

# **DOCUMENTO DE REQUISITOS BOLÃO WEB**

**Versão 2.0**

Integrantes	RA
EVERTON RODRIGO DA SILVA	54385
GUILHERME GANASSIN	61528
MARCELO QUADRELLI PINHEIRO	60933

## SUMARIO

1 INTRODUÇÃO.....	3
1.1 Objetivo do Documento.....	3
1.2 Escopo do Produto.....	3
2 DESCRIÇÃO GERAL DO PRODUTO.....	4
2.1 Interfaces com o Produto.....	4
2.2 Requisitos Funcionais.....	4
[RF001] Gerenciar Operadores.....	4
[RF002] Gerenciar Campeonatos.....	4
[RF003] Gerenciar Times.....	5
[RF004] Gerenciar Partidas.....	5
[RF005] Gerenciar apostadores.....	5
[RF006] Cadastro Externo de Apostador.....	6
[RF007] Gerenciar Palpite.....	6
[RF008] Processar resultado e distribuir pontos.....	6
[RF009] Ranking de apostadores.....	6
[RF010] Consulta fácil de palpites.....	7
2.3 Requisitos Não Funcionais.....	7
[NF001] Plataforma Web.....	7
3 ESTUDO DE VIABILIDADE.....	7
3.1 Quanto às contribuições à empresa.....	7
3.2 Quanto à viabilidade econômica.....	8
3.3 Quanto à viabilidade técnica.....	8
3.4 Quanto à legalidade.....	8
3.5 Quanto às alternativas.....	8
4 AVALIAÇÃO DOS RISCOS.....	9
4.1 Riscos de Projeto.....	9
4.2 Risco técnico.....	9
5 DIAGRAMA DE PACOTES.....	10
5.1 Diagrama de Pacotes Genérico.....	10
5.2 Diagrama de Pacotes com ênfase na atuação de Frameworks.....	10
6 ARQUITETURA DE SOFTWARE.....	11
6.1 Diagrama de Arquitetura de Software.....	11
7 FRAMEWORKS.....	12
7.1 JSF.....	12
7.2 BootsFaces.....	12
7.3 Hibernate.....	13
8 MODELO DE CASOS DE USO.....	13
8.1 Identificação e Priorização dos casos de uso.....	13
Diagrama de Casos de Uso.....	14
8.2 Detalhamento dos Casos de Uso.....	15
9 MODELO DE ANÁLISE.....	25
10 PLANEJAMENTO DAS ITERAÇÕES.....	29
11 DIAGRAMA DE CLASSES DE PROJETO.....	30
11.1 Diagrama de Classes com Pacotes.....	30
11.2 Diagrama de Classes de Projeto.....	30

# 1 INTRODUÇÃO

Este documento especifica os requisitos do **Bolão Web**, um sistema onde usuários poderão fazer apostas sobre o placar de jogos de futebol e time vencedor dos principais campeonatos de futebol do mundo. Esta documentação irá fornecer aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação da aplicação.

O Bolão Web será uma aplicação disponível na rede mundial de computadores (*internet*) através do endereço **<http://bolao.anguloweb.com.br>**, publicamente, através de um registro de usuário. O seu principal objetivo é angariar visibilidade e acessos ao domínio da empresa **AnguloWeb Soluções Para Internet Ltda.** na internet, promovendo engajamento e diversão com o bom e velho bolão, porém, de uma maneira inovadora através de um sistema web.

## 1.1 Objetivo do Documento

Este documento tem como público alvo todos os envolvidos no projeto, desde sua análise até a concepção, com o intuito de guiar a todos em direção a seu objetivo e, conseqüentemente, sua conclusão satisfatória.

## 1.2 Escopo do Produto

A Anguloweb é uma empresa de natureza *startup*, com foco no desenvolvimento de soluções web personalizadas e sob encomenda.

O objetivo do projeto é elevar a audiência de seu domínio na internet, gerando visibilidade e credibilidade à empresa, para alavancar o crescimento da mesma através da obtenção de novos clientes e projetos.

O projeto tem como foco a diversão, e com integração as redes sociais, pretendemos obter engajamento dos usuários, levando a um aumento exponencial dos novos registros e, conseqüentemente, dos acessos.

O objeto do projeto é a criação de uma plataforma web onde haja interação entre usuários através de um sistema de “Bolão”, onde usuários cadastrados podem palpar sobre jogos de futebol previamente cadastrados, recebendo assim pontuações por seus acertos. Sua pontuação pode ser vista por um ranking dos melhores apostadores.

## 2 DESCRIÇÃO GERAL DO PRODUTO

### 2.1 Interfaces com o Produto

Ator	Descrição
Administrador	Usuário do sistema que deverá moderar e gerenciar os usuários e definir políticas de uso e pontuação das apostas.
Operador	Usuário do sistema que deverá manter os cadastros da aplicação atualizados e atender as solicitações dos usuários apostadores. A principal função do operador será atualizar o sistema com os dados das partidas de futebol disponíveis para aposta.
Apostador	Usuário final do sistema que fará as apostas do bolão e receberá pontos de acordo com seus resultados.
Visitante	Usuário final da sistema interessado somente em visualizar informações como, resultado de jogos, situação do ranking, ver campeonatos.

### 2.2 Requisitos Funcionais

#### [RF001] Gerenciar Operadores

O sistema deverá permitir incluir, alterar, excluir, e consultar os usuários operadores. O cadastro deve conter um identificador único gerado pelo sistema, nome, e-mail e senha.

<b>Prioridade:</b>	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável
<b>Ator(es):</b>	Administrador		
<b>Requisitos associados:</b>			

#### [RF002] Gerenciar Campeonatos

O sistema deverá permitir incluir, alterar, excluir, e consultar os campeonatos. O cadastro deve conter um identificador único gerado pelo sistema, nome, escopo (regional, estadual, nacional, continental ou mundial), data de início, data de fim, tipo (pontos ou mata-mata) e observações.

<b>Prioridade:</b>	<input checked="" type="checkbox"/> Essencial	<input type="checkbox"/> Importante	<input type="checkbox"/> Desejável
<b>Ator(es):</b>	Administrador, Operador		

<b>Requisitos associados:</b>	
-------------------------------	--

### [RF003] Gerenciar Times

O sistema deverá permitir incluir, alterar, excluir, e consultar os times participantes de cada campeonato ou torneio. O cadastro deve conter um identificador único gerado pelo sistema, tipo (time ou seleção), nome e local de origem. O local de origem depende do campo tipo: Se for time, considera-se um campo composto por cidade, estado e país. Se for uma seleção, considera-se apenas o país.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Administrador, Operador</i>					
<b>Requisitos associados:</b>						

### [RF004] Gerenciar Partidas

O sistema deverá permitir incluir, alterar, excluir, e consultar as partidas. O cadastro deve conter um identificador único gerado pelo sistema, o identificador de ambos os times, a data e hora da partida e a quantidade de gols marcados por cada time.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Administrador, Operador</i>					
<b>Requisitos associados:</b>	<i>[RF004]</i>					

### [RF005] Gerenciar apostadores

Através do administrador ou operador responsável, o sistema deve permitir o cadastro de usuários apostadores. O cadastro deve conter um identificador único gerado pelo sistema, nome, sobrenome, apelido, e-mail, data de nascimento, senha e foto.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Administrador e Operador</i>					
<b>Requisitos associados:</b>						

### [RF006] Cadastro Externo de Apostador

O sistema deverá permitir o cadastro dos novos usuários apostadores em sua página inicial através de conexão com redes sociais (Facebook).

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Apostador</i>					
<b>Requisitos associados:</b>						

### [RF007] Gerenciar Palpite

O sistema deverá permitir o apostador incluir, alterar, excluir, e consultar seus palpites de resultado para cada jogo. A alteração e exclusão podem ser feitas até o momento de início do jogo.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Apostador</i>					
<b>Requisitos associados:</b>						

### [RF008] Processar resultado e distribuir pontos

Ao fim de cada partida, os usuários operadores serão responsáveis em registrar o resultado do jogo, e ao salvar o sistema deverá processar esse resultado e distribuir os pontos aos apostadores que deram um palpite em determinada partida. O apostador recebe dois pontos caso acerte o placar exato e um ponto caso acerte o time vencedor. O Administrador também possui autonomia de realização desta tarefa.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Administrador, Operador</i>					
<b>Requisitos associados:</b>						

### [RF009] Ranking de apostadores

O sistema deverá disponibilizar de forma pública uma página com o ranking dos apostadores classificado em ordem da quantidade de pontos de

forma decrescente. A página deve conter a classificação, nome e quantidade de pontos de cada apostador. Caso o apostador esteja logado, a posição dele deve estar destacada.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>	<i>Administrador, Visitante</i>					
<b>Requisitos associados:</b>						

## [RF010] Consulta fácil de palpites

O sistema deverá possuir uma tela inicial para o apostador constando seus palpites, os resultados dos jogos já realizados, dados de registro, posição no ranking e quantidade de pontos.

<b>Prioridade:</b>		Essencial	<b>X</b>	Importante		Desejável
<b>Ator(es):</b>	<i>Apostador</i>					
<b>Requisitos associados:</b>	RF008					

## 2.3 Requisitos Não Funcionais

### [NF001] Plataforma Web

O sistema deverá ser implantado em plataforma web, para que seja possível acesso de qualquer lugar, sem a necessidade de instalação do software e também através de dispositivos móveis.

<b>Prioridade:</b>	<b>X</b>	Essencial		Importante		Desejável
<b>Ator(es):</b>						

## 3 ESTUDO DE VIABILIDADE

### 3.1 Quanto às contribuições à empresa

O projeto visa, para o cliente, gerar visibilidade ao seu domínio, um crescente em termos de usuários e credibilidade perante aos mesmos. Visando também garantir futuros novos clientes e projetos. Pois a partir do objetivo traçado, o usuário final poderá interagir e, muitas vezes, se informar sobre o esporte predileto da maioria dos brasileiros, o futebol.

### **3.2 Quanto à viabilidade econômica**

O projeto acaba por ter um custo relativamente baixo, pois é feito através de um sistema web, deixando de lado problemas mais custosos e físicos. Ressaltando que mesmo por tal motivo ainda há custos tal qual: recursos humanos, manutenção.

### **3.3 Quanto à viabilidade técnica**

A tecnologia atual oferece uma gama de recursos que podem contribuir com a construção do projeto. Estamos em um momento que está sendo muito discutido desenvolvimento web, armazenamento em nuvem e tantos outros assuntos relativos ao tema.

Em relação a alguma restrição de custo de processamento, cremos que não haverá maiores problemas, no nosso entendimento os requisitos para o uso do sistema, em termos de hardware e estrutura, estão dentro da faixa de preço e processamento que o mercado brasileiro oferece - necessitando apenas acesso à internet e um navegador web.

A questão do prazo é um ponto importante, pois como campeonatos e demais torneios geralmente são eventos com data estabelecida, existe um prazo válido para a entrega do projeto. Mas considerando as necessidades e o tempo hábil, cremos que o tempo é suficiente.

### **3.4 Quanto à legalidade**

O projeto é baseado em um sistema de palpites, que não visa lucro direto ao cliente ou, de forma alguma, prejuízo ao usuário. Tratando-se apenas de uma opção de descontração e informação. Em nenhum momento foi constatada alguma infração ou ilegalidade em tempo de desenvolvimento do projeto.

### **3.5 Quanto às alternativas**

O desenvolvimento web nos oferece muitas possibilidades, dentre a que mais se destacou durante o estudo foi a linguagem JAVA, pois ela nos oferece codificação no servidor, um ponto importante para o desenvolvimento que visamos. Revisados todos os nossos fatores que colocamos em consideração, chegamos a decisão em usar esta tecnologia, em um SGBD MySQL. A escolha se deu devido à familiaridade entre os desenvolvedores do projeto, por não ser um projeto de grande porte, oferecer mais rapidez em seu desenvolvimento, entre outros fatores menos específicos.



## **4 AVALIAÇÃO DOS RISCOS**

### **4.1 Riscos de Projeto**

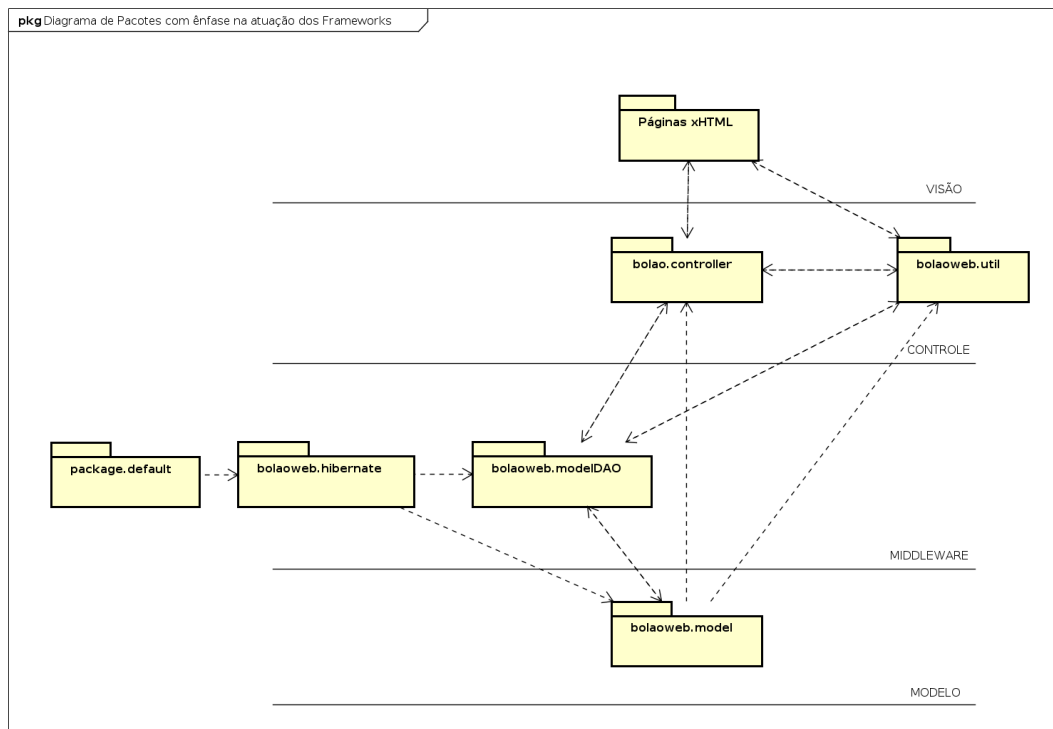
- Falta de tempo, uma vez que somos todos universitários e trabalharemos no contra turno.
- Como os nossos usuários são pessoas aleatórias, não temos como fazer um estudo muito aproximado do que irá agradá-lo e a falta desse estudo poderá gerar uma constante mudança no escopo do projeto.

### **4.2 Risco técnico**

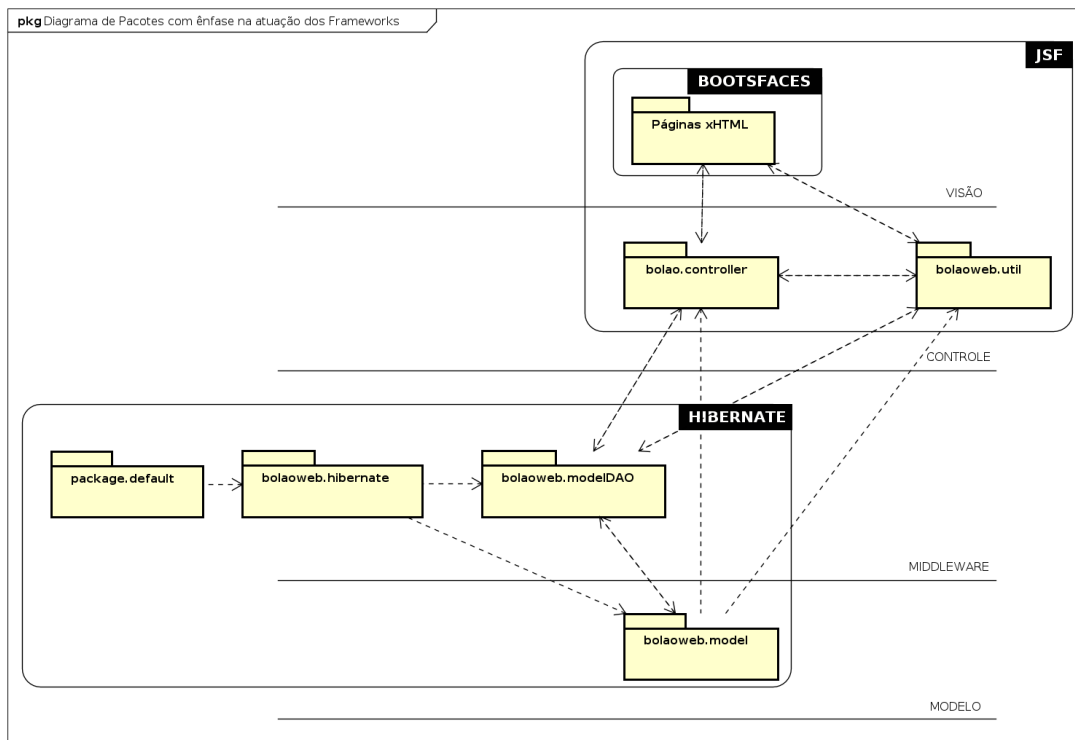
- Falta de informação, caso aconteça de não conseguirmos informações com tempo suficiente para cadastrar os jogos no Bolão WEB e os apostadores fazerem suas apostas.

## 5 DIAGRAMA DE PACOTES

### 5.1 Diagrama de Pacotes Genérico



### 5.2 Diagrama de Pacotes com ênfase na atuação de Frameworks



## 6 ARQUITETURA DE SOFTWARE

Definimos que o tipo de arquitetura de sistema que utilizaremos para o desenvolvimento do sistema web é o MVC. Onde especificamos que:

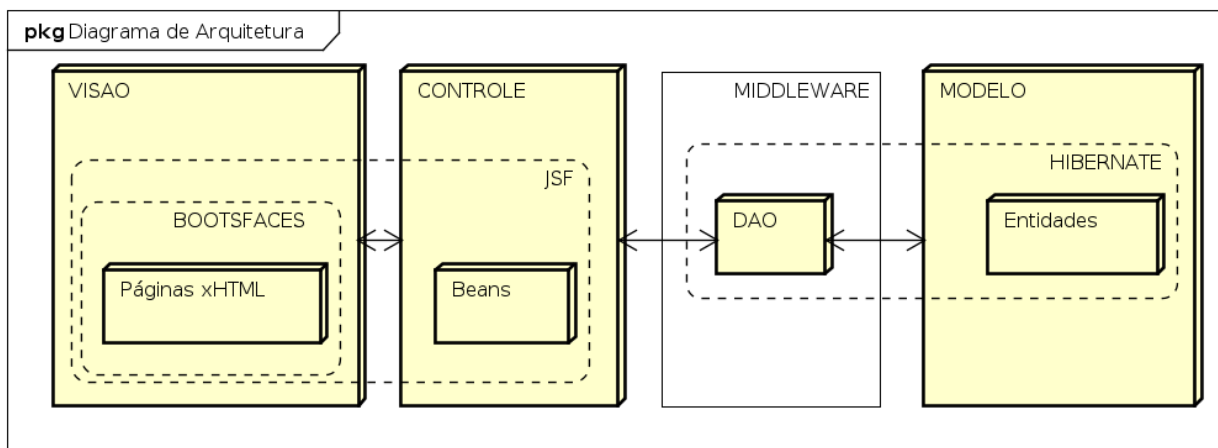
**Camada de Visão:** Será a camada responsável com a interação com o usuário, a partir de um navegador. Responsável pela entrada de dados, requisitar os serviços ao controle e apresentar os resultados;

**Camada de Controle:** Será a responsável em definir e realizar as chamadas de método dentro da aplicação, fazer o controle dos dados recebidos da camada de apresentação e se comunicar com as classes de persistência quando solicitado algum dado presente nesta.

**Camada de Modelo:** Será a responsável em armazenar as regras de negócio e os dados recebidos da camada de aplicação.

No diagrama a seguir, consideramos também uma subcamada de Middleware, localizada entre as camadas de Controle e Modelo. A função principal desta subcamada é portar as classes DAO, que são responsáveis por intermediar o acesso do controle as entidades do Modelo. Esta subcamada de Middleware também é responsável por prover suporte e configuração das classes de persistencia.

### 6.1 Diagrama de Arquitetura de Software



## 7 FRAMEWORKS

### 7.1 JSF

O JavaServer Faces (JSF) é uma tecnologia que nos permite criar aplicações Java para Web utilizando componentes visuais pré-prontos, de forma que o desenvolvedor não se preocupe com o Javascript e HTML, o que define o JSF como um framework. Basta adicionar os componentes que forem necessários ao projeto e eles serão renderizados e exibidos em formato HTML.

Com a característica stateful podemos criar formulários de várias páginas e navegar nos vários passos dos componentes com o estado das telas sendo mantidos automaticamente.

A separação de camadas de apresentação e de aplicação também é uma característica muito marcante da arquitetura do JSF. Pensando no modelo MVC, o JSF possui uma camada de visualização bem separada do conjunto de classes de modelo.

O JSF ainda tem a vantagem de ser uma especificação do Java EE, isto é, todo servidor de aplicações Java tem que vir com uma implementação dela e há diversas outras disponíveis.

Em nosso projeto, o JSF será usado na construção de toda a interface gráfica de interação com o usuário, a utilização deste framework visa a maior eficiência nesta fase da implementação e o principal objetivo é a diminuição do tempo gasto na implementação da interface gráfica.

O JSF irá fazer o mapeamento das páginas xhtml, recebendo as ações pretendidas pelo usuário do sistema, enviando as mesmas para as classes BEANs (responsável pela integração com as classes de persistência) também mapeadas pelo mesmo.

### 7.2 BootsFaces

O BootsFaces é um framework para desenvolvimento de estilos de páginas HTML que surge como uma opção do famoso Bootstrap para desenvolvimento JavaWeb com suporte JSP.

O Bootstrap por si só, já tras como características diversas vantagens, tais como, vasta documentação, otimização para layouts responsivos e a praticidade para o desenvolvimento de layouts padronizados. Com a popularidade do Bootstrap, o BootsFaces foi criado para ser uma poderosa ferramenta JSF que tras o melhor do Bootstrap e que tem foco no desenvolvimento rápido e fácil de aplicações Java Web.

Necessariamente, o BootsFaces se concentra na estrutura de layout da página, permitindo a possibilidade de implementar sistemas bem desenhados com pouco esforço.

Outra característica importante do BootsFaces que influenciou diretamente na escolha do mesmo para nosso projeto é o fato de ser bem leve e consumir pouco espaço. Comparado com outros frameworks JSF, o BootsFaces pode ser até 8 vezes menor.

Infelizmente, por se tratar de uma tecnologia nova, não há disponibilização de temas previamente prontos (como no caso do PrimeFaces) e a customização dos componentes do template padrão disponibilizado pelo BootsFaces não é possível de ser realizada devido a utilização do Maven, ou seja, sempre que a dependência é atualizada, as alterações dos componentes é sobrescrita pela sua versão padronizada.

### **7.3 Hibernate**

O Hibernate é um framework para o mapeamento objeto-relacional. Este framework facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (XML) ou anotações.

O objetivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional. Em especial, no desenvolvimento de consultas e atualizações dos dados.

Sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL). O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual da conversão dos dados resultante, mantendo o programa portátil para quaisquer bancos de dados SQL, porém causando um pequeno aumento no tempo de execução.

Em nosso sistema o Hibernate será utilizado para gerenciar o acesso às informações contidas no banco de dados composto pelas classes de modelo.

O Hibernate, conforme o padrão de projeto Singleton, fará a instanciação única das classes, gerando um “canal” de acesso ao banco de dados.

## **8 MODELO DE CASOS DE USO**

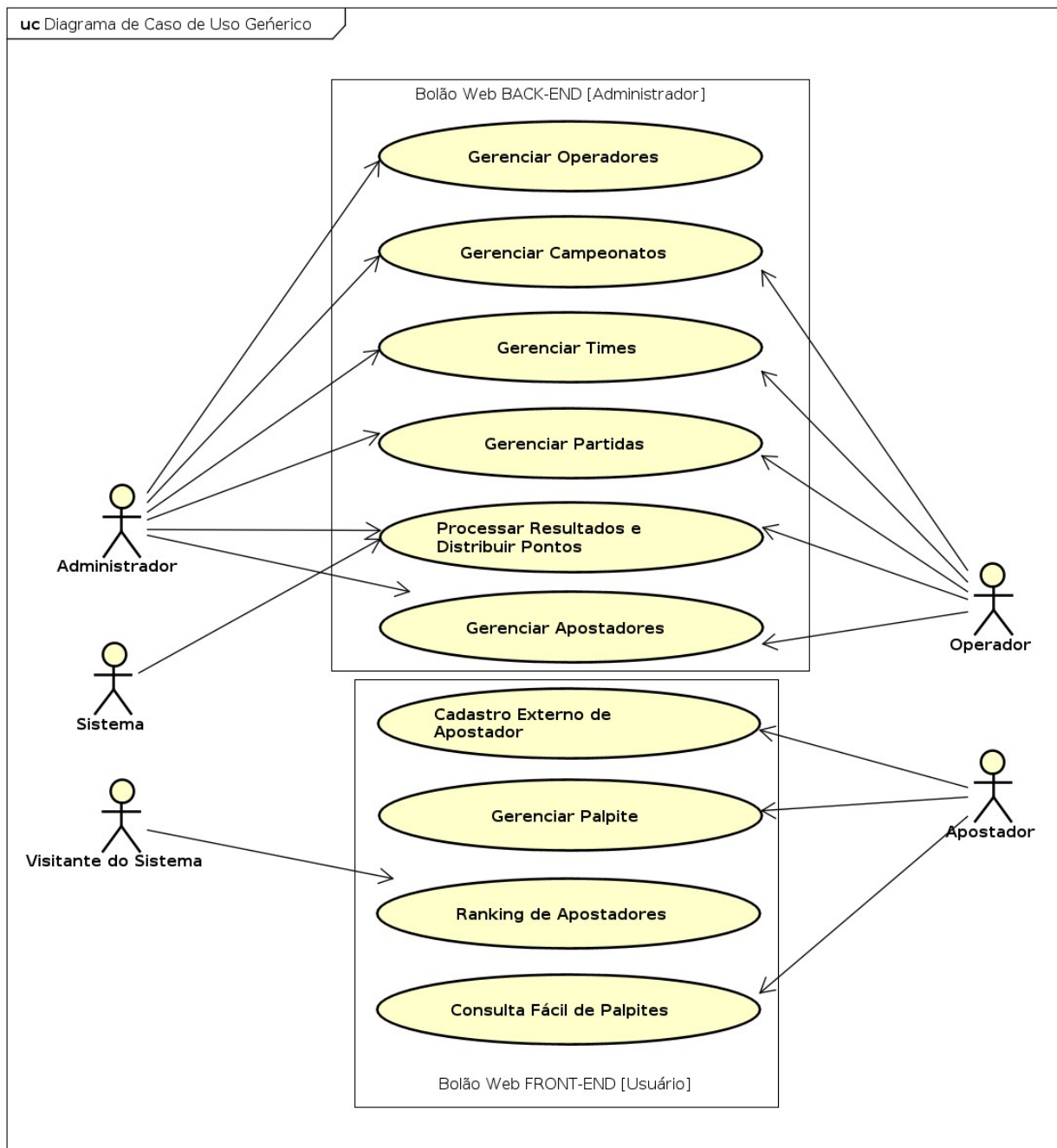
### **8.1 Identificação e Priorização dos casos de uso**

Casos de uso priorizados de forma ordinal:

- [RF001] Gerenciar Operadores - Guilherme
- [RF002] Gerenciar Campeonatos - Marcelo
- [RF003] Gerenciar Times - Everton
- [RF004] Gerenciar Partidas - Everton
- [RF005] Gerenciar Apostador - Guilherme
- [RF006] Cadastro Externo de Apostador- Everton

[RF007] Gerenciar Palpite - Marcelo  
[RF008] Processar resultado e distribuir pontos - Everton  
[RF009] Ranking de apostadores - Guilherme  
[RF010] Consulta fácil de palpites - Marcelo

## Diagrama de Casos de Uso



**Obs.:** O diagrama de caso de uso acima representa todos os casos de uso do sistema relacionados com seus respectivos atores. Nos detalhes a seguir, foi implementado um diagrama para cada caso de uso representando seus detalhes específicos.

## 8.2 Detalhamento dos Casos de Uso

### [RF001] USE CASE<gerenciar operadores> - Guilherme

**Objetivo:**

Permite o gerenciamento e manipulação de usuários responsáveis pela alimentação das informações operacionais do sistema. Estas informações operacionais são pertinentes ao lançamento de resultados, gerenciamento de times, gerenciamento de partidas, gerenciamento de campeonatos e processamento de apostas.

**Pré-condições:**

O usuário deve estar logado no sistema e possuir permissão de administrador.

**Iniciado por:**

Qualquer usuário administrador no intuito de efetuar uma inclusão, alteração ou consulta por usuários operacionais.

**Fluxo principal:**

1. O usuário administrador acessa a opção de controle de usuários operacionais;
2. O sistema mostra a tela para escolha da rotina que será;
3. O usuário escolhe a opção para inclusão/alteração de novo usuário operacional;
4. O sistema mostra o formulário de cadastro/alteração;
5. O usuário informa os dados de inclusão de novo usuário, com atenção especial aos atributos obrigatórios;
6. O sistema valida a corretude dos atributos, e verifica o preenchimento dos atributos obrigatórios;
7. O sistema grava as informações na devida tabela do BD;
8. O sistema envia uma mensagem para a tela do usuário informando o sucesso da operação e atualiza o usuário como ativo.

**Fluxo alternativo 1: Dados pessoais incorretos**

No item 6, caso as informações não corresponderem aos padrões de validação o sistema informa através de mensagem de erro a ocorrência e indica quais campos precisam ser corrigidos. O fluxo principal é retomado com as informações preenchidas corretamente.

**Fluxo alternativo 2: Problema com o banco de dados**

No item 7, caso o banco de dados apresente algum problema ou as informações de persistencia não forem definidas corretamente, o sistema não grava os dados e exibe uma mensagem de erro na tela indicando o problema.

**Fluxo alternativo 3: Exceder tempo**

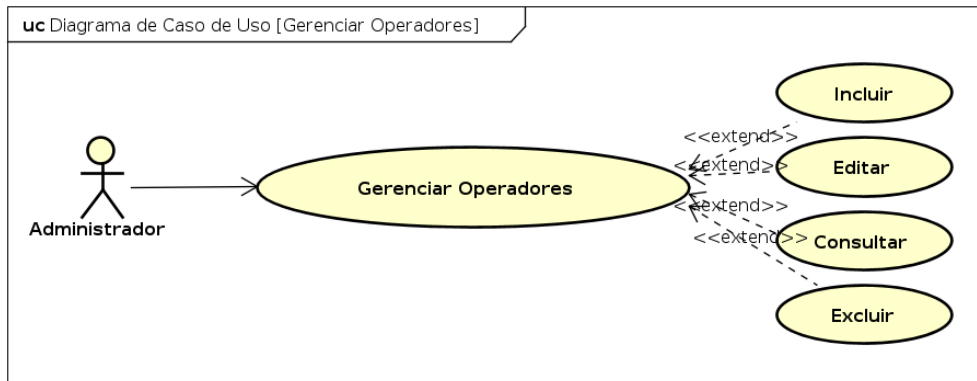
No item 5, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

**Fluxo alternativo 4: Cancelar cadastro.**

Em qualquer momento, o usuário pode cancelar, e o sistema retornando ao início no Menu Principal.

**Pós-condições:**

Um usuário administrador deve ser criado e estar devidamente ativo, ou o processo é abortado na ocorrência do Fluxo alternativo 2.



## [RF002]USE CASE<gerenciar campeonatos> - Marcelo

### Objetivo:

O sistema deverá permitir manipular os dados referente aos campeonatos através da página do Bolão. O cadastro deve conter um identificador único gerado pelo sistema, nome, escopo (regional, estadual, nacional, continental ou mundial), data de início, data de fim, tipo (pontos ou mata mata) e observações.

### Pré-Condições:

O cadastramento e demais gerenciamento de Campeonato é uma atividade exclusiva para usuários Administradores ou Operadores.

### Iniciado Por:

O Administrador ou o Operador, que estiver designado a cadastrar os campeonatos, fará o cadastro manualmente através do site.

### Fluxo principal:

1. O usuário seleciona a opção de cadastrar/alteração campeonato
2. O sistema apresenta um formulário de cadastro/alteração
3. O usuário preenche o formulário de campeonato e submete o formulário
4. O sistema valida os dados cadastrais
5. O sistema reapresenta as informações na tela e pede confirmação
6. O usuário confirma os dados
7. O sistema efetua a inclusão dos dados informados no cadastro de campeonato
8. O sistema agradece o cadastramento e convida o usuário a cadastrar um novo campeonato.

### Fluxo alternativo 1: Corrigir dados cadastrais

No item 4, o sistema ao validar os dados cadastrais, detecta que alguns dados considerados como obrigatórios não foram preenchidos ou que dados foram preenchidos incorretamente.

5a. O sistema reapresenta os dados cadastrais indicando o(s) erro(s)

5b O usuário corrige o(s) erro(s) cadastrais

Retorna ao fluxo principal no item 6

### Fluxo alternativo 2: Exceder tempo



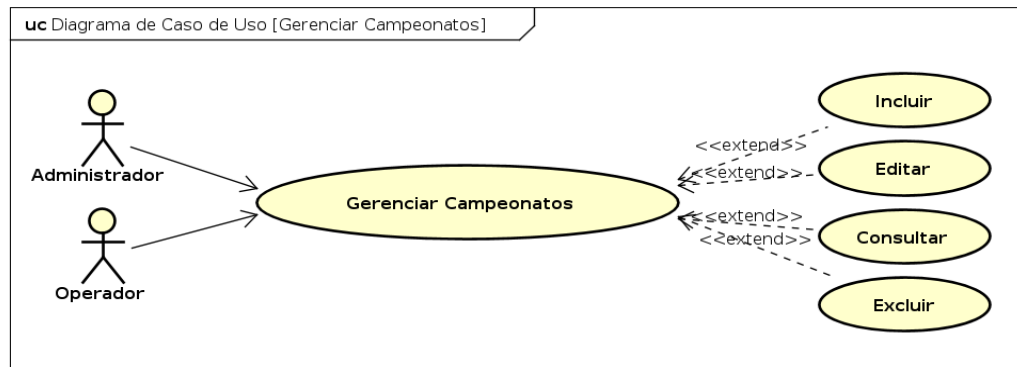
No item 6, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

### **Fluxo alternativo 3: Cancelar cadastro.**

Em qualquer momento, o usuário pode cancelar, e o sistema retornando ao início no Menu Principal.

### **Pós-condições:**

O campeonato foi cadastrado no sistema ou o sistema exibiu a sua página inicial, por cancelamento da operação.



## **[RF003] USE CASE<gerenciar times> - Everton**

### **Objetivo:**

Permitir ao usuário o gerenciamento e manipulação das informações das equipes no sistema.

### **Pré-Condições:**

Usuário logado no sistema e com permissões ao nível operacional ou de administrador.

### **Iniciado Por:**

Usuário com nível de permissão devidamente logado no sistema, com a intenção de incluir, alterar, excluir ou consultar as equipes no sistema.

### **Fluxo principal:**

1. O usuário seleciona a opção de gerenciamento de times;
2. O sistema retorna ao usuário uma tela de formulário para inserção/alteração;
3. O usuário informa que tipo de equipe será manipulada, se clube ou federação (país);
4. O usuário informa o nome, dados de origem da equipe e seleciona a opção para gravação no sistema;
5. O sistema gera o identificador do time inscrito, caso inclusão, ou gera "decremento" do identificador, caso exclusão;
6. O sistema salva as informações no banco de dados;
7. O sistema exibe uma mensagem de confirmação da operação.

### **Fluxo alternativo 1: Cancelar cadastro.**

Em qualquer momento, o usuário pode cancelar o cadastro da equipe, com o sistema retornando ao início do formulário com os campos vazios.

### **Fluxo alternativo 2: Exceder tempo**

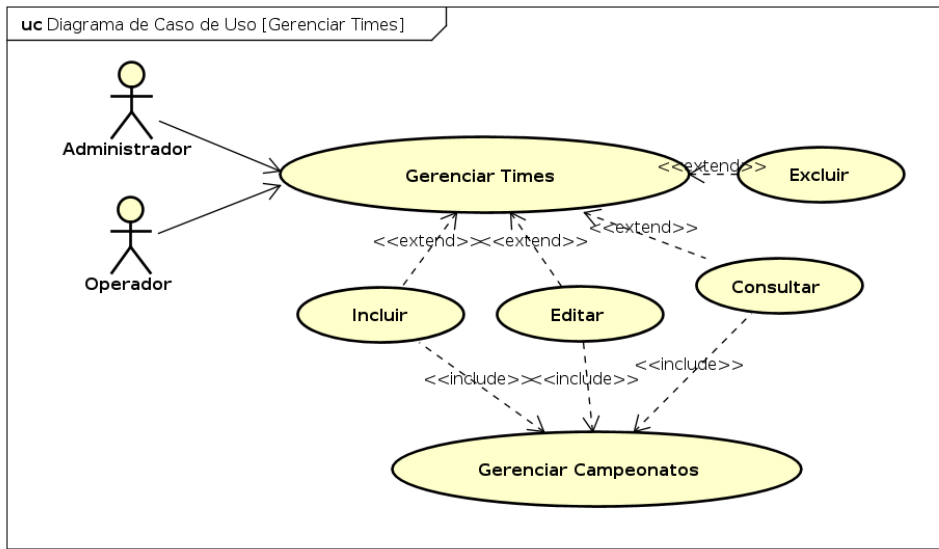
No item 3 ou 4, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

### Fluxo alternativo 3: Dados da equipe

No item 6, o sistema ao validar encontra uma inconsistência, sendo um dado inválido, time já inserido ou a falta de um campo requerido.

### Pós-condições:

Um time foi criado, excluído, alterado ou consultado caso o usuário não tenha abortado o processo.



## [RF004] USE CASE <gerenciar partidas> - Everton

### Objetivo:

Permitir ao usuário administrador ou operador toda a manipulação de informações relativas as partidas.

### Pré-condições:

O usuário deve estar logado no sistema, deve existir ao menos um campeonato cadastrado e ao menos dois times cadastrados.

### Iniciado por:

Usuário administrador ou operador com a intenção de incluir, alterar, excluir ou consultar as partidas.

### Fluxo principal:

1. O usuário acessa a opção de gestão de partidas.
2. O sistema exibe o formulário de cadastro.
3. O usuário seleciona o campeonato na qual a partida faz parte, o time da casa e o time visitante, a data e hora da partida.
4. O sistema gera o identificador da partida.
5. O sistema salva as informações no banco de dados.
6. O sistema exibe uma mensagem informando o sucesso da operação.

### Fluxo alternativo 1: Exceder tempo

No item 3, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

**Fluxo alternativo 2: Cancelar cadastro.**

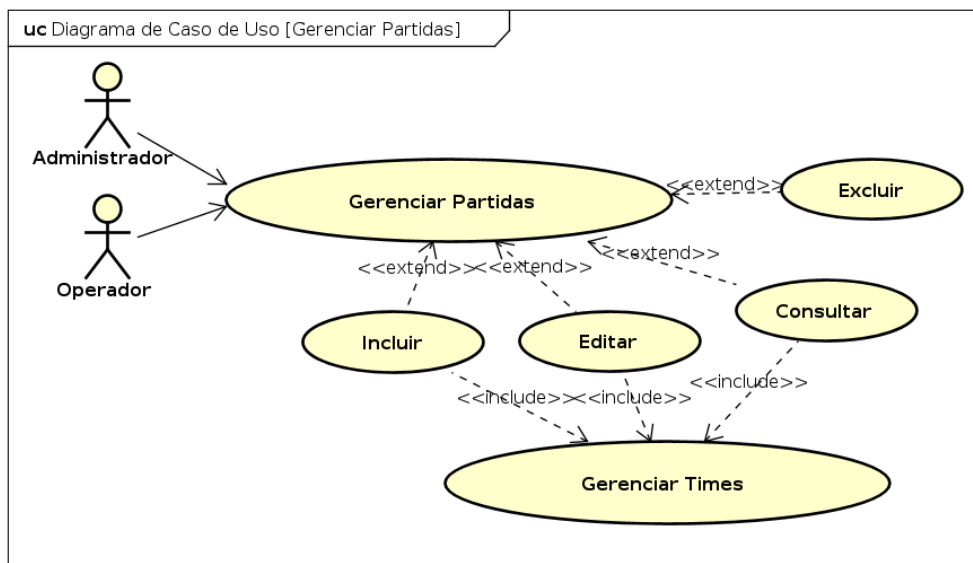
Em qualquer momento, o usuário pode cancelar, e o sistema retornando ao início no Menu Principal.

**Fluxo alternativo 3: Data e hora da partida são inferiores a atual**

Antes do item 4, caso o sistema identifique que a data e hora da partida são menor que a atual (partida já ocorreu), o sistema deve obrigar o usuário a informar o placar do hoje antes de salvar o registro.

**Pós-condições:**

Uma partida foi criada, excluída, alterada ou consultada caso o usuário não tenha abortado o processo.

**[RF005] USE CASE<gerenciar apostador> - Guilherme****Objetivo:**

O sistema deverá permitir os processos de cadastro e gerenciamento do usuário apostador.

**Pré-Condições:**

O usuário deve estar logado.

**Iniciado Por:**

Pelo Administrador ou Operador, quando necessário, através do sistema administrador do Bolão.

**Fluxo principal:**

1. O usuário seleciona a opção de gerenciar cadastro de apostador;
2. O sistema apresenta o formulário para preenchimento dos dados;
3. O usuário incluir/altera os dados;
4. O sistema valida os dados do formulário;
5. O sistema reapresenta os dados ao usuário e pede confirmação
6. O usuário confirma a operação
7. O sistema grava no banco de dados as informações;
8. O sistema notifica o sucesso da operação e volta à página inicial do sistema.

**Fluxo alternativo 1: Corrigir dados**

No item 4, o sistema ao validar os dados, detecta que alguns dados considerados como obrigatórios não foram preenchidos ou que dados foram preenchidos incorretamente.

5a. O sistema reapresenta os dados indicando o(s) erro(s)

5b O usuário corrige o(s) erro(s)

Retorna ao fluxo principal no item 6

**Fluxo alternativo 2: Exceder tempo**

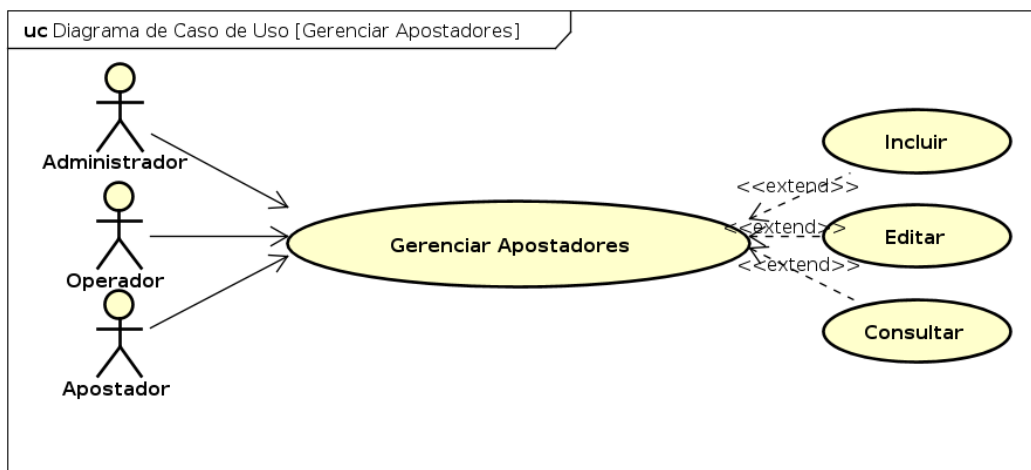
No item 3, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

**Fluxo alternativo 3: Cancelar cadastro.**

Em qualquer momento, o usuário pode cancelar, e o sistema retornando ao início no Menu Principal.

**Pós-condições:**

Os dados do usuário foram alterados no sistema ou o sistema exibiu a sua página inicial, por cancelamento da operação.

**[RF006] USE CASE<cadastro externo de apostador> - Everton****Objetivo:**

O sistema deverá permitir os processos de cadastro e autenticação através de redes sociais.

**Pré-Condições:**

O usuário possuir conta no Facebook.

**Iniciado Por:**

Pelo Apostador, quando necessário, através da tela principal de autenticação do sistema.

**Fluxo principal:**

1. O apostador seleciona a opção para se cadastrar;
2. O sistema disponibiliza um frame solicitando login e senha do Facebook do usuário;
3. O apostador preenche as informações de login e senha;
4. O sistema verifica se o usuário já foi cadastrado anteriormente no sistema;
5. O sistema requisita a entidade Facebook a autenticação do usuário;

6. O sistema retorna a confirmação da autenticação e solicita se o usuário deseja confirmar;
7. O apostador confirma a operação;
8. O sistema grava os dados pessoais do usuário no banco de dados de acordo com as informações repassadas pelo Facebook.
9. O sistema encaminha para a tela do Menu Principal;

**Fluxo alternativo 1: Exceder tempo**

No item 3, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

**Fluxo alternativo 2: Cancelar cadastro.**

Em qualquer momento, o usuário pode cancelar, e o sistema retornando ao início no Menu Principal.

**Fluxo alternativo 3: Já possui cadastro no Bolão Web.**

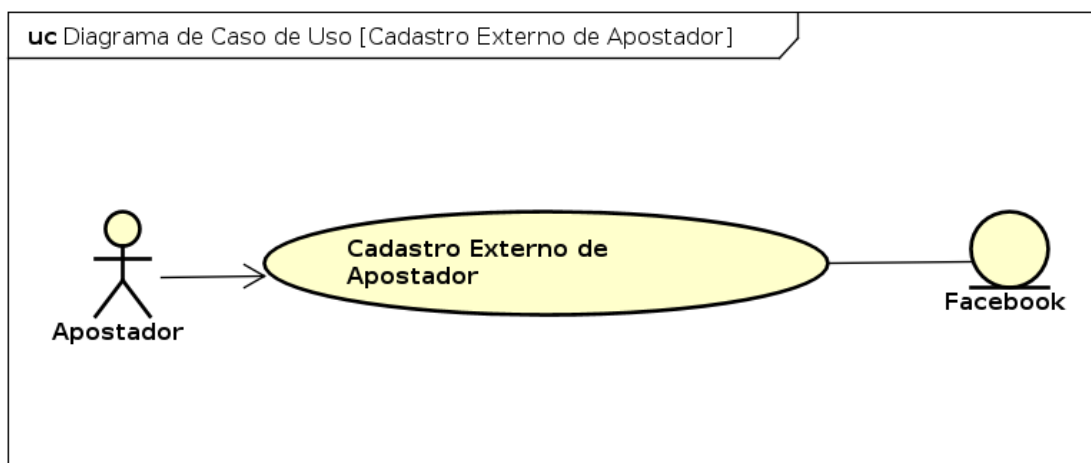
No item 4, caso o sistema retorne uma posição positiva sobre o cadastro do usuário, o fluxo continua a partir do item 9.

**Fluxo alternativo 4: Não possui cadastro no Facebook.**

No item 6, caso o sistema receba o retorno de que o usuário não possui conta no Facebook, o fluxo é interrompido e uma mensagem é exibida solicitando que tal providência seja tomada.

**Pós-condições:**

Os dados do usuário são gravados no sistema, o usuário possui permissão para acesso ao Menu Principal do sistema com todas as funcionalidades que lhe tem direito.



**[RF007] USE CASE<gerenciar palpites> - Marcelo**

**Objetivo:**

Permitir que o usuário tente adivinhar o placar dos jogos que estiverem disponíveis no site do Bolão.

**Pré-Condições:**

O usuário deve estar logado no sistema.

**Iniciado por:**

De acordo com cada partida cadastrada pelo Administrador no sistema, o usuário poderá definir seu palpite até um dia antes da rodada corrente.

Fluxo principal:

1. O usuário acessa a página inicial do site do Bolão;
2. <<include>>Autenticar usuário;
3. O usuário seleciona a opção no sistema de fazer o palpite;
4. O usuário seleciona a partida na qual deseja cadastrar um palpite;
5. O usuário informa o palpite da partida selecionada;
6. O sistema mostra as informações dos dados inseridos e das partida selecionadas;
7. O usuário confirma os dados passados ao sistema;
8. O sistema realiza a inclusão dos dados informados no cadastro de palpite na partida desejada;
9. O sistema confirma o palpite e convida o usuário a fazer um novo ou retornar a página inicial do site.

### Fluxo alternativo 1: Corrigir dados do palpite

No item 7, o sistema ao validar os dados do palpite, detecta que alguns dados considerados como obrigatórios não foram preenchidos ou que dados foram preenchidos incorretamente.

5a. O sistema reapresenta os dados do palpite indicando o(s) erro(s);

5b. O usuário corrige o(s) erro(s) do palpite;

Retorna ao fluxo principal no item 6.

### Fluxo alternativo 2: Cancelar operação

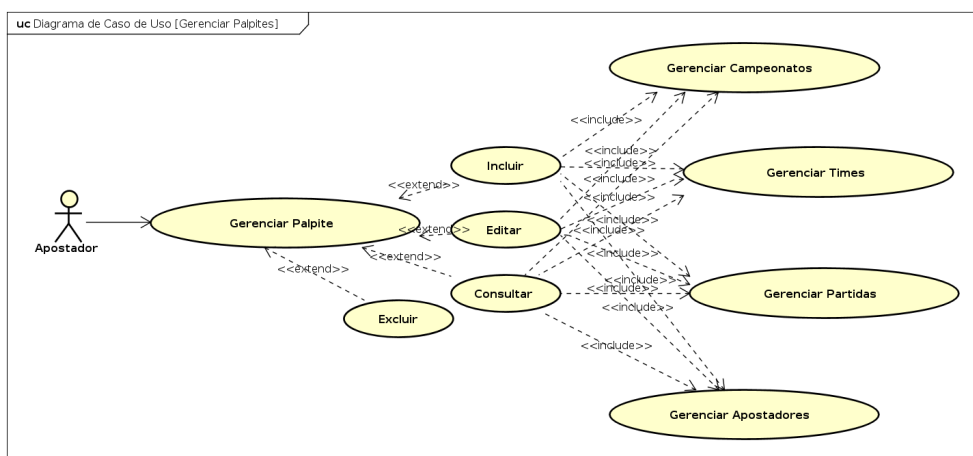
Nos itens 3 ou 4 ou 5 ou 7, o usuário cancela a operação. O sistema retorna à página inicial do sistema.

### Fluxo alternativo 3: Exceder tempo

No item 7, caso o usuário não confirme os dados e fique mais de 30 minutos inativo, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.

### Pós-condições:

O palpite do usuário foi cadastrado no sistema ou o sistema exibiu a sua página inicial, por cancelamento da operação.



## [RF008] USE CASE<processar resultados e distribuir pontos> - Everton

Objetivo:

Ao fim de uma partida, os operadores são alertados e devem cadastrar o placar e o sistema deve percorrer todos os palpites da respectiva partida distribuindo aos apostadores dois pontos caso acerte o placar exato e um ponto caso acerte o time vencedor.

**Pré-condições:**

Existir uma partida com pelo menos um palpite e com data e hora inferior a atual.

**Iniciado por:**

Processo do sistema que verifica constantemente se existe alguma partida já ocorrida.

**Fluxo principal:**

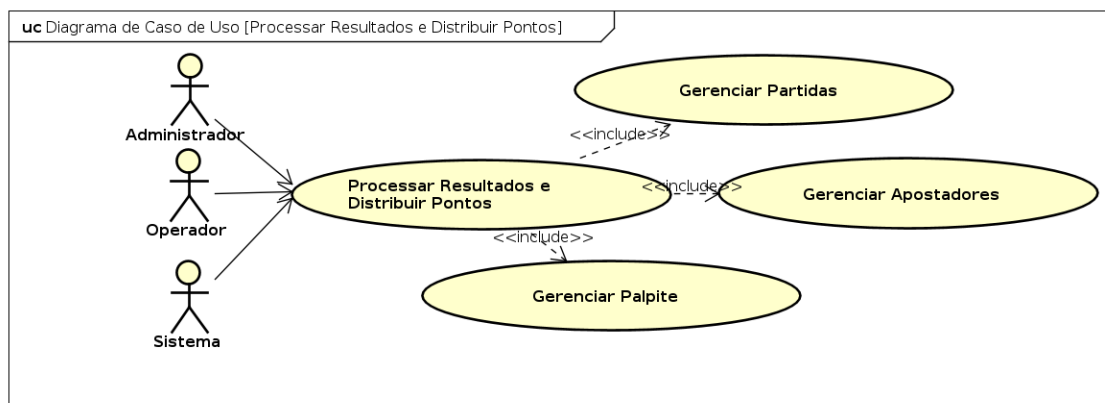
1. O sistema identifica uma partida com data e hora inferior a atual.
2. O sistema notifica a todos os operadores por e-mail com um link direto para a partida.
3. O usuário recebe o e-mail e acessa o link.
4. O usuário informa o placar do jogo e seleciona a opção para salvar os dados.
5. O sistema atualiza a partida no banco de dados.
6. O sistema exibe uma mensagem informando o sucesso da operação.
7. O sistema carrega todos os palpites relacionados a essa partida.
8. O sistema, para cada palpite, soma aos pontos do apostador dois pontos caso acerte o placar exato e um ponto caso acerte o time vencedor e envia um e-mail ao apostador informando de seus resultados.

**Fluxo alternativo: Cadastro do placar já efetuado**

Após o item 3, caso a partida já possua cadastro de placar, é exibida uma mensagem informando o usuário e solicitando que ele valide e confirme o placar. Após isso, o processo é finalizado.

**Pós-condições:**

A partida já realizada fica com o placar cadastrado e os apostadores que deram palpites nela, recebem os seus pontos.



**[RF009] USE CASE<ranking de apostadores> - Guilherme**  
**Objetivo:**

O sistema deverá permitir consultar o ranking dos Apostadores através da página do Bolão. A consulta é permitida a qualquer visitante do site. No ranking deve constar classificação, nome e pontos de cada apostador em ordem decrescente da classificação.

**Pré-Condições:**

Nenhuma pré-condição, o usuário simplesmente deve ir ao sistema e realizar a consulta.

**Iniciado por:**

Qualquer visitante ou usuário do site.

**Fluxo principal:**

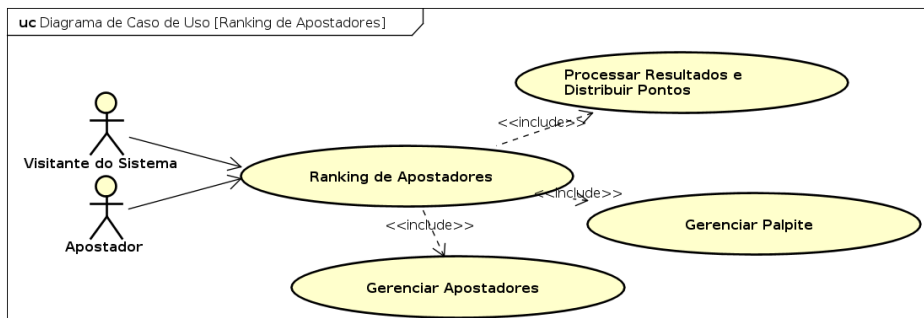
1. O usuário acessa a página iniciado do sistema, o site do Bolão.
2. O usuário seleciona a opção de consultar ranking de apostadores
3. O sistema apresenta o ranking

**Fluxo alternativo: Mostrar posição**

No item 3, caso o usuário logado seja um apostador, a posição dele deverá estar em destaque.

**Pós-condições:**

O ranking foi exibido na tela.

**[RF010] USE CASE<consulta fácil de palpites> - Marcelo****Objetivo:**

Permitir ao usuário realizar consultados palpites já inseridos no sistema, assim como as partidas já realizadas da rodada, posição no ranking e a quantidade de pontos.

**Pré-Condições:**

Usuário logado no sistema e com permissões ao nível de apostador.

**Iniciado Por:**

Usuário com nível de permissão devidamente logado no sistema.

**Fluxo principal:**

1. O sistema exibe a tela inicial do site;
2. O usuário seleciona a consulta fácil de palpites;
3. O sistema retorna uma página onde constam as informações de palpites já realizados, pontuação e a posição no ranking geral;
4. O usuário confere os dados e seleciona voltar, retornando a página inicial do sistema.

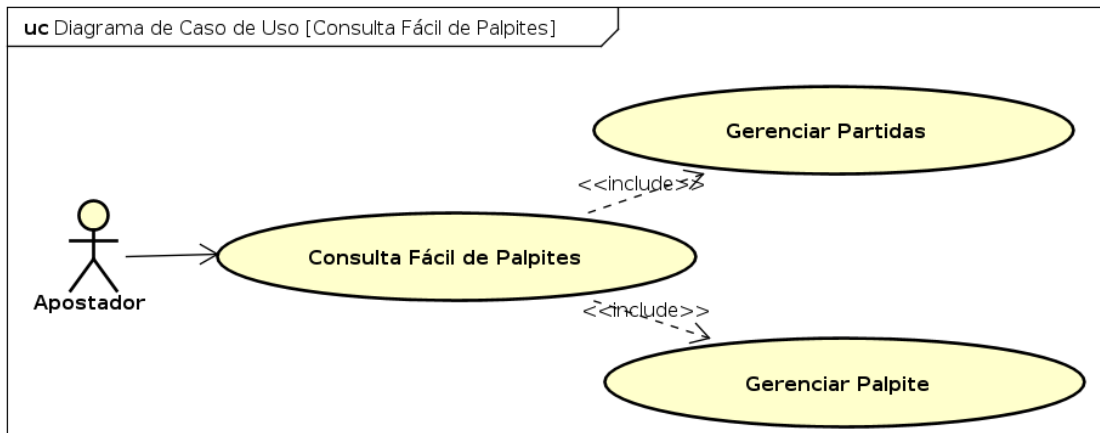
**Fluxo alternativo: Exceder tempo**

No item 7, caso o usuário não retorne nenhuma ação dentro de 10 minutos, a operação é cancelada pelo sistema. Nesse caso o sistema retorna à sua página inicial.



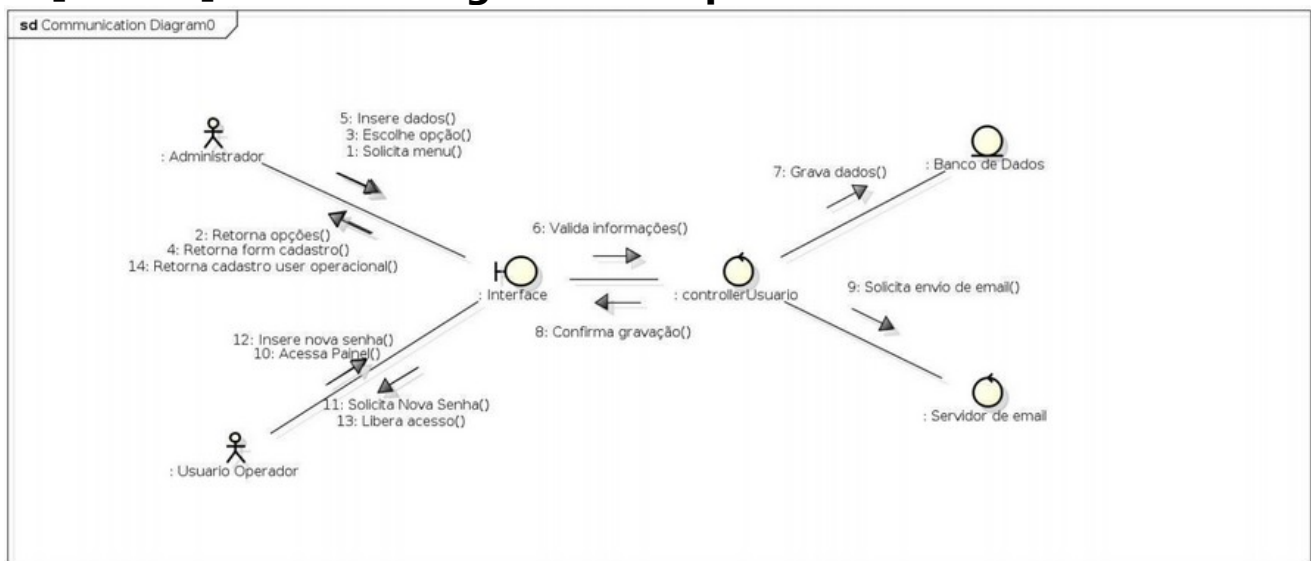
### Pós-condições:

O sistema retorna a página inicial do sistema após o usuário conferir os dados ou exceder o tempo limite.



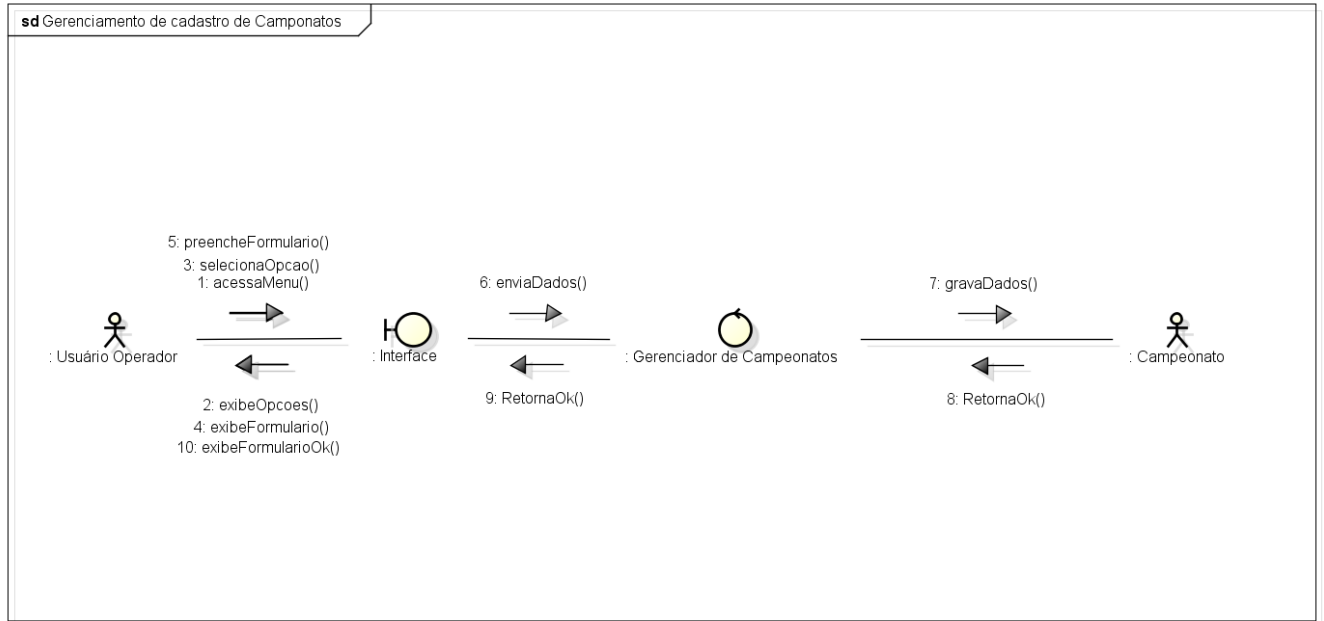
## 9 MODELO DE ANÁLISE

### [RF001] USE CASE<gerenciar operadores> - Guilherme



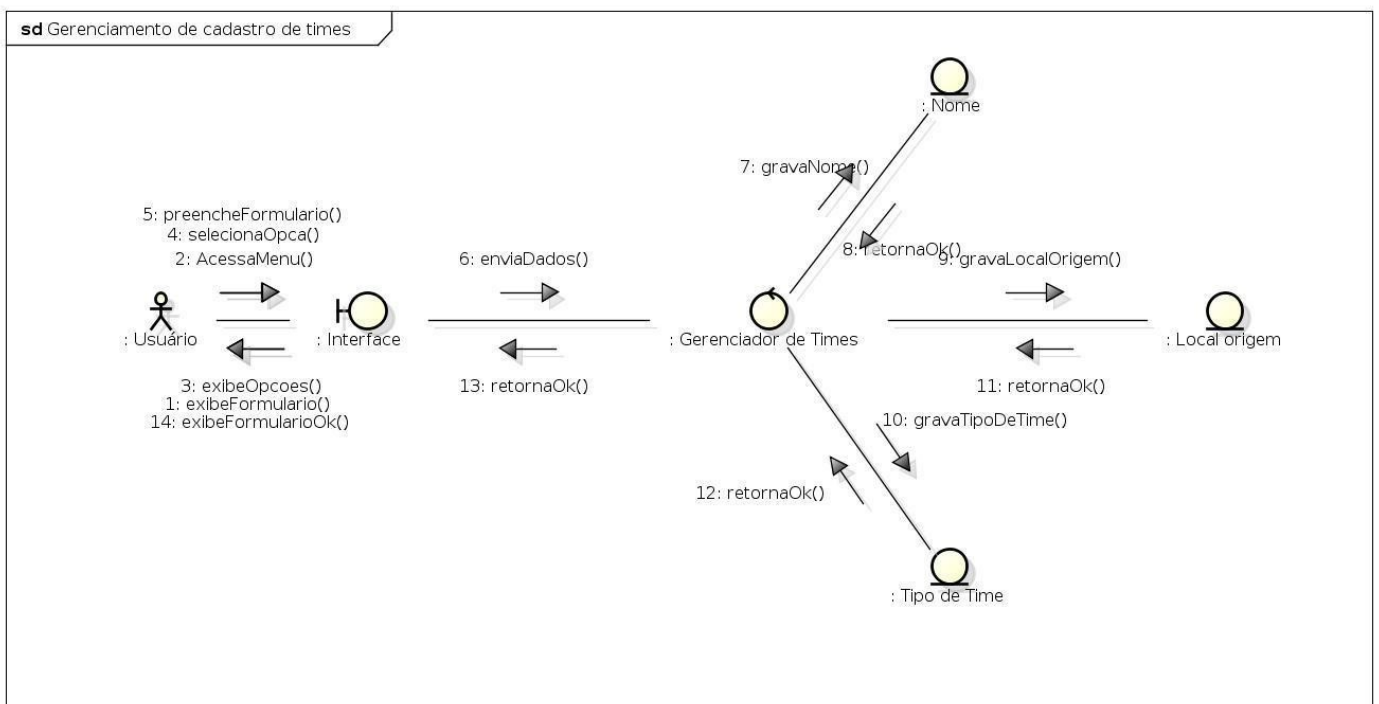
powered by Astah

## [RF002] USE CASE<gerenciar campeonatos> - Marcelo



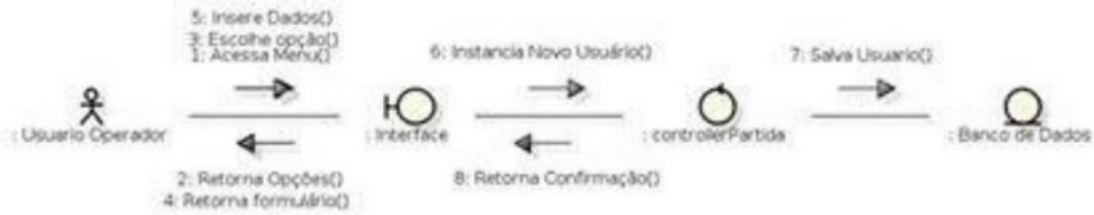
powered by Astah

## [RF003] USE CASE<gerenciar times> - Everton



powered by Astah

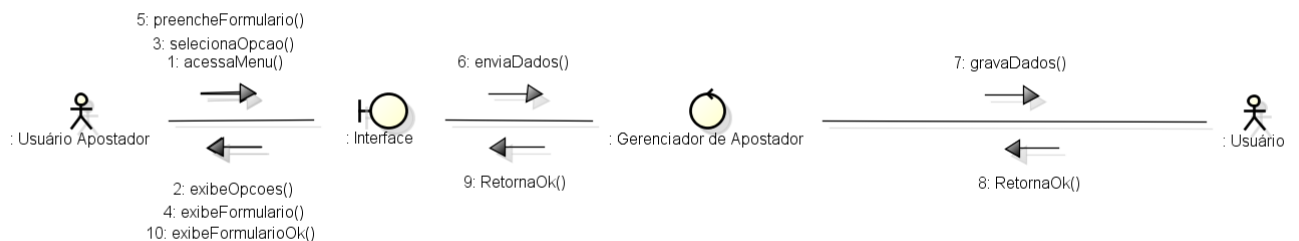
## [RF004] USE CASE<gerenciar partidas> - Everton



powered by Astah

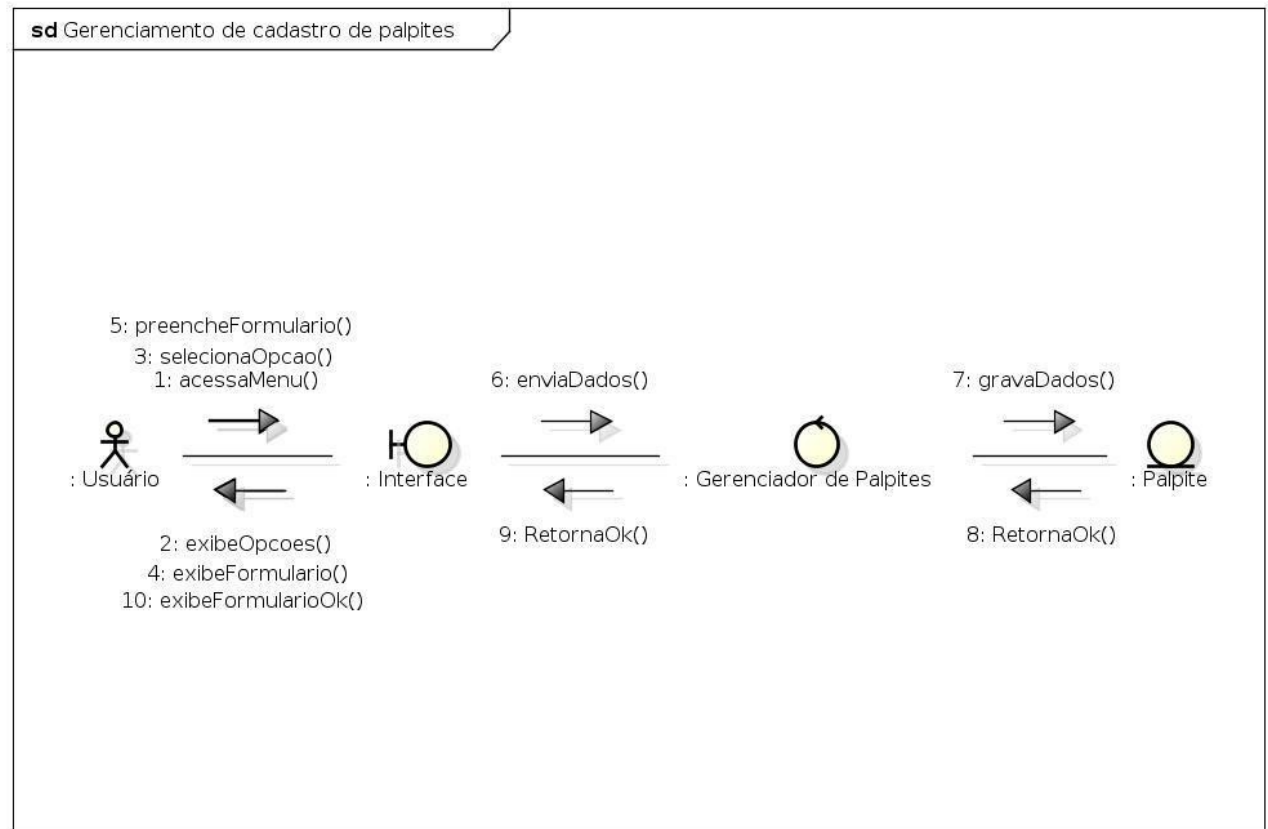
## [RF005] USE CASE<gerenciar apostador> - Guilherme

sd Gerenciamento de cadastro de apostador



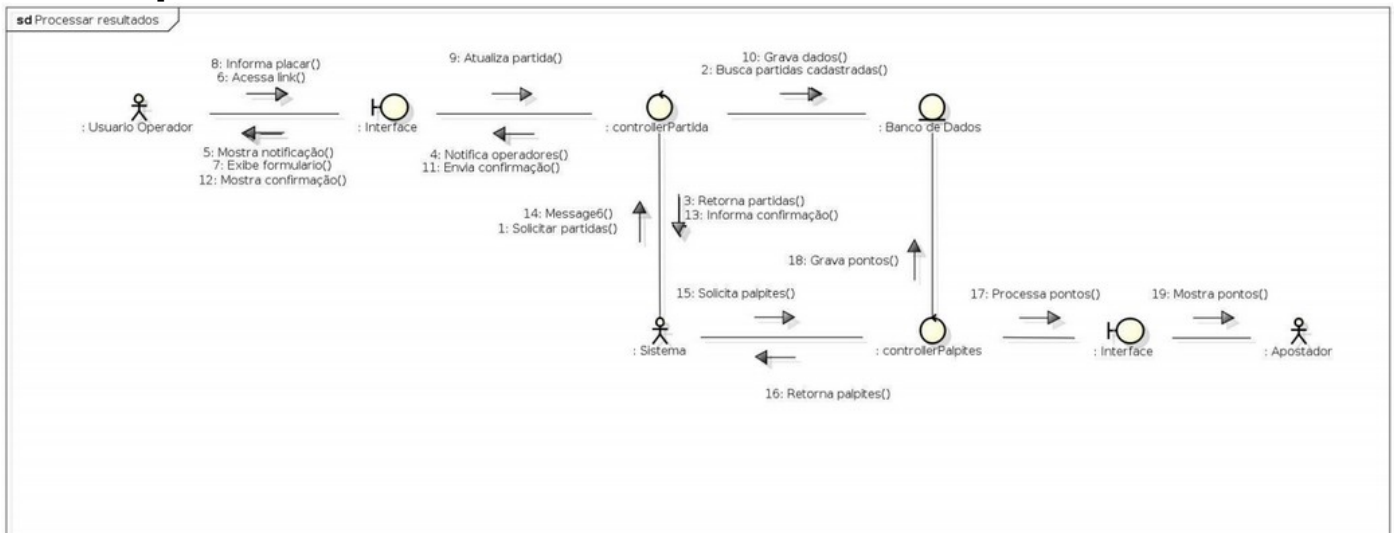
powered by Astah

## [RF007] USE CASE<gerenciar palpites> - Marcelo



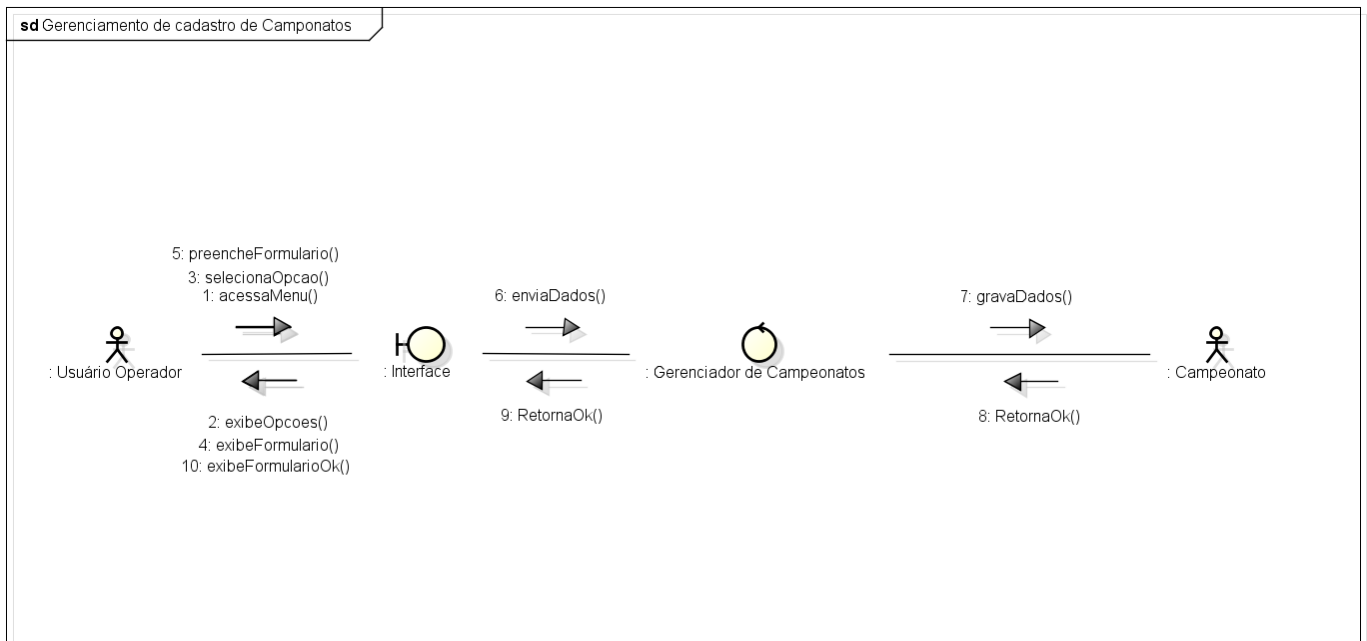
powered by Astah

## [RF008] USE CASE<processar resultados e distribuir pontos> - Everton

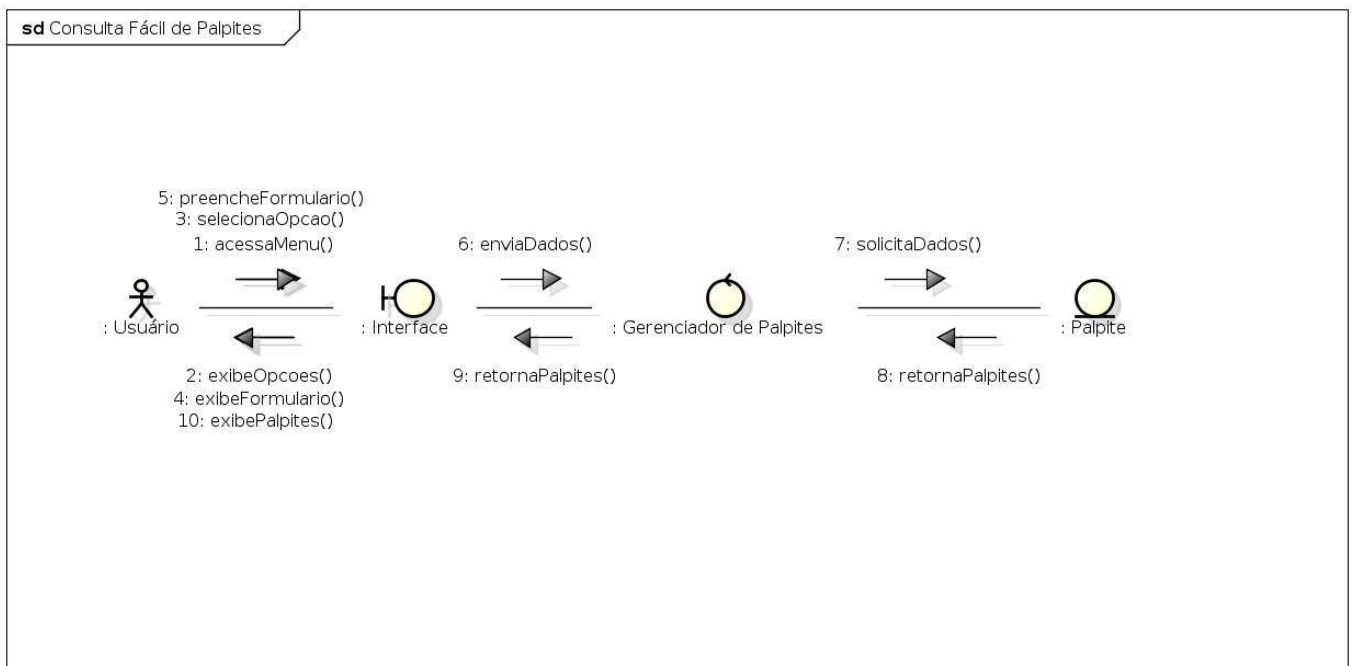


powered by Astah

## [RF009] USE CASE<ranking de apostadores> - Guilherme



## [RF010] USE CASE<consulta fácil de palpites> - Marcelo



## 10 PLANEJAMENTO DAS ITERAÇÕES

### ITERAÇÃO 1:

USE CASE<gerenciar administradores> - Everton  
 USE CASE<gerenciar operadores> - Guilherme

USE CASE<gerenciar campeonatos> - Marcelo  
 USE CASE<gerenciar times> - Everton  
 USE CASE<gerenciar partidas> - Everton

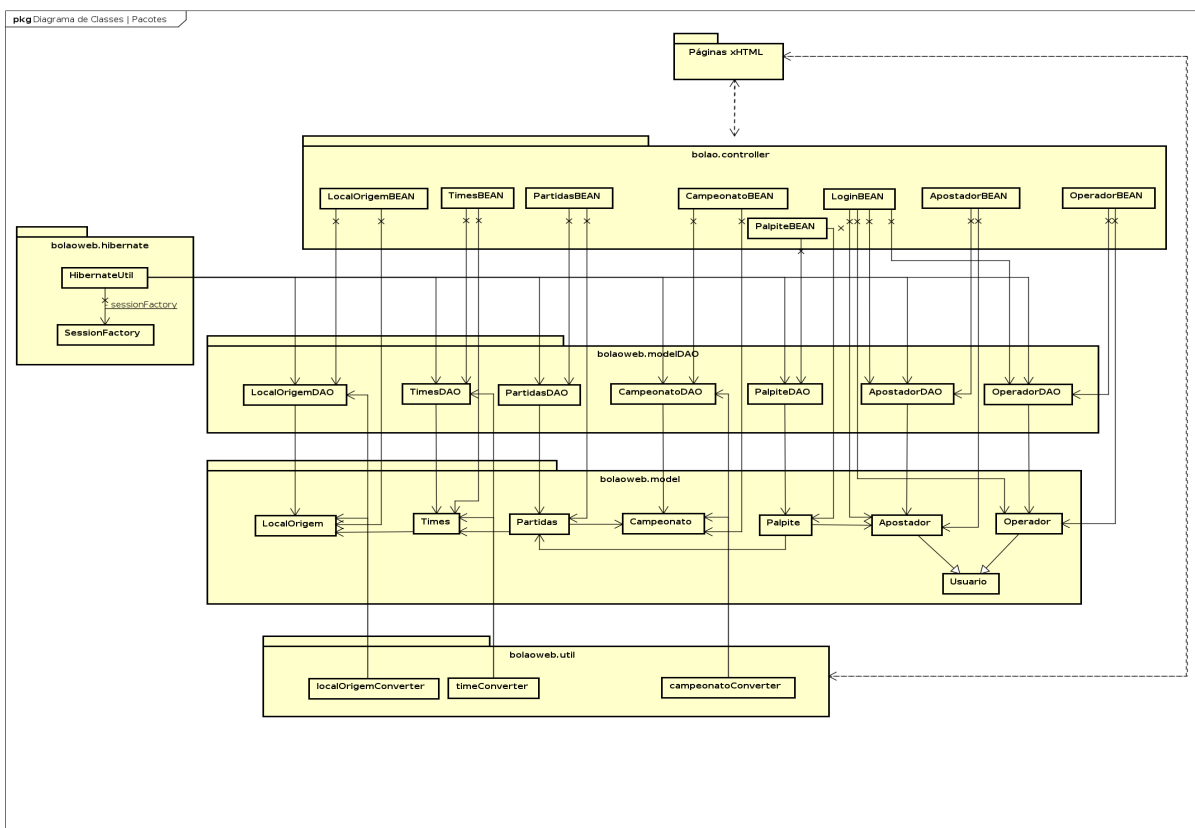
## ITERAÇÃO 2:

USE CASE<gerenciar apostador> - Everton  
 USE CASE<cadastro externo de apostador> - Guilherme  
 USE CASE<gerenciar palpite> - Marcelo  
 USE CASE<processar resultados e distribuir pontos> - Everton  
 USE CASE<consultar ranking de apostadores> - Guilherme  
 USE CASE<consulta fácil de palpites> - Marcelo

# 11 DIAGRAMA DE CLASSES DE PROJETO

## 11.1 Diagrama de Classes com Pacotes

O Diagrama de Classes com Pacotes apresenta uma visão geral de todas as classes do sistema, enfatizando seus respectivos pacotes e o relacionamento entre classes porém abstraindo informações referentes a atributos e operações.



## 11.2 Diagrama de Classes de Projeto

O Diagrama de Classes de Projeto demonstra todas as classes do sistema e o relacionamento dentre elas, enfatizando também todos os atributos e operações.

*Pelo fato do diagrama estar muito grande impossibilitando a visualização correta, o mesmo foi enviado no mesmo diretório da documentação no GitHub em formato **\*.png** e **\*.asta**.*