadeMandante(double[] mandante, double[]
visitante){
    double probMandante = 0d;
    for (int i=0; i<4; i++){
        for (int j=i+1; j<5; j++){
            probMandante += mandante[j] * visitante[i];

```

```

    }
}
return new BigDecimal(probMandante);
}

private BigDecimal calculaProbabilidadeEmpate(double[] mandante, double[]
visitante){
    double probEmpate = 0d;
    for (int i=0; i<5; i++){
        probEmpate += mandante[i] * visitante[i];
    }
    return new BigDecimal(probEmpate);
}

private BigDecimal calculaProbabilidadeVisitante(double[] mandante, double[]
visitante){
    double probVisitante = 0d;
    for (int i=0; i<4; i++){
        for (int j=i+1; j<5; j++){
            probVisitante += mandante[i] * visitante[j];
        }
    }
    return new BigDecimal(probVisitante);
}

private double[] geraProbabilidades(BigDecimal media){
    double [] prob = new double[5];
    for (int i = 0; i<5; i++){
        prob[i] = this.calculaProbabilidade(media, i);
    }
    return prob;
}

private Partida ajustaProbabilidades(Partida p){
    BigDecimal probMandante = p.getProbMandante();
    BigDecimal probEmpate = p.getProbEmpate();
    BigDecimal probVisitante = p.getProbVisitante();
    BigDecimal probabilidade = probMandante.add(probEmpate);
    probabilidade = probabilidade.add(probVisitante);
    p.setProbMandante(probMandante.divide(probabilidade, new MathContext(4)));
    p.setProbEmpate(probEmpate.divide(probabilidade, new MathContext(4)));
    p.setProbVisitante(probVisitante.divide(probabilidade, new MathContext(4)));
    return p;
}

private int fatorial(int x){
    if(x<=1) return 1;
    else{
        for (int i=x-1; i>1; i--){
            x*=i;
        }
        return x;
    }
}

private double calculaProbabilidade(BigDecimal media, int gols){
    return media.pow(gols).multiply(new BigDecimal(Math.exp(-
media.doubleValue()), new MathContext(4))).divide(new BigDecimal(this.fatorial(gols), new
MathContext(4)), BigDecimal.ROUND_HALF_EVEN).doubleValue();
}

private void atualizaParametros(){
    if(partida.getGolsMandante() != null && partida.getGolsVisitante() != null){
        int golMandante = partida.getGolsMandante();

```



```

        if (golMandante>3) golMandante = 4;
        int golVisitante = partida.getGolsVisitante();
        if (golVisitante>3) golVisitante = 4;
        Parametro parametroMandante =
this.buscaParametro(partida.getTemporadaId().getCampeonatoId().getIdCampeonato(),
partida.getIdClubeMandante().getIdClube());
        Parametro parametroVisitante =
this.buscaParametro(partida.getTemporadaId().getCampeonatoId().getIdCampeonato(),
partida.getIdClubeVisitante().getIdClube());

parametroMandante.setGolsFavorMandante(this.calculaParametro(parametroMandante.getGolsFavorMandante(), golMandante));

parametroMandante.setGolsContraMandante(this.calculaParametro(parametroMandante.getGolsContraMandante(), golVisitante));

parametroVisitante.setGolsFavorVisitante(this.calculaParametro(parametroVisitante.getGolsFavorVisitante(), golVisitante));

parametroVisitante.setGolsContraVisitante(this.calculaParametro(parametroVisitante.getGolsContraVisitante(), golMandante));
        this.salvaParametro(parametroMandante);
        this.salvaParametro(parametroVisitante);
    }
}

private BigDecimal calculaParametro(BigDecimal parametro, int gol){
    BigDecimal pMultiplicado = parametro.multiply(new BigDecimal(4d));
    BigDecimal pSomado = pMultiplicado.add(new BigDecimal(gol));
    BigDecimal pDividido = pSomado.divide(new BigDecimal(5d));
    return pDividido;
}

public List getProbabilidadesMandante(){
    List probabilidades = new ArrayList();
    for (int v=0; v<4; v++){
        for (int m=v+1; m<5; m++){
            double p = this.calculaProbabilidade(partida.getMediaMandante(), m) *
this.calculaProbabilidade(partida.getMediaVisitante(), v);
            Probabilidade prob = new Probabilidade(m,v,p);
            probabilidades.add(prob);
        }
    }
    return this.sort(probabilidades);
}

public List getProbabilidadesEmpate(){
    List probabilidades = new ArrayList();
    for (int g=0; g<5; g++){
        double p = this.calculaProbabilidade(partida.getMediaMandante(), g) *
this.calculaProbabilidade(partida.getMediaVisitante(), g);
        Probabilidade prob = new Probabilidade(g,g,p);
        probabilidades.add(prob);
    }
    return this.sort(probabilidades);
}

public List getProbabilidadesVisitante(){
    List probabilidades = new ArrayList();
    for (int m=0; m<4; m++){
        for (int v=m+1; v<5; v++){
            double p = this.calculaProbabilidade(partida.getMediaMandante(), m) *
this.calculaProbabilidade(partida.getMediaVisitante(), v);

```

```

        Probabilidade prob = new Probabilidade(m,v,p);
        probabilidades.add(prob);
    }
}
return this.sort(probabilidades);
}

public List sort(List lista){
    Collections.sort(lista, new ProbComparator());
    return lista;
}
}

```

11.47. Admin - PartidaConverter.java

```

/*
 * PartidaConverter.java
 *
 * Created on 21 de Agosto de 2007, 18:31
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Partida;

/**
 *
 * @author Carlos
 */
public class PartidaConverter implements Converter {

    /** Creates a new instance of PartidaConverter */
    public PartidaConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uIComponent,
String string) {
        if (string == null) {
            return null;
        }
        Integer id = new Integer(string);
        PartidaController controller = (PartidaController)
facesContext.getApplication().getVariableResolver().resolveVariable(
            facesContext, "partidaAdmin");

        return controller.findPartida(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uIComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if(object instanceof Partida) {
            Partida o = (Partida) object;
            return "" + o.getIdPartida();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Partida");
        }
    }
}

```

```
}  
}
```

11.48. Admin - TemporadaController.java

```
/*  
 * TemporadaController.java  
 *  
 * Created on 21 de Agosto de 2007, 18:32  
 *  
 * To change this template, choose Tools | Template Manager  
 * and open the template in the editor.  
 */  
  
package chutegol.admin;  
  
import java.util.ArrayList;  
import java.util.Collection;  
import java.util.List;  
import javax.faces.application.FacesMessage;  
import javax.faces.context.FacesContext;  
import javax.faces.model.DataModel;  
import javax.faces.model.ListDataModel;  
import javax.faces.model.SelectItem;  
import javax.persistence.EntityManager;  
import javax.persistence.EntityManagerFactory;  
import javax.persistence.Persistence;  
import javax.persistence.Query;  
import persistencia.Campeonato;  
import persistencia.Temporada;  
  
/**  
 *  
 * @author Carlos  
 */  
public class TemporadaController {  
  
    /** Creates a new instance of TemporadaController */  
    public TemporadaController() {  
        emf = Persistence.createEntityManagerFactory("chutegolPU");  
    }  
  
    private Temporada temporada;  
  
    private DataModel model;  
  
    private EntityManagerFactory emf;  
  
    private EntityManager getEntityManager() {  
        return emf.createEntityManager();  
    }  
  
    private int batchSize = 20;  
  
    private int firstItem = 0;  
  
    public Temporada getTemporada() {  
        return temporada;  
    }  
  
    public void setTemporada(Temporada temporada) {  
        this.temporada = temporada;  
    }  
  
    public DataModel getDetailTemporadas() {
```

```

        return model;
    }

    public void setDetailTemporadas(Collection<Temporada> m) {
        model = new ListDataModel(new ArrayList(m));
    }

    public String createSetup() {
        this.temporada = new Temporada();
        return "temporadaAdmin_create";
    }

    public String create() {
        EntityManager em = getEntityManager();
        try {
            em.getTransaction().begin();
            em.persist(temporada);
            em.getTransaction().commit();
            addSuccessMessage("Temporada was successfully created.");
        } catch (Exception ex) {
            try {
                addErrorMessage(ex.getLocalizedMessage());
                em.getTransaction().rollback();
            } catch (Exception e) {
                addErrorMessage(e.getLocalizedMessage());
            }
        } finally {
            em.close();
        }
        return "temporadaAdmin_list";
    }

    public String detailSetup() {
        setTemporadaFromRequestParam();
        return "temporadaAdmin_detail";
    }

    public String editSetup() {
        setTemporadaFromRequestParam();
        return "temporadaAdmin_edit";
    }

    public String edit() {
        EntityManager em = getEntityManager();
        try {
            em.getTransaction().begin();
            temporada = em.merge(temporada);
            em.getTransaction().commit();
            addSuccessMessage("Temporada was successfully updated.");
        } catch (Exception ex) {
            try {
                addErrorMessage(ex.getLocalizedMessage());
                em.getTransaction().rollback();
            } catch (Exception e) {
                addErrorMessage(e.getLocalizedMessage());
            }
        } finally {
            em.close();
        }
        return "temporadaAdmin_list";
    }

    public String destroy() {
        EntityManager em = getEntityManager();

```

```

    try {
        em.getTransaction().begin();
        Temporada temporada = getTemporadaFromRequestParam();
        temporada = em.merge(temporada);
        em.remove(temporada);
        em.getTransaction().commit();
        addSuccessMessage("Temporada was successfully deleted.");
    } catch (Exception ex) {
        try {
            addErrorMessage(ex.getLocalizedMessage());
            em.getTransaction().rollback();
        } catch (Exception e) {
            addErrorMessage(e.getLocalizedMessage());
        }
    } finally {
        em.close();
    }
    return "temporadaAdmin_list";
}

public Temporada getTemporadaFromRequestParam() {
    EntityManager em = getEntityManager();
    try{
        Temporada o = (Temporada) model.getRowData();
        o = em.merge(o);
        return o;
    } finally {
        em.close();
    }
}

public void setTemporadaFromRequestParam() {
    Temporada temporada = getTemporadaFromRequestParam();
    setTemporada(temporada);
}

public DataModel getTemporadas() {
    EntityManager em = getEntityManager();
    try{
        Query q = em.createQuery("select object(o) from Temporada as o");
        q.setMaxResults(batchSize);
        q.setFirstResult(firstItem);
        model = new ListDataModel(q.getResultList());
        return model;
    } finally {
        em.close();
    }
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Temporada findTemporada(Integer id) {

```

```

EntityManager em = getEntityManager();
try{
    Temporada o = (Temporada) em.find(Temporada.class, id);
    return o;
} finally {
    em.close();
}
}

public javax.faces.model.SelectItem[] getCampeonatoIds() {
    EntityManager em = getEntityManager();
    try{
        List <Campeonato> l = (List <Campeonato>) em.createQuery("select o
from Campeonato as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Campeonato x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    try{
        int count = ((Long) em.createQuery("select count(o) from Temporada as
o").getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

public int getBatchSize() {
    return batchSize;
}

public String next() {
    if (firstItem + batchSize < getItemCount()) {
        firstItem += batchSize;
    }
    return "temporadaAdmin_list";
}

public String prev() {
    firstItem -= batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "temporadaAdmin_list";
}
}

```

11.49. Admin - TemporadaConverter.java

```
/*
 * TemporadaConverter.java
 *
 * Created on 21 de Agosto de 2007, 18:32
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.admin;

import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Temporada;

/**
 *
 * @author Carlos
 */
public class TemporadaConverter implements Converter {

    /** Creates a new instance of TemporadaConverter */
    public TemporadaConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uComponent,
String string) {
        if (string == null) {
            return null;
        }
        Integer id = new Integer(string);
        TemporadaController controller = (TemporadaController)
facesContext .getApplication() .getVariableResolver() .resolveVariable (
            facesContext, "temporadaAdmin");

        return controller.findTemporada(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if(object instanceof Temporada) {
            Temporada o = (Temporada) object;
            return "" + o.getIdTemporada();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Temporada");
        }
    }
}
}
```

11.50. User - ClubeController.java

```
/*
 * ClubeController.java
 *
 * Created on 16 de Setembro de 2007, 13:25
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */
```

```

package chutegol.user;

import java.util.ArrayList;
import java.util.Collection;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.Query;
import persistencia.Clube;

/**
 *
 * @author Carlos
 */
public class ClubeController {

    /** Creates a new instance of ClubeController */
    public ClubeController() {
        emf = Persistence.createEntityManagerFactory("chutegolPU");
    }

    private Clube clube;

    private DataModel model;

    private EntityManagerFactory emf;

    private EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    private int batchSize = 10;

    private int firstItem = 0;

    public Clube getClube() {
        return clube;
    }

    public void setClube(Clube clube) {
        this.clube = clube;
    }

    public DataModel getDetailClubes() {
        return model;
    }

    public void setDetailClubes(Collection<Clube> m) {
        model = new ListDataModel(new ArrayList(m));
    }

    public Clube getClubeFromRequestParam() {
        EntityManager em = getEntityManager();
        try{
            Clube o = (Clube) model.getRowData();
            o = em.merge(o);
            return o;
        } finally {
            em.close();
        }
    }
}

```



```

}

public void setClubeFromRequestParam() {
    Clube clube = getClubeFromRequestParam();
    setClube(clube);
}

public DataModel getClubes() {
    EntityManager em = getEntityManager();
    try{
        Query q = em.createQuery("select object(o) from Clube as o");
        q.setMaxResults(batchSize);
        q.setFirstResult(firstItem);
        model = new ListDataModel(q.getResultList());
        return model;
    } finally {
        em.close();
    }
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Clube findClube(Integer id) {
    EntityManager em = getEntityManager();
    try{
        Clube o = (Clube) em.find(Clube.class, id);
        return o;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    try{
        int count = ((Long) em.createQuery("select count(o) from Clube as
o").getSingleResult()).intValue();
        return count;
    } finally {
        em.close();
    }
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

```

```

    public int getBatchSize() {
        return batchSize;
    }

    public String next() {
        if (firstItem + batchSize < getItemCount()) {
            firstItem += batchSize;
        }
        return "clube_list";
    }

    public String prev() {
        firstItem -= batchSize;
        if (firstItem < 0) {
            firstItem = 0;
        }
        return "clube_list";
    }
}

```

11.51. User - ClubeConverter.java

```

/*
 * ClubeConverter.java
 *
 * Created on 16 de Setembro de 2007, 13:25
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.user;

import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Clube;

/**
 *
 * @author Carlos
 */
public class ClubeConverter implements Converter {

    /** Creates a new instance of ClubeConverter */
    public ClubeConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uComponent,
String string) {
        if (string == null) {
            return null;
        }
        Integer id = new Integer(string);
        ClubeController controller = (ClubeController)
facesContext .getApplication() .getVariableResolver() .resolveVariable (
            facesContext, "clube");

        return controller.findClube(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uComponent,
Object object) {
        if (object == null) {
            return null;
        }
    }
}

```

```

    }
    if(object instanceof Clube) {
        Clube o = (Clube) object;
        return "" + o.getIdClube();
    } else {
        throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Clube");
    }
}
}
}

```

11.52. User - PartidaController.java

```

/*
 * PartidaController.java
 *
 * Created on 16 de Setembro de 2007, 13:25
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.user;

import chutegol.util.ProbComparator;
import chutegol.util.Probabilidade;
import java.math.BigDecimal;
import java.math.MathContext;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Date;
import java.util.List;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;
import javax.faces.model.SelectItem;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.NoResultException;
import javax.persistence.Persistence;
import javax.persistence.Query;
import persistencia.Clube;
import persistencia.Parametro;
import persistencia.ParametroPK;
import persistencia.Partida;
import persistencia.Temporada;

/**
 *
 * @author Carlos
 */
public class PartidaController {

    /** Creates a new instance of PartidaController */
    public PartidaController() {
        emf = Persistence.createEntityManagerFactory("chutegolPU");
    }

    private Partida partida;

    private Temporada temporada;

```

```

private DataModel model;

private EntityManagerFactory emf;

private EntityManager getEntityManager() {
    return emf.createEntityManager();
}

private int batchSize = 10;

private int firstItem = 0;

public Partida getPartida() {
    return partida;
}

public void setPartida(Partida partida) {
    this.partida = partida;
}

public Temporada getTemporada() {
    return temporada;
}

public void setTemporada(Temporada temporada) {
    this.temporada = temporada;
}

public DataModel getDetailPartidas() {
    return model;
}

public void setDetailPartidas(Collection<Partida> m) {
    model = new ListDataModel(new ArrayList(m));
}

public String detailSetup() {
    setPartidaFromRequestParam();
    if (partida.getMediaMandante() == null) {
        addErrorMessage("Parâmetros ainda não calculados para a partida");
        return "partida_list";
    } else return "partida_detail";
}

public Partida getPartidaFromRequestParam() {
    EntityManager em = getEntityManager();
    try{
        Partida o = (Partida) model.getRowData();
        o = em.merge(o);
        return o;
    } finally {
        em.close();
    }
}

public void setPartidaFromRequestParam() {
    Partida partida = getPartidaFromRequestParam();
    setPartida(partida);
}

public DataModel getPartidas() {
    EntityManager em = getEntityManager();
    try{
        Query q = em.createQuery("select object(o) from Partida as o");
        q.setMaxResults(batchSize);
    }
}

```

```

        q.setFirstResult(firstItem);
        model = new ListDataModel(q.getResultList());
        return model;
    } finally {
        em.close();
    }
}

public DataModel getPartidasTemporada() {
    EntityManager em = getEntityManager();
    if (temporada == null) return this.getPartidas();
    else{
        try{
            int temporadaId = this.temporada.getIdTemporada();
            Query q = em.createNamedQuery("Partida.findByTemporada");
            q.setParameter("temporada", temporada);
            q.setMaxResults(batchSize);
            q.setFirstResult(firstItem);
            model = new ListDataModel(q.getResultList());
            return model;
        } finally {
            em.close();
        }
    }
}

public String partidasTemp(){
    return "partida_list";
}

public static void addErrorMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_ERROR,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage(null, facesMsg);
}

public static void addSuccessMessage(String msg) {
    FacesMessage facesMsg = new FacesMessage(FacesMessage.SEVERITY_INFO,
msg, msg);
    FacesContext fc = FacesContext.getCurrentInstance();
    fc.addMessage("successInfo", facesMsg);
}

public Partida findPartida(Integer id) {
    EntityManager em = getEntityManager();
    try{
        Partida o = (Partida) em.find(Partida.class, id);
        return o;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getClubeMandanteIds() {
    EntityManager em = getEntityManager();
    try{
        List <Clube> l = (List <Clube>) em.createQuery("select o from Clube
as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Clube x : l) {
            select[i++] = new SelectItem(x);
        }
    }
}

```

```

        return select;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getClubeVisitanteIds() {
    EntityManager em = getEntityManager();
    try{
        List <Clube> l = (List <Clube>) em.createQuery("select o from Clube
as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Clube x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public javax.faces.model.SelectItem[] getTemporadaIds() {
    EntityManager em = getEntityManager();
    try{
        List <Temporada> l = (List <Temporada>) em.createQuery("select o
from Temporada as o").getResultList();
        SelectItem select[] = new SelectItem[l.size()];
        int i = 0;
        for(Temporada x : l) {
            select[i++] = new SelectItem(x);
        }
        return select;
    } finally {
        em.close();
    }
}

public int getItemCount() {
    EntityManager em = getEntityManager();
    if (temporada == null){
        try{
            int count = ((Long) em.createQuery("select count(o) from Partida
as o").getSingleResult()).intValue();
            return count;
        } finally {
            em.close();
        }
    } else{
        try{
            Query q = em.createNamedQuery("Partida.countByTemporada");
            q.setParameter("temporada", temporada);
            int count = ((Long) q.getSingleResult()).intValue();
            return count;
        }finally{
            em.close();
        }
    }
}

public int getItemCountDate() {
    EntityManager em = getEntityManager();
    Date hoje = new Date();
    if (temporada == null){

```

```

        try{
            Query q = em.createNamedQuery("Partida.countByDataPartida");
            q.setParameter("dataPartida", hoje);
            int count = ((Long) q.getSingleResult()).intValue();
            return count;
        } finally {
            em.close();
        }
    }
} else{
    try{
        Query q = em.createNamedQuery("Partida.countByDataTemporada");
        q.setParameter("dataPartida", hoje);
        q.setParameter("temporada", temporada);
        int count = ((Long) q.getSingleResult()).intValue();
        return count;
    } finally{
        em.close();
    }
}
}

public String hoje(){
    firstItem = getItemCountDate() + 1;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "partida_list";
}

public int getFirstItem() {
    return firstItem;
}

public int getLastItem() {
    int size = getItemCount();
    return firstItem + batchSize > size ? size : firstItem + batchSize;
}

public int getBatchSize() {
    return batchSize;
}

public String next() {
    if (firstItem + batchSize < getItemCount()) {
        firstItem += batchSize;
    }
    return "partida_list";
}

public String prev() {
    firstItem -= batchSize;
    if (firstItem < 0) {
        firstItem = 0;
    }
    return "partida_list";
}

public String first(){
    firstItem = 0;
    return "partida_list";
}

public String last(){
    firstItem = getItemCount() - batchSize;
    if (firstItem < 0) {

```

```

        firstItem = 0;
    }
    return "partida_list";
}

private int fatorial(int x){
    if(x<=1) return 1;
    else{
        for (int i=x-1; i>1; i--){
            x*=i;
        }
        return x;
    }
}

private double calculaProbabilidade(BigDecimal media, int gols){
    return media.pow(gols).multiply(new BigDecimal(Math.exp(-
media.doubleValue()), new MathContext(2))).divide(new BigDecimal(this.fatorial(gols), new
MathContext(2)), BigDecimal.ROUND_HALF_EVEN).doubleValue();
}

public List getProbabilidadesMandante(){
    List probabilidades = new ArrayList();
    for (int v=0; v<4; v++){
        for (int m=v+1; m<5; m++){
            double p = this.calculaProbabilidade(partida.getMediaMandante(), m) *
this.calculaProbabilidade(partida.getMediaVisitante(), v);
            Probabilidade prob = new Probabilidade(m,v,p);
            probabilidades.add(prob);
        }
    }
    return this.sort(probabilidades);
}

public List getProbabilidadesEmpate(){
    List probabilidades = new ArrayList();
    for (int g=0; g<5; g++){
        double p = this.calculaProbabilidade(partida.getMediaMandante(), g) *
this.calculaProbabilidade(partida.getMediaVisitante(), g);
        Probabilidade prob = new Probabilidade(g,g,p);
        probabilidades.add(prob);
    }
    return this.sort(probabilidades);
}

public List getProbabilidadesVisitante(){
    List probabilidades = new ArrayList();
    for (int m=0; m<4; m++){
        for (int v=m+1; v<5; v++){
            double p = this.calculaProbabilidade(partida.getMediaMandante(), m) *
this.calculaProbabilidade(partida.getMediaVisitante(), v);
            Probabilidade prob = new Probabilidade(m,v,p);
            probabilidades.add(prob);
        }
    }
    return this.sort(probabilidades);
}

public List sort(List lista){
    Collections.sort(lista, new ProbComparator());
    return lista;
}
}

```


11.53. User - PartidaConverter.java

```
/*
 * PartidaConverter.java
 *
 * Created on 16 de Setembro de 2007, 13:25
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package chutegol.user;

import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.convert.Converter;
import persistencia.Partida;

/**
 *
 * @author Carlos
 */
public class PartidaConverter implements Converter {

    /** Creates a new instance of PartidaConverter */
    public PartidaConverter() {
    }

    public Object getAsObject(FacesContext facesContext, UIComponent uComponent,
String string) {
        if (string == null) {
            return null;
        }
        Integer id = new Integer(string);
        PartidaController controller = (PartidaController)
facesContext.getApplication().getVariableResolver().resolveVariable(
        facesContext, "partida");

        return controller.findPartida(id);
    }

    public String getAsString(FacesContext facesContext, UIComponent uComponent,
Object object) {
        if (object == null) {
            return null;
        }
        if(object instanceof Partida) {
            Partida o = (Partida) object;
            return "" + o.getIdPartida();
        } else {
            throw new IllegalArgumentException("object:" + object + " of type:" +
object.getClass().getName() + "; expected type: persistencia.Partida");
        }
    }
}
}
```

11.54. Útil - Probabilidade.java

```
/*
 * probabilidade.java
 *
 * Created on 11 de Setembro de 2007, 17:39
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
```

```

*/

package chutegol.util;

/**
 *
 * @author Carlos
 */

public class Probabilidade implements Comparable {

    private int placarMandante;
    private int placarVisitante;
    private double prob;

    /** Creates a new instance of probabilidade */
    public Probabilidade() {
    }

    public Probabilidade(int m, int v, double p){
        this.setPlacarMandante(m);
        this.setPlacarVisitante(v);
        this.setProb(p);
    }

    public int getPlacarMandante() {
        return placarMandante;
    }

    public void setPlacarMandante(int placarMandante) {
        this.placarMandante = placarMandante;
    }

    public int getPlacarVisitante() {
        return placarVisitante;
    }

    public void setPlacarVisitante(int placarVisitante) {
        this.placarVisitante = placarVisitante;
    }

    public double getProb() {
        return prob;
    }

    public void setProb(double prob) {
        this.prob = prob;
    }

    public int compareTo(Object o1){
        Probabilidade p2 = (Probabilidade) o1;
        if( prob > p2.getProb() ){
            return -1;
        } else if(prob < p2.getProb()){
            return 1;
        } else { //se zero
            if(placarMandante > p2.getPlacarMandante()){
                return -1;
            }else{
                if(placarMandante < p2.getPlacarMandante()){
                    return 1;
                }else{
                    if(placarVisitante > p2.getPlacarVisitante()){
                        return -1;
                    }else{

```

