



**INSTITUTO FEDERAL
SANTA CATARINA**

**CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA
TECNOLOGIA DA INFORMAÇÃO**

DECIO LUCIANO PLACIDO

**Modelagem de uma aplicação
WEB para gerenciamento e apoio
ao desenvolvimento do futebol
amador no Brasil**

**Florianópolis - SC
2019**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA - IFSC
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA
INFORMAÇÃO**

**ACADÊMICO:
DECIO LUCIANO PLACIDO**

**MODELAGEM DE UMA APLICAÇÃO WEB PARA GERENCIAMENTO E APOIO
AO DESENVOLVIMENTO DO FUTEBOL AMADOR NO BRASIL**

Trabalho de Conclusão de Curso
submetido ao Instituto Federal de
Educação, Ciência e Tecnologia de Santa
Catarina, como parte dos requisitos para
obtenção do título de Tecnólogo em
Gestão da Tecnologia de Informação.

Professor orientador:
Prof. Herval Daminelli, Esp.

**FLORIANÓPOLIS – SC
JANEIRO/2019**

Ficha de identificação da obra elaborada pelo autor.

Placido, Decio Luciano
Modelagem de uma aplicação WEB para gerenciamento
e apoio ao desenvolvimento do futebol amador no Brasil / Decio
Luciano Placido ; orientação de Herval Daminelli.
- Florianópolis, SC, 2019.
75 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. CST
em Gestão da Tecnologia da Informação. Departamento
Acadêmico de Saúde e Serviços.
Inclui Referências.

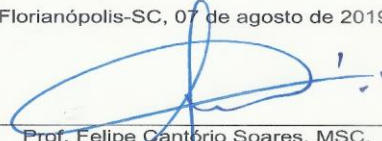
1. Modelagem de aplicação WEB. 2. Aplicativo mobile.
3. Futebol amador. 4. Ligas. 5. Clubes. I. Daminelli,
Herval. II. Instituto Federal de Santa Catarina. Departamento
Acadêmico de Saúde e Serviços. III. Título.

**MODELAGEM DE UMA APLICAÇÃO WEB PARA GERENCIAMENTO E APOIO AO
DESENVOLVIMENTO DO FUTEBOL AMADOR NO BRASIL**

DECIO LUCIANO PLACIDO

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 07 de agosto de 2019.



Prof. Felipe Cantório Soares, MSC.
Coordenador do CST em Gestão da Tecnologia da Informação
Instituto Federal de Santa Catarina


Banca Examinadora:



Prof. Herval Daminielli, Esp.
Orientador
Instituto Federal de Santa Catarina



Prof. Felipe Cantório Soares, MSC.
Instituto Federal de Santa Catarina



Underléa Cabreira Corrêa, MSC.
Instituto Federal de Santa Catarina

Dedico este trabalho a Deus, a minha família, colegas de classe, ao meu orientador e demais amigos que influenciaram neste projeto.

AGRADECIMENTOS

A Deus que me ilumina a caminhada, aos meus amados pais João e Maria Inês, queridos irmãos e filhos João Pedro e Joseph Anthony, a minha esposa Sílvia, obrigado pela paciência, incentivo, força e carinho. Valeu a pena esperar. Agradeço ainda, ao professor Herval Daminelli que aceitou fazer parte desse projeto, de maneira fundamental, aos professores que fizeram parte dessa trajetória, aos colegas de classe e amigos que contribuíram de alguma forma para que fosse possível alcançar esta conquista.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis.”

José de Alencar, escritor e político brasileiro

RESUMO

Este trabalho apresenta a modelagem e o desenvolvimento de uma aplicação *web*, construído inicialmente, através de uma plataforma de desenvolvimento de aplicativos, com intuito de comprovar as necessidades vislumbradas pelo autor e validar a ideia de produção de um projeto que abrange um sistema mais amplo de gerenciamento de campeonatos de futebol amador, organizados por ligas independentes em todo o país. O projeto contempla tecnologias, conceitos e fundamentos atuais, estudados ao longo do curso de Gestão de Tecnologia da Informação e utilizados em larga escala, por profissionais e empresas do setor de tecnologia e inovação.

Palavras-chave: Modelagem de Aplicação web. Aplicativo Mobile. Futebol amador. Ligas. Clubes.

ABSTRACT

This work presents the development of a web application design, initially built, through a platform for development of applications, in order to verify the authors needs and validate the idea of building a project that covers a wider system of amateur soccer championships management, organized by independent leagues all over the country. The project contemplates current technologies, concepts and fundamentals, studied in the course of Information Technology Management and used in large scale, by professionals and companies in the technology and innovation sector.

Keywords: Web Application Design. Mobile App. Amateur Football. Leagues. Clubs.

LISTA DE TABELAS

Tabela 1: Custos do projeto	29
Tabela 2: Descrição de caso de uso “Manter Liga”	49
Tabela 3: Descrição de caso de uso “Manter Clube”	51
Tabela 4: Descrição de caso de uso “Manter Campeonato”	53
Tabela 5: Descrição de caso de uso “Manter Jogos”	55
Tabela 6: Descrição de caso de uso “Manter Atleta”	57
Tabela 7: Descrição de caso de uso “Manter Árbitros”	59
Tabela 8: Descrição de caso de uso “Manter Patrocinador”	61

LISTA DE IMAGENS

Figura 1: Requisitos não funcionais.....	26
Figura 2: Visão geral dos Diagramas UML.....	32
Figura 3: Arquitetura do software distribuído.....	35
Figura 4: Arquitetura MVC.....	36
Figura 5: Comparação de planos Fábrica de Aplicativos.....	38
Figura 6: Arquitetura de Acesso Banco de dados e aplicação.....	41
Figura 7: Diagrama de Caso de Uso	48
Figura 8: Diagrama de Classe.....	63
Figura 9: Diagrama Entidade-Relacionamento.....	64
Figura 10: Telas do aplicativo.....	65

LISTA DE ABREVIATURAS

- API** Application Programming Interface (Interface de Programação de Aplicativos)
- CBF** Confederação Brasileira de Futebol
- CRM** Customer Relationship Management (Gerenciamento do Relacionamento com o Cliente)
- CSS** Cascading Style Sheets (Folhas de Estilo em Cascata)
- DER** Diagrama Entidade Relacionamento
- ERP** Enterprise Resource Planning (Planejamento de Recursos Empresariais)
- FIFA** Federação Internacional de Futebol Associada
- HTML** Hypertext Markup Language (Linguagem de Marcação de Hipertexto)
- MVC** Model-View-Controller (Modelo-Visão-Controlador)
- PDI** Plano Diretor de Informática
- PHP** PHP Hypertext Preprocessor
- SAP** Systeme Anwendungen und Produkt in der Datenverarbeitung (Sistemas, Aplicativos e Produtos para Processamento de Dados)
- SGBD** Sistema Gerenciador de Banco de Dados
- SPAC** São Paulo Athletic Club
- SQL** Structured Query Language (Linguagem de Consulta Estruturada)
- UML** Unified Modeling Language (Linguagem de Modelagem Unificada)
- XML** Extensible Markup Language (Linguagem de Marcação Extensível)
- Y2K** Termo designado para o Bug do Milênio

SUMÁRIO

1. INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO DO TEMA	14
1.2 JUSTIFICATIVA.....	17
1.3 PROBLEMA.....	18
1.4 OBJETIVOS.....	18
1.4.1 Objetivo Geral	19
1.4.2 Objetivos Específicos	19
1.5 CRONOGRAMA	20
2. METODOLOGIA	21
2.1. CLASSIFICAÇÃO DA PESQUISA	21
2.2. ORGANIZAÇÃO DO DOCUMENTO.....	21
2.3. DELIMITAÇÃO DE VARIÁVEIS DE PESQUISA	21
2.4. DELIMITAÇÃO DA PESQUISA	20
3. REFERENCIAL TEÓRICO	21
3.1. GESTÃO DA TECNOLOGIA DA INFORMAÇÃO	21
3.2. FUTEBOL.....	22
3.2.1. Histórico	23

	14
3.2.2 Modo de Jogar	24
3.3. ENGENHARIA DE SOFTWARE	26
3.3.1. Requisitos	26
3.4 Linguagens de programação Orientadas a Objetos e o Java	28
3.5 CSS	30
3.6 HTML	30
3.7 BANCO DE DADOS	30
3.7.1 Fundamentos	30
3.8. Engenharia de Requisitos	31
3.8.1. Análise Requisitos	30
3.9. PROTOTIPAÇÃO	33
3.10. SOFTWARE BASEADO EM COMPONENTES	35
3.11. SOFTWARE DISTRIBUÍDO	36
3.11.1. Arquitetura cliente-servidor	36
3.12. FRAMEWORKS	38
3.12.1. Fábrica de Apps	38
3.13. PROCESSO DE DESENVOLVIMENTO	41
3.14 ERP	41
3.15 CRM	44

4. DESENVOLVIMENTO	47
4.1. REQUISITOS FUNCIONAIS.....	47
4.2. REQUISITOS NÃO FUNCIONAIS	48
4.3. FERRAMENTAS SIMILARES.....	48
4.3.1 iEquipes	48
4.3.2 Starling Software	48
4.3.3 Progression	49
4.4. TECNOLOGIAS EMPREGADAS.....	48
4.5. DIAGRAMA DE CASO DE USO	48
4.6. DIAGRAMA DE CLASSES	62
4.7. DIAGRAMA ENTIDADE RELACIONAMENTO	64
4.8. O SISTEMA	65
5. CONCLUSÃO	72
REFERÊNCIAS BIBLIOGRÁFICAS	73

1. INTRODUÇÃO

A Tecnologia da Informação (TI) vem ocupando a cada dia que passa, um espaço maior nas organizações, tornando-se imprescindível na execução de processos e parte importante no amparo às tomadas de decisão.

Nesta era digital, essa questão fica ainda mais evidente, uma vez que os usuários prezam por agilidade, eficiência e uma boa experiência do cliente, tanto na navegação quanto na interação com o sistema.

Segundo Campos Filho (1994), a Tecnologia da Informação qualifica as práticas de trabalho do cotidiano nas organizações, sendo considerado, também, o ponto de partida para as inovações que se tornam cruciais à sua sobrevivência nos negócios.

1.1 CONTEXTUALIZAÇÃO DO TEMA

De acordo com Rodrigues Filho e Gomes (2004), a Tecnologia da Informação passou a ser considerado um elemento indispensável que contribui na evolução dos negócios nas organizações e a utilização da Tecnologia de Informação eximiu-se de realizar apenas o processamento e organização da informação, passando a ser o instrumento que oferece suporte à gestão da informação.

Porém, temos, também, que muitas empresas não utilizam a TI em sua totalidade, vendo a área apenas como um setor de suporte. Como ressalta Foina (2006), muitos dos benefícios prometidos pela TI nem sempre são visualizados pela empresa, devido, também, à complexidade da área e a falhas no gerenciamento do setor, que necessita de uma gestão ativa e alinhada com as estratégias da empresa.

Com o grande aumento de uso de computadores e celulares pelos usuários, muitas empresas têm buscado alternativas para otimizar seus processos e alcançar, de outras maneiras, seus clientes.

Neste contexto, o presente trabalho analisa como é possível estruturar um aplicativo que facilite as operações de times e torneios de futebol do Brasil, principalmente aqueles que têm poucos recursos. Normalmente, os times mais

periféricos do futebol brasileiro participam de ligas que têm poucos recursos e os procedimentos são muitas vezes manuais e informais. Isso prejudica a transparência, é um procedimento que pode ser lento e não permite o acompanhamento em tempo real das partidas e dos resultados do torneio. Todos estes pontos, contudo, podem ser alterados com a utilização de um aplicativo *web* desenvolvido para trabalhar com plataforma cliente *mobile*.

Um sistema *web* pode permitir que os administradores dos times e dos torneios insiram resultados e parâmetros de seus times e sobre os respectivos resultados em um sistema *web*, o qual é acessado por dispositivos clientes. Os dispositivos clientes podem ser plataformas *mobile* capazes de acessar informações em tempo real sobre os clubes e sobre os torneios em que participam.

Neste sentido, este trabalho traz uma ferramenta para a utilização das tecnologias da informação e comunicação como apoio ao gerenciamento e desenvolvimento de competições e clubes de futebol amador no Brasil. Este estudo apresenta e compreende o desenvolvimento do projeto de modelagem de um sistema *web* de baixo custo capaz de captar informações sobre torneios esportivos em tempo real.

1.2 JUSTIFICATIVA

O esporte amador atrai milhões de pessoas ao redor do mundo, seja para práticas individuais ou coletivas, para melhoramento da condição física, como instrumento de socialização entre pessoas e povos, para indivíduos e grupos que buscam o desafio da competição, ou apenas como um hobby.

Atualmente, percebem-se grandes investimentos em alguns poucos clubes e atletas profissionais, federações e confederações ao redor do globo. No entanto, o panorama do esporte amador, e nesse caso o futebol amador, nos mostra uma situação bastante distinta.

No Brasil, segundo uma estimativa da CBF, existem mais de 9.000 clubes amadores espalhados pelo país e, desses, somente 450 são filiados (Lance, 2018).

Já, na Inglaterra, o esporte possui 7.000 clubes, entre semiprofissionais e amadores, e é separado em 20 divisões (Lance, 2018).

O gerenciamento dos resultados e jogos de diversas ligas e campeonatos no país é feita, muitas vezes, de forma manual em papel, tabelas ou planilhas eletrônicas, sendo que a disponibilização destas informações ocorre normalmente via conversas informais ou formais, possuindo um alcance pequeno. O processo pode ser muitas vezes repetitivo, descontraído ou incorreto.

Temos, então, um alto percentual de competições que não utiliza ou oferece nenhum tipo de processo automatizado dentro da sua gestão.

Outro ponto de destaque é a agilização de processos que a TI pode trazer para os clubes e gerenciamento de jogos, permitindo a automatização de tarefas manuais, melhor custo/benefício para a administração de um clube e a maior facilidade com a venda de publicidade e a integração com mídias sociais.

Neste sentido, é possível citar fundamentos teóricos discutidos em sala, nas diversas disciplinas que compõem o curso, como Empreendedorismo, Sistemas de Informação, Tomada de Decisão, Modelagem de Dados, para citar apenas algumas que guiam o gestor a perceber oportunidades de mercado.

Alinhando tais conhecimentos adquiridos no decorrer da formação acadêmica às experiências vividas no ambiente de clubes e competições de futebol amador, pode-se verificar e comprovar as dificuldades e necessidades listadas nesta justificativa.

1.3 PROBLEMA

Atualmente não há uma ferramenta desenvolvida especificamente para as ligas de futebol amador que criam e organizam competições entre os clubes e que ofereça a automatização dos processos e a disponibilização das informações que ocorrem durante tais torneios. Acarretando em atraso, falta de transparência e credibilidade nos dados apresentados.

Diante dos fatos apresentados, faz-se necessário a produção de uma solução tecnológica que satisfaça tais anseios.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Criar a modelagem de uma ferramenta de aplicação *web* para plataformas *mobile* com o objetivo de automatizar os processos para o gerenciamento das informações relacionadas às competições entre agremiações do futebol amador.

1.4.2 Objetivos Específicos

De modo a atingir o objetivo geral, temos os seguintes objetivos específicos:

- Realizar o levantamento de requisitos para desenvolvimento do sistema de gerenciamento de campeonatos, clubes e atletas;
- Identificar métodos e processos manuais nas organizações que podem ser automatizados;
- Gerar a documentação e diagramas responsáveis por permitir o desenvolvimento da modelagem do sistema.

1.5 CRONOGRAMA

Ações	Janeiro	Fevereiro	Março	Abril	Maiο
Pesquisa	X	X			
Análise de Requisitos	X	X			
Prototipagem das Telas		X			
Definição Desenvolvimento			X	X	X
Testes					X

2. METODOLOGIA

Neste capítulo, apresentamos a classificação da pesquisa, a forma como este trabalho está organizado, bem como algumas delimitações da pesquisa e suas variáveis.

Comentado [A1]: Espaçamento entrelinhas

2.1. CLASSIFICAÇÃO DA PESQUISA

A presente pesquisa pode ser inserida em três classificações, de acordo com Silva e Menezes (2005): bibliográfica, exploratória e pesquisa aplicada.

Trata-se de uma pesquisa bibliográfica devido ao levantamento bibliográfico que deve ser realizado para que esse trabalho tenha o embasamento teórico adequado e para que o pesquisador possa se basear em dados já fundamentados.

Já, quanto à pesquisa exploratória, essa classificação dá-se de acordo com os objetivos, pois envolve levantamento bibliográfico e torna o problema claro, para que possa se encontrar soluções para o mesmo.

E, por fim, quanto à pesquisa aplicada, essa dá-se em vista de sua natureza, pois tem como objetivo gerar conhecimentos dirigidos para uma aplicação prática, para solução de problemas específicos.

2.2. ORGANIZAÇÃO DO DOCUMENTO

Esse documento tem a seguinte estrutura: No capítulo 3 temos a fundamentação teórica em que esse trabalho está baseado, apresentamos alguns sistemas existentes para elencar os requisitos necessários. No capítulo 4, segue-se o desenvolvimento deste projeto, tem-se os requisitos, casos de uso e diagramas. Por fim, no capítulo 5 apresentamos as conclusões e sugestões de trabalhos futuros.

2.3. DELIMITAÇÃO DE VARIÁVEIS DE PESQUISA

Para o desenvolvimento da modelagem do sistema, foi necessário fazer o levantamento dos requisitos funcionais e não funcionais, que dão base às funções que o sistema irá executar, bem como seus padrões de qualidade.

Também foi feito um levantamento dos sistemas já existentes, de forma a verificar se algum deles atendia os requisitos elencados para esse projeto e, como nenhum atendia, foram definidas as tecnologias que seriam utilizadas para o desenvolvimento dos requisitos levantados.

Também, foram desenvolvidos os diagramas necessários para total entendimento das regras de negócio e particularidades do sistema, entre eles: Diagrama de Caso de Uso, Diagrama de Classes, Diagrama de entidade e relacionamento.

2.4. DELIMITAÇÃO DA PESQUISA

A pesquisa é delimitada pela modelagem e desenvolvimento de um sistema para Web. Para isso, foram abordados diversos conceitos que permitiram o desenvolvimento dessa modelagem, tal como engenharia e arquitetura de software e boas práticas de desenvolvimento.

As ferramentas utilizadas foram escolhidas com base em seu valor (gratuitas), fácil acesso e uso, incluindo como adicionais uma comunidade consolidada para desenvolvimento, bem como a documentação existente para a mesma.

Este trabalho tem como objetivo o desenvolvimento do projeto e do projeto do sistema, sendo que a documentação aqui apresentada permite o desenvolvimento da ferramenta completa.

3. REFERENCIAL TEÓRICO

O referencial teórico abordou alguns temas centrais de interesse para a presente pesquisa, tais como: Um panorama da situação atual das competições, das ligas, dos clubes e atletas de futebol amador no país, a Gestão da Tecnologia da Informação como meio de desenvolvimento de atividade social, esportiva e cultural, Ferramentas de Desenvolvimento que possam ajudar na busca por estas soluções, Serviços e Gerenciamento de Projetos com foco nas boas práticas da TI. Temas que são de relevância para o entendimento e resultado final do trabalho.

3.1. GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

A concorrência e otimização do tempo fazem com que hoje exista uma necessidade constante pela procura de inovações tecnológicas, que, de acordo com Chede (2004), permitem uma melhoria e automatização dos processos de uma empresa, de forma a manter a mesma em um nível elevado ou estável de produtividade e competitividade.

Dessa forma, a Tecnologia da Informação (TI) possui cada vez mais um destaque maior dentro das empresas, pois é em sua estrutura que são baseados os sistemas de informações específicos e os serviços tecnológicos, para os quais são representados os processos gerenciais com a finalidade de se obter amparo à tomada de decisão.

De acordo, então, com Rodrigues Filho e Gomes (2004), a Tecnologia da Informação passou a ser considerada um elemento indispensável que contribui na evolução dos negócios nas organizações e a utilização da Tecnologia de Informação eximiu-se de realizar apenas o processamento e organização da informação, passando a ser o instrumento que oferece suporte à gestão da informação.

Porém, muitas empresas ainda não utilizam a Tecnologia da Informação de forma plena, utilizando a área de TI como mero setor de suporte de equipamentos e sistemas. Como destaca Foina (2006), muitos dos benefícios prometidos pela TI nem

sempre são visualizados pela empresa, devido, também, à complexidade da área e a falhas no gerenciamento do setor, que necessita de uma gestão ativa e alinhada com as estratégias da empresa.

Por sua vez, Meireles (2001) descreve a gestão de sistemas de tecnologias de informação como sendo uma ação administrativa norteada por critérios delimitados pelo Plano Diretor de Informática (PDI), englobando os elementos que expressam a estratégia estabelecida, quase sempre anualmente, do planejamento para a TI. Geralmente, o PDI encontra-se associado à modernização, ampliação ou expansão dos sistemas de informação da organização, sendo este o desdobramento do Planejamento Estratégico da TI.

Com o passar dos anos, a tendência é que as empresas cada vez busquem por mais processos de TI que automatizem seus processos e ofereçam dados para tomada de decisão. Diversas ferramentas já disponibilizam e oferecem essas funcionalidades, como por exemplo ERP's e CRM's.

3.2. FUTEBOL

O futebol é um dos esportes mais famosos no mundo, e segundo Murad (2007), ele também representa o esporte mais popular do planeta, pois envolve bilhões de pessoas, que vão desde jogadores amadores, torcedores e demais profissionais envolvidos.

Uma das grandes vantagens do futebol, e que contribui para que ele seja tão famoso e praticado é justamente ele não requerer materiais ou equipamentos de alto custo. No Brasil, ele é o esporte mais popular, segundo uma pesquisa realizada pela instituição Deloitte (2001) e estima-se que o número de adeptos no país chegue em 13,1 milhões, segundo a FIFA.

Com o número de atletas e expectadores crescendo dia após dia, observa-se o crescimento de competições e torneios amadores, geralmente envolvendo premiações. Segundo Polli (2015), a organização desses eventos não é uma tarefa fácil, devido à quantidade de times e atletas.

3.2.1. Histórico

A origem mais antiga encontrada do futebol data dos séculos III e II a.C., na China, que consistia em chutar uma bola com os pés para uma pequena rede, e uma de suas variações era a de ter que passar a bola por adversários para conseguir chegar na rede.

No Japão, encontramos um jogo chamado *kemaki*, que consiste em manter uma bola no ar passando-a entre os jogadores.

Começamos a ter notícias de um jogo cujo objetivo era passar a bola para o outro lado em um campo retangular na Grécia e Roma nos anos 700, 800 a.C.

Porém, o futebol nem sempre teve a fama que tem hoje, chegando a ser proibido por 500 anos na Inglaterra pelo Rei Eduardo III, devido a sua extrema violência e falta de regras (Hurtado, 2006).

No início de 1848 e nos anos seguintes, começou-se o processo de unificar as regras do esporte, sendo elas bem parecidas com as atuais, principalmente no que se refere a proibição de tocar a bola com as mãos e movê-la somente com os pés (The Cambridge University Rules November, 1863).

Segundo Giglio (2003), o futebol chegou ao Brasil somente em 1894, trazido por Charles Miller, que trazia na sua bagagem duas bolas de futebol. Apesar desta ser a data oficial, existem relatos de jogos de futebol com datas anteriores a essa, de acordo com Proni (2000).

Um dos problemas levantados por somente uma pessoa ser a introdutória do futebol em um país com nossas dimensões também é levantado por Giglio (2003), porém não podemos tirar o mérito de Charles Miller, pois, como os demais membros da elite britânica, ele frequentava o São Paulo Athletic Club (SPAC), e isso popularizou a prática do esporte fora das escolas (Santos Neto, 2002).

O futebol era inicialmente jogado somente pela elite, e somente alguns anos depois é que passou a fazer parte de grande parcela da população, que viu em suas regras fáceis e na sua jogabilidade em qualquer lugar um novo esporte, que logo ganhou força.

3.2.2 Modo de Jogar

O futebol pode ser praticado tanto em quadras abertas quanto fechadas, por homens ou mulheres. São necessárias 2 equipes, cada uma com 11 pessoas cada. O objetivo geral é marcar pontos fazendo a bola entrar no gol do time adversário utilizando somente os pés, pernas, tronco e a cabeça, o uso das mãos não é permitido.

Cada partida tem 90 minutos, sendo dividida em 2 tempos de 45 minutos cada, sendo que cada partida deve ser fiscalizada por um árbitro e ganha o time que fizer mais gols, já em caso de empate pode ser dado um tempo extra de 15 minutos no final dos tempos.

Caso um jogador cometa alguma falta ele pode receber 2 tipos de cartão, um amarelo como um aviso ou então um vermelho que o expulsa do jogo. Dois cartões amarelos equivalem a um vermelho, e caso algum jogador cometa uma falta dentro da grande área de sua defesa, deve ser marcado um pênalti em favor do time adversário, que consiste em um chute da bola direto para o gol por um jogador.

3.3. ENGENHARIA DE SOFTWARE

A engenharia de software é responsável por toda a documentação que auxiliará o desenvolvimento ou prototipação do sistema, incluindo seus requisitos, diagramas, telas, entre outros.

Uma engenharia de software bem desenvolvida auxiliará todo o processo de desenvolvimento, de forma que os requisitos sejam criados ou alterados de forma ordenada, tornando o desenvolvimento do software profissional, conforme definido por Sommerville (2011).

3.3.1. Requisitos

Os requisitos são a forma como será passado para os analistas, desenvolvedores e restante da equipe o que o sistema deverá fazer, fazendo com que a equipe técnica possa focar no desenvolvimento em si.

Segundo Ávila e Spínola (2007) os requisitos são as características ou descrição do que o sistema deve fazer para atender os objetivos propostos. Eles

Comentado [A2]: Décio, seria importantíssimo, aqui, inserires um novo tópico, tipo 3.2.3, discorrendo um pouco sobre o futebol amador no Brasil, hoje: o que são as ligas amadoras, quais são, por estados, como se processam as competições entre elas, como sobrevivem, etc. este tópico será, muito provavelmente, um item de questionamento da banca

restringem as funções do sistema e também parte do software responsável por exibir ou solucionar algum problema do mundo real.

Segundo os autores, os requisitos ainda são essenciais para que seja definida uma concordância entre o cliente e o fornecedor, sobre o que será feito, como o sistema deverá se comportar e como ele deve fazer isso. Os requisitos, também, servem como parâmetro para que o produto final possa ser validado e para que se reduzam os custos do desenvolvimento, caso sejam bem definidos, de forma a não gerar retrabalho.

Os requisitos, ainda, podem ser definidos em 2 tipos: funcionais e não funcionais. Os requisitos funcionais se referem ao que o sistema deve fazer, e como deve fazer. Enquanto que os requisitos não funcionais se referem à qualidade do software, ou seja, tempo de resposta, confiabilidade, simplicidade, segurança, disponibilidade, entre outros.

É importante, também, que os dois tipos de requisitos sejam considerados, pois a falta de qualquer um deles pode impactar no uso do sistema. Por exemplo, de nada adianta ter-se todas as funções bem implementadas e funcionando corretamente caso o sistema seja difícil de utilizar ou então o carregamento dos dados seja muito demorado.

Na Figura 1 apresentamos uma imagem que relaciona os tipos de

requisitos não funcionais, definidos para o bom funcionamento do sistema.

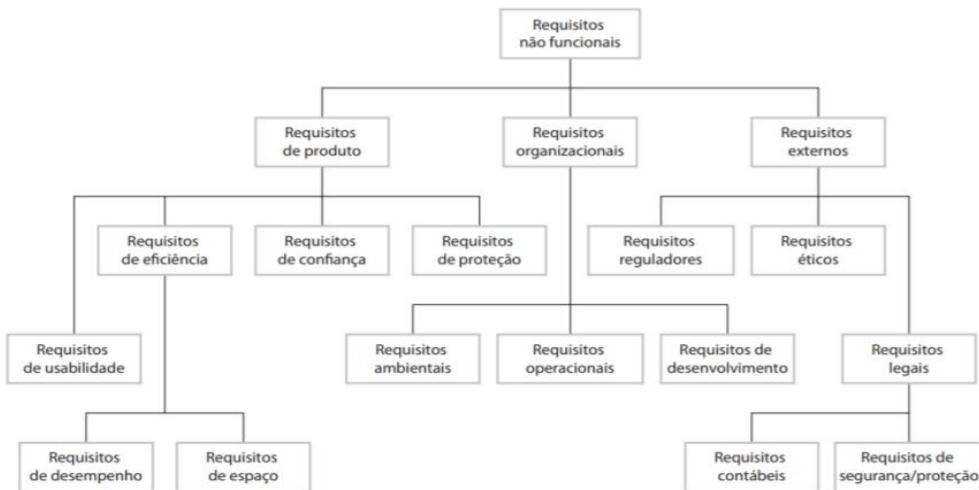


Figura 1: Requisitos não funcionais (Sommerville, 2011).

Por fim, temos também os requisitos de domínio. O objetivo deles é o de encontrar classes ou padrões que possam resultar na reutilização de código, melhorando a manutenção do sistema, a agilidade do desenvolvimento e a redução de custos, de acordo com Pressman (2011).

Desta forma, também é importante definir a arquitetura que será utilizada no desenvolvimento, pois a mesma impactará nos fatores descritos acima.

3.4 Linguagens de programação Orientadas a Objetos e o Java

As linguagens de programação orientadas a objetos trazem um considerável número de mecanismos que facilitam a escrita do código e ao mesmo tempo promovem a reutilização do mesmo. As vantagens deste paradigma de linguagem de programação são ilustradas abaixo (DEITEL, 2008).

- A codificação é mais próxima da realidade, o que aumenta a produtividade do desenvolvedor. É mais fácil para o mesmo desenvolver uma classe pessoa com as funcionalidades necessárias do que codificar todas as funções e todos os respectivos atributos no mesmo código que todo o resto do sistema.

- A manutenção é mais simples, pois é mais fácil encontrar as classes que devem ser atualizadas. Isto é consideravelmente mais fácil do que pesquisar em um conjunto de arquivos de bibliotecas de funções, o código que deve ser alterado.

- A reutilização de código é maior, pois é possível reutilizar as classes necessárias para sistemas que serão desenvolvidos no futuro. Por exemplo, um sistema que gerencia um time de futebol que tem a classe Pessoa, esta classe pode ser utilizada em um sistema de gerenciamento de torneios de futebol no futuro.

- Para programar em um sistema orientado a objetos é necessário instanciar as classes necessárias e além disso é possível utilizar métodos e atributos públicos, protegidos e privados. Estes mecanismos, além de facilitar a reutilização, aumentam a segurança do código. Muitas vezes, os códigos estruturados podem ser burlados por meio da inserção de métodos no sistema. Isto não é possível em sistemas desenvolvidos por uma linguagem orientada a objetos.

- É possível utilizar a herança do código para se reaproveitar parte de um sistema. Por exemplo, a classe Pessoa do sistema que gerencia times de futebol pode ser utilizada em um sistema de gerência de torneios, para facilitar a escrita do código da classe Juiz, a qual pode herdar as características da classe Pessoa.

Dentre as linguagens de Programação Orientadas a Objetos, a mais utilizada é o Java, que é uma linguagem de programação de alto nível e propósito geral desenvolvida pela Sun Microsystems. A linguagem de programação Java foi desenvolvida por uma pequena equipe de engenheiros, conhecida como Green Team, que iniciou a linguagem em 1991.

Originalmente chamado OAK, a linguagem Java foi projetada para dispositivos portáteis e decodificadores. OAK não teve sucesso e, em 1995, Sun mudou o nome para Java e modificou o idioma para aproveitar a crescente rede mundial de computadores.

Mais tarde, em 2009, a Oracle Corporation adquiriu a Sun Microsystems e assumiu a propriedade de dois importantes ativos de software da Sun: Java e Solaris.

3.5 CSS

CSS significa Cascading Style Sheets, com ênfase em “Style”. Enquanto o HTML é usado para estruturar um documento da *web* (definindo itens como manchetes e parágrafos, e permitindo que você incorpore imagens, vídeos e outras mídias), CSS vem e especifica o estilo do seu documento - *layouts* de página, cores e fontes são determinados com CSS. Pense no HTML como a base (cada casa tem uma) e CSS como as escolhas estéticas (há uma grande diferença entre uma mansão vitoriana e uma casa moderna de meados do século).

3.6 HTML

HTML significa Hypertext Markup Language. Ele permite ao usuário criar e estruturar seções, parágrafos, cabeçalhos, links e *blockquotes* para páginas da Web e aplicativos.

HTML não é uma linguagem de programação, o que significa que não é possível criar funcionalidades dinâmicas. Em vez disso, torna possível organizar e formatar documentos, de forma semelhante ao Microsoft Word.

3.7 BANCO DE DADOS

3.7.1 Fundamentos

Um banco de dados é uma coleção de informações que é organizada para que possa ser facilmente acessada, gerenciada e atualizada.

Os dados são organizados em linhas, colunas e tabelas e são indexados para facilitar a localização de informações relevantes. Os dados são atualizados, expandidos e excluídos à medida que novas informações são adicionadas. Os bancos de dados processam as cargas de trabalho para criar e atualizar-se, consultando os dados que contêm e executando aplicativos contra eles.

Os bancos de dados de computadores geralmente contêm agregações de registros ou arquivos de dados, como transações de vendas, catálogos de produtos e inventários e perfis de clientes.

3.8. ENGENHARIA DE REQUISITOS

Os requisitos são um dos itens mais complexos de se definir pelos engenheiros e analistas, pois, muitas vezes, nem o cliente sabe o que quer, e como quer, o que significa que, no decorrer do projeto, muitos requisitos são alterados. Caso essas mudanças não sejam feitas de forma ordenada, causarão um grande impacto na qualidade do software, seu custo e tempo de desenvolvimento.

Muitos engenheiros e desenvolvedores desejam iniciar o desenvolvimento antes mesmo da completa definição dos requisitos. Porém, se analisarmos a Tabela 1, podemos perceber como os custos com retrabalho tendem a ser muito maiores do que aguardar para iniciar. Como retrabalho, podemos citar a correção de falhas no desenvolvimento e também mudança nos requisitos.

	% do Custo de Desenvolvimento	% dos erros introduzidos	% dos erros encontrados	Custo relativo de correção
Análise de Requisitos	5	55	18	1
Projeto	25	30	10	1 – 1.5
Codificação e teste de unidade	50			
Teste	10	10	50	1 – 5
Validação e Documentação	10			
Manutenção		5	22	10 – 100

Tabela 1: Custos do projeto (Ávila e Spínola, 2007).

De acordo com Sommerville (2011), a engenharia de requisitos contempla pelo menos 4 áreas-base, sendo elas: estudo de viabilidade, análise de requisitos, especificações e validação. O estudo de viabilidade tem por objetivo avaliar se o sistema trará retorno para a empresa, sendo que esse retorno não necessariamente precisa ser financeiro, pode ser com relação a diminuição de custos ou aumento de produtividade. A análise de requisitos tem por objetivo estabelecer o que o sistema

deve fazer e suas restrições. As especificações convertem os requisitos em funcionalidades e por fim, a validação verifica se os requisitos atendem as necessidades do cliente e o que foi estabelecido.

3.8.1. Análise Requisitos

Após ser feito um estudo de viabilidade do sistema, deve-se efetuar a análise de requisitos. Esse é um processo particular de cada empresa. Porém, deve ser acompanhado de perto pelos responsáveis pelo projeto e pelo cliente, com *feedback* constante, de acordo com Sommerville (2011).

Nessa atividade, os engenheiros de software trabalham com clientes e usuários finais do sistema para obter informações sobre o domínio da aplicação, os serviços que o sistema deve oferecer, o desempenho do sistema, restrições de hardware e assim por diante. (SOMMERVILLE, 2011).

Um item que auxilia nos requisitos é a prototipação, pois, assim, certifica-se que todos os envolvidos estão trabalhando na mesma ideia-base, e também permite que interfaces e funcionalidades sejam modificadas antes do início do desenvolvimento. Dentro da análise de requisitos, temos alguns fatores-chave, de acordo com Sommerville (2011), tais como: descoberta dos requisitos, cenários, casos de uso e, por fim, a UML.

A descoberta dos requisitos consiste em entender qual o problema do cliente, ou então, qual problema o sistema irá solucionar. Para isso, é necessário um amplo esforço de comunicação com todos os envolvidos. Nessa etapa pode-se fazer uso de entrevistas, pesquisas de campo, entre outros.

Os cenários contemplam colocar os envolvidos dentro de uma simulação do funcionamento do sistema, sendo que isso pode ser feito através de protótipos. Os cenários podem ser uma sequência de eventos, ou mesmo, um texto simples e direto.

Cenários são largamente utilizados na Engenharia de Requisitos para descrever o comportamento do sistema de forma independente do projeto de software e, também, ajudar a testar modelos e especificações durante a validação de requisitos. Também, complementam a modelagem de metas do sistema, considerando essas abstrações que descrevem as intenções dos usuários em alto nível. Essas intenções, quando representadas por cenários, dão exemplos concretos de como o sistema deve funcionar para que os objetivos dos usuários sejam alcançados. (GUSMÃO; MARINHO; MURÍLO, 2011, p. 28).

Os casos de uso, segundo Melo (2009), têm por objetivo mostrar o comportamento do sistema e sua interação com os atores, ou seja, com os usuários envolvidos em cada caso de uso/funcionalidades. O grande ponto positivo dos casos de uso é que eles possibilitam a ligação das ações do sistema com os atores, sem que seja necessária nenhuma referência interna. Os casos de uso também demonstram o resultado esperado das ações, podendo, também, demonstrar os cenários alternativos.

Um ator (actor) representa um papel executado por um usuário ou por outro sistema que interaja com o sistema modelado. O ator não vai representar a pessoa e sim o papel que essa pessoa encena. Dessa forma, pode ser que a secretária Maria possa interagir com o sistema com dois ou mais papéis, o de secretária e o de vendedora. (MELO, 2009).

Por fim, a UML (*Unified Modeling Language*) é a linguagem considerada pelo mercado padrão para a modelagem. É uma linguagem visual que tem como

objetivo modelar, documentar e descrever projetos de software baseados em orientação a objetos. A UML também é útil para medir o sucesso de um software, conforme Pressman (2011), pois demonstra como os usuários irão interagir com o sistema e, respeitando os diagramas e a modelagem proposta, o nível de satisfação do usuário tende a ser mais elevado.

De acordo com Guedes (2011), os diagramas de UML são divididos em 6 categorias, contendo os seguintes itens demonstrados abaixo e melhor apresentados na Figura 2.

- Diagrama de Estruturas: Diagramas de classe, Diagrama de Estrutura Composta, Diagrama de Objetos, Diagrama de Componentes, Diagrama de Implantação e Diagrama de Pacotes.
- Diagramas de Comportamento: Casos de uso, Diagrama de Atividade, Diagrama de Máquina de Estados, Diagrama de Sequência, Diagrama de Comunicação, Visão Geral de Interação e Tempo.

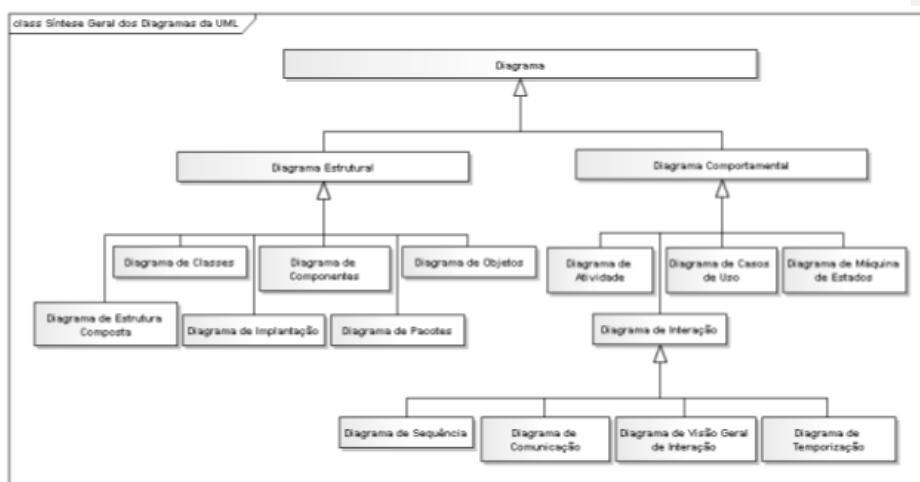


Figura 2: Visão geral dos Diagramas UML (Guedes, 2011).

3.9. PROTOTIPAÇÃO

De acordo com Sommerville (2011), um protótipo é uma forma inicial de demonstrar uma versão antes que a mesma seja desenvolvida ou concluída. Seu principal objetivo é antecipar possíveis alterações e, também, garantir que todos os membros da equipe têm o mesmo entendimento do projeto.

É importante que os protótipos também tenham seus objetivos definidos, seja de verificar a viabilidade do projeto, antecipar alterações ou mesmo apresentar uma pré-versão do que será desenvolvido para os clientes. Devem ser utilizados com moderação, pois também incluem custos e tempo para serem fabricados.

3.10. SOFTWARE BASEADO EM COMPONENTES

Um software baseado em componentes possui uma manutenção mais simplificada, pois podemos considerar que o desenvolvimento é feito através de blocos, que juntos formam o sistema como um todo. Para que esse tipo de desenvolvimento possa ser aplicado, é necessário que haja pouca ou nenhuma interação entre os blocos, de forma que o desenvolvimento de um afete o menos possível o outro.

O desenvolvimento baseado em componentes permite que o sistema final seja tratado como vários "mini sistemas", diminuindo sua complexidade e permitindo que cada componente empregado seja focado em apenas uma funcionalidade ou um conjunto de funcionalidades semelhantes. Deste modo, estas funcionalidades podem ser reutilizadas em diversas aplicações através do acesso ao componente. (OLIVEIRA, 2012).

Segundo Oliveira (2012), os maiores benefícios do uso do desenvolvimento por componentes é o de maior produtividade de desenvolvimento, maior qualidade, redução de custos e também uma maior padronização, já que todos os membros da

equipe precisam seguir um determinado padrão.

3.11. SOFTWARE DISTRIBUÍDO

Segundo o autor Sommerville (2011), uma aplicação distribuída é aquela que pode ser acessada por diversos computadores em locais diversos, ou seja, é um sistema mais complexo do que um sistema centralizado, onde o acesso parte de somente um local. Os sistemas distribuídos são essenciais para que se possa alterar as informações do sistema sem ter que fazer uma alteração diretamente no software.

A implementação pode ser feita em qualquer linguagem de programação, e pode ser acessada por qualquer dispositivo com acesso à internet. Na Figura 3, ilustramos melhor o funcionamento desse tipo de software.

3.11.1. Arquitetura cliente-servidor

A arquitetura cliente/servidor é um modelo de computação no qual o servidor hospeda, entrega e gerencia a maioria dos recursos e serviços a serem consumidos pelo cliente. Esse tipo de arquitetura tem um ou mais computadores clientes conectados a um servidor central através de uma conexão de rede ou de internet. Este sistema compartilha recursos de computação.

Dessa forma, por exemplo, é possível executar o programa ou aplicativo de diversas máquinas ou celulares, e o servidor enviará e responderá as requisições de todos. Sendo assim, o sistema pode ser acessado de diversas formas e por diversos usuários, independentemente do local onde se encontram, desde que possuam uma

conexão de dados com a internet.

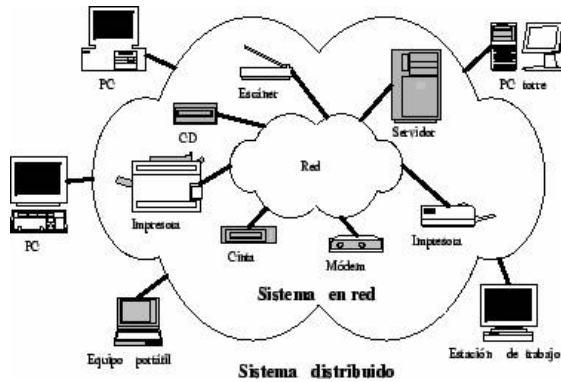


Figura 3: Arquitetura do software distribuído (Díaz, 2012).

Esse tipo de aplicação também precisa ser separado em camadas, ou seja, deixar a parte de processamento mais pesada para o servidor, que usualmente é a máquina com melhor desempenho. Dessa forma, todo o armazenamento e processamento de dados é feito no servidor, que alimenta a aplicação, seja ela mobile ou web.

Existem vários tipos de separação em camadas. A mais usual é a MVC, ou seja, *model-view-controller*. Nesse tipo de arquitetura, a camada *model* é responsável por conter toda a regra de negócio, enquanto a *view* apresenta a parte visual por onde o usuário interage com o sistema e, por fim, a camada *controller* é a responsável por fazer a interação entre a camada *view* e a camada *model*. Essa estrutura pode ser

Comentado [A3]: Havia um erro aqui.

visualizada na Figura 4.

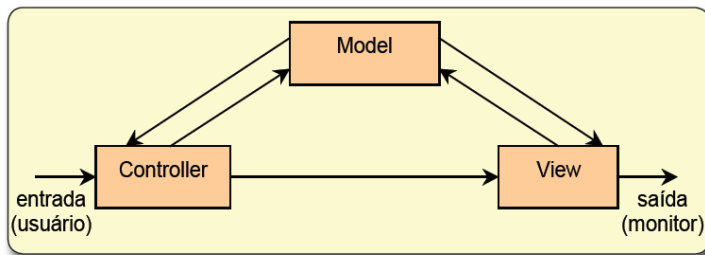


Figura 4: Arquitetura MVC (Couto e Filho, 2012).

3.12. FRAMEWORKS

Muitos desenvolvedores e equipes fazem uso de *frameworks* como forma de agilizar o processo de desenvolvimento e também de padronizar o mesmo, pois utilizam a reutilização de código e servem de esqueleto-base para as aplicações. Podem ser vistos, também, como um aglomerado de funções e metodologias para o desenvolvimento.

Como toda aplicação, o uso de *frameworks* apresenta vantagens e desvantagens. Como principais desvantagens podemos citar a curva de aprendizado e também ter que trabalhar com base no padrão do *framework*, sendo que, muitas vezes, pode ficar 'travado' devido à forma de trabalho de cada um. Como vantagens, podemos citar a padronização de código e desenvolvimento.

Hoje em dia, os principais *frameworks* no mercado são: Zend Framework, Laravel, CodeIgniter, Symfony e CakePHP. Para o desenvolvimento mobile podemos citar o Fábrica de Apps, que foi utilizado nesse trabalho.

3.12.1. Fábrica de Apps

A Fábrica de Aplicativos foi criada em 2013, e sua proposta é a de desenvolver aplicativos em minutos sem a necessidade de criar código. Nele, já existem mais de 400 mil aplicativos criados em mais de 80 países.

A proposta do *framework* é a de criar o aplicativo com base nos

componentes já existentes na plataforma, e existem os planos gratuito e pago, sendo o gratuito mais limitado e não permitindo, por exemplo, a publicação do app nas lojas de Play store e Apple store. Porém, o aplicativo pode ser baixado via QRcode no site da plataforma. Também existe uma limitação quanto à quantidade de páginas que podem ser criadas na versão gratuita.

Apresentamos mais algumas diferenças entre as versões paga e gratuita na Figura 5. Como grande vantagem, podemos ter, então, o desenvolvimento ágil de um aplicativo ou um protótipo para validação de uma ideia. Porém, como desvantagem, temos a limitação do plano gratuito quanto a funcionalidades, quantidade de telas e pelas atualizações no aplicativo terem que ser feitas pela plataforma.

Esta plataforma foi escolhida para o desenvolvimento do protótipo deste trabalho pois a sua versão gratuita satisfaz as necessidades do mesmo. A sua praticidade, flexibilidade e o aumento de produtividade que ela agrega são ideais para o desenvolvimento da proposta de projeto deste trabalho.

	GRÁTIS	ESPECIAL	PREMIUM
Conteúdo do App			
Quantidade de apps que podem ser criados	ilimitado	ilimitado	ilimitado
Quantidade de apps que podem ser publicados	0	1	ilimitado
Quantidade de abas no app	10 abas	200 abas	200 abas
Visualização de usuários	50 primeiros acessos	ilimitado	ilimitado
Templates e Modelos prontos	✓	✓	✓
Domínio app.vc	✓	✓	✓
Integração com página web	✓	✓	✓
Livre da nossa propaganda no app	✗	✓	✓
Integração com propagandas do Google (Admob)	✗	✓	✓
Integração com Google Analytics	✗	✓	✓
Publicação e Acesso			
Web App	✓	✓	✓
Publicação Google Play (Android)	✗	✓	✓
Publicação App Store (iPhone)	✗	✓	✓
Acesso ilimitado ao app	✓	✓	✓
Downloads ilimitados	✗	✓	✓
Notificações push ilimitadas	✗	✓	✓
Logins e cadastros ilimitados	✗	✓	✓
Subconta de acesso	✗	✗	✓
Suporte			
Suporte via e-mail	✓	✓	✓
Suporte via chat	✓	✓	✓
Acesso a Central de Ajuda	✓	✓	✓

Figura 5: Comparação de planos Fábrica de Aplicativos (Fábrica, 2019).

3.13. PROCESSO DE DESENVOLVIMENTO

Independente do desenvolvimento ser para *web* ou para *mobile*, alguns processos são os mesmos, como por exemplo: a criação do design, do *front-end* e do *back-end*. Esses são os principais itens de qualquer desenvolvimento.

O design consiste em projetar onde ficará cada item do *layout* como textos, botões e imagens. Também é o responsável pelas cores a serem utilizadas. O *web design* é uma área de estudo caracterizada pelo desenvolvimento de interfaces com o usuário em *web sites* e aplicações *web*. Podemos considerar o *web design* como parte do design gráfico, que é muito mais abrangente, pois envolve o desenho de qualquer tipo de *layout*, *banner* ou mídia computadorizada. (SILVA, 2013)

O *front-end* é o responsável por transformar o *layout* criado na etapa do *design* de uma forma que possa ser utilizado pelos usuários, com a interação dos botões, textos, entre outros. Então, podemos dizer que o *front-end* é o responsável por transformar em código o *design*, utilizando a linguagem HTML e a marcação de estilos CSS no caso das aplicações *web* e em XML no caso das aplicações *mobile*.

Por fim, temos o *back-end*, responsável pelas regras de negócio e funcionalidades da aplicação. Nessa etapa, é utilizada uma linguagem de programação como PHP, JAVA, Python, Asp.net, entre outros.

Dentro do *back-end* temos também os banco de dados, que são os responsáveis por armazenar os dados da aplicação, os mais utilizados são o MySQL e SQL. Dados podem ser vistos como uma representação de fatos, conceitos ou instruções de uma maneira normalizada, podendo ser adaptados à comunicação, interpretação e processamento. Já a informação pode ser vista como todo o conjunto de dados devidamente ordenados e organizados de forma significativa. Os

bancos de dados, por sua vez, foram concebidos com o objetivo de possibilitar o armazenamento de informações em sistemas de arquivos permanentes, com o intuito de possibilitar posteriores acesso e manipulação de informações, de forma organizada e estruturada. (CAYRES, 2015).

A Figura 6 ilustra o processo de comunicação das linguagens de programação com um banco de dados. Dessa forma, temos os usuários que fazem requisições para um software gerenciador de banco de dados. Esse software utiliza um programa para consultar os dados armazenados em um determinado banco de dados.

O desenvolvimento do projeto de banco de dados segue 3 etapas: o desenvolvimento do modelo entidade relacionamento, o modelo relacional e, por fim, o modelo físico. O modelo entidade relacionamento começa a fazer o mapeamento das entidades que serão as tabelas com outras entidades, de forma a analisar a ligação entre as mesmas e como o modelo deve ser desenvolvido, de maneira a facilitar o reuso de código e evitar a duplicação de informações. Após esse modelo, é desenvolvido o modelo relacional, que já inclui as respectivas colunas de cada tabela com o tipo de dados esperado. E, por fim, é desenvolvido o modelo físico e implementado em algum gerenciador de banco de dados, como MySQL, Postgres, SQL Server, Oracle, entre outros.

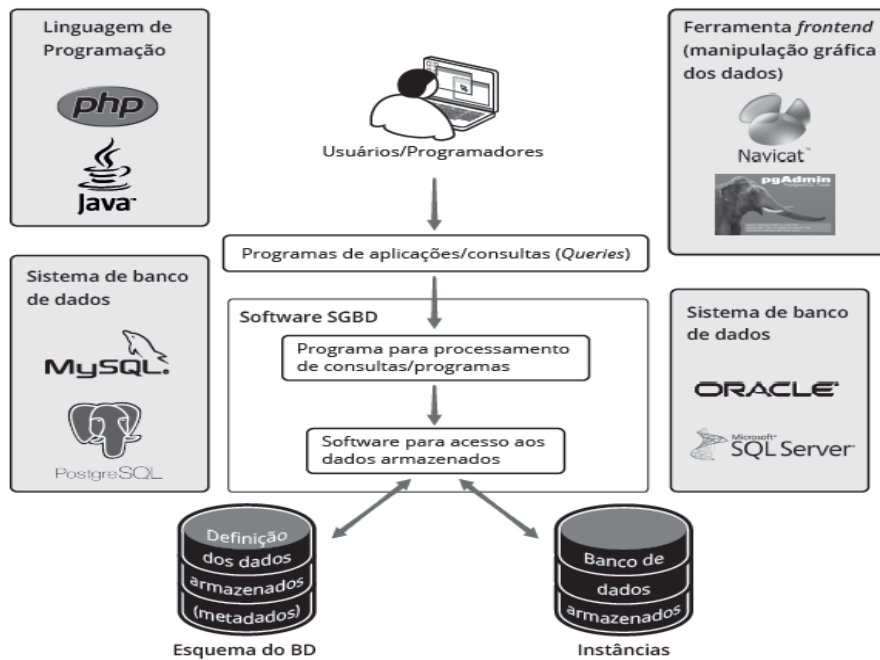


Figura 6: Arquitetura de Acesso Banco de dados e aplicação (Cayres, 2015).

3.14 ERP

O ERP (Enterprise Resource Planning, planejamento de recursos empresariais) é um software de gerenciamento de processos de negócios que permite que uma organização use um sistema de aplicativos integrados para gerenciar os negócios e automatizar muitas funções de *back office* relacionadas à tecnologia, serviços e recursos humanos.

O *software* ERP normalmente integra todas as facetas de uma operação - incluindo planejamento, desenvolvimento, fabricação, vendas e marketing - em um único banco de dados, aplicativo e interface do usuário.

O *software* ERP é considerado um tipo de aplicativo corporativo, que é um software projetado para ser usado por empresas maiores e geralmente requer equipes

dedicadas para personalizar e analisar os dados, além de lidar com atualizações e implantação. Por outro lado, os aplicativos ERP de pequenas empresas são soluções de *software* de gerenciamento de negócios leves, muitas vezes personalizadas para um setor de negócios específico ou vertical.

O caso de negócios para implementar um sistema ERP pode ser visto ao examinar qualquer uma das três histórias da Nestlé. A Nestlé SA é a empresa controladora do gigante de fabricação de doces e está sediada na Suíça (Konicki, 2000). Em 2000, a Nestlé SA decidiu que queria alavancar seu tamanho e começar a agir como o gigante que é. Para isso, assinou um contrato de US \$ 200 milhões com a SAP para implantar um sistema de ERP para seus 230.000 funcionários em 80 países em todo o mundo (Olson, 2004). Além dessa quantia, a Nestlé SA também se comprometeu com mais US \$ 80 milhões a serem gastos em consultoria, manutenção e atualizações (Konicki, 2000). Executivos da Nestlé SA perceberam que a empresa precisava padronizar seus processos de negócios se quisesse ser competitiva. O lançamento estava programado para levar três anos para os maiores sites da Nestlé SA, com os outros seguindo. Incluídos na implementação estavam os módulos financeiro, contas a pagar, contas a receber, planejamento, gerenciamento de produção, suprimento, compras diretas, cadeia de suprimento, planejamento de demanda, atendimento e inteligência de negócios do mySAP.com (Konicki, 2000).

Antes da decisão do ERP da Nestlé SA, a Nestlé UK já havia implementado um sistema ERP. A subsidiária britânica da Nestlé SA implementou o SAP R / 3 durante um período de cinco anos em 18 fábricas no Reino Unido (Glick, 2001). Essa implementação foi concluída em 1999 e foi um dos maiores sistemas de ERP do Reino Unido, com mais de 6.000 usuários (Glick, 2001). Assim como na implantação da Nestlé SA, os objetivos da implementação da Nestlé no Reino Unido foram centrados em alavancar o tamanho da organização, bem como reforçar a cadeia de suprimentos e reprojeter práticas e processos de trabalho (Glick, 2001).

A terceira história de implementação do ERP da Nestlé envolve a Nestlé USA. A Nestlé USA é a subsidiária norte-americana de US \$ 8,1 bilhões da Nestlé SA.

Em 1997, a Nestlé USA iniciou seu próprio projeto de ERP conhecido como Melhor (Excelência Empresarial através da Tecnologia de Sistemas). Programado para ser executado ao longo de seis anos, terminando no primeiro trimestre de 2003, este projeto foi orçado em mais de US \$ 200 milhões e implementaria cinco módulos SAP: compras, finanças, vendas e distribuição, contas a pagar e contas a receber. Semelhante às outras duas divisões da Nestlé, o objetivo por trás dessa implementação do ERP era a unificação.

Além disso, o projeto resolveria os problemas do Y2K da Nestlé USA (Worthen, p. 3). No caso da Nestlé USA, o ERP era parte da visão que o Presidente e CEO da Nestlé USA, Joe Weller, chamou de "One Nestlé", responsável por "transformar as marcas separadas em uma empresa altamente integrada"). Antes da implementação, a Nestlé USA tinha nove diferentes registros gerais e 28 pontos de entrada de clientes (Worthen, pág. 2). O objetivo do projeto ERP era reduzir esses números para um. Uma das visões mais interessantes sobre o problema da Nestlé EUA é a história da baunilha. Antes da implementação do ERP, a Nestlé USA não atuava como uma única empresa. Em vez disso, cada local agia em seu próprio nome e era livre para tomar suas próprias decisões comerciais. "Em 1997, uma equipe que examinou os vários sistemas da empresa descobriu, entre muitas outras redundâncias preocupantes, que as marcas da Nestlé EUA pagavam 29 preços diferentes pela baunilha - para o mesmo fornecedor".

Essa situação surgiu do fato de que cada fábrica realizava seus próprios negócios com o fornecedor e o fornecedor ajustava o preço por fábrica com base no que eles achavam que a fábrica pagaria. A situação foi piorada pelo fato de cada fábrica se referir à baunilha de maneira diferente. Enquanto uma fábrica poderia se referir à baunilha como 1234, outra fábrica se referia a ela como 7890. Isso tornava quase impossível para os indivíduos na sede corporativa fazer comparações entre as fábricas para ver os custos de fabricação.

3.15 CRM

CRM ou *Customer Relationship Management* é uma estratégia para gerenciar os relacionamentos e interações de uma organização com clientes e clientes em potencial. Um sistema de CRM ajuda as empresas a permanecerem conectadas aos clientes, agilizar processos e melhorar a lucratividade.

Quando as pessoas falam sobre CRM, elas geralmente se referem a um sistema de CRM, uma ferramenta usada para gerenciamento de contatos, gerenciamento de vendas, produtividade e muito mais. O objetivo de um sistema de CRM é simples: melhorar os relacionamentos comerciais.

4. DESENVOLVIMENTO

A modelagem da aplicação em questão tem por objetivo automatizar os processos para o gerenciamento das informações relacionadas às competições entre agremiações do futebol amador.

A otimização do acesso às informações é essencial para uma sociedade globalizada, onde as informações devem estar disponíveis de forma simples, fácil e sempre que necessário. O aplicativo apresenta informações sobre campeonatos, tais como artilheiros, tabela de jogos, cartões e suspensões, bem como demais estatísticas e notícias relacionadas.

Com base nas informações obtidas e nas necessidades levantadas, temos a definição do objetivo do protótipo e os requisitos que foram definidos para que o mesmo possa ser alcançado.

4.1. REQUISITOS FUNCIONAIS

O sistema deve conter os seguintes requisitos funcionais:

- Cadastro da liga contendo em anexo os contratos digitalizados, histórico e demais informações como nome, fundação, símbolo, entre outros.
- Cadastro do clube, com informações similares às da liga, e também com a planilha de *Scout* (olheiro).
- Cadastro de campeonatos com o nome, data e hora de início, categoria, equipes participantes, número de rodadas, fases e cadastro de jogos.
- Cadastro de jogos com informações completa a respeito de placares, cartões, tempo de jogo, jogadores, pênaltis, entre outros.
- Cadastro de atleta, contendo informações gerais, anexos de contrato bem como cadastros de antropometria (medidas de partes do corpo humano).
- Cadastro de árbitro com informações pessoais e sua formação.
- Cadastro de patrocinador contendo informações pessoais e do plano de campanha, como o plano contratado, período e data de início e fim.
- As informações devem ser disponibilizadas de forma gratuita para os

usuários, não sendo necessário nenhum login e senha para acessá-la.

4.2. REQUISITOS NÃO FUNCIONAIS

O sistema deve compreender os seguintes requisitos não funcionais:

- Deve ser compatível com as novas versões de dispositivos *mobile*.
- O layout deve se ajustar aos mais diversos formatos de tela de dispositivos.
- Deve ser facilmente exportado para as principais lojas de disponibilização de aplicativos no mercado (Apple store e Google play), o que demandaria a compra da licença da plataforma Fábrica de Software.
- As informações devem carregar em menos de 10 segundos.

4.3. FERRAMENTAS SIMILARES

Na pesquisa realizada, foram encontrados três sistemas de gerenciamento de times com uma proposta similar à apresentada, sendo eles: iEquipes, Starling Software e Progression.

Não encontramos um aplicativo similar ao proposto 100% mobile, sendo que, em sua maioria, as ferramentas são web e o acesso é somente por planos pagos, dificultando os testes e análise das ferramentas.

4.3.1 iEquipes

O iEquipes, assim como o Progression, não possui um cadastro para *trial* (versão para teste por tempo limitado) ou mesmo gratuito, de forma a possibilitar a avaliação das funcionalidades.

4.3.2 Starling Software

Ferramenta 100% desktop, de maneira que não encontramos no mesmo uma proposta similar a nossa. Sendo assim, por ter natureza e aplicabilidade diferente

de nosso protótipo, este não será analisado.

4.3.3 Progression

O mesmo possui site em português, porém não possui representantes no Brasil, conforme informa no seu site, e também não disponibiliza um plano gratuito para testes.

4.4. TECNOLOGIAS EMPREGADAS

Conforme descrito na seção de referencial teórico, optamos por utilizar a plataforma Fábrica de Aplicativos para o desenvolvimento do sistema, de forma a otimizar o tempo de desenvolvimento e obter uma melhor validação do projeto. Nele, são feitas todas as alterações e inclusões de conteúdo no aplicativo, através de painéis e tipos de dados em cada tela.

4.5. DIAGRAMA DE CASO DE USO

Os diagramas de caso de uso demonstram o funcionamento de sistema a partir dos usuários, apresentando quais são os atores envolvidos em cada funcionalidade.

Os casos de uso são utilizados para capturar os requisitos do sistema, ou seja, referem-se aos serviços, tarefas ou funcionalidades identificadas como necessários ao software e que podem ser utilizados de alguma maneira pelos atores que interagem com o sistema, sendo usados para expressar e documentar os comportamentos pretendidos para as funções deste. (GUEDES, 2011).

Para o desenvolvimento do diagrama foi utilizada a ferramenta online e

gratuita Lucidchart, e o diagrama pode ser visualizado na Figura 7.

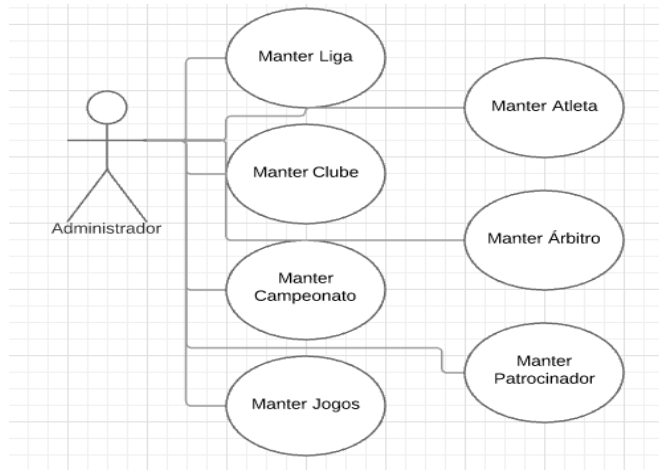


Figura 7: Diagrama de Caso de Uso (do Autor).

Nome do Caso de Uso	Manter Liga
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de liga no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Cadastro de responsável, incluir anexos, histórico e contrato
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter liga.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 2: Descrição de caso de uso Manter Liga.

Nome do Caso de Uso	Manter Liga
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de clube no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Cadastro de responsável, incluir anexos, histórico e planilha Scout
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter clube.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 3: Descrição de caso de uso Manter Clube.

Nome do Caso de Uso	Manter Campeonato
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de campeonato no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Campeonato cadastrado
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter campeonato.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 4: Descrição de caso de uso Manter Campeonato.

Nome do Caso de Uso	Manter Jogos
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de jogos no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Jogo, times, tempos de jogos e jogadores cadastrados
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter jogos.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 5: Descrição de caso de uso Manter Jogos.

Nome do Caso de Uso	Manter Atletas
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de atletas no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Atletas cadastrados
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter atletas.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 6: Descrição de caso de uso Manter Atletas.

Nome do Caso de Uso	Manter Árbitros
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de árbitros no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Árbitro cadastrado
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter árbitro.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 7: Descrição de caso de uso Manter Árbitro.

Nome do Caso de Uso	Manter Patrocinador
Caso de Uso Geral	
Ator Principal	Administrador
Atores Secundários	
Resumo	Descreve a etapa de cadastros de patrocinador no sistema
Pré-condições	Usuário logado e com permissão adequada
Pós-condições	Patrocinador, campanha e período da campanha cadastrado
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciono opção de manter patrocinador.	
	2. Apresento a tela de cadastro selecionada
3. Seleciono adicionar	
	4. Apresento tela de cadastro
5. Preenche os dados do formulário	
	6. Salva dados do formulário

Restrições/ Validações	1. Usuário precisa estar logado.
Fluxo Exceção	
Ações do Ator	Ações do Sistema
	1. Sistema informa que cadastro já se encontra cadastrado para esse usuário. 2. Abre formulário de cadastro

Tabela 8: Descrição de caso de uso Manter Patrocinador.

4.6. DIAGRAMA DE CLASSES

O diagrama de classes demonstra o sistema como um todo, já apresentando a ligação entre as tabelas e seus respectivos campos e atributos. O

diagrama pode ser visualizado na Figura 8.

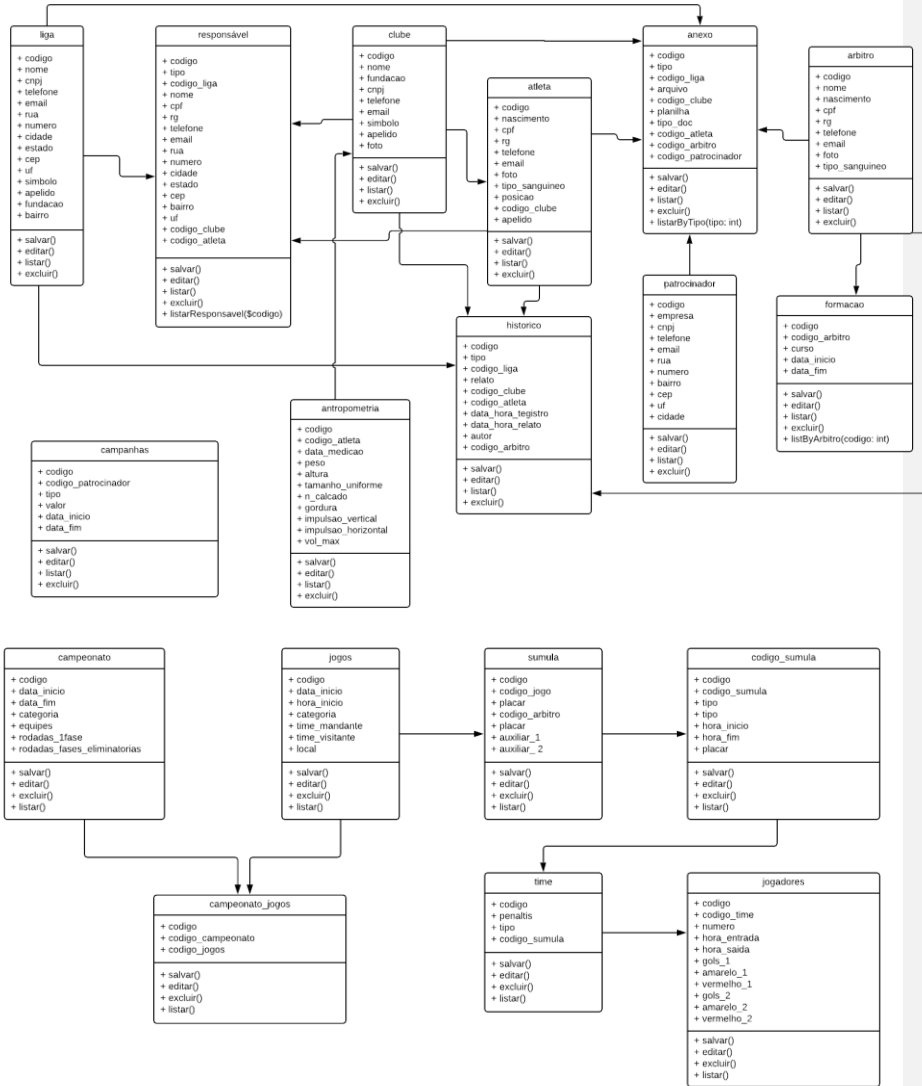


Figura 8: Diagrama de Classes.

4.7. DIAGRAMA ENTIDADE – RELACIONAMENTO

O diagrama entidade-relacionamento busca apresentar o funcionamento geral do sistema de forma mais abrangente. Ele serve como base para a criação do modelo lógico e encontra-se disponível na Figura 9.

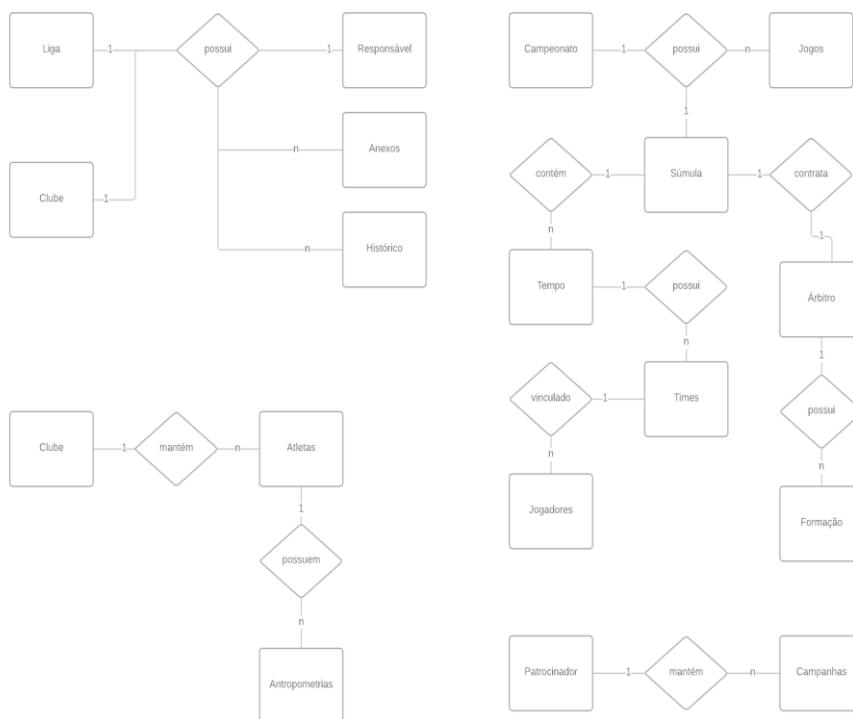


Figura 9: Diagrama entidade – relacionamento.

4.8. O SISTEMA

Abaixo pode-se visualizar os prints de tela do aplicativo mostrando basicamente todas as suas funcionalidades, as quais serão citadas pela sequência: A tela 1 é a Tela Principal para o cliente. Mostra o menu com as abas Master 40 e Campeonatos de Base, que são as categorias. Mais abaixo, a aba Notícias e a última, a aba referente ao cliente, a Liga organizadora. A tela 2, que é a tela principal dentro da categoria escolhida, introduz o Master 40 e suas abas, “O Campeonato”, “Estatísticas” com números do torneio da categoria, “Clubes” participantes e “Estádios” onde os clubes da categoria, mandam seus jogos. Aparecem, aqui, também, Os “Elencos” de cada clube participante e a “Galeria” com imagens do torneio. A tela 3 mostra o submenu da aba “Notícias”, com informações sobre a Abertura do Campeonato, Recessos, Julgamentos, entre outras questões.

Na página 67, temos a tela 4, que é a tela “O campeonato”, com as abas “Tabela de Jogos”, “Cartões e Suspensões” e “Artilheiros”. Na tela 5, observa-se a classificação e a tabela de jogos. A tela 6 contém o controle de Cartões e Suspensões.

Na página 68, temos a tela 7, onde há a tabela de artilheiros. A tela 8 contém o submenu “Estatísticas” e suas abas “Os Campeões”, “Arbitragem” e “Números do Campeonato”. A tela 9 é a aba “Campeões”, onde está a lista de todos os campeões do torneio.

Na página 69, temos a tela 10, que é a aba “Arbitragem” contendo a lista de todos os árbitros e auxiliares. A tela 11 apresenta os “Números do Campeonato” com um resumo de dados pertinentes à competição, como total de gols marcados, média de gols por rodada e média de gols por jogo. A tela 12 refere-se a aba “Clubes” e apresenta a relação de todos os clubes participantes com seus escudos.

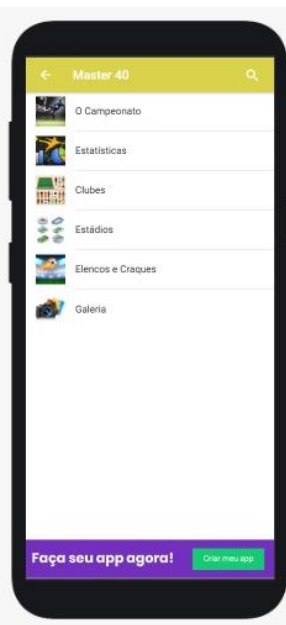
Na página 70, continuando na aba “Clubes”, ao clicar em um escudo da tela 12, abrirá a tela 13, contendo o escudo do clube e um histórico com dados da agremiação, tais como, História, Fundação, Local, Contatos, Mídias Sociais. A tela 14 apresenta o submenu da aba “Estádios”, mostrando uma lista com todos os estádios onde os clubes mandam seus jogos, contendo, em cada item, o endereço completo

do estádio e um link que direciona para um mapa que pode ser visualizado na tela 15.

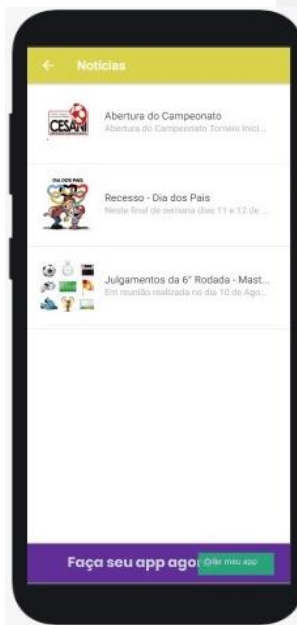
Na página 71, temos a tela 16 que se refere a aba “Elencos e Craques”. Nesta tela, aparecem todos os clubes participantes. Ao clicar em um dos clubes, abrirá a tela 17, a qual apresentará uma foto do elenco e, abaixo, todos os atletas, comissão técnica e dirigentes desta equipe. A tela 18 é a última e apresenta o conteúdo da aba “Galeria”, com imagens que ilustram o andamento desta edição do torneio.



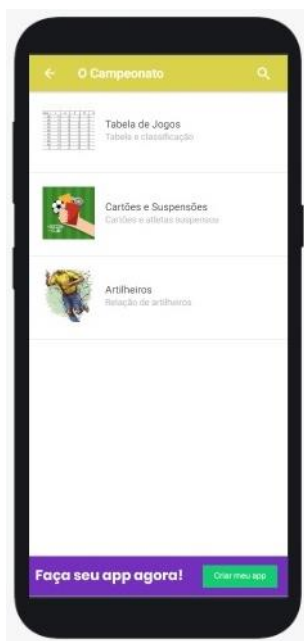
Tela 1



Tela 2



Tela 3



Tela 4



Tela 5



Tela 6



Tela 7



Tela 8



Tela 9



Tela 10



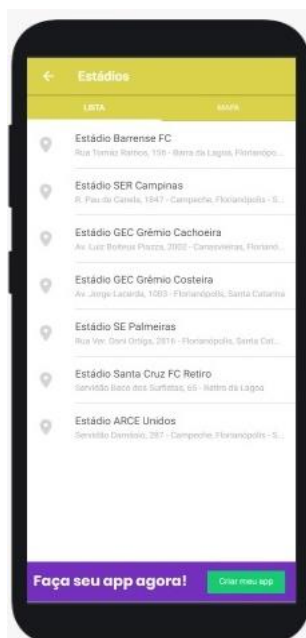
Tela 11



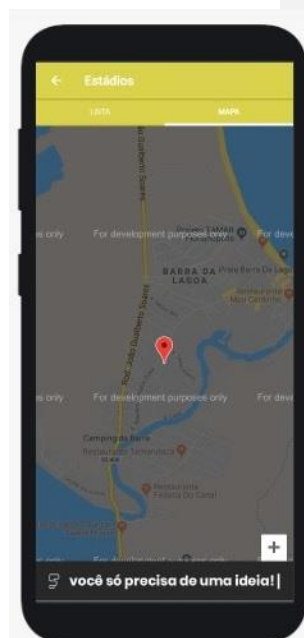
Tela 12



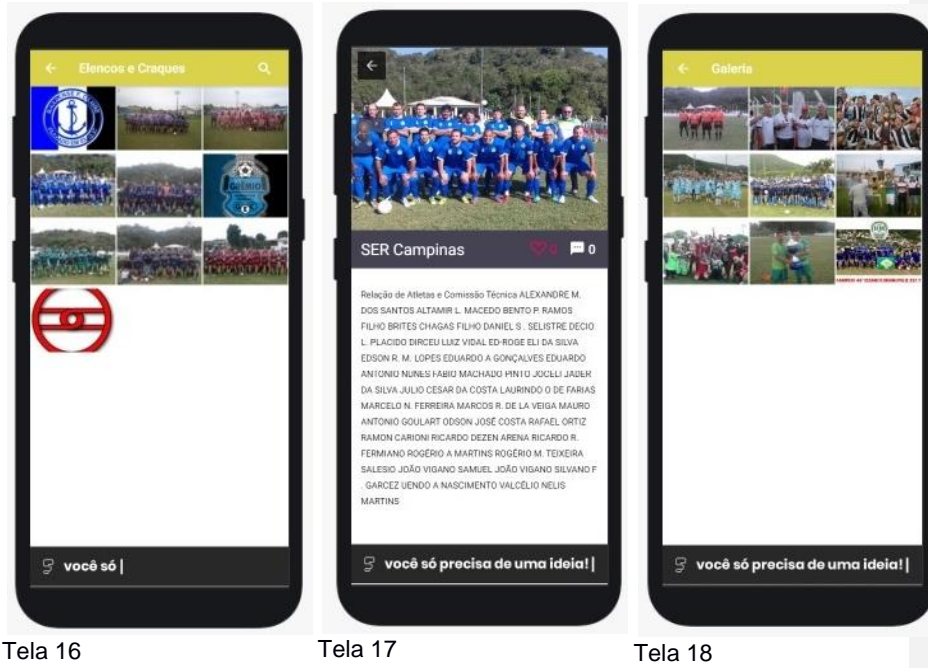
Tela 13



Tela 14



Tela 15



Tela 16

Tela 17

Tela 18

Figura 10: Telas do aplicativo.

5. CONCLUSÃO

Esse trabalho teve como objetivo o desenvolvimento de um aplicativo web para automatizar os processos para o gerenciamento das informações relacionadas às competições entre agremiações do futebol amador, apresentando os requisitos necessários para tal, bem como os diagramas de caso de uso, entidade-relacionamento e classes.

O sistema foi colocado no ar, com as principais funcionalidades implementadas, por tratar-se de uma versão inicial, estando disponível para *download*, visualização e utilização através da plataforma Fábrica de Aplicativos com algumas limitações. Algumas funcionalidades estão implementadas de forma mais simples, visando a validação e verificação dos requisitos.

Através da pesquisa bibliográfica, também foi possível um maior aprofundamento nas tecnologias utilizadas. Pincelamos uma breve introdução na história do futebol em nosso país e, com uma rápida análise, pudemos comparar nosso projeto com o que está disponível em sistemas concorrentes, percebendo quais funcionalidades estão em falta no mercado.

O desenvolvimento do sistema através da plataforma Fábrica de Aplicativos possibilitou a análise da mesma, verificando-se suas limitações e vantagens como ferramenta para desenvolvimento de uma versão funcional para validação de requisitos e validação, em última instância, do próprio projeto.

Como trabalho futuro, desejamos a implementação efetiva do sistema em uma plataforma mobile, através da utilização de uma linguagem de programação multiplataforma, tal como o *Ionic 4*. Dessa forma, será possível ter um maior domínio de todas as características do aplicativo a ser desenvolvido e de suas funcionalidades.

REFERÊNCIAS BIBLIOGRÁFICAS

ÁVILA, Ana Luiza; SPÍNOLA, Rodrigo O. **Introdução à Engenharia de Requisitos**. Rev. Engenharia de Software Magazine, DevMedia, ano 1, ed. 1, p. 46-52, 2007.

CAYRES, Paulo Henrique. **Modelagem de Banco de Dados**. Rio de Janeiro: RNP/ESR, 2015.

CHEDE, Cezar Taurion. **Grid Computing: Um Novo Paradigma Computacional**. Rio de Janeiro: Brasport Livros e Multimídia Ltda., 2004.

OLSON, David L. **Managerial Issues of Enterprise Resource Planning Systems**. McGraw Hill/Irwin, New York, 2004.

COUTO, Thiago Barbosa; FILHO, Antonio Mendes da Silva. **Desenvolvimento de aplicação web usando arquitetura MVC: Explorando o framework CodeIgniter com arquitetura MVC**. Rev. Engenharia de Software Magazine, DevMedia, ano 4, ed. 48, p. 47-55, 2012.

DÍAS, Raul Vega. **Un middleware basado en componentes Java reutilizables**. Universidad de las Palmas de Gran Canaria. 2012.

FOINA, Paulo Rogério. **Tecnologia de Informação: Planejamento e Gestão**. 2. ed. São Paulo: Atlas, 2006.

GIGLIO, Sérgio Settani. **Futebol – Arte ou Futebol – Força? O Estilo Brasileiro em Jogo**. Universidade Estadual de Campinas. Campinas, 2003.

GLICK, Bryan. **7 Days: SAP Software Gives Nestle Sweet Returns**. *Computing*, Apr 12, 2001; pg. 4

GUEDES, Gilleanes T. A. **UML 2 - Uma abordagem prática**, 2a ed., São Paulo: Novatec, 2011.

GUSMÃO, Cristine; MARINHO, Francisco; MURÍLO, Sérgio. **Uso de cenários para**

especificação de requisitos de qualidade e avaliação de arquitetura. Rev. Engenharia de Software Magazine, DevMedia, ano 4, ed. 41, p. 25-30, 2011.

LAKATOS, Eva Maria. **Metodologia científica.** 5.ed. São Paulo, Atlas, 2008.

Lance. **Brasil perde número de clubes e 90% dos registrados na CBF jogam pouco.** Disponível em <<https://www.lance.com.br/futebol-nacional/brasil-perde-numero-clubes-dos-registrados-cbf-joga-pouco.html>>. Acesso em 19 de janeiro de 2019. 2018.

MEIRELES, Manuel. **Sistemas de Informações.** São Paulo: Arte & Ciência, 2001.

MELO, Ana Cristina. **UML – Casos de Uso: Modelando os casos de uso como contrato entre usuários e desenvolvedores.** Rev. Engenharia de Software Magazine, DevMedia, ano 1, ed. 9, p. 20-26, 2009.

OLIVEIRA, Renato Gonçalves de. **Desenvolvimento baseado em componentes.** Rev. Java Magazine, DevMedia, ano 10, ed. 110, p. 40-45, 2012.

POLLI, Vínicius Macedo. **Futchamps sistema gerenciador de competições futebolísticas.** Curitiba, 2015.

PRESSMAN, Roger. **Engenharia de Software: uma abordagem profissional, 7ª ed.,** Porto Alegre: McGraw-Hill Brasil, 2011.

PRONI, Marcelo W. **A metamorfose do futebol.** Campinas, Editora da UNICAMP, 2000.

RODRIGUES FILHO, José e Gomes, Natanael Pereira. **Tecnologia da Informação no Governo Federal.** RAP. 2004

SANTOS NETO, José M. dos. **Visão do jogo – primórdios do futebol no Brasil.**

São Paulo, Cosac & Naify, 2002.

SILVA, Adam. **Web Designer: o que faz, onde estuda e qual o salário, 2013**. Disponível em: <<http://www.adamsilva.com.br/profissoes/web-designer>>. Acesso em: 17 abril de 2019.

SILVA, Edna Lúcia da; Menezes, Eстера Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4a ed. Florianópolis: Universidade Federal de Santa Catarina - UFSC, 2005. Disponível em: <https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf>. Acesso em: 21 março 2019.

SOMMERVILLE, Ian, **Engenharia de Software**, 9a ed., São Paulo: Pearson Prentice Hall, 2011.

APÊNDICE F – Declaração de finalização de trabalho de curso

INSTITUTO FEDERAL
SANTA CATARINAMINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o(a) estudante DECTO LUCIANO PLACIDO,
matricula n° 0822100045, do Curso Superior de
TECNOLOGIA EM GESTÃO DA TECN. DA INFORM. defendeu o trabalho intitulado
MODELAGEM DE UMA APLICAÇÃO WEB PARA GERENCIAMENTO E APOIO
AO DESENVOLVIMENTO DO FUTEBOL AMADOR NO BRASIL,
o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de
Santa Catarina, Câmpus Florianópolis.

Florianópolis, 01 de AGOSTO de 2019.

Renival Damianielli
Prof. Orientador do TCC