TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

THÁLLYS LISBOA SIMÕES E IGOR PEREIRA DE ALMEIDA

**TRABALHO 2**

PROPOSTA DE ESPECIFICAÇÃO DE SOFTWARE

JANUÁRIA – MINAS GERAIS

2016

THÁLLYS LISBOA SIMÕES E IGOR PEREIRA DE ALMEIDA

**TRABALHO 2**

PROPOSTA DE ESPECIFICAÇÃO DE SOFTWARE

Trabalho apresentado ao Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) do IFNMG – Instituto Federal do Norte de Minas Gerais – Campus Januária, para a disciplina [Programação de Sistemas Corporativos].

Prof. Petrônio Cândido Silva

Januária - Minas Gerais

2016

[1. MINIMUNDO 4](#_Toc469607098)

[2. OBJETIVOS 4](#_Toc469607099)

[2.1 OBJETIVO GERAL 4](#_Toc469607100)

[2.2 OBJETIVOS ESPECÍFICOS 4](#_Toc469607101)

[3. DOCUMENTO DE REQUISITOS 5](#_Toc469607102)

[3.1 Proposta do documento 5](#_Toc469607103)

[3.1.1 Proposito do documento 5](#_Toc469607104)

[3.1.2 Escopo do produto 5](#_Toc469607105)

[3.1.3 Abreviações 5](#_Toc469607106)

[3.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS 5](#_Toc469607107)

[4. MODELAGEM CONCEITUAL (DER) 10](#_Toc469607108)

[5. DIAGRAMA DE PACOTES UML ( ) 11](#_Toc469607109)

[6. DIAGRAMAS DE CASO DE USO 11](#_Toc469607110)

[7. MODELO LÓGICO DO BANCO DE DADOS (NA 3FN) 22](#_Toc469607111)

[8. DIAGRAMA LÓGICO ORIENTADO À OBJETOS (DIAGRAMA DE CLASSES UML) DA CAMADA DE LÓGICA DO NEGÓCIO (DOMAIN MODEL) 23](#_Toc469607112)

[9. DESCRIÇÃO DA ARQUITETURA DE SOFTWARE 23](#_Toc469607113)

[10. DIAGRAMA DE PACOTES 24](#_Toc469607114)

[11. DIAGRAMA DE SEQUÊNCIA 24](#_Toc469607115)

[11.1 Diagrama de sequência de login. 24](#_Toc469607116)

[11.2 Diagrama de sequência de cadastro de jogador 25](#_Toc469607117)

[11.3 Diagrama de sequência de alugar dependências 25](#_Toc469607118)

[11.4 Diagrama de sequência de cadastrar patrocinador 26](#_Toc469607119)

[11.5 Diagrama de sequência de verificar despesas 26](#_Toc469607120)

[12. DEFINIÇÃO DAS TECNOLOGIAS QUE SERÃO UTILIZADAS NA IMPLEMENTAÇÃO 27](#_Toc469607121)

[13. DEFINIÇÃO DO AMBIENTE ALM (IDE, CONTROLE DE VERSÃO, GERENCIAMENTO DE PROJETOS, BUGS, FERRAMENTAS CASE 28](#_Toc469607122)

[13.1. IDE 28](#_Toc469607123)

[13.2. Controle de versão 29](#_Toc469607124)

[13.3. Gerenciamento de projetos 29](#_Toc469607125)

[13.4. Ferramentas CASE 29](#_Toc469607126)

[14. DIAGRAMA DE COMPONENTES 30](#_Toc469607127)

[15. DIAGRAMA DE IMPLANTAÇÃO 30](#_Toc469607128)



**1. MINIMUNDO**

Um clube de futebol deseja se modernizar e para tal, a implantação de um sistema capaz de gerenciar de forma eficaz e produtiva todas as dependências ... "como um todo" se faz necessária.

O clube de futebol deseja armazenar e manter o cadastro/informações de todos os funcionários do clube. Um funcionário possui um código que o identifica, nome, cargo, salário, endereço e RG. Um funcionário pode possuir um chefe, que por sua vez também é um funcionário. Este clube também possui várias dependências (Exemplo: campos, hotel, hospital, sala de jogos) que devem ser cadastradas no sistema. Desta maneira o sistema deve cadastrar quais dependências poderão ser alugadas por outros clubes ou seleções de outros países. A dependência possui um código, nome e capacidade. Deve ser registrada a data das visitas de outras equipes ou pessoas para controle.

O clube ainda guarda todas modalidades disponíveis (Exemplo: Sub-20, Juniores, Sub- 17). Cada modalidade tem código, nome e ainda possui um funcionário que o coordena por um determinado período. A modalidade esportiva tem descrição.

É necessário que se armazene ainda todos os atletas do clube. Cada atleta possui um código, nome, RG, endereço, código da federação e foto. Cada atleta pode participar de um ou mais equipes e é registrado o período que um atleta fez parte de uma equipe. As equipes estão relacionadas com as modalidades e podem conter um ou vários atletas. Uma equipe possui código e descrição.

Uma equipe disputa campeonatos, os quais possuem um código e descrição. O clube armazena os seus títulos conquistados em cada campeonato. São guardados o código e a data do título. A participação de uma equipe em um campeonato recebe patrocínio. Um patrocinador pode patrocinar várias equipes em vários campeonatos. Um patrocinador deve ter código, nome e logomarca.

Todas as finanças e rendas do clube serão cadastradas afim de realizar um controle de gastos e gestão de todo clube (contratações, venda de atletas, empréstimos,).

**2. OBJETIVOS**

**2.1 OBJETIVO GERAL**

Desenvolver um sistema com o objetivo de auxiliar um clube de futebol, na organização de suas principais atividades, seja elas de controle gestão e controle financeiro, de forma clara e eficiente, mostrar e listar todo o traçado do clube, desde sua diretoria, até a disputa de campeonatos, gestão de patrocinadores.

**2.2 OBJETIVOS ESPECÍFICOS**

- Levantar as informações sobre gestão do clube para inserir no sistema.

- Tornar de fácil acesso, todos os principais dados, do clube.

- Implementar o sistema para uso dos principais funcionários

- Gerenciamento do clube, por completo.

**3. DOCUMENTO DE REQUISITOS**

**3.1 Proposta do documento**

**3.1.1 Proposito do documento**

A proposta desse documento é a especificação dos requisitos para o sistema Master Club. O Master Club é um sistema que auxiliará as principais tarefas para o funcionamento de um clube de acordo com as suas necessidades. Os documentos serão armazenados em um banco de dados, onde apenas os usuários só terão acesso se os mesmos estiverem devidamente cadastrados.

**3.1.2 Escopo do produto**

O sistema deve auxiliar de maneira eficaz no gerenciamento do clube, fazendo tarefas como: Cadastro de funcionários, Cadastro de dependências, fazer exclusão de dados que não são mais necessários, Atualizações, Modificações de informações, Consultar, gerar relatórios e resgatar.

**3.1.3 Abreviações**

Durante a apresentação do documento algumas abreviações serão utilizadas.

Abreviações:

* RF: Requisitos funcionais
* RNF: Requisitos não funcionais

**3.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS**

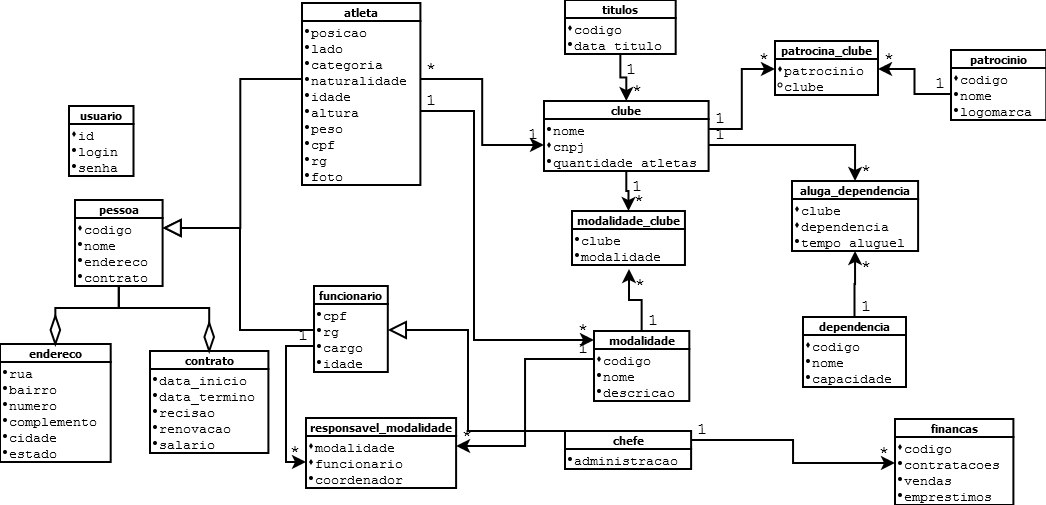
**Responsáveis**

|  |  |
| --- | --- |
| **NOME** | **ASSINATURA** |
| Igor Pereira de Almeida | Igor |
| Thállys Lisboa | Thállys |

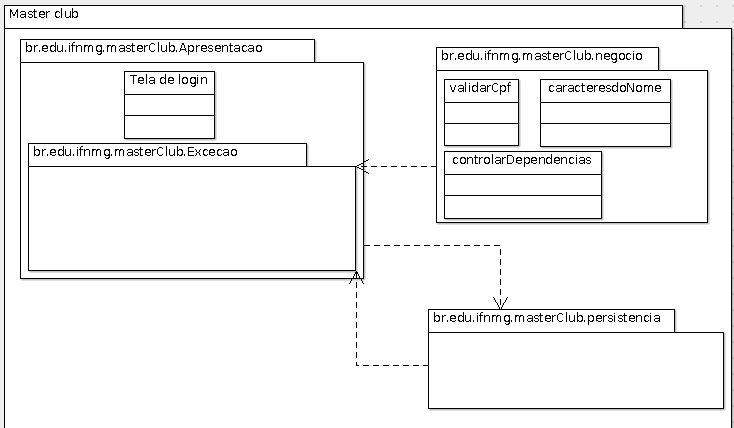
**Requisitos funcionais e não funcionais**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RF-1 Tela de Login | | | | |
| O sistema deve ter um controle de acesso autenticado com senha, para que cada tipo de usuário tenha acesso apenas a funcionalidade que é libera, para o seu uso. Ex (O administrador não terá acesso a alguns dados que o gerente tem e vice-versa). | | | | |
| **Requisitos não funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| RNF-1.1-Campo Usuário | O campo Usuário deve ter no mínimo 4 caracteres e no máximo 10. |  | ( ) | ( X ) |
| RNF-1.2-Campo Senha | A senha não pode ter menos de 8 caracteres. |  | ( ) | ( X ) |
| RNF-1.3-Formato dos caracteres do campo senha. | Os caracteres do campo senha devem ser formatados de forma a não mostrar de forma legível a senha digitada. |  | ( ) | ( X ) |
| RNF-1.4-Restrições dos campos | Os campos da tela de login não podem ficar em branco ( se isso ocorrer o usuário deverá receber uma mensagem mostrando que os campos estão em branco e que para continuar ele necessita de preenche-los). |  | ( ) | ( X ) |
| RNF-1.5-Campos preenchidos de forma incorreta. | O sistema deve informar ao usuário através de mensagem (“Usuário e/ou senha incorretos favor digitar novamente”) se um dos campos foi preenchido de forma incorreta |  | ( ) | ( X ) |
| RNF-1.6-Mensagem de sucesso | O usuário ao clicar no botão entrar o sistema deve antes de deixar acessar mostrar uma mensagem de acesso feito com êxito. |  | ( X ) | ( ) |
| RF-2 Tela principal | | | | |
| A tela principal é onde estarão os menus para os principais serviços que serão ofertados pelo sistema. | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| RNF-2.1-Menu cadastro | O menu cadastro irá armazenar todos os cadastros que são necessários dentro do software. |  | ( ) | ( X ) |
| RNF-2.2-Menu Departamentos | Menu responsável pelo controle dos departamentos devidamente cadastrados no sistema. |  | ( ) | ( X ) |
| RNF-2.3-Menu campeonatos | O usuário irá usar esse menu para cadastrar os campeonatos em que o clube será acionado para o uso das suas localidades. |  | ( ) | ( X ) |
| RNF-2.4-Menu financeiro | Ao clicar no menu financeiro o usuário deverá ter todas as opções sobre finanças de um clube. |  | ( ) | ( X ) |
| RNF-2.5-Controle do menu financeiro | O usuário deve estar logado como administrador ou como gerente para ter acesso as funcionalidades desse menu. |  | ( X ) | ( ) |
| RNF-2.5-Menu Sobre | Ao clicar sobre esse menu deve ser apresentada informações sobre o clube para o usuário. |  | ( ) | ( X ) |
| RNF-2.6-Menu sair | Quando o usuário clicar sobre esse menu o mesmo deverá fazer o logout. |  | ( ) | ( X ) |
| RNF-2.6- Mensagem menu sair | O usuário ao clicar nesse menu deverá, ser exibido uma mensagem perguntando se o usuário realmente deseja sair do sistema e com as opções sim ou não. |  | ( ) | ( X ) |
|  |  |  |  |  |
| RF-3 Cadastro de jogadores | | | | |
| O sistema deve fazer o cadastro de todos os jogadores do clube. | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| RNF-3.1-Nome | O nome do jogador deve ter entre 4 e 255 caracteres. |  | ( ) | ( X ) |
| RNF-3.2-Categoria | Deve ser inserida uma categoria para o jogador exemplo base. |  | ( ) | ( X ) |
| RNF-3.3-Naturalidade | O jogador deve indicar a qual nacionalidade ele pertence. |  | ( ) | ( X ) |
| RNF-3.4-posição | O jogador deve identificar em qual posição ele joga. Obs jogador pode ter mais de uma posição |  | ( ) | ( X ) |
| RNF-3.4-Pé/chute | Opção de chute do jogador |  | ( X ) | ( ) |
| RNF-3.5-Peso | O jogador deve inserir qual o seu peso no momento. |  | ( X ) | ( ) |
| RNF-3.6-Idade | O jogador deve identificar no seu cadastro a sua idade |  | ( ) | ( X ) |
| RNF-3.7-Altura | O jogador deve mostrar ao sistema qual a sua altura |  | ( ) | ( X ) |
| RNF-3.8-Inicio | O início marca onde começa a valer o contrato. |  | ( ) | ( X ) |
| RNF-3.9-Termino | Determina o tempo de termino do contrato |  | ( ) | ( X ) |
| RNF-3.10-Salario | Valor a ser pago ao atleta contratado durante o tempo de contrato |  | ( ) | ( X ) |
| RNF-3.11-Botão salvar | Deverá salvar todas as alterações feitas se os campos estiverem todos preenchidos. |  | ( ) | ( X ) |
| RNF-3.12-Erro salvar | Se algum dos campos não estiver preenchido deve-se mostrar uma mensagem, “campo tal não foi preenchido de forma correta favor preenche-lo”. |  | ( ) | ( X ) |
| RNF-3.13-Botão sair | Deverá sair do cadastro de jogadores e não salvar as alterações feitas |  | ( ) | ( X ) |
| RF-4 Cadastro de membros da comissão técnica | | | | |
| O sistema deve fazer o cadastro de todos os membros da comissão técnica do clube. | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| RNF-4.1-Nome | O membro da comissão deve ter um nome que tenha de 4 até 255 caracteres |  | ( ) | ( X ) |
| RNF-4.2-Naturalidade | O membro da comissão deve informar qual a sua naturalidade |  | ( ) | ( X ) |
| RNF-4.3-Peso | O membro da comissão deve informar o seu peso atual |  | ( ) | ( X ) |
| RNF-4.4-Altura | A altura do membro deve ser medida no ato do contrato |  | ( ) | ( X ) |
| RNF-4.5-Inicio | O membro só pode começar a atuar pelo clube após a data de início estipulada no contrato |  | ( ) | ( X ) |
| RNF-4.6-Término | Após o fim do contrato o membro da comissão não pode mais atuar pelo clube a menos que haja uma renovação do contrato |  | ( ) | ( X ) |
| RNF-4.7-Função | O membro da comissão deve exercer a sua função estabelecida em contrato |  | ( ) | ( X ) |
| RNF-4.8-Botão salvar | Deverá salvar todas as alterações feitas se os campos estiverem todos preenchidos. |  | ( ) | ( X ) |
| RNF-4.9-Erro salvar | Se algum dos campos não estiver preenchido deve-se mostrar uma mensagem, “campo tal não foi preenchido de forma correta favor preenche-lo”. |  | ( ) | ( X ) |
| RNF-4.10-Botão sair | Deverá sair do cadastro do membro e não salvar as alterações feitas |  | ( ) | ( X ) |
| RF-5 Cadastro de funcionários | | | | |
| Todos os funcionários do clube devem estar devidamente cadastrados | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| RNF-5.1-Nome | Todos os funcionários devem ter um nome com no mínimo 4 caracteres e no máximo 255 caracteres. |  | ( ) | ( X ) |
| RNF-5.2-Cargo | O usuário deve ter um cargo a ser exercido dentro do clube |  | ( ) | ( X ) |
| RNF-5.3-Naturalidade | O usuário deve informar qual a sua naturalidade |  | ( ) | ( X ) |
| RNF-5.4-Peso | O funcionário deve informar o seu peso no ato do cadastro |  | ( ) | ( X ) |
| RNF-5.5-Idade | O funcionário recém contratado deve ter uma idade mínima de 18 anos. |  | ( ) | ( X ) |
| RNF-5.6-Altura | O funcionário deverá verificar a sua altura. |  | ( ) | ( X ) |
| RNF-5.7-Inicio | O funcionário deverá cumprir com suas funções a partir do início do contrato. |  | ( ) | ( X ) |
| RNF-5.8-Termino | Após o termino do contrato o funcionário não tem mais a obrigação de fazer seus serviços diários a menos que haja a renovação do contrato |  | ( ) | ( X ) |
| RNF-5.9-Salario | O funcionário deve receber o salario definido no contrato |  | ( ) | ( X ) |
| RNF-5.10-Botão salvar | Deverá salvar todas as alterações feitas se os campos estiverem todos preenchidos. |  | ( ) | ( X ) |
| RNF-5.11-Erro salvar | Se algum dos campos não estiver preenchido deve-se mostrar uma mensagem, “campo tal não foi preenchido de forma correta favor preenche-lo”. |  | ( ) | ( X ) |
| RNF-5.12-Botão sair | Deverá sair do cadastro de funcionários e não salvar as alterações feitas |  | ( ) | ( X ) |
| RF-6 Cadastro de patrocinadores | | | | |
| Todos os clubes devem ter ao menos um patrocinador | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| RNF-6.1-Marca | Ao ser assinado o contrato fica obrigado o patrocinador a oferecer sua marca ao clube |  | ( ) | ( X ) |
| RNF-6.2-País | A marca deverá informar sob pena de multa o seu país de origem. |  | ( ) | ( X ) |
| RNF-6.3-Valor | O patrocinador deverá pagar um valor anual ou mensal estipulado no contrato |  | ( ) | ( X ) |
| RNF-6.4-Tipo de patrocínio | O patrocinador deverá informar ao clube qual o seu tipo de patrocínio. |  | ( ) | ( X ) |
| RNF-6.5-Inicio | O patrocinador deverá auxiliar o clube a partir da data de início definida no contrato. |  | ( ) | ( X ) |
| RNF-6.7-Termino | Ao finalizar o contrato do patrocinador com o clube o mesmo não terá mais suas obrigações |  | ( ) | ( X ) |
| RNF-6.8-Botão salvar | Deverá salvar todas as alterações feitas se os campos estiverem todos preenchidos. |  | ( ) | ( X ) |
| RNF-6.9-Erro salvar | Se algum dos campos não estiver preenchido deve-se mostrar uma mensagem, “campo tal não foi preenchido de forma correta favor preenche-lo”. |  | ( ) | ( X ) |
| RNF-6.10-Botão sair | Deverá sair do cadastro de patrocinadores e não salvar as alterações feitas |  | ( ) | ( X ) |

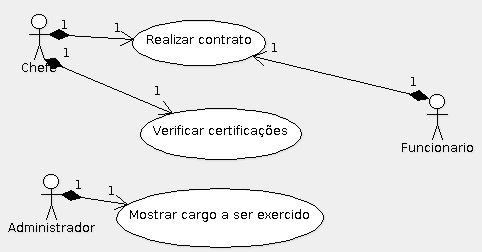
**4. MODELAGEM CONCEITUAL (DER)**

****

**5. DIAGRAMA DE PACOTES UML ( )**

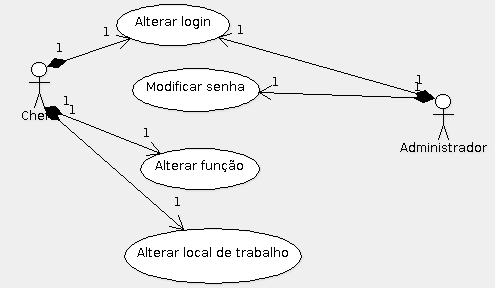


**6. DIAGRAMAS DE CASO DE USO**

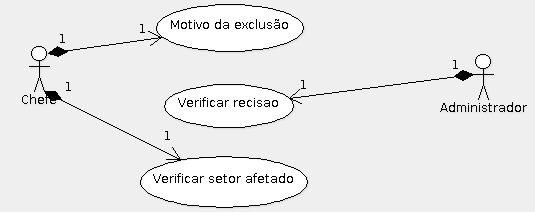
UC 1 Gerenciado de funcionários

UC 1.1 Cadastrar funcionário

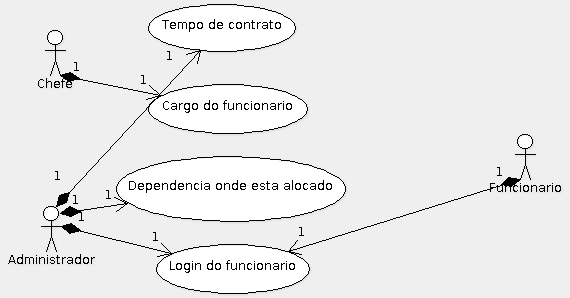
UC 1.2 Alterar funcionário



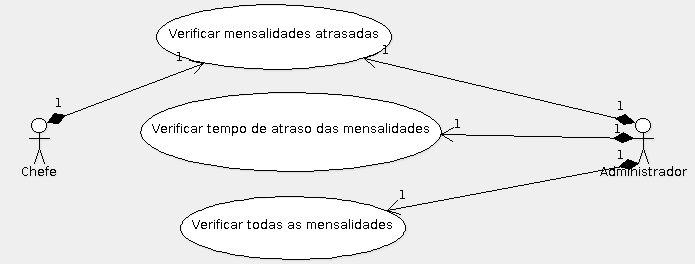
UC 1.3 Excluir funcionário



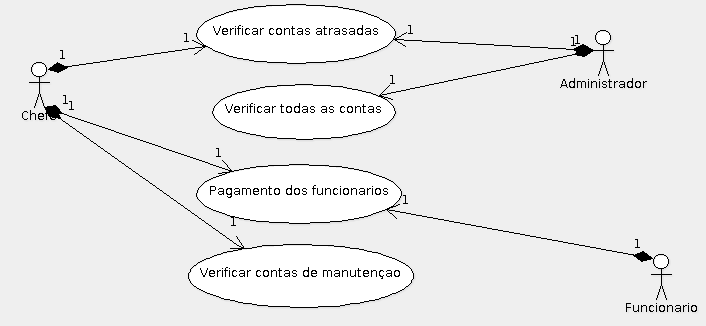
UC 1.4 Informações do funcionário



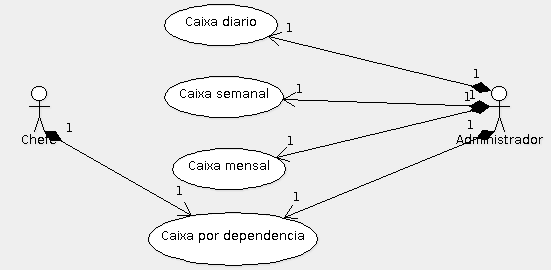
UC 2 Controle de finanças

UC 2.1 Verificar mensalidades das dependências

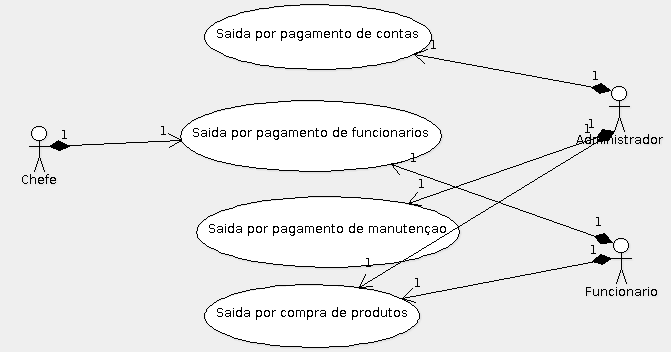
UC 2.2 Contas a pagar



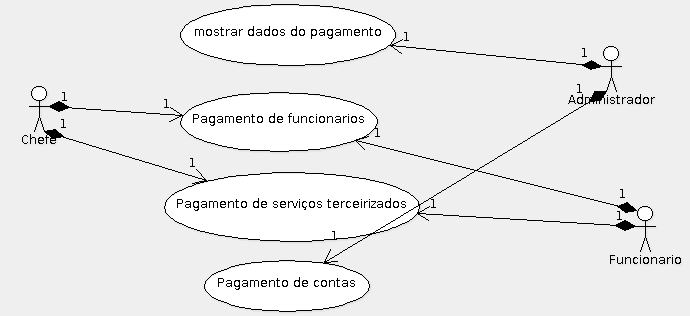
UC 2.3 Caixa



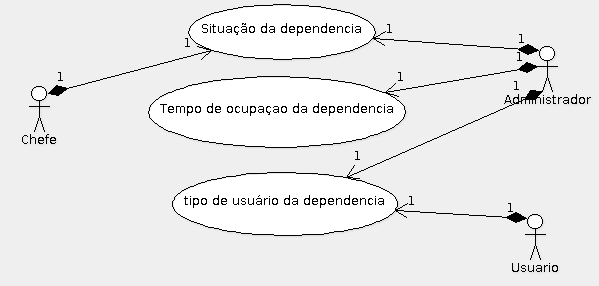
UC 2.4 Mostrar saídas

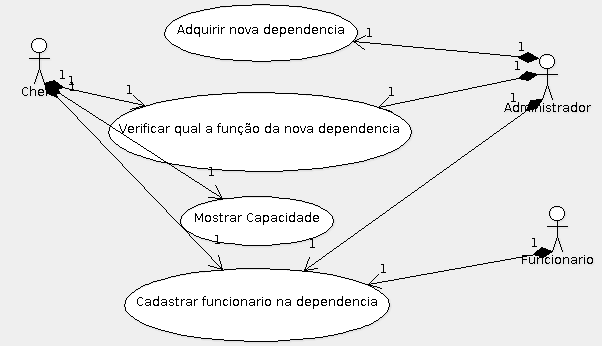


UC 2.5 Efetuar pagamento

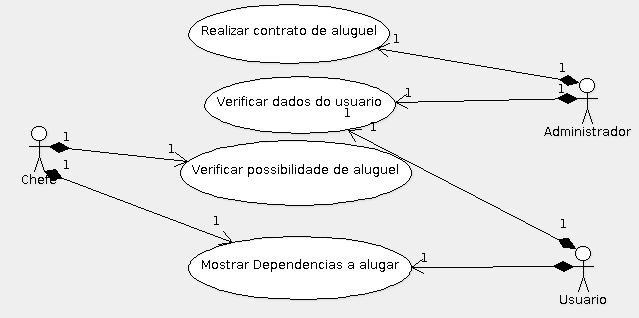


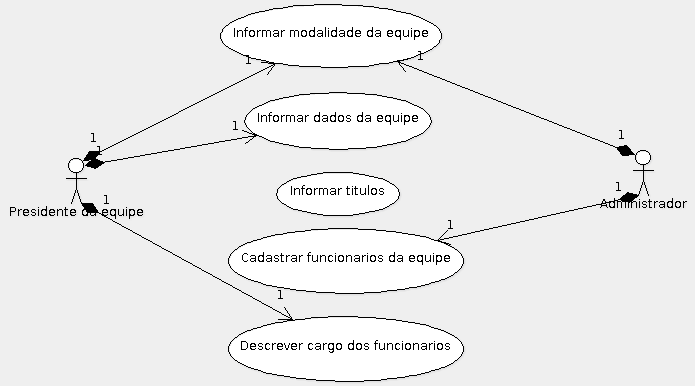
UC 3 Gerenciador de dependências

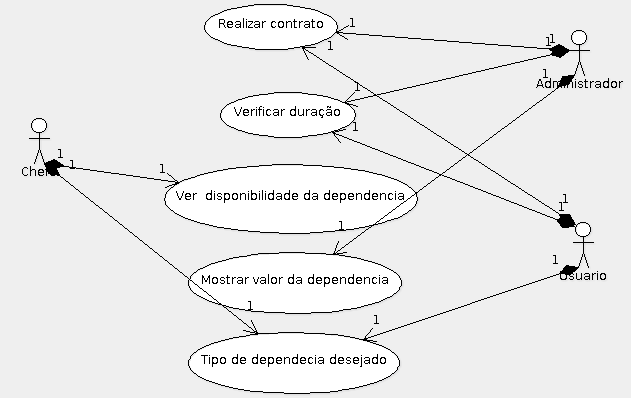
UC 3.1 Cadastrar dependência

UC 3.2 Verificar disponibilidade

UC 3.3 Realizar aluguel



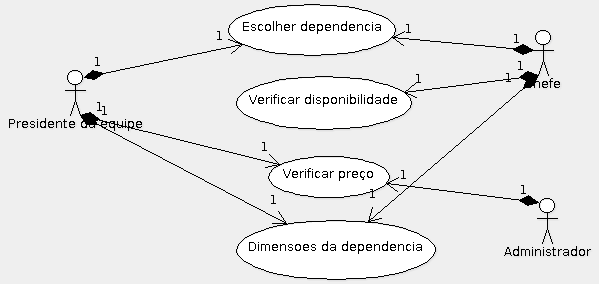
UC 3.4 Liberar para campeonato



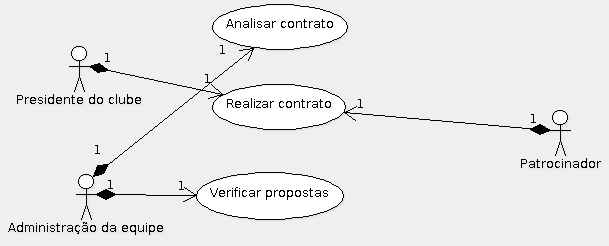
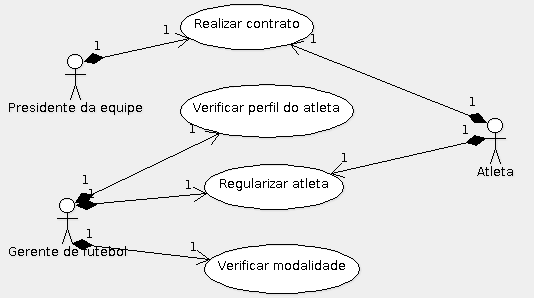
UC 4 Administrar equipes

UC 4.1 Cadastrar equipe

UC 4.2 Alugar dependência

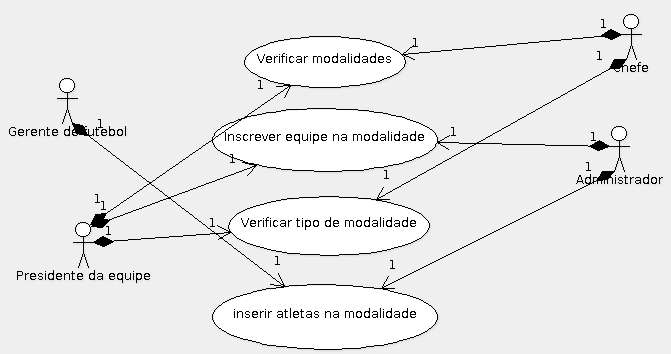


UC 4.3 Cadastrar atletas

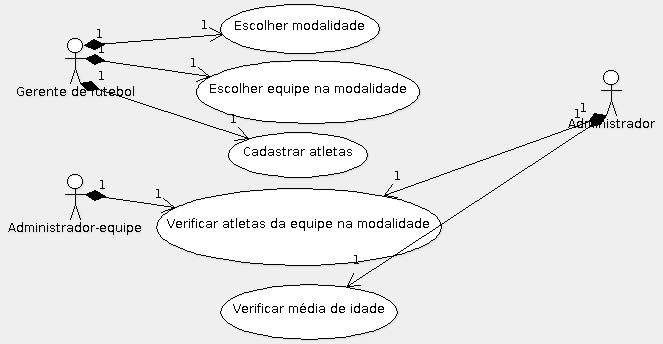
UC 4.4 Cadastrar patrocinadores

UC 5 Controle de modalidade

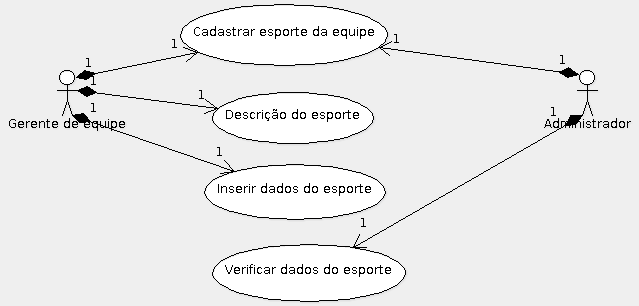
UC 5.1 Cadastrar modalidade



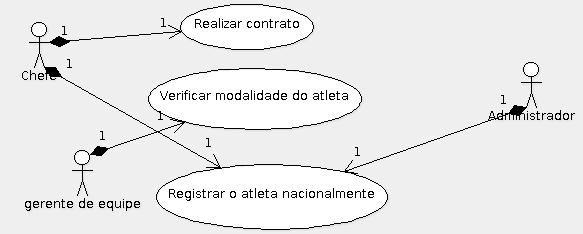
UC 5.2 Verificar média de idade dos atletas



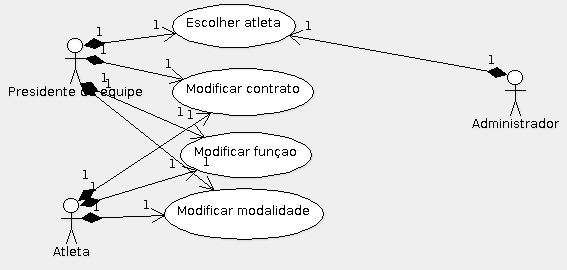
UC 5.3 Verificar esporte da equipe

UC 6 Gerenciar atletas

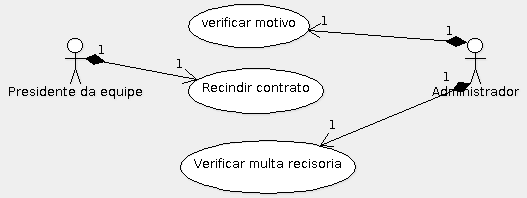
UC 6.1 Cadastrar atleta



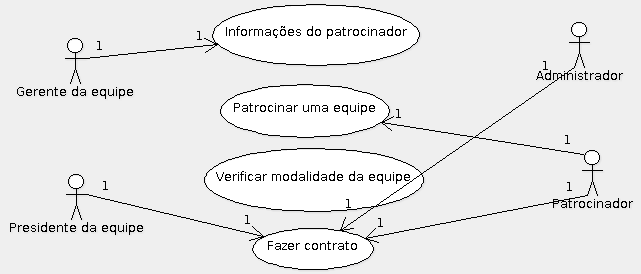
UC 6.2 Modificar atleta



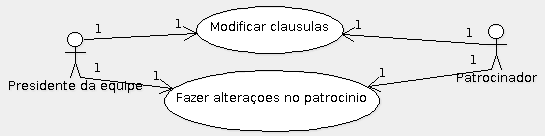
UC 6.3 Deletar atleta

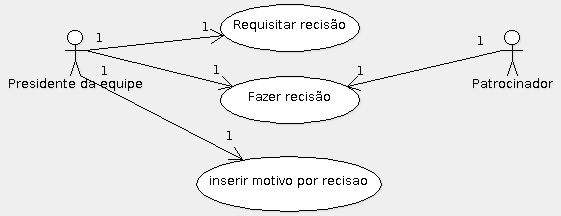
UC 7 Gerenciar patrocinadores

UC 7.1 Cadastrar patrocinador

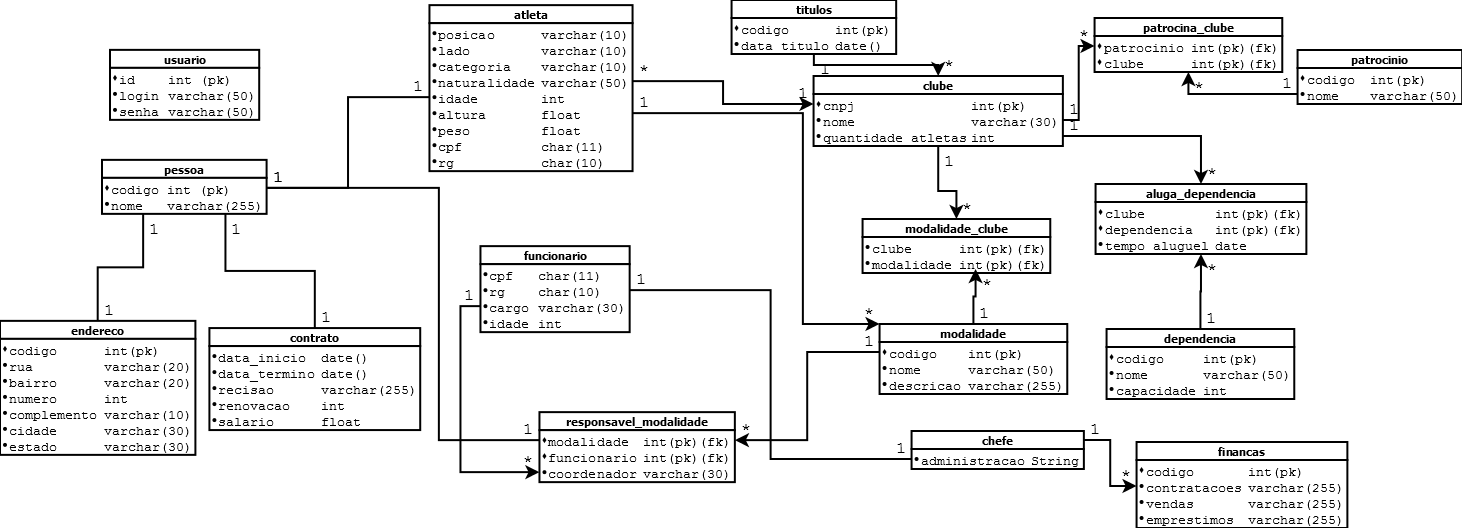


UC 7.2 Modificar patrocinador

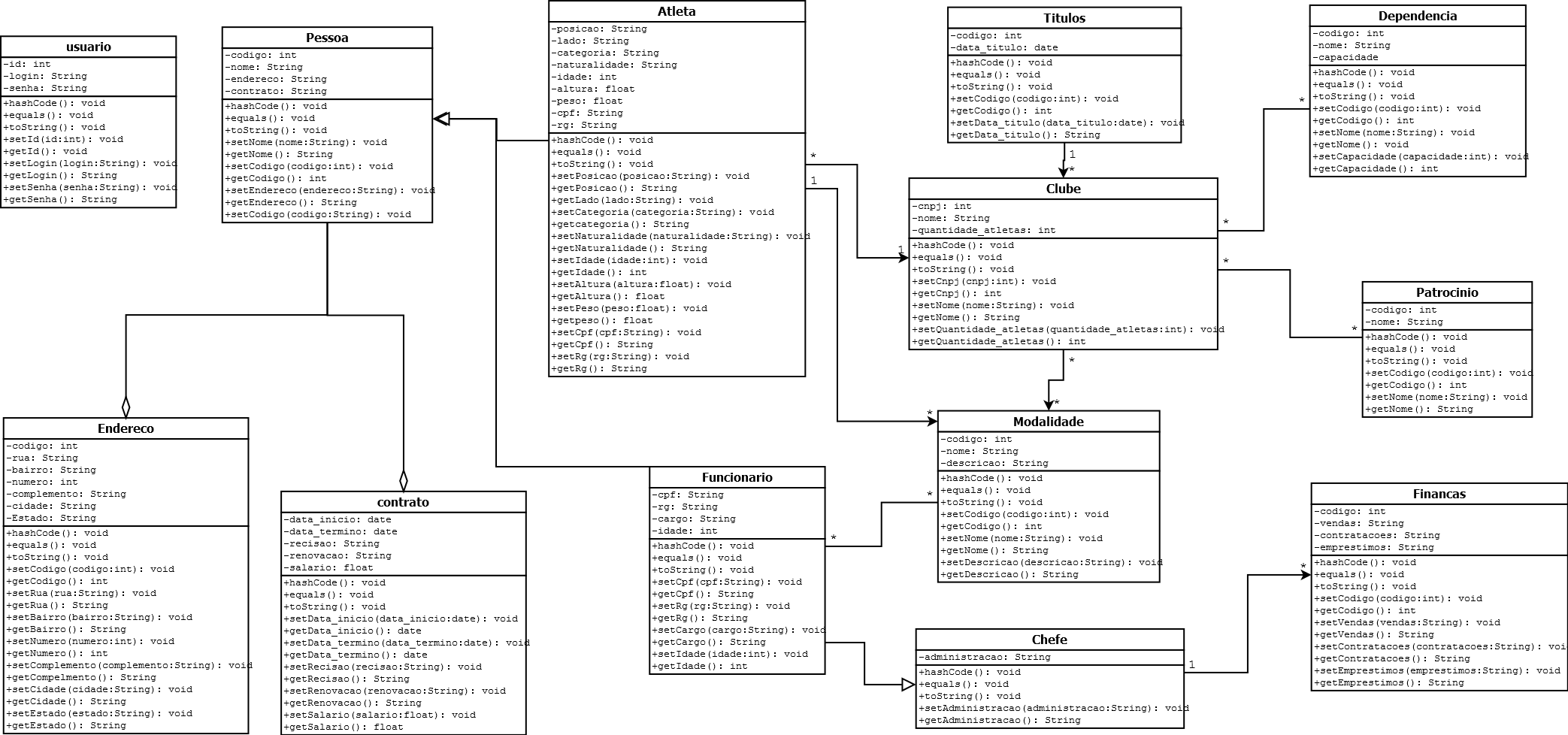


UC 7.3 Deletar patrocinador

1. **MODELO LÓGICO DO BANCO DE DADOS (NA 3FN)**

****

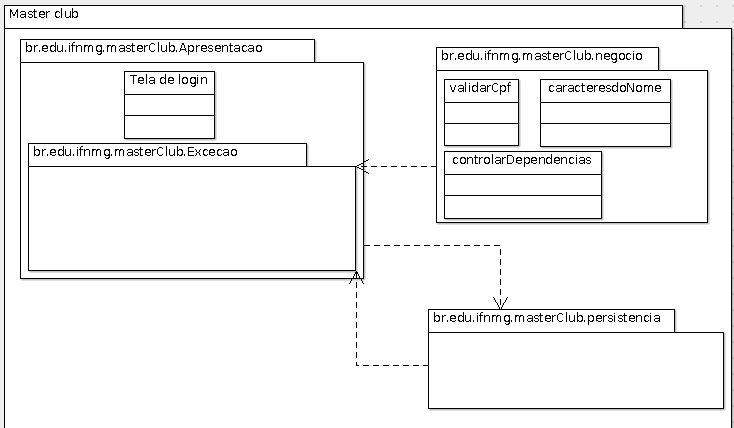
1. **DIAGRAMA LÓGICO ORIENTADO À OBJETOS (DIAGRAMA DE CLASSES UML) DA CAMADA DE LÓGICA DO NEGÓCIO (DOMAIN MODEL)**

****

1. **DESCRIÇÃO DA ARQUITETURA DE SOFTWARE**

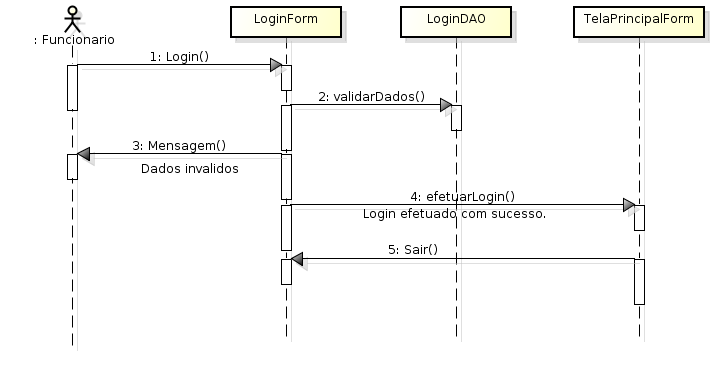
O método de arquitetura que será utilizado no desenvolvimento do software é modelo de arquitetura em três camadas. Esse método aborda a forma que o sistema será modularizado e suas camadas: Aplicação, que contém as regras do negócio, a camada de Acesso à Dados, que contém toda a lógica de interação com o SGBD e a camada de Apresentação, que contém toda a lógica de interface e interação com o usuário. Utilizar dessas camadas facilita no reaproveitamento do código, manutenção e edição dos recursos, teve melhor divisão das tarefas e mantém o código sempre limpo, além de facilitar a organização de cada parte do sistema, facilitando os desenvolvedores a identificarem erros nos locais em que os mesmos acontecem, pois, cada camada tem sua função, basta só identificar os eventos que ocorrem em cada uma.

1. **DIAGRAMA DE PACOTES**

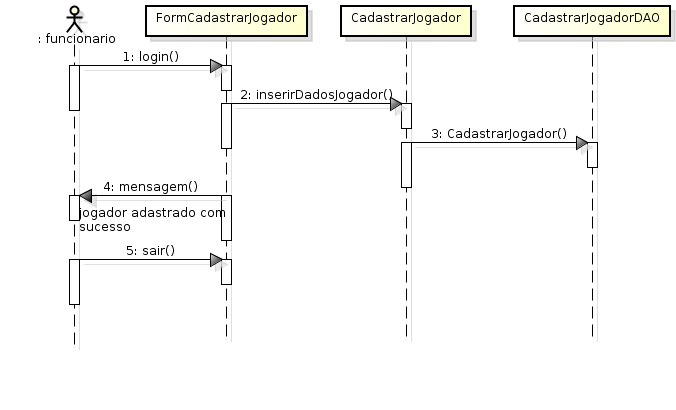
****

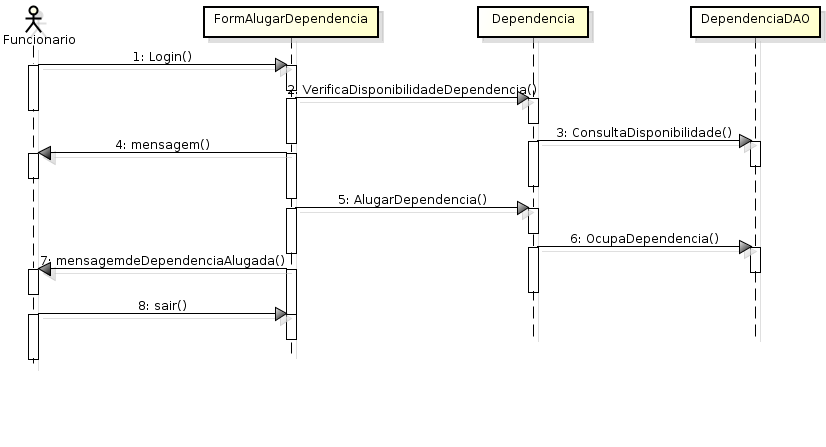
1. **DIAGRAMA DE SEQUÊNCIA**

**11.1 Diagrama de sequência de login.**

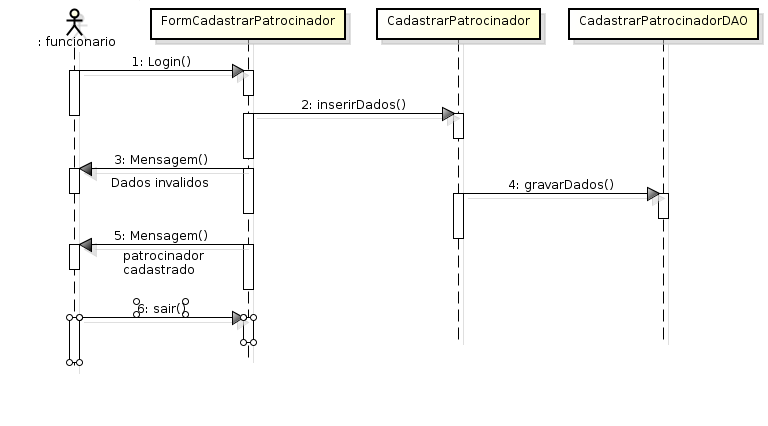
****

**1.2 Diagrama de sequência de cadastro de jogador**

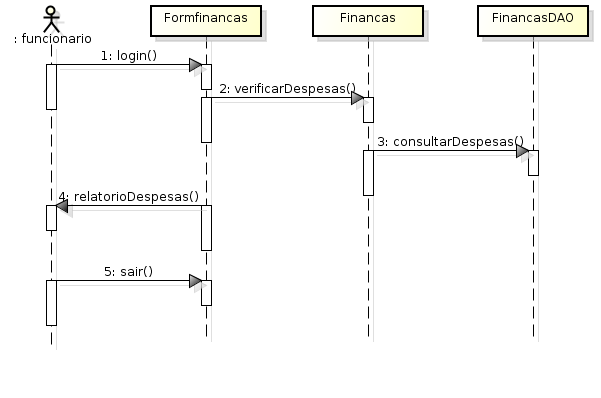
**11.3 Diagrama de sequência de alugar dependências**

****

**11.4 Diagrama de sequência de cadastrar patrocinador**

****

**11.5 Diagrama de sequência de verificar despesas**



1. **DEFINIÇÃO DAS TECNOLOGIAS QUE SERÃO UTILIZADAS NA IMPLEMENTAÇÃO**

O java representa um modelo de linguagem simplificado, orientada a objetos, multitarefa, interpretada, arquitetura neutra, portátil, robusta, e de alto desempenho. A tecnologia Java é composta de uma linguagem de programação e de uma plataforma que utiliza de máquina virtual (SANDRE, 2010)

Utilizar da plataforma java faz com que programas possam ser executados em qualquer plataforma, ou computador.É mais viável desenvolver um software que possa ser executado em “qualquer lugar”, independente da máquina do cliente. (FURGERI, Sérgio; 2012; p.18).

**Implementação do código usando Netbeans(java) versão 8.1**: O NetBeans é um ambiente de desenvolvimento integrado (IDE) Java desenvolvido pela empresa Sun Microsystems. O NetBeans IDE é um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS.

**GitHub:** O Git é um sistema de controle de versão, projetado basicamente para facilitar a vida de quem quer executar projetos em equipe, permitindo que um grupo de pessoas trabalhem juntas. Entretanto é muito utilizado por quem trabalha sozinho, devido a possibilidade de controlar as versões do projeto. O Git dá suporte aos projetos mantendo-os seguros não só no desenvolvimento, mas na manutenção após ele estar em produção como afirma LIMA (2012).

A atualização do projeto é feita com um único comando, e no caso de atualizações, este comando envia somente os arquivos modificados, economizando muito tempo na transferência de arquivos, e aumentando a segurança da atualização, já que não existe o risco de enviar arquivos desatualizados ou incompletos.

O GitHub é um repositório para softwares que usa o controle de versionamento Git. Com ele é possível gerar projetos inovadores e envolver participantes melhorando a produtividade do software de forma simples e dinâmica.

**JDBC:** Java Database Connectivity ou JDBC é um conjunto de classes e interfaces (API) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional, sendo o responsável por armazenar dados.

**MySQL** é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

**ArgoUML**: O ArgoUML é uma aplicação open source que usa UML(Unified Modeling Language.) para modelar o desenho de softwares. Ele roda na maior parte das plataformas por ser implementado em Java. Ele dá suporte para todos os tipos de diagramas UML padrão e inclui suporte cognitivo.

1. **DEFINIÇÃO DO AMBIENTE ALM (IDE, CONTROLE DE VERSÃO, GERENCIAMENTO DE PROJETOS, BUGS, FERRAMENTAS CASE**

**13.1. IDE**

O IDE que será utilizado será o netbeans na versão 8.1. O IDE netbeans oferece todos os recursos para o desenvolvimento de um bom software. Além disso você pode usar o IDE netbeans como SDK para o desenvolvimento de aplicativos usando a plataforma do netbeans, isso por que ele vem com assistentes e modelos que servem de auxilio durante todo o desenvolvimento.

O netbeans serve para solucionar as mais variadas situações por oferecer suporte a outras plataformas, como *Java Enterprise Edition (EE)* , *Java Standard Edition (SE)* e ainda da suporte para desenvolvimento de aplicativos móveis.

O netbeans é muito utilizado por ter suporte a diversas linguagens e como o editor do IDE é extensível pode ser acrescentadas outras linguagens.

Entre as vantagens do IDE podemos citar também que ele é um IDE que é suportado em sistemas operacionais como MAC, Windows e Linux, importante em desenvolvimentos onde a mais de um colaborados

**13.2. Controle de versão**

A ferramenta utilizada no controle de versão do software foi o GIT.

O GIT é uma ferramenta que serve para facilitar o controle do que é desenvolvido no software e também porque auxilia em projetos que são desenvolvidos por equipes uma vez que mesmo não estando em ambientes físico próximos conseguem atualizar o projeto de forma individual. O GIT é utilizado também em projetos que são desenvolvidos de forma individual, isso se deve ao fato da facilidade de “controlar” as versões desenvolvidas no projeto.

Um sistema de controle de versão faz o papel de “juntar” as partes do projeto, de modo que, cada membro da equipe faça uma parte, e, utilizando este sistema seja possível juntar tudo no final.

**13.3. Gerenciamento de projetos**

O gerenciamento foi feito seguindo os quatro principais pilares da gerencia de projeto.

* Qualidade
* custo
* preço
* tempo

**13.4. Ferramentas CASE**

A sigla CASE significa “Computer Aided Software Engeneering”, em português: “Engenharia de Software Auxiliada por Computador”.

Ferramentas CASE são de extrema importância no desenvolvimento de um projeto na área de TI, elas dão todo o auxilio necessário para se esboçar como o projeto deve ser desenvolvido.

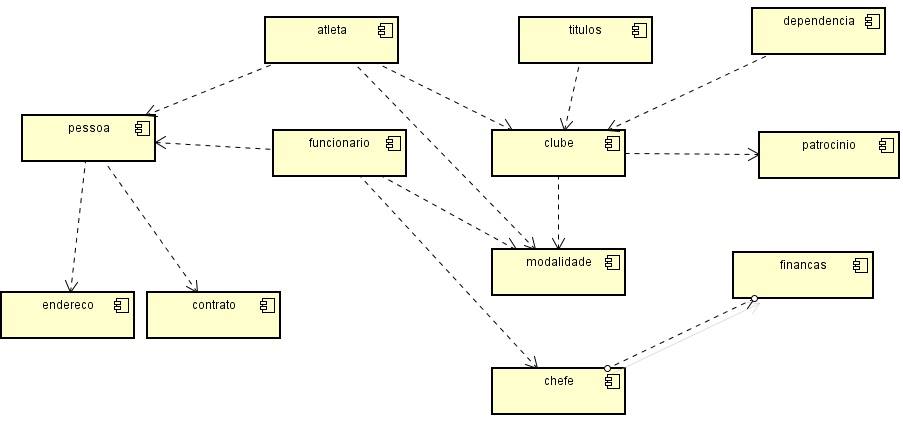
As ferramentas CASE utilizadas para desenvolver o projeto foram o astah professional e o argoUML.

Astah: o astah foi utilizado por ser uma ferramenta CASE com uma interface intuitiva dar suporte a 9 tipos de diagramas, são eles:

* Diagrama de Estruturas: Diagrama de Classes, Diagrama de Estruturas Compostas, Diagrama de Componentes e Diagrama de Implantação.
* Diagrama de Comportamentos: Diagrama de Atividades, Diagrama de Interação, Diagrama de Casos de Uso e Diagrama de Máquina de Estados.
* Diagrama de Interação : Diagrama de Sequência e Diagrama de Comunicação.

ArgoUML: é uma aplicação [*open sourc*](https://pt.wikipedia.org/wiki/Open_source)*e* que usa UML para modelar o desenho de *software* de [computador](https://pt.wikipedia.org/wiki/Computador). A aplicação corre na maior parte das plataformas uma vez que é implementada em [Java](https://pt.wikipedia.org/wiki/Java_(linguagem_de_programação)). Está distribuída sob a [licença BSD](https://pt.wikipedia.org/wiki/Licença_BSD). Providencia suporte para quase todos os tipos de diagrama da UML padrão e inclui suporte cognitivo.

1. **DIAGRAMA DE COMPONENTES**

****

1. DIAGRAMA DE IMPLANTAÇÃO

