CENTRO UNIVERSITÁRIO CAMPOS DE ANDRADE – UNIANDRADE

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUCIANO FURTADO CORREIA

**SOCCER MANAGER**

Curitiba, 2014

CENTRO UNIVERSITÁRIO CAMPOS DE ANDRADE - UNIANDRADE

LUCIANA ALINE DA SILVA

WESLEY JUNIOR DE PAULA

**SOCCER MANAGER**

Este trabalho trata-se de software de controle e administração para escolas de futebol. Será apresentado na disciplina de Projeto Final, tendo como foco principal, a conclusão do curso de Analise e Desenvolvimento de Sistemas.

Curitiba, 2014

Sumário

[**1.** **Introdução** 5](#_Toc404898106)

[**2.** **Ideia** 6](#_Toc404898107)

[**2.1.** **Objetivos** 7](#_Toc404898108)

[2.1.1. Objetivo Geral 7](#_Toc404898109)

[2.1.2. Objetivo Específico 7](#_Toc404898110)

[**3.** **Análise de Requisitos** 7](#_Toc404898111)

[**3.1.** **Ator PROFESSOR** 9](#_Toc404898112)

[**3.2.** **Ator PROPRIETÁRIO** 10](#_Toc404898113)

[**3.3.** **Regras de negócio para o sistema:** 11](#_Toc404898114)

[**3.4.** **Requisitos não-funcionais:** 11](#_Toc404898115)

[**3.5.** **Descrição de Caso de Uso** 11](#_Toc404898116)

[**4.** **Protótipos de telas** 16](#_Toc404898117)

[**4.1.** **Cadastro de Aluno** 16](#_Toc404898118)

[**4.2.** **Cadastro de Funcionário** 19](#_Toc404898119)

[**4.3.** **Cadastro de Competições e eventos** 19](#_Toc404898120)

[**4.4.** **Agenda** 22](#_Toc404898121)

[**5.** **Diagrama de Classe** 22](#_Toc404898122)

[**6.** **Premissas:** 23](#_Toc404898123)

[**7.** **Restrições:** 23](#_Toc404898124)

[**8.** **Limites do Projeto:** 24](#_Toc404898125)

[**9.** **Funcionalidades** 24](#_Toc404898126)

[**10.** **Cronograma** 24](#_Toc404898127)

[**11.** **Ambiente Tecnológico** 26](#_Toc404898128)

[**11.1.** **Metodologia de desenvolvimento RUP** 26](#_Toc404898129)

[**11.2.** **Linguagem Groovy/Grails** 28](#_Toc404898130)

[**11.3.** **PostgreSQL** 30](#_Toc404898131)

[**11.4.** **Netbeans** 30](#_Toc404898132)

[**11.5.** **iReport** 31](#_Toc404898133)

[**11.6.** **Apache Tomcat** 31](#_Toc404898134)

[**11.7.** **Charisma Bootstrap** 32](#_Toc404898135)

[**11.8.** **Pencil Project** 32](#_Toc404898136)

[**12.** **Conclusão** 34](#_Toc404898137)

[**13.** **Referências** 35](#_Toc404898138)

Lista de Imagens

[Figura 1 - Caso de uso que representa os atores Professor e Proprietário. 16](#_Toc404898139)

[Figura 2 - Protótipo para a tela de cadastro de aluno. 16](#_Toc404898140)

[Figura 3 - Protótipo para a tela de cadastro de saúde do aluno. 17](#_Toc404898141)

[Figura 4 - Protótipo para tela de cadastro de horários de aulas do aluno. 17](#_Toc404898142)

[Figura 5 - Protótipo para a tela da lista das mensalidades pagas por aluno. 18](#_Toc404898143)

[Figura 6 - Protótipo para a tela do histórico das competições do aluno. 18](#_Toc404898144)

[Figura 7 - Protótipo para a tela do histórico dos jogos do aluno. 19](#_Toc404898145)

[Figura 8 - Protótipo de tela do cadastro de funcionário. 19](#_Toc404898146)

[Figura 9 - Protótipo para a tela de cadastros das competições e eventos. 20](#_Toc404898147)

[Figura 10 - Protótipo de tela para montar os times para participar dos jogos. 20](#_Toc404898148)

[Figura 11 - Protótipo de tela da galeria de fotos. 21](#_Toc404898149)

[Figura 12 - Protótipo de tela de cadastro das súmulas dos jogos. 21](#_Toc404898150)

[Figura 13 -Protótipo de tela para a galeria de imagens para súmulas. 22](#_Toc404898151)

[Figura 14 - Protótipo de tela da agenda de compromissos. 22](#_Toc404898152)

[Figura 15 - Imagem do Diagrama de Classes 23](#_Toc404898153)

[Figura 16 - Ciclo de Vida RUP 28](#_Toc404898154)

1. **Introdução**

Este trabalho será apresentado para a conclusão do curso de Análise e Desenvolvimento de Sistemas. Servirá para representar a documentação de desenvolvimento do software proposto.

O software será de gerenciamento para escolas de futebol, área que encontra-se carente de softwares específicos. Será projetado com base na metodologia de desenvolvimento de software RUP e desenvolvido no ambiente NetBens IDE 8.0, com Apache Tomcat 8.0.9, estas são as principais aplicações que serão utilizadas no decorrer do projeto e no desenvolvimento do software.

A ideia:

“Desde muito tempo atrás, mais especificamente 25 anos, eu vejo meu pai envolvido com futebol.

Desde pequena eu, Luciana, participava com ele disso, como sou a mais velha, fui como o filho homem que o pai leva junto. Cresci nesse meio.

Há, mais ou menos, 10 anos atrás, começou devagar a surgir uma ideia, meu pai resolveu fazer um curso de arbitragem para o Futsal paranaense, feito o curso, ele se destacou como um bom árbitro, e me carregou junto, eu sendo ‘filho mais velho’ virei anotadora.

Assim, durante 8 anos, eu fui em jogos quase todos os finais de semana da minha vida.

Mas como ser apenas árbitro não era suficiente, ele comprou uma escola de futebol, e novamente começou a fazer sucesso, ganhando vários dos campeonatos que participava. E eu junto, foi quando além de gostar, comecei a jogar e entender a emoção que homens sentem. Jogava no masculino, porque na época eu era a única corajosa. Infelizmente parei, porque tinha que trabalhar, mas sempre participo de algumas coisas que ele inventa para promover a escola.

Quando comecei a faculdade, na minha primeira semana de aula, ele me encarou e disse: ‘Quando você vai fazer um sistema pra escola?’, sério, como se eu fosse uma programadora de peso com alto grau de experiência. Quase tive uma parada cardíaca e respiratória, enchi os pulmões e com toda a coragem que eu consegui fingir ter, respondi: ‘Quando eu tiver base para tal programa, afinal terá que ser o melhor.’

Ele não engoliu a coragem, mas se conformou.

Sendo assim, escolhi um sonho, que em parte também é meu, e resolvi fazer como projeto de encerramento de curso. Assim surgiu a ideia, minha e de meu pai.

Sendo assim, o principal motivo para o desenvolvimento é: facilitar a vida do meu pai, que já tem alguns empregos.”

Analisando os fatores tanto externos, outras escolas, quanto internos, pela dificuldade próxima, surgiu a necessidade desenvolvê-lo.

O software Soccer Manager possui etapas que foram planejadas para serem visualizadas juntas. As informações a seguir contêm a explicação dessas etapas:

1. **Ideia**

A ideia veio de uma necessidade no ambiente familiar, e quando avaliada amplamente, descobriu-se uma grande necessidade deixada de lado por várias pessoas.

Vivemos num mundo onde a tecnologia fica cada vez mais necessária, em todos os lugares há acesso às mais recentes informações, que são disparadas em segundos e visualizadas em todos os lugares.

Não poderia ser diferente no esporte, atualmente, as mais diversas tecnologias são utilizadas seja para transmissão ou recepção de informação. Como não há a possibilidade de fugir da realidade, apenas aos grandes investidores detêm acesso às melhores.

Então, pensando que também os pequenos querem se informatizar, a ideia passou de simples, para algo mais complexo e melhor ajustado a nossa realidade.

Desenvolver um software para escolas de futebol será considerado uma inovação, pois nas pesquisas realizadas, não existem softwares específicos para esse fim.

Poucas escolas utilizam algum tipo de software, ainda assim esse software utilizado não satisfaz a necessidade, por não conter informações precisas.

Algumas escolas se organizam através de planilhas e anotações manuscritas, que causa atraso e confusão quando não há a devida organização.

A partir desta informação, surgiu a necessidade de modificar a realidade e auxiliar o gerenciamento destas, criando um software específico para as escolas de futebol.

* 1. **Objetivos**

O que é?

Software web para o gerenciamento de processos para escolas de futebol.

* + 1. Objetivo Geral

Fornecer um software de gerenciamento, exclusivo para escolas de futebol.

* + 1. Objetivo Específico
* Facilitar a administração das escolas de futebol, pois esse tipo de instituição tem processos característicos.
* Automatizar os processos para melhor integração das informações.

1. **Análise de Requisitos**

A análise de requisitos, foi colhida através de entrevistas com pessoas que possuem ou trabalham em escolas de futebol em Curitiba.

Essas pessoas relataram suas principais dificuldades e o que facilitaria sua rotina nesses ambientes.

Nesta entrevista, houve questionamentos variados, a seguir algumas questões que foram levantadas:

* Há quanto tempo você trabalha nesse ramo?
  + Houve período de a partir de 1 ano até 6 anos.
* Quantos alunos há na escola?
  + Respostas a partir de 50 alunos.
* Qual o tempo médio de permanência dos alunos nas aulas?
  + Percebeu-se que há alunos desde que a escola abriu, como alunos muito recentes, o tempo de permanência varia muito, pois depende muito de fatores externos.
* Como se faz para conseguir novos alunos?
  + O principal meio é a partir do boca-a-boca, pois uma criança traz a outra, os pais também são grandes influenciadores, pois, geralmente, frequentam os locais onde há as escolas e acabam matriculando seus filhos.
* A escola recebe apenas alunos do sexo masculino?
  + Todas as entrevistadas possuem alunos de ambos os sexos matriculados.
* Qual o meio utilizado para arquivar a matrícula do aluno que se matricula?
  + Algumas escolas possuem uma ficha de matrícula em que os pais e ou responsáveis do aluno preenchem, outras escolas é um processo simples, mas que traz alguns problemas para o proprietário, por não possuir ficha a matrícula é feita mais por um acordo entre os pais e o proprietário.

Geralmente as fichas são guardadas em pastas e arquivadas, várias escolas não possuem um local apropriado para guardar esse tipo de documentação.

* Qual o principal meio para o controle de frequência do aluno.
  + Algumas possuem chamadas feitas pelo horário de aula, outras, como possuem poucos alunos, o professor já sabe quais os alunos que não estão presentes.
* Como é feito o controle das competições?
  + Os professores que participam sempre já sabem as datas e os locais, eles mantêm sempre comunicação entre si. Quando a competição é pela federação, as informações sobre a competição ficam disponíveis no site dessa instituição.
* Você acha que um software específico para sua escola te ajudaria?
  + Os que concordaram, gostaram da ideia pois ajudará no seu trabalho. Alguns disseram que seria bom, mas não utilizariam, pois não podem investir em novas tecnologias, outras já tem processos próprios e não querem mudar.
* Aos que concordaram com o software:
* Como você acha que o software irá te ajudar na escola?
  + Muitos responderam que os papéis talvez não irão diminuir, mas eles terão acesso às informações de seus alunos onde eles estiverem.

A maioria das perguntas foi de quais as funcionalidades que essas pessoas gostariam que fosse implantado no sistema, às principais foram analisadas e formaram os casos de uso.

Através de suas respostas, casos de uso foram feitos, a partir disto, definidas as principais requisições dos futuros usuários.

Foram identificados 2 atores principais:

* Professor ou auxiliar (usuário direto: terá senha e login para entrar no sistema, manipula dados e coleta informações.)
* Proprietário ou administrador (usuário direto: terá login e senha para acessar o sistema, será o principal usuário, poderá incluir, manipular e excluir dados do sistema.)
  1. **Ator PROFESSOR**

Esse caso de uso delimita as ações que o usuário com o perfil de professor pode realizar no sistema.

* + 1. Requisitos funcionais:

Acompanhar o desenvolvimento do aluno matriculado.

Consultar o histórico do aluno.

Montar o time para participar de competições.

Gerar relatórios das aulas.

Incluir ou excluir alunos das aulas.

* + 1. Ações para: Ator Professor

Montar time: montar o time para as competições que irão participar.

Avaliar aluno: a avaliação do aluno tanto nas aulas quanto nas competições.

Acompanhar competições: participar e registrar as informações das competições.

Administrar aulas: pode criar um relatório de qual foi o andamento das aulas, incluir ou excluir os alunos das aulas.

* + 1. Regras de negócio

RN1 - O professor poderá apenas montar um time com alunos que estão nas categorias gravadas no sistema, com idade igual ou inferior ao limite da categoria.

RN2 - Para avaliar o aluno, o professor contará com uma lista de alunos que estão cadastrados no horário de aula equivalente, podendo adicionar ou excluir aluno do horário.

RN3 - O professor poderá incluir ou excluir alunos das aulas.

RN4 – O professor tem acesso para visualizar e/ou alterar o cadastro dos alunos.

RN5 - A ação remover ou incluir aluno para a aula, valerá apenas para aquela aula do dia.

* 1. **Ator PROPRIETÁRIO**

Esse caso de uso delimita as ações que o usuário com o perfil de proprietário pode realizar no sistema.

* + 1. Requisitos funcionais:

O proprietário terá acesso a todos os cadastros, podendo fazer alterações, inclusões e exclusões.

Tem acesso para dar baixa em recebimentos das mensalidades.

O proprietário pode acompanhar o histórico dos dados dos alunos gravados no sistema.

* + 1. Ações para: Ator PROPRIETÁRIO

Cadastrar Aluno: matricular os alunos.

Cadastrar Funcionário: cadastrar funcionários para controle.

Cadastrar Competições: cadastra as novas competições.

Cadastrar Súmulas: Guardar as informações de todas competições.

Regras de negócio:

RN1 - O proprietário terá acesso a todo sistema, considera-se como administrador do sistema.

RN2 - O proprietário fará a parte de administração da escola, delimitando a área de atuação ao sistema pelo professor.

RN3 - No cadastro de aluno, o proprietário poderá estipular os horários e dias em que o aluno participará das aulas, também o valor das mensalidades e data de vencimento.

RN4 - O proprietário terá acesso ao histórico de cada aluno matriculado tanto ativo, quanto inativo.

RN5 - O proprietário pode imprimir recibos no sistema, para recebimentos de mensalidades.

* 1. **Regras de negócio para o sistema:**

RN1 - Ao cancelar a matrícula de um aluno, o cadastro dele fica inativo no sistema, ainda disponível para consulta ou posterior reativação.

* 1. **Requisitos não-funcionais:**

As telas que contém listas não podem exceder 3ms para apresentar a lista.

O banco de dados fica disponível no servidor, o cliente não tem acesso ao mesmo.

O backup dos dados será automático, com o período que o cliente decidir, o padrão será todo dia, horário: 23:58h. Será realizado com o agendador de tarefas do seu sistema operacional.

Para acesso externo à rede, a velocidade de carregamento das páginas, principalmente àquelas que possuem mais informações, dependerá da velocidade da internet e transferência de dados disponível no dispositivo de acesso.

O a forma de acesso externa depende de configurações no servidor, estando sujeito à disponibilidade de acesso à internet. O sistema será instalado num computador que servirá como o servidor, acessos internos serão realizados pelo ip e rede interna, acessos externos serão realizados se o servidor estiver conectado à internet.

* 1. **Descrição de Caso de Uso**

Caso de uso - Fazer Login

Fluxo Normal:

1 - Usuário entra na página de login.

2 - Usuário inseri o login e a senha.

3 - Sistema valida o login e a senha.

4 - Usuário entra no sistema.

5 - Fim do caso de uso.

Fluxo de Exceção:

3a - Sistema não valida o login e a senha

3b - Retorna ao passo 2.

Caso de uso - Cadastrar aluno:

Fluxo Normal:

1 - Usuário acessa a página de cadastro de novo aluno.

2 - Inserir as informações do novo aluno.

3 - Salvar as informações.

4 - Sistema retorna as informações cadastradas disponpiveis apenas para a visualização.

5 - Fim do caso de uso.

Fluxo alternativo:

4a - Usuário verifica que faltou informações.

4b - Retorna ao passo 2.

Fluxo de Exceção:

3a - Informações obrigatórias não foram inseridas, RN1.

3b - Sistema retorna erro nos campos que faltam serem preenchidos.

3c - Retorna ao passo 2.

Regras de Negócio:

RN1 - As informações obrigatórias serão: Nome, Idade, Nome Pai, Nome Mãe, Categoria, Telefone.

Caso de uso - Cadastrar Funcionário:

Fluxo Normal:

1 - Usuário acessa a página de cadastro de novo usuário.

2 - Inserir as informações do novo funcionário.

3 - Salvar as informações do funcionário.

4 - Sistema retorna as informações cadastradas apenas para visualização.

5 - Fim do caso de uso.

Fluxo Alternativo:

4a - Usuário verifica que faltaram informações.

4b - Retorna ao passo 2.

Fluxo de Exceção:

3a - Informações obrigatórias não foram preenchidas, RN1.

3b - Sistema retorna erro nos campos que faltam serem preenchidos.

3c - Retorna ao passo 2.

Regras Negócio:

RN1 - As informações obrigatórias serão: Nome, CPF, Telefone.

Caso de uso - Cadastrar Time

Fluxo normal:

1 - Usuário acessa a cadastro de novo time.

2 - Inserir as informações para cadastrar novo time.

3 - Sistema libera a lista de alunos disponíveis para o time (RN1).

4 - Usuário seleciona os alunos para compor o time.

5 - Usuário salva as informações do novo time.

6 - Sistema retorna as informações salvas para visualização.

7 - Fim do caso de uso.

Fluxo Alternativo:

6a - Usuário identifica informações faltantes.

6b - Retorna ao passo 2.

Regras de Negócio:

RN1 - O sistema irá liberar a lista para selecionar os alunos disponíveis na categoria que o time será formado.

Caso de uso - Cadastrar Jogos/Competições:

Fluxo Normal:

1 - Usuário entra no cadastro de novo jogo.

2 - Inserir as informações para o novo cadastro.

3 - Usuário salva as informações.

4 - Sistema retorna as informações salvas para visualização.

5 - Fim do caso de uso.

Fluxo Alternativo:

4a - Usuário verifica informações faltantes.

4b - Retorna ao passo 2.

Caso de uso – Cadastrar Informações Jogo (Súmula)

Fluxo Normal:

1 - Usuário entra no cadastro de visualizar jogo.

2 – Entra na página para incluir as informações do jogo.

3 - Usuário salva as informações.

4 - Sistema retorna as informações salvas para visualização.

5 - Fim do caso de uso.

Fluxo Alternativo:

4a - Usuário verifica informações faltantes.

4b - Retorna ao passo 2.

Caso de uso – Receber Mensalidade

Fluxo Normal:

1 - Usuário entra na tela visualizar Aluno.

2 - Usuário entra em Mensalidades.

3 - Insere as informações referentes à mensalidade.

4 - Usuário salva as informações.

5 - Sistema retorna a lista das mensalidades pagas pelo Aluno.

6 - Fim caso de uso.

Fluxo Alternativo:

5a - Usuário pode imprimir recibo referente ao pagamento.

5b - Usuário percebe informações faltantes.

5c - Retorna ao passo 2.

Caso de uso - Imprimir Recbo de Recebimento

Fluxo Normal:

1 - Usuário entra na tela de visualizar Aluno.

2 - Usário entra em Mensalidades.

3 - Usuário clica em imprimir recibo.

4 - Fim caso de uso.

Caso de uso – Incluir informações dos cadastros básicos

O usuário vai cadastrar as informações básicas, que são outras funcionalidades do sistema, como incluir Cidades, Estado, País, Função, Galeria de Imagens, Empresa.

Fluxo Normal:

1 – Usuário entra no cadastro.

2 – Usuário inclui as informações do cadastro. (RN1)

3 – Usuário salva as informações inseridas.

4 – O sistema retorna as informações salvas.

5 – Fim caso de uso

Fluxo Alternativo:

3a – Usuário confere informações faltantes.

3b – Retorna ao passo 2.

Regras de Negócio:

RN1 – Se for a galeria de imagens, o usuário fará o upload das imagens.

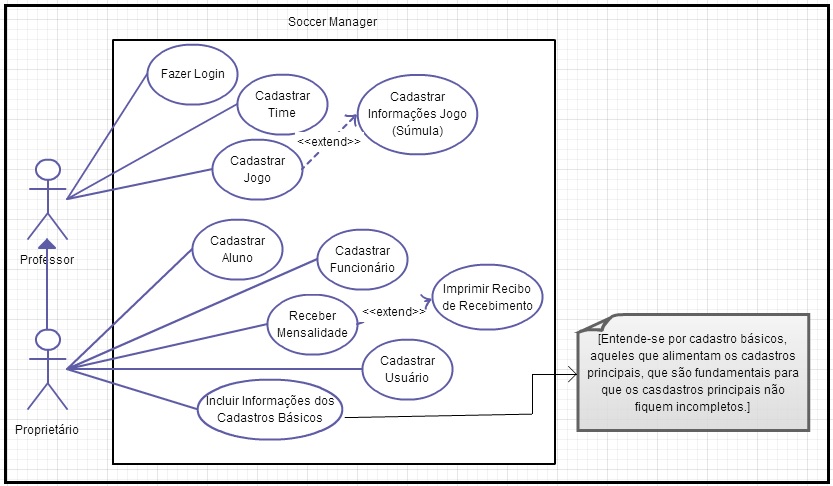


Figura 1 - Caso de uso que representa os atores Professor e Proprietário.

1. **Protótipos de telas**

São os protótipos das telas apresentados aos entrevistados para o melhor entendimento de como o sistema pode agir:

* 1. **Cadastro de Aluno**

Tela de cadastro de aluno.

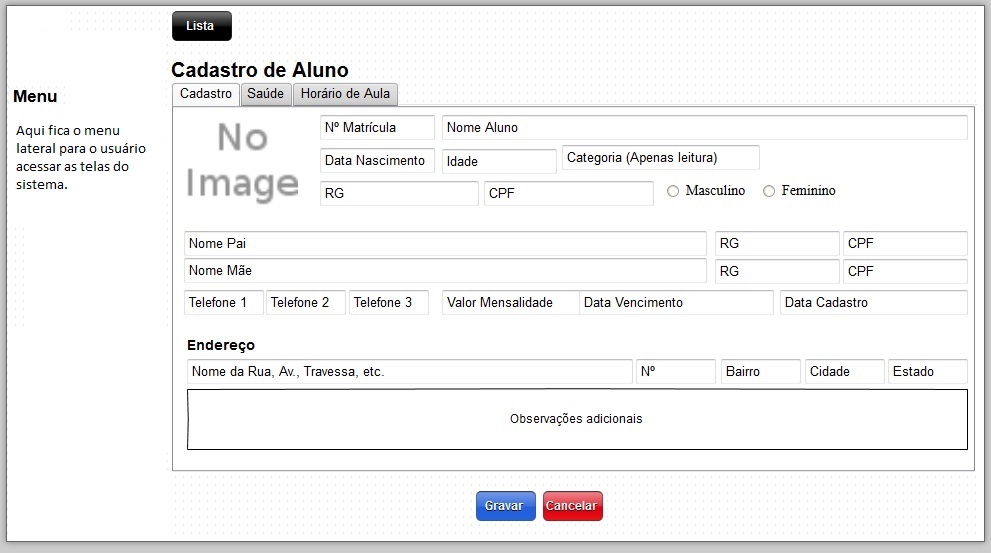


Figura 2 - Protótipo para a tela de cadastro de aluno.

Tela para cadastrar informações sobre a saúde do aluno.

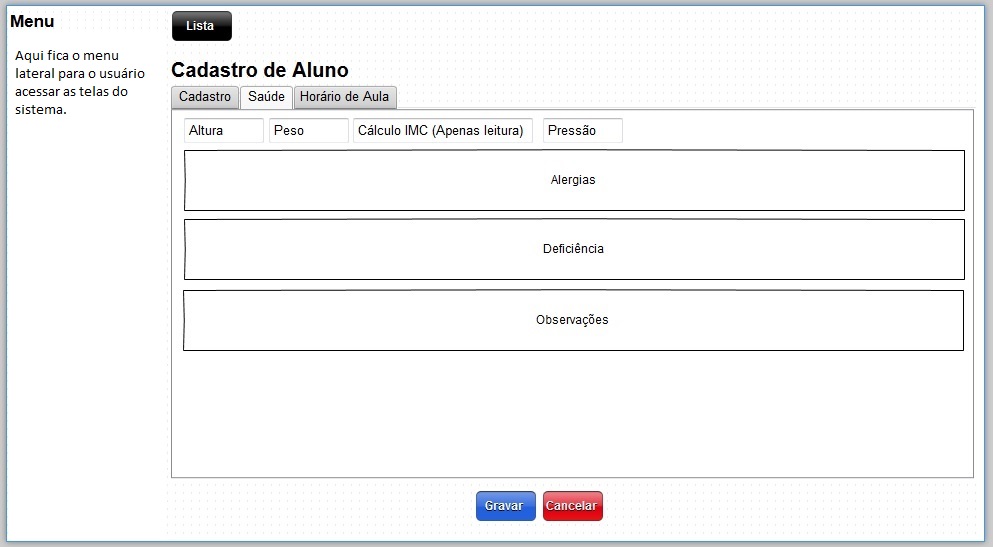


Figura 3 - Protótipo para a tela de cadastro de saúde do aluno.

Tela de cadastro de horário de aulas do aluno.

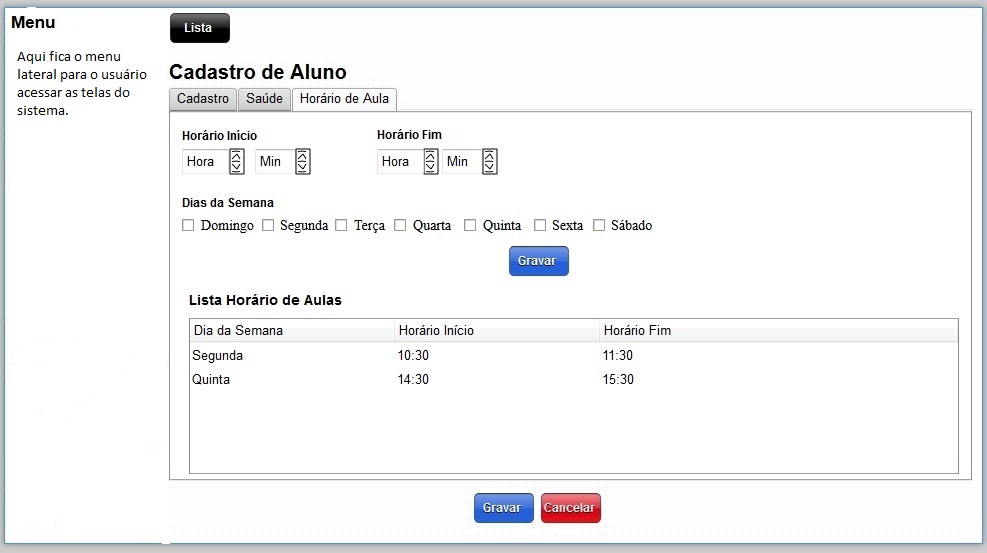


Figura 4 - Protótipo para tela de cadastro de horários de aulas do aluno.

Tela onde visualiza-se a lista das mensalidades pagas por aluno, no campo seleciona o nome do aluno, e as informações são carregadas na tabela.

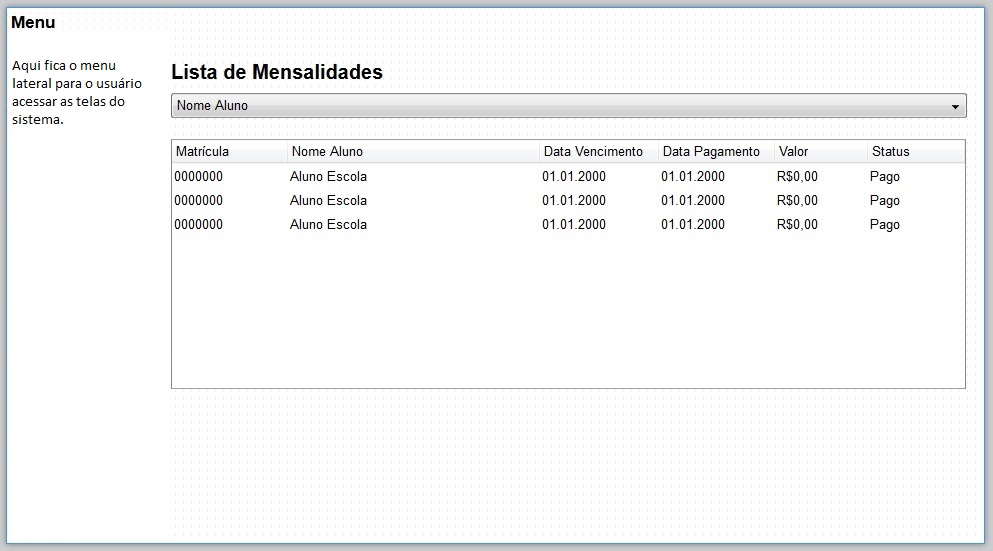


Figura 5 - Protótipo para a tela da lista das mensalidades pagas por aluno.

Tela do histórico do aluno, a pesquisa é feita pelo período do mês e traz informações por aluno.

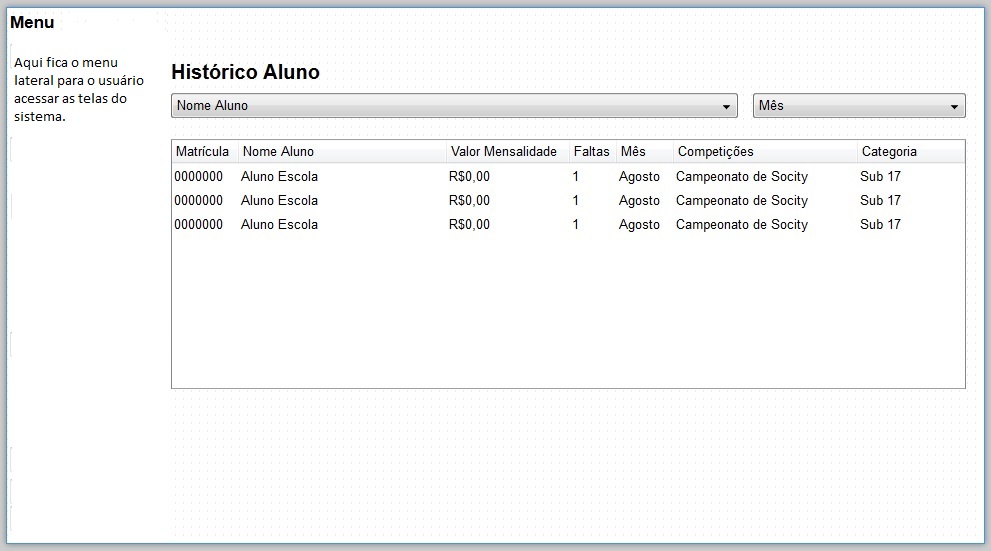


Figura 6 - Protótipo para a tela do histórico das competições do aluno.

Tela do histórico das competições, a pesquisa é feita por nome do aluno e mostra informações dos jogos em que aluno participou.

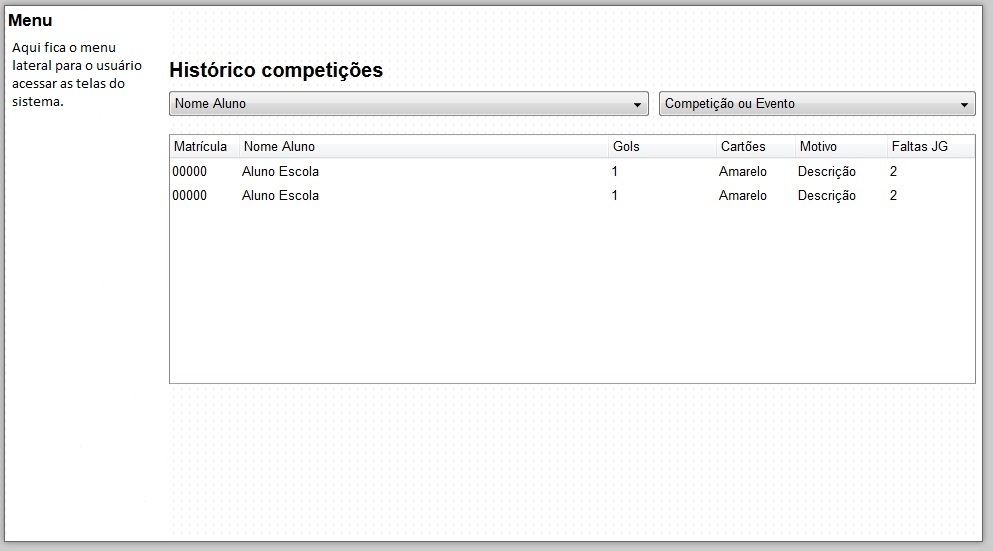


Figura 7 - Protótipo para a tela do histórico dos jogos do aluno.

* 1. **Cadastro de Funcionário**

Tela de cadastro de funcionário.



Figura 8 - Protótipo de tela do cadastro de funcionário.

* 1. **Cadastro de Competições e eventos**

Tela para cadastro das competições, jogos e eventos.



Figura 9 - Protótipo para a tela de cadastros das competições e eventos.

Tela para montar os times que participarão dos jogos em competições ou eventos esportivos.

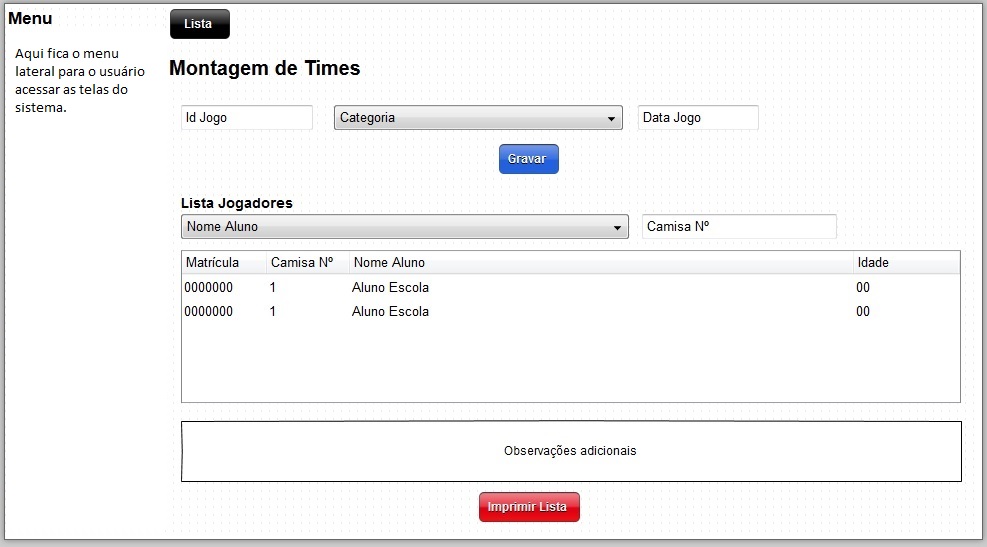


Figura 10 - Protótipo de tela para montar os times para participar dos jogos.

Tela que visualiza os álbuns das fotos que foram adicionadas ao sistema.

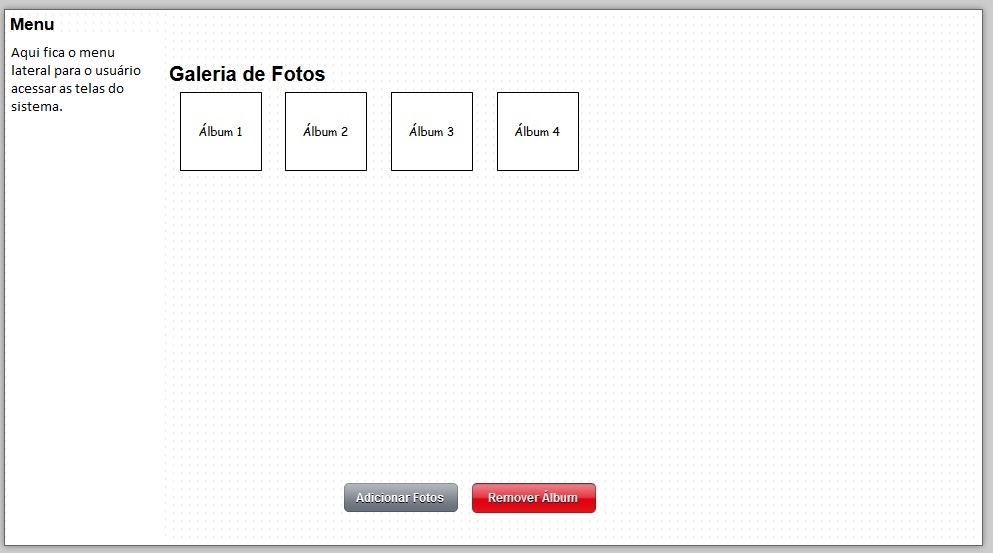


Figura 11 - Protótipo de tela da galeria de fotos.

Tela para cadastrar as informações das súmulas.

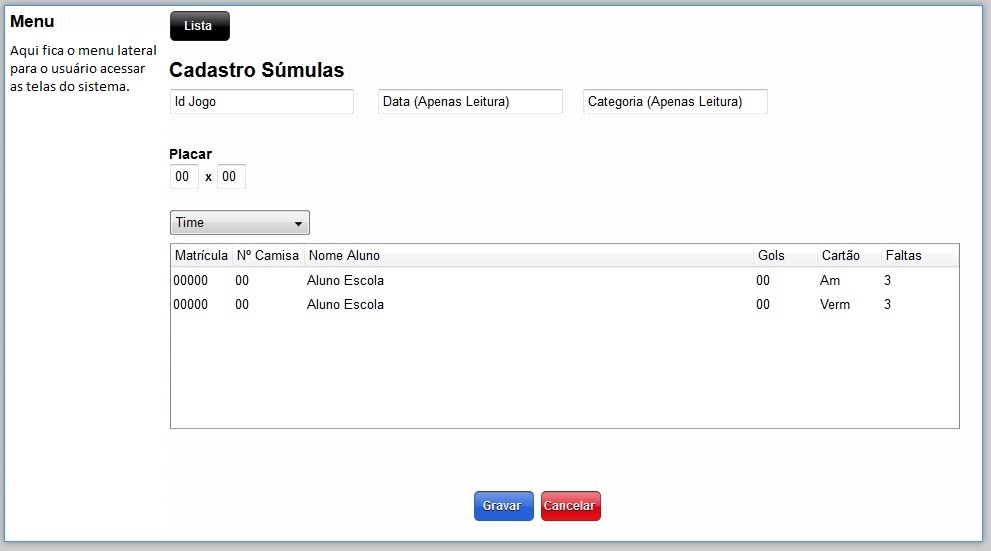


Figura 12 - Protótipo de tela de cadastro das súmulas dos jogos.

Tela para visualizar as imagens das súmulas que foram adicionadas ao sistema.

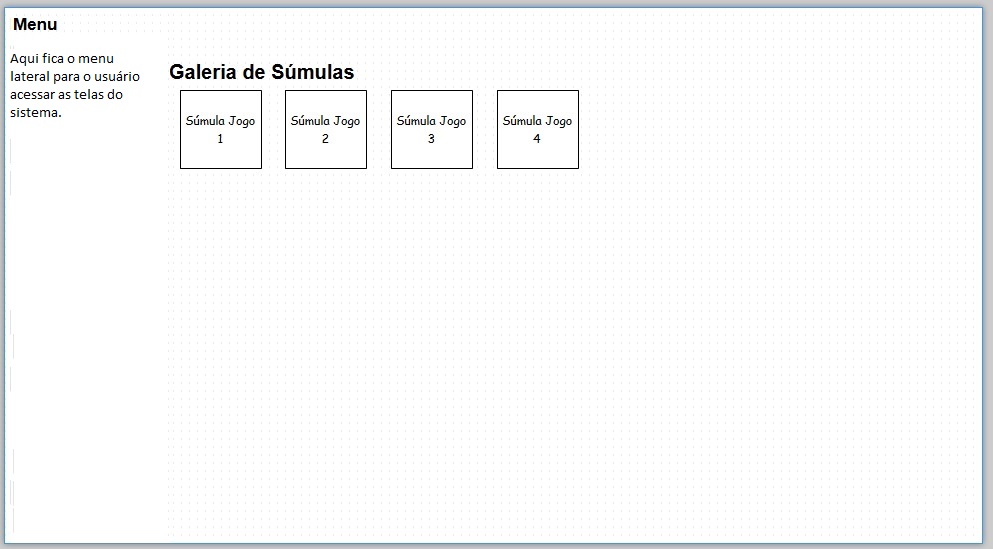


Figura 13 -Protótipo de tela para a galeria de imagens para súmulas.

* 1. **Agenda**

Tela de agenda, visualiza-se os compromissos gravados.

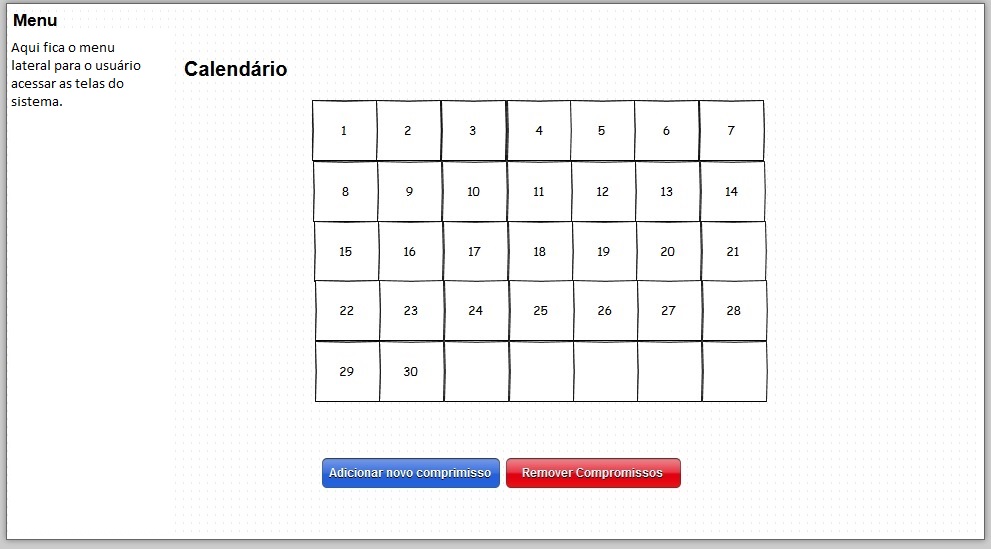


Figura 14 - Protótipo de tela da agenda de compromissos.

1. **Diagrama de Classe**

Diagrama de Classe simples, pois no RUP não é obrigatório, auxilia no desenvolvimento do sistema.

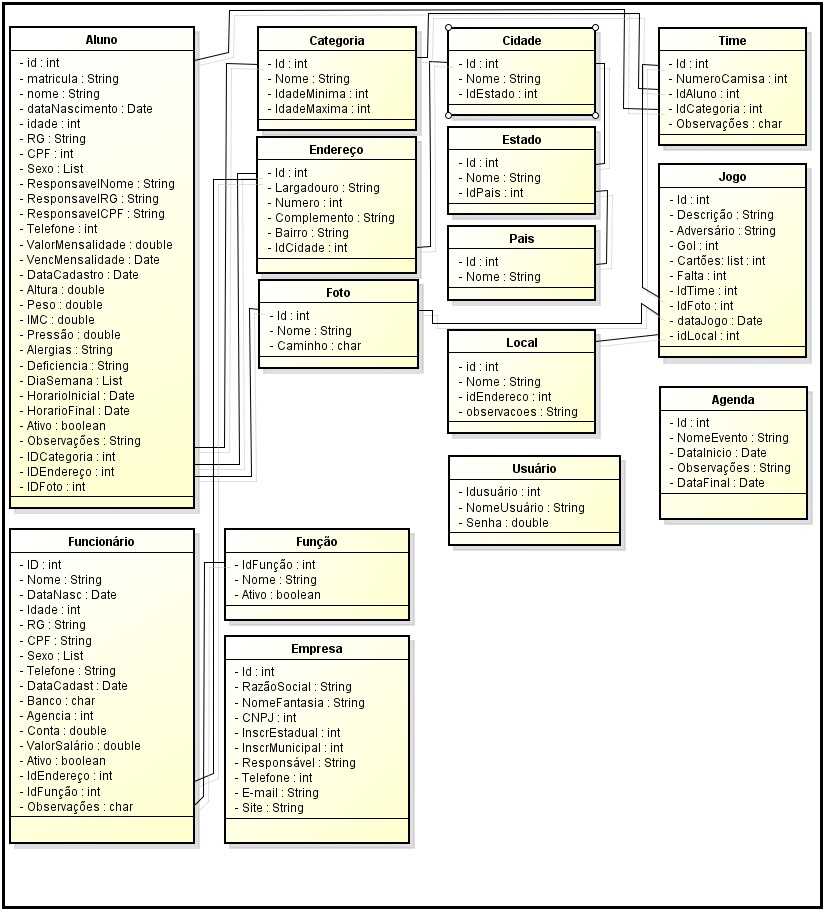


Figura 15 - Imagem do Diagrama de Classes

1. **Premissas:**

O software será desenvolvido até 01/12/2014.

O software terá disponibilidade em português.

Suporte técnico será exclusivamente por e-mail.

Atualizações serão online, o usuário será informado através de um e-mail, este conterá os locais do sistema que foram feitas as correções e modificações.

O sistema não gera nota fiscal.

O sistema não tem um módulo financeiro para contas à pagar e à receber.

1. **Restrições:**

O sistema não funcionará no navegador Internet Explorer, pois nesse navegador certas opções de design podem apresentar mal funcionamento, podendo provocar má interpretação por parte do usuário.

O servidor terá que possuir conexão com a internet para que acessos por dispositivos móveis, que estão fora rede local, sejam realizados.

1. **Limites do Projeto:**

O sistema funcionará a partir das versões dos navegadores:

- Chrome: 30.0.1599.101

- Mozilla Firefox: 28.0

- Opera: 18.00 Build 1284.63

Versões anteriores pode apesentar mal funcionamento e/ou limitar o uso. Os testes do software serão feitos a partir destas versões.

Não emite de nota fiscal.

O software será desenvolvido por duas pessoas.

1. **Funcionalidades**

* Cadastros
  + Alunos
  + Categoria
  + Funcionário
  + Função
  + Cidade
  + Estado
  + País
* Competições
  + Time
  + Jogo
  + Local
  + Galeria de imagens
* Configurações
  + Usuário
  + Empresa

1. **Cronograma**

O cronograma está dividido em data inicial e final, onde são as datas planejadas, e data início realizado (Realização) e data final realizado (Finalização).

As fases são a concepção, utilizada para definir e elaborar a ideia; elaboração, usada para definir a estratégia, com elaboração do cronograma e definição da documentação; construção, para o desenvolvimento do sistema; e por fim a transição, utilizada para realizar a interação com o usuário.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cronograma** | | | | | | |
| Etapas | REF | Data inicial | Data final | Realização | Finalização | Predec. |
| **1. Concepção** |  |  |  |  |  |  |
| *1.1 Definir ideia* |  |  |  |  |  |  |
| 1.1.1 Definir ações | 1 | 01/05/14 | 10/05/14 |  |  | - |
| 1.1.2 Verificar as possibilidades da construção da ideia proposta | 2 | 11/05/14 | 15/05/14 |  |  | 1 |
| 1.1.3 Finalizar definição da ideia | 3 | 16/05/14 | 16/05/14 |  |  | 2 |
| 1.1.4 Análise inicial da viabilidade da construção da ideia | 4 | 17/05/14 | 31/05/14 |  |  | 3 |
| *1.2 Análise de requisitos* |  |  |  |  |  |  |
| 1.2.1 Visitar escolas e buscar informações. | 5 | 01/06/14 | 23/07/14 | 22/06/14 | 28/06/14 | 4 |
| 1.2.2 Definir os diagramas de casos de uso. | 6 | 24/07/14 | 31/07/14 | 25/07/14 | 05/08/14 | 5 |
| 1.2.3 Criar os protótipos de tela. | 7 | 01/08/14 | 13/08/14 | 05/08/14 | 23/08/14 | 6 |
| 2.1.5 Revisar protótipos de tela e diagramas de casos de uso | 8 | 13/08/14 | 13/08/14 | 26/08/14 | 27/08/14 | 7 |
| 2.1.6 Definir riscos do projeto | 9 | 14/08/14 | 14/08/14 | 27/08/14 | 31/08/14 | 8 |
| **2. Elaboração** |  |  |  |  |  |  |
| *2.1 Cronograma* |  |  |  |  |  |  |
| 2.1.1 Definir tarefas e atividades para o período | 10 | 04/08/14 | 12/08/14 | 05/08/14 | 05/08/14 | 6 |
| 2.1.2 Definir as datas de execução | 11 | 04/08/14 | 12/08/14 | 05/08/14 | 05/08/14 | 10 |
| 2.1.3 Definir as predecessoras | 12 | 13/08/14 | 14/08/14 | 05/08/14 | 05/08/14 | 11 |
| 2.1.4 Preencher com as datas de realização (última será o dia da apresentação) | 13 | 04/08/14 | 02/12/14 | 05/08/14 |  | 12 |
| 2.1.5 Definir diagramas de classe | 14 | 15/08/14 | 20/08/14 | 30/08/14 | 31/08/14 | 13 e 9 |
| 2.1.6 Analisar a sequência das classes para desenvolvimento | 15 | 21/08/14 | 24/08/14 | 01/09/14 | 03/09/14 | 14 |
| 2.1.7 Revisar e finalizar a documentação | 16 | 24/08/14 | 31/08/14 | 30/08/14 | 07/09/14 | 15 |
| **3.Construção** |  |  |  |  |  |  |
| *3.1 Desenvolvimento* |  |  |  |  |  |  |
| 3.1.1 Definir etapas para programação | 17 | 01/09/14 | 04/09/14 | 03/09/14 | 05/09/14 | 16 |
| 3.1.2 Configurar ambiente tecnológico | 18 | 05/09/14 | 10/09/14 | 12/08/14 | 20/08/14 | 17 |
| 3.1.3 Desenvolvimento do sistema | 19 | 11/09/14 | 31/10/14 | 15/09/14 | 01/12/14 | 18 |
| 3.1.4 Testes | 20 | 01/11/14 | 06/11/14 |  |  | 19 |
| 3.1.5 Correções originadas dos testes | 21 | 01/11/14 | 10/11/14 |  |  | 20 |
| 3.1.6 Correções originadas do usuário | 22 | 22/11/14 | 26/11/14 |  |  | 21 |
| **4. Implementação ou Elaboração.** |  |  |  |  |  |  |
| *4.1 Implementação usuário* |  |  |  |  |  |  |
| 4.1.2 Apresentação ao usuário | 23 | 11/11/14 | 11/11/14 |  |  | 22 |
| 4.1.3 Treinamento para o usuário | 24 | 11/11/14 | 11/11/14 |  |  | 23 |
| 4.1.4 Período de testes do usuário | 25 | 12/11/14 | 22/11/14 |  |  | 24 |
| 4.1.5 Testes para correções | 26 | 22/11/14 | 26/11/14 |  |  | 25 |
| 4.1.6 Finalizar o teste do usuário | 27 | 26/11/14 | 26/11/14 |  |  | 22 e 26 |
| 4.1.7 Conversa final com o usuário | 28 | 27/11/14 | 27/11/14 |  |  | 27 |
| **5. Apresentação projeto** |  | **02/12/14** |  |  |  |  |
| *5.1 Preparação para apresentação* |  |  |  |  |  |  |
| 5.1.1 Revisão da documentação e envio para banca | 29 | 28/11/14 | 28/11/14 | 27/11/14 | 28/11/14 | 18 e 28 |
| 5.1.4 Finalizar projeto | 30 | 30/11/14 | 01/12/14 | 28/11/14 | 01/12/14 | 29 |
| 5.1.5 Dia apresentação | 31 | 02/12/14 |  |  | 02/11/14 | 30 |

1. **Ambiente Tecnológico**

O ambiente tecnológico trata-se da apresentação das ferramentas que foram utilizadas para o planejamento, desenvolvimento e implantação do software.

* 1. **Metodologia de desenvolvimento RUP**

O RUP é uma metodologia de desenvolvimento de software que usa uma abordagem de planejamento e execução iterativa e incremental seu principal objetivo é entregar o produto que o cliente realmente deseja e que será útil e com qualidade.

Iterativo pois suas técnicas avaliam a estratégia de planejamento de retrabalho, onde revisões e melhorias de partes do sistema são pré-definidos.

Em cada iteração:

* São identificados e especificados os casos de uso mais relevantes.
* É feita a análise e projeto dos casos de uso, usando-se a arquitetura como guia.
* São implementados componentes que realizam o que foi projetado.
* Verifica-se se os componentes satisfazem os casos de uso escolhidos.

E incremental, pois define uma estratégia de planejamento estagiado, onde várias partes do sistema são desenvolvidas em paralelo, e ajustadas quando completas. O sistema é desenvolvido com uma integração única.

O RUP é moldável ao tamanho do projeto de software, ele une as características boas das outras metodologias.

Usa a abordagem orientada a objetos e UML (UML é uma linguagem usada para especificar, modelar e documentar os artefatos de um sistema), principalmente os casos de uso, que são utilizados para especificar os requisitos, e durante a análise, projeto e implementação, são realizados. Na fase de teste, verifica-se se o sistema realiza o que foi descrito no caso de uso.

Está baseado na arquitetura, é definida nas primeiras iterações, que devem ser bem definidas, e o desenvolvimento consiste em complementá-la. A arquitetura organiza o desenvolvimento e ajuda a identificar oportunidades de reuso.

Também destaca-se a influência do cliente no processo de desenvolvimento, acredita-se que, com uma maior participação do cliente, algumas atividades poderiam ser mais produtivas e até o volume de artefatos poderia ser reduzido.

O RUP foi escolhido por principalmente se adequar ao tamanho de qualquer projeto de desenvolvimento, além disso, por oferecer características boas de outras metodologias. Em seu ciclo de vida, possui o loop que percebeu-se necessário para construção do projeto, o cronograma do projeto foi baseado nas fazes do RUP.

* + 1. Ciclo de vida

As fases são divididas por iterações. Cada uma delas indica a ênfase que é dada no projeto em um dado instante. Para capturar a dimensão do tempo de um projeto, o RUP divide o projeto em quatro fases diferentes:

* Concepção: estabelecer o escopo e viabilidade econômica do projeto, obtém uma visão do produto final a partir de um caso de uso definido.
* Elaboração: elimina principais riscos, ocorre o planejamento de atividades e recursos necessários, onde são definidas funcionalidades e a arquitetura a ser desenvolvida.
* Construção: implementação do software, construção do código.
* Transição: o produto é passado aos usuários. Nesta fase ocorre treinamento dos usuários (e possíveis mantenedores) e a avaliação do produto (“beta-testing”).

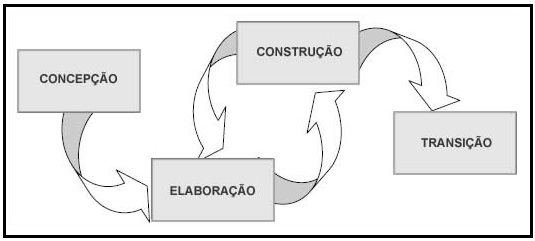


Figura 16 - Ciclo de Vida RUP

Fluxo de atividade básico:

* Requisitos
* Análise e projeto
* Implementação
* Testes
* Distribuição
  1. **Linguagem Groovy/Grails**

Grails é um framework para construção de aplicações para web através da linguagem de programação Groovy (uma linguagem dinâmica para a plataforma Java). Foi desenvolvido em código aberto e pretende ser um framework de alta produtividade graças à utilização do paradigma da programação por convenção que preserva o desenvolvedor dos detalhes de configuração.

Tem sua origem no Ruby on Rails, ele é um meta-framework, ou seja, uma junção de vários frameworks, que interage com a linguagem Groovy. As aplicações criadas utilizando o framework Grails são desenvolvidas com base no padrão de arquitetura MVC (Model-View-Controller), é um modelo de arquitetura de software que separa a representação da informação da interação do usuário com ele. O modelo (model) consiste nos dados da aplicação, regras de negócios, lógica e funções. Uma visão (view) pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões do mesmo dado, como um gráfico de barras para gerenciamento e uma visão tabular para contadores. O controlador (controller) faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão. As ideias centrais por trás do MVC são a reusabilidade de código e separação de conceitos.

Grails utiliza o mesmo conceito de produtividade do Ruby on Rails, que é baseado em “Conversão ao invés de Configuração”, e elimina quase que por completo a necessidade de configurações em arquivos XML, técnica que traz facilidade e produtividade.

O “G” do Grails vem de Groovy, que é uma linguagem de script, é uma linguagem para a Máquina Virtual do Java (JMV). Groovy é totalmente orientada a objeto mais simples e fácil de entender, trabalha de forma transparente com todos os objetos e bibliotecas Java existentes e compila diretamente para Java bytecode tanto no desenvolvimento de aplicações como no modo script. É ágil e dinâmica ela é baseada em Python, Ruby e Smalltalk e tem sua sintaxe similar ao Java, além disso é possível também “integrar” aplicações Java e Groovy de forma transparente. O Groovy simplifica a implementação por “adicionar” as suas classes os métodos Gets e Sets, economizando tempo e esforço.

Grails traz uma integração total entre Groovy e Java. Que além de possibilitar que você reutilize código Java existente naturalmente, pois roda na mesma JVM, permite também que ele se beneficie das melhores tecnologias Java existente como: Hibernate para persistência no banco de dados implementando ORM (banco de dados relacionais), Spring Framework e o SiteMesh para administrador layouts. Inclusive quando utiliza-se Grails, utiliza todas essas tecnologias descritas acima, além de várias outras que não foram citadas, tudo “debaixo dos panos”.

Ao invés de aprender todas essas tecnologias, Grails traz uma fina camada de abstração sobre elas, que faz subir o nível do desenvolvimento. Com isso é claro, traz mais produtividade, flexibilidade, robustez, facilidade em programar.

A escolha dessa linguagem deve-se por ser simples e de fácil manuseio, o MVC é uma forma clara de divisão na programação, traz a organização e delimita áreas de desenvolvimento. Além disso, Groovy consegue harmonia entre HTML, CSS, JavaScript, entre outras opções, o que tornou o código fonte flexível e organizado.

* 1. **PostgreSQL**

O PostgreSQL é um software que controla o armazenamento, recuperação, exclusão, segurança e integridade dos dados em um banco de dados. Um banco de dados relacional armazena dados em tabelas. Tabelas são organizadas em colunas, e cada coluna armazena um tipo de dados (inteiro, números reais, strings de caracteres, data, etc.). Os dados de uma simples “instância” de uma tabela são armazenados como uma linha. É desenvolvido como projeto de código aberto.

O PostgreSQL foi pioneiro em vários conceitos que somente se tornaram disponível muito mais tarde em alguns sistemas de banco de dados comerciais.

Devido à sua licença liberal, o PostgreSQL pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja particular, comercial ou acadêmica, livre de encargos.

O PostgreSQL será utilizado para manter as informações que o usuário cadastrar no sistema. Ficará instalado no computador principal ou servidor, o usuário não terá acesso ao banco de dados por medidas de segurança. O backup dos dados será realizado através de um agendador de tarefas que os sistemas operacionais possuem, normalmente o backup é agendado uma vez por dia, mas fica a critério do usário.

* 1. **Netbeans**

O NetBeans é um projeto open source de sucesso, com uma grande base de utilizadores. A Sun Microsystems fundou o projeto NetBeans em junho de 2000 e continua a ser o seu principal patrocinador.

Atualmente, existem dois produtos: o IDE NetBeans (NetBeans IDE) e a Plataforma NetBeans (NetBeans Platform).

O NetBeans IDE é um ambiente de desenvolvimento - uma ferramenta para programadores, que permite escrever, compilar, depurar e instalar programas. O IDE é completamente escrito em Java, mas pode suportar qualquer linguagem de programação. Existe também um grande número de módulos para estender as funcionalidades do IDE NetBeans. É também um produto livre, sem restrições à sua forma de utilização.

Também está disponível a Plataforma NetBeans; uma base modular e extensível que pode ser usada como infraestrutura para a criação de grandes aplicações de desktop.

Ambos os produtos são open source e livres para uso comercial e não comercial. O código fonte está disponível para ser usado através da licença Common Development and Distribution License (CDDL).

O sistema será desenvolvido completamente no NetBeans, pois ele na sua versão completa a ser adquirida no site da empresa, possui o pacote com a linguagem Groovy, que é utilizada no desenvolvimento do projeto.

* 1. **iReport**

O iReport é uma ferramenta que permite definir o design do relatório dentro de um ambiente gráfico, contento “todos” os recursos que a biblioteca JasperReports oferece.

É possível definir relatórios com designs modernos e complexos sem se quer escrever uma linha de código XML, que é todo gerado automaticamente.

O ambiente oferece atalhos para tarefas de compilação e visualização do relatório, permitindo a realização de testes.

O iReport tem a capacidade de criar relatórios com várias extensões, por exemplo: PDF, Open Office, DCOX, etc.

No sistema, os relatórios serão criados a partir do iReport terão a extensão .PDF, para posterior consulta e impressão.

* 1. **Apache Tomcat**

Um servidor de aplicação é uma plataforma sobre a qual roda a porção servidora de um aplicativo.

Do ponto de vista do software um servidor de aplicação consiste de um agrupamento de funções de negócio e de serviços que integrados satisfazem as necessidades dos usuários.

Um Servidor de Aplicações (em inglês Applications Server), é um servidor que disponibiliza um ambiente para a instalação e execução de certas aplicações, centralizando e dispensando a instalação nos computadores clientes. Os servidores de aplicação também são conhecidos por middleware.

O objetivo do servidor de aplicações é disponibilizar uma plataforma que separe do desenvolvedor de software algumas das complexidades de um sistema computacional. No desenvolvimento de aplicações comerciais, por exemplo, o foco dos desenvolvedores deve ser a resolução de problemas relacionados ao negócio da empresa, e não de questões de infraestrutura da aplicação. O servidor de aplicações responde a algumas questões comuns a todas as aplicações, como segurança, garantia de disponibilidade, balanceamento de carga e tratamento de exceções.

O servidor de aplicação será instalado num computador principal ou servidor, para que todos os usuários internos e externos possam ter acesso ao sistema.

* 1. **Charisma Bootstrap**

Simplificando, o Bootstrap é uma coleção de vários elementos e funções personalizáveis para projetos da web, empacotados previamente em uma única ferramenta. Ao projetar um site com o Bootstrap, os desenvolvedores podem escolher quais elementos querem usar. E, o mais importante, podem ter a certeza de que os elementos escolhidos não conflitarão entre si.

Esses elementos personalizáveis contidos no Bootstrap são uma combinação de HTML, CSS e Java Script. O Bootstrap tem diversas funções, como a capacidade de resposta móvel e várias opções de plug-in jQuery com muitos recursos.

O Bootstrap possui modelos para serem incorporados aos sistemas, o Charisma é um desses modelos, escolhido por possuir um design simples, limpo e totalmente responsivo, que se adapta aos aparelhos, facilitando aos usuários que poderão acessar e visualizar as funções em quaisquer dispositivos. Pode ser encontrado facilmente nas pesquisas por ser gratuito. Ele incorpora CSS e JavaScript, com isso, torna-se mais fácil a manipulação do ambiente e padroniza os campos inseridos.

* 1. **Pencil Project**

É uma ferramenta que auxilia o desenvolvedor e o cliente, pois projeta protótipos de tela.

Pencil Project é construído com a finalidade de fornecer uma ferramenta de prototipagem GUI livre e de código aberto que as pessoas podem facilmente instalar e usar para criar maquetes de plataformas de desktop populares.

Essa ferramenta foi utilizada para construir os protótipos de tela, a função principal é conscientizar o usuário de como o sistema irá se comportar, facilitando a compreensão para a entrevista. Quando o usuário apenas imagina como será, os detalhes são deixados para trás. Vendo a tela, o usuário tem uma noção maior e consegue perceber até os pequenos detalhes.

1. **Conclusão**

O Soccer Manager, trata-se de um sistema de cadastro e administração para escolas de futebol. As necessidades, funcionalidades e facilidades, foram implementadas com base nos levantamentos de requisitos, obtidos através de pesquisas realizadas em algumas escolas de futebol em Curitiba.

Esse sistema oferece uma plataforma que auxilia na administração de diversas atividades deste ramo de negócio. Através do Soccer Manager, é possível realizar cadastros de alunos e funcionários, além de organizar dados dos eventos de forma fácil e eficiente, através de uma interface amigável e altamente intuitiva.

Esse software é uma aplicação voltada para web, mas pode ser implantando localmente, sem perder nenhuma de suas funcionalidades.

Todo o projeto foi desenvolvido com ferramentas open source, o que diminui consideravelmente o custo do mesmo, tendo poucas distinções das ferramentas pagas, tendo o resultado esperado na conclusão do projeto.

Dessas ferramentas open source foram usadas algumas como a IDE NetBeans (ferramenta de desenvolvimento) PostgreSQL (ferramenta de manipulação da base de dados), entre outras diversas.

Esse sistema é uma solução gratuita, que inicialmente não será comercializada, mas conforme demanda poderá ser estudada essa possibilidade.

1. **Referências**

As referências estão dispostas em ordem cronológica de acesso aos sites.

* IBM Rational Unified Process. Disponível em: < http://pt.wikipedia.org/wiki/IBM\_Rational\_Unified\_Process > Acesso em 11/07/2014
* SOUZA, Givanaldo Rocha de, Levantamento de Requisitos. Disponível em: <http://docente.ifrn.edu.br/givanaldorocha/disciplinas/engenharia-de-software-licenciatura-em-informatica/levantamento-de-requisitos > Acesso 11/07/2014
* Modelagem de Casos de Uso (Parte 2). Disponível em: <http://www.ic.unicamp.br/~ariadne/mc426/cap02-2.pdf> Acesso 11/07/2014
* O que é um Recibo, e para que serve? Disponível em: <http://www.sempretops.com/diversos/recibo-pronto-modelos/> Acesso 19/07/2014
* BR, Claudio. Os sete equívocos dos casos de uso. Disponível em: <http://blog.claudiobr.com/os-sete-equivocos-dos-casos-de-uso/> Acesso 19/07/2014
* PINTO, Evandro M. Regra de Negócio Não é Requisito de Software. Disponível em: <http://wm2info.com.br/blog/2012/01/17/regra-de-negocio-nao-e-requisito-de-software/> Acesso 19/07/2014
* Visão Geral do RUP. Disponível em: <http://www.cin.ufpe.br/~if717/slides/3-visao-geral-do-rup.pdf> Acesso 22/07/2014
* Apache Tomcat. Disponível em: <http://pt.wikipedia.org/wiki/Apache\_Tomcat> Acesso 22/07/2014
* Apache Tomcat. Disponível em: <http://tomcat.apache.org/> Acesso 22/07/2014
* Índice de massa corporal. Disponível em: <http://pt.wikipedia.org/wiki/%C3%8Dndice\_de\_massa\_corporal> Acesso 22/07/2014
* KRUCHTEN, Philippe. What Is the Rational Unified Process? Disponível em: <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jan01/WhatIstheRationalUnifiedProcessJan01.pdf> Acesso 22/07/2014
* Rational Unified Process. Disponível em: <http://www.wthreex.com/rup/v711\_sp\_ptbr/index.htm> Acesso 23/07/2014
* Desenvolvimento iterativo e incremental. Disponível em: <http://pt.wikipedia.org/wiki/Desenvolvimento\_iterativo\_e\_incremental> Acesso 23/07/2014
* Conceito: Adaptando um Processo para um Projeto Pequeno. Disponível em: <http://www.wthreex.com/rup/v711\_sp\_ptbr/rup/guidances/concepts/tailoring\_a\_process\_for\_a\_small\_project,2.721222001330515E-305.html> Acesso 23/07/2014
* Conceito: Práticas Dinâmicas e RUP. Disponível em: <http://www.wthreex.com/rup/v711\_sp\_ptbr/rup/guidances/concepts/agile\_practices\_and\_rup,2.4535616775795524E-305.html> Acesso 23/07/2014
* RUP x Metodologias Ágeis: uma análise crítica. Disponível em: <http://unisinos-eslp.blogspot.com.br/2011/04/rup-x-metodologias-ageis-uma-analise.html> Acesso 23/07/2014
* STEFFEN, Juliana Berossa. O que são essas tais de metodologias Ágeis? Disponível em: <https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas\_o\_que\_s\_c3\_a3o\_essas\_tais\_de\_metodologias\_\_c3\_a1geis?lang=en> Acesso 24/07
* RUP – Rational Unified Process. Disponível em: <http://tec4rup.wordpress.com/2010/05/13/rup-fases/> Acesso 24/07/2014
* Rup x métodos ágeis. Disponível em: <http://www.trabalhosfeitos.com/ensaios/Rup-x-M%C3%A9todos-%C3%81geis/322735.html> Acesso 24/07/2014
* NETO, Jurmir Canal. Introdução ao Grails: FRAMEWORK WEB BASEADO EM MVC. Disponível em: <http://www.profissionaisti.com.br/2010/06/introducao-ao-grails-framework-web-baseado-em-mvc/> Acesso 24/07/2014
* Grails: um guia rápido e indireto. Disponível em: <http://www.itexto.net/devkico/?page\_id=220> Acesso 24/07/2014
* Instalando e configurando o Grails. Disponível em: <http://santograils.org/category/basico/> Acesso 24/07/2014
* User Guide. Disponível em: <http://groovy.codehaus.org/User+Guide> Acesso 24/07/2014
* FREITAS, Sávio. GROOVY – O QUE É? COMO FUNCIONA? VALE A PENA? Disponível em: <http://saviofreitas.wordpress.com/2011/03/20/groovy-o-que-e-como-funciona-vale-a-pena/> Acesso 24/07/2014
* Framework. Disponível em: <http://pt.wikipedia.org/wiki/Framework> Acesso 24/07/2014
* Groovy. Disponível em: <http://pt.wikipedia.org/wiki/Groovy> Acesso 24/07/2014
* Básico do Groovy para quem for aprender Grails. Disponível em: <http://www.itexto.net/devkico/?p=231> Acesso 24/07/2014
* Grails. Disponível em: <http://pt.wikipedia.org/wiki/Grails> Acesso 24/07
* Grails. Disponível em: <http://www.cpscetec.com.br/adistancia/grails> Acesso 24/07/2014
* Groovy, Portuguese Home. Disponível em: <http://groovy.codehaus.org/Portuguese+Home> Acesso 25/07/2014
* Ruby on Rails. Disponível em: <http://pt.wikipedia.org/wiki/Ruby\_on\_Rails> Acesso 25/07/2014
* Learn Grails. Disponível em: <https://grails.org/learn> Acesso 25/07/2014
* MVC. Disponível em: <http://pt.wikipedia.org/wiki/MVC> Acesso 25/07/2014
* PostgreSQL. Disponível em: <http://pt.wikipedia.org/wiki/PostgreSQL> Acesso 25/07/2014
* O que é o PostgreSQL? Disponível em: <http://pgdocptbr.sourceforge.net/pg82/intro-whatis.html> Acesso 25/07/2014
* Banco de dados relacional. Disponível em: <http://pt.wikipedia.org/wiki/Banco\_de\_dados\_relacional> Acesso 25/07/2014
* NETO, Arilo Cláudio Dias.Disponível em: <http://www.devmedia.com.br/bancos-de-dados-relacionais-artigo-revista-sql-magazine-86/20401> Acesso 25/07/2014
* Introdução e Histórico. Disponível em: <https://wiki.postgresql.org/wiki/Introdu%C3%A7%C3%A3o\_e\_Hist%C3%B3rico> Acesso 25/07/2014
* Padrões de Projeto : O modelo MVC - Model View Controller. Disponível em: <http://www.macoratti.net/vbn\_mvc.htm> Acesso 25/07/2014
* ALMEIDA, Rodrigo Rebouças de. Model-View-Controller (MVC). Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm> Acesso 25/07/2014
* BASTOS, Daniel Flores. O que é Model-view-controller (MVC)? Disponível em: <http://www.oficinadanet.com.br/artigo/desenvolvimento/o\_que\_e\_model-view-controller\_mvc> Acesso 25/07/2014
* MACEDO, Alexandre. Relatórios em Java – JasperReports e iReport. Disponível em: <http://www.k19.com.br/artigos/relatorios-em-java-jasperreports-e-irepor/> Acesso 25/07/2014
* Document Views. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/documentviews/relatorios.htm> Acesso 25/07/2014
* About. Disponível em: <http://www.postgresql.org/about/> Acesso 28/07/2014
* TOFFOLI, Giulio. iReport Designer Getting Started. Disponível em: <http://community.jaspersoft.com/wiki/ireport-designer-getting-started> Acesso 28/07/2014
* O que é o NetBeans? Disponível em: <https://netbeans.org/index\_pt\_PT.html> Acesso 28/07/2014
* NetBeans IDE Features. Disponível em: <https://netbeans.org/features/index\_pt\_BR.html> Acesso 28/07/2014
* NetBeans. Disponível em: <http://pt.wikipedia.org/wiki/NetBeans> Acesso 28/07/2014
* O que é o NetBeans? Disponível em: <http://www.oficinadanet.com.br/artigo/1061/o\_que\_e\_o\_netbeans> Acesso 28/07/2014
* Conheça o Apache Tomcat. Disponível em: <http://www.devmedia.com.br/conheca-o-apache-tomcat/4546> Acesso 28/07/2014
* MEDEIROS, Higor. Introduzindo o servidor de aplicação Apache Tomcat. Disponível em: <http://www.devmedia.com.br/introduzindo-o-servidor-de-aplicacao-apache-tomcat/27939> Acesso 28/07/2014
* PORTA, Giovanni La. O Software Apache Tomcat. Disponível em: <http://www.frameworksystem.com/blog/o-software-apache-tomcat/> Acesso 28/07/2014
* MALHERBI, Eduardo. Prototipação de Sistemas utilizando a Ferramenta Balsamiq Mockup. Disponível em: <http://www.devmedia.com.br/prototipacao-de-sistemas-utilizando-a-ferramenta-balsamiq-mockup/27232> Acesso 05/08/2014
* Pencil Project. Disponível em: <http://pencil.evolus.vn/WikiIndex.html> Acesso 05/08/2014
* Como Calcular o meu IMC? Disponível em: <http://www.calculoimc.com.br/como-calcular> Acesso 11/08/2014
* MARTINS, Vidal. Servidores de Aplicação. Disponível em: <http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1016> Acesso 02/09/2014
* Servidor de aplicação. Disponível em: <http://pt.wikipedia.org/wiki/Servidor\_de\_aplica%C3%A7%C3%A3o> Acesso 02/09
* UTTERBACK, Benjamin. O que é o Bootstrap? – Verdades e mitos: Parte 1 de 2. Disponível em: <http://www.prestashop.com/blog/pt/2014/03/06/o-que-e-o-bootstrap-verdades-e-mitos-parte-1-de-2/> Acesso 02/09/2014