

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**DOUGLAS HENRIQUE BATISTA**

**APLICATIVO DE COMUNIDADE VIRTUAL PARA A PRÁTICA DE  
ESPORTES VOLTADA PARA DISPOSITIVOS MÓVEIS**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PATO BRANCO  
2014**

**DOUGLAS HENRIQUE BATISTA**

**APLICATIVO DE COMUNIDADE VIRTUAL PARA A PRÁTICA DE  
ESPORTES VOLTADA PARA DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso apresentado como requisito à o Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco.

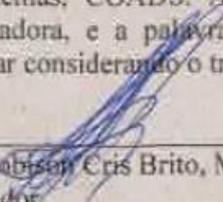
Orientador: Robison Cris Brito.

**PATO BRANCO  
2014**

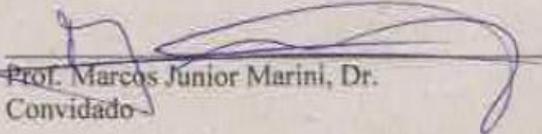
ATA Nº: 250

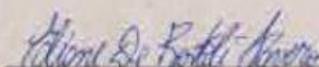
**DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO DOUGLAS HENRIQUE BATISTA.**

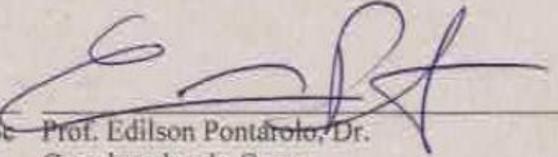
Às 15:30 hrs do dia 17 de dezembro de 2014, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Robison Cris Brito (Orientador), Eliane Maria de Bortoli Fávero (Convidada) e Marcos Junior Marini (Convidado), para avaliar o Trabalho de Diplomação do aluno Douglas Henrique Batista, matrícula 1374532, sob o título **Implementação de uma Comunidade Virtual para a Prática de Esportes Voltada para Dispositivos Móveis**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:05 hrs foi encerrada a sessão.

  
\_\_\_\_\_  
Prof. Robison Cris Brito, M.Sc.  
Orientador

  
\_\_\_\_\_  
Profa. Eliane Maria de Bortoli Fávero, M.Sc.  
Convidada

  
\_\_\_\_\_  
Prof. Marcos Junior Marini, Dr.  
Convidado

  
\_\_\_\_\_  
Profa. Eliane Maria de Bortoli Fávero, M.Sc.  
Coordenadora do Trabalho de Diplomação

  
\_\_\_\_\_  
Prof. Edilson Pontarolo, Dr.  
Coordenador do Curso

## RESUMO

BATISTA, Douglas Henrique. **Aplicativo de Comunidade Virtual para a Prática de Esportes Voltada para Dispositivos Móveis**. 2014. 57f. Trabalho de Conclusão de Curso. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2014.

Atualmente a falta de tempo tem dificultado a prática de esportes pelas pessoas, sendo mais difícil ainda reunir-se com outras pessoas interessadas no mesmo esporte. O uso de aplicativos móveis pelas pessoas das mais variadas idades, nível de formação ou classe vêm tomando cada dia mais frequente, principalmente aqueles relacionados às redes sociais. Aliado à esses dois fatos, as tecnologias de desenvolvimento para ambientes móveis como o Android também têm sido bastante utilizadas, considerando sua portabilidade com diversos tipos de aparelhos celulares, bem como sua base na tecnologia Java, o que facilita o desenvolvimento. Sendo assim, este trabalho teve por objetivo desenvolver um aplicativo baseado na tecnologia Android, denominado NoPodium, caracterizado como uma comunidade virtual voltada para a prática de esportes, o qual facilite a busca por pessoas e eventos de práticas esportivas, estreitando a relação entre as pessoas que praticam determinado esporte no qual o usuário poderá interagir com outros esportistas e convidar seus amigos relacionados para eventos esportivos criados no aplicativo, fazendo com que a comunicação e organização de eventos esportivos seja facilitada. Para isso, foram realizados estudos sobre redes sociais e comunidades virtuais, as diferentes plataformas móveis e em específico a plataforma Android, além de aplicativos semelhantes já desenvolvidos. Depois dos estudos foram extraídos os requisitos e necessidades do aplicativo, gerando-se a modelagem para o sistema. Por fim o aplicativo NoPodium foi implementado, estando contido nesse trabalho as telas e parte do código desenvolvido com suas explicações.

**Palavras-chave:** Aplicativo Móvel. Redes Sociais. Comunidades Virtuais. Prática Esportiva. Android.

## ABSTRACT

BATISTA, Douglas Henrique. **Application of Virtual Communities to the Practice of Sports Focused on Mobile Devices**. 2014. 57f. Final Course Work. Course of Technology in Analysis and Systems Development. Universidade Tecnológica Federal do Paraná, Campus Pato Branco. Pato Branco, 2014.

Currently the lack of time has hindered the practice of sports by people, and even harder to meet with others interested in the same sport. The use of mobile applications by people of various ages, education level or class are taking every day more frequent, especially those related to social networks. Allied to these two facts, the development of technologies for mobile environments such as Android have also been widely used, considering its portability with various types of mobile phones and their base on Java technology, which facilitates the development. Thus, this work aims to develop an application based on Android technology, featured as a virtual community focused on sports, which facilitates the search for people and sports activities events, enhancing the relationship between people who practice a particular sport, which the user can interact with other athletes and invite your related friends to sporting events created in the application, making better the communication and organization of sporting events. For this, studies were conducted on social networks and virtual communities, the different mobile platforms and in specific the Android platform, and similar applications already developed. After studies were extracted requirements and needs of the application, generating the modeling of the system. Finally, the application has been deployed, and is contained in this work the screens and the developed code with your explanations.

**Palavras-chave:** Mobile application. Social Networks. Virtual Communities. Practice of Sports. Android

## LISTA DE FIGURAS

Figura 1- Arquitetura em camadas Android. ....	16
Figura 2 - Telas do Aplicativo Eu Atleta.....	18
Figura 3 - Tela do Site Peladeiro.com.....	19
Figura 4 - Tela do Aplicativo Strava.....	19
Figura 5 – Tela do Aplicativo Nike Runnig. ....	20
Figura 6 - Visão geral do sistema .....	26
Figura 7 - Diagrama de casos de uso .....	30
Figura 8 - Modelo Conceitual.....	33
Figura 9 - Diagrama de entidade e relacionamento.....	33
Figura 10 - Estruturação na IDE.....	38
Figura 11 – Tela de Login.....	38
Figura 12 - Formulário de cadastro de usuário.....	40
Figura 13 - Tela Inicial.....	41
Figura 14 - Formulário de cadastro de evento.....	43
Figura 15 - Tela das Informações do Perfil .....	45
Figura 16 - Lista de Amigos .....	47
Figura 17 - Listagem de Eventos .....	50
Figura 18 - Tela de Pesquisa de Evento.....	51
Figura 19 - Tela de Exibição do Evento .....	52

## LISTA DE QUADROS

Quadro 1 - Ferramentas e tecnologias utilizadas.....	21
Quadro 2 - Iterações definidas .....	22
Quadro 3 - Requisito Funcional - Cadastrar Usuário.....	27
Quadro 4 - Requisito Funcional – Visualização do perfil.....	27
Quadro 5 - Requisito Funcional - Cadastrar Evento .....	27
Quadro 6 - Requisito Funcional – Convidar para Evento.....	28
Quadro 7 - Requisito Funcional – Visualização dos Eventos.....	28
Quadro 8 - Confirmar Participação em Eventos .....	28
Quadro 9 - Requisito Funcional – Cadastrar Locais .....	28
Quadro 10 - Requisito Funcional – Solicitar amizade.....	28
Quadro 11 - Requisito Funcional – Troca de Mensagens entre os usuários.....	29
Quadro 12 - Requisito Funcional – Localizar outros usuários.....	29
Quadro 13 - Requisito Funcional – Publicar Conteúdo .....	29
Quadro 14 - Caso de Uso Expandido – Solicitar Amizade .....	31
Quadro 15 - Caso de Uso Expandido – Criar Evento.....	31
Quadro 16 - Caso de Uso Expandido – Publicar Conteúdo .....	32
Quadro 17 - Caso de Uso Expandido – Participar do Evento .....	32
Quadro 18 - Mapeamento entidade – Amizade.....	34
Quadro 19 - Mapeamento entidade – Mensagem.....	34
Quadro 20 - Mapeamento entidade – UsuarioPublicacao .....	34
Quadro 21 - Mapeamento entidade – Usuario.....	35
Quadro 22 - Mapeamento entidade – UsuarioEsporte.....	35
Quadro 23 - Mapeamento entidade – Cidade.....	35
Quadro 24 - Mapeamento entidade – Uf .....	35
Quadro 25 - Mapeamento entidade – Esporte.....	35
Quadro 26 - Mapeamento entidade – EventoPublicacao .....	36
Quadro 27 - Mapeamento entidade – ParticipanteEvento.....	36
Quadro 28 - Mapeamento entidade – Evento .....	36
Quadro 29 - Mapeamento entidade – Local.....	37
Quadro 30 - Mapeamento entidade - Imagem.....	37

## SUMÁRIO

1 INTRODUÇÃO.....	7
1.1 CONSIDERAÇÕES INICIAIS.....	7
1.2 OBJETIVOS.....	8
1.2.1 Objetivo Geral.....	8
1.2.2 Objetivos Específicos.....	8
1.3 JUSTIFICATIVA.....	9
1.4 ESTRUTURA DO TRABALHO.....	9
2 REFERENCIAL TEÓRICO.....	10
2.1 REDES SOCIAIS.....	10
2.2 COMUNIDADES VIRTUAIS.....	12
2.3 DISPOSITIVOS MÓVEIS.....	13
2.3.1 Plataformas.....	13
2.3.2 Desenvolvimento Mobile.....	14
2.3.3 Desenvolvimento para Android.....	15
2.3.4 Exemplos de aplicações <i>Mobile</i> existentes.....	17
2.4 TRABALHOS RELACIONADOS.....	17
3 MATERIAIS E MÉTODOS.....	21
3.1 MATERIAIS.....	21
3.2 MÉTODO.....	21
4 RESULTADOS E DISCUSSÕES.....	24
4.1 DESCRIÇÃO DO SISTEMA.....	24
4.2 MODELAGEM DO SISTEMA.....	25
4.3 IMPLEMENTAÇÃO DO APLICATIVO.....	37
5 CONSIDERAÇÕES FINAIS.....	54
REFERÊNCIAS.....	55

# 1 INTRODUÇÃO

## 1.1 CONSIDERAÇÕES INICIAIS

O dia a dia das pessoas tem se tornado cada vez mais agitado, o tempo tem sido escasso e problemas de saúde tem surgido com mais frequência, pois em função dessa agitação, as pessoas deixam de cuidar de sua saúde e bem-estar, abrindo mão e seu tempo de lazer para ocupar-se mais com o trabalho. É preciso conciliar o tempo e afazeres com hábitos e ações que melhorem a saúde e bem-estar.

Uma solução para fugir dessa rotina é a prática esportiva, pois são diversos os benefícios do esporte para o ser humano, dentre eles (IHAA, 2011): o bem-estar, o fortalecimento dos ossos, a diminuição do *stress* e ansiedade, ajuda na qualidade do sono, o aumento da capacidade mental, a diminuição do colesterol, evita a depressão, diminui o risco de doenças crônicas, retarda o envelhecimento, entre outros.

Porém uma dificuldade encontrada na hora de praticar um esporte é reunir pessoas para a sua prática, além de encontrar um tempo e local apropriado. Mesmo para os esportes que são individuais, como a corrida de rua, é sempre bom ter alguém acompanhando, mas isso não é fácil, pois nem sempre as pessoas mais próximas possuem interesse no esporte, havendo a necessidade de reunir-se com pessoas muitas vezes desconhecidas, mas com isso perde-se muito tempo até reunir as pessoas.

O uso de aplicativos móveis pelas pessoas das mais variadas idades, nível de formação ou classe vêm se tornando cada dia mais frequente, principalmente aqueles relacionados às redes sociais. Uma relação entre pessoas sempre é formada por um interesse comum e uma rede social basicamente se caracteriza por essas múltiplas relações entre indivíduos ou grupos que geralmente possuem algo em comum. Tecnologias de desenvolvimento para ambientes móveis como o Android também tem sido bastante utilizadas, considerando sua portabilidade com

diversos tipos de aparelhos celulares, bem como sua base na tecnologia Java, o que facilita o desenvolvimento.

Sendo assim, o presente trabalho objetiva a implementação de um aplicativo móvel, caracterizado como uma comunidade virtual voltada para a prática de esportes, o qual facilite a busca por pessoas e práticas esportivas, estreitando a relação entre os usuários que praticam determinado esporte. Este aplicativo permite as funcionalidades de busca de eventos esportivos por cidade, busca de outros usuários, permite o compartilhamento de imagens, com funcionalidades online e offline.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

- Implementar um aplicativo móvel baseado em redes sociais para facilitar a realização de esportes coletivos.

### 1.2.2 Objetivos Específicos

- Facilitar a identificação de eventos esportivos coletivos por parte do público interessado.
- Permitir a inscrição de indivíduos em práticas de esportes coletivos.
- Permitir o acompanhamento por parte de indivíduos interessados das práticas de esportes coletivos que estiverem sendo realizadas.
- Permitir a interação entre os praticantes de esportes.
- Proporcionar aos usuários a busca por locais adequados para a prática esportiva.

### 1.3 JUSTIFICATIVA

Com o avanço da sociedade, as pessoas vivem em um mundo cada vez mais ágil e globalizado, e no meio disso falta tempo para cuidar da saúde e do bem-estar, elementos que se pode encontrar através da prática esportiva. Porém, muitas vezes falta comunicação sobre eventos e atividades que pode-se participar, portanto o aplicativo desenvolvido tem o objetivo de facilitar essa comunicação entre praticantes e admiradores do esporte, permitindo o acesso a eventos criados por usuários de forma que praticantes de um mesmo esporte possam se conhecer e praticá-lo juntos.

O aplicativo será desenvolvido para a plataforma móvel Android, justamente pensando no grande crescimento e utilização e ainda visando a facilidade de implementação na plataforma.

Esse trabalho trará uma contribuição importante para os estudos das plataformas móveis, assim como para o entendimento do funcionamento de uma rede social, esclarecendo as principais diferenças referente a uma comunidade virtual. Em linhas gerais o conhecimento obtido na etapa de desenvolvimento e estudo acerca da tecnologia *mobile* será de grande valia para o acadêmico, futuros acadêmicos e profissionais, acrescentando um conhecimento significativo acerca da programação para dispositivos móveis.

### 1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos. No Capítulo 2 é apresentado o referencial teórico apresentando os conceitos de redes sociais, comunidades virtuais, desenvolvimento para dispositivos móveis, Android, e aplicações móveis existentes. No Capítulo 3 estão os materiais e o método utilizados para a modelagem e a implementação do sistema. Os resultados da realização deste trabalho são apresentados no Capítulo 4 onde traz as telas e parte do código desenvolvido para a aplicação. Por fim está a conclusão, seguida das referências bibliográficas.

## 2 REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento necessário à proposta deste trabalho que tem como foco o desenvolvimento de um aplicativo móvel baseado no conceito de redes sociais. Como o sistema será móvel, o referencial teórico irá tratar também do funcionamento desse tipo de aplicativo, apresentando exemplos e aplicativos equivalentes ao que pretende-se desenvolver.

### 2.1 REDES SOCIAIS

Desde o início da história humana é necessidade do ser humano se relacionar uns com os outros, trocar ideias, experiências de vida, fatos do cotidiano, entre outros. Ao longo do tempo muitas invenções foram feitas para melhorar essa relação entre as pessoas (ex., a invenção das cartas, do telefone, da televisão, do celular), tudo isso para que o ser humano pudesse ter meios para se comunicar e relacionar. É com essa proposta que surgiram as redes sociais.

Uma rede é um agrupamento de indivíduos, organizações ou agências organizadas em bases não hierárquicas em torno de questões ou preocupações, as quais atuam proativamente e sistematicamente baseadas no compromisso e confiança (WHO 1998).

Uma relação entre pessoas sempre é formada por um ponto em comum, os indivíduos são ligados por uma causa, em que dois ou mais estão interessados, e uma rede social basicamente se caracteriza por essas muitas relações entre indivíduos ou grupos que geralmente possuem algo em comum. Por isso essas redes estão em um crescimento contínuo, pois a cada dia são criadas novas relações, aumentando o número de conexões entre os indivíduos.

Essas relações são definidas por Minarelli (2001) em três níveis:

- Grupo Primário: pessoas próximas que possuem um forte envolvimento emocional;
- Grupo Secundário: relações formais e menos íntimas;

- Grupo de Referência: pessoas que ajudam os indivíduos a moldar nossas características pessoais, pois seus valores e suas expectativas ordenam nosso padrão de comportamento.

Atualmente, através da Internet é possível estabelecer essas relações virtualmente, em um mundo virtual sem fronteiras, porém isso não descaracteriza as relações, que não deixam de ser reais. Esse ambiente *online* permite o agrupamento de pessoas que possuem interesses em comum, formando efetivamente as redes sociais *online*.

O crescimento dessas redes sócias se dá pelo fato da era tecnológica em que se vive, considerando que atualmente a maioria das pessoas no globo tem acesso à Internet, fazendo com que mais pessoas se insiram no contexto de rede social e formem suas relações.

De acordo com o estudo americano *Generation M2: Media in the Lives of 8-to18-Years-Olds* feita em 2010, em média 40% dos jovens entre 8 e 18 anos entram em alguma rede social online diariamente e dedicam em média uma hora por dia (PEREIRA et. al., 2011).

Muitas redes sociais estão disponíveis na Internet, porém as mais utilizadas, no Brasil, segundo pesquisa realizada pela Confederação do Transporte no Brasil em 2013 (PESQUISA, 2014) são o Facebook, Youtube e o Twitter, nessa ordem. Ainda estima-se que há em média 65 milhões de usuários brasileiros no Facebook.

E como em qualquer segmento social existente a rede social também possui sua própria linguagem, palavras e expressões que a caracterizam, conforme diz Cassiano (2011): “Postar, twittar, curtir, compartilhar, seguir! Essas são algumas das linguagens criadas e ações realizadas nas redes sociais existentes”. As redes sociais já fazem parte e ajudam no desenvolvimento da cultura dos brasileiros, como sendo um dos principais meios de comunicação no país.

Segundo Recuero (2009), as redes sociais são compostas por dois elementos: atores (grupos, pessoas, organizações ou comunidades) e as relações que eles desenvolvem entre si (de trabalho, de estudo, de amizade, etc.). As redes sociais são diferentes entre si, porém possuem algumas características semelhantes, dentre elas o fato de todas serem compostas por perfis criados pelos usuários e ser possível visualizar a rede de amigos desse usuário.

## 2.2 COMUNIDADES VIRTUAIS

Dentro do contexto de redes sociais encontram-se as comunidades virtuais, as quais de acordo com Cummings et. al. (2002, p.103),

...são definidas como grupos de pessoas distribuídas e que contribuem voluntariamente à informação em um espaço social comum, suportado por uma comunicação em rede. Tais comunidades são organizadas, frequentemente, em torno de um interesse comum ou de um atributo demográfico.

Uma definição abrangente do termo comunidade virtual é apresentada por Preece (2000). Para o autor, comunidade virtual é “qualquer espaço social virtual onde as pessoas se reúnem para obter e oferecer informações ou apoio, para aprender ou para encontrar companhia”, independente de localização geográfica (local, nacional ou internacional) e tamanho (pequeno ou grande).

De acordo com estudo realizado, Haetinger (2005) constata que a formação e a manutenção das comunidades virtuais ocorrem principalmente em função da motivação dos participantes, da disponibilidade de cada um em colaborar e cooperar, do intercâmbio de suas criações e descobertas e do compromisso estabelecido entre eles. Para a formação, também são relevantes as afinidades, curiosidades ou os objetivos comuns e o desejo de compartilhar competências, habilidades e valores.

Dessa forma, pode-se dizer que uma comunidade virtual se caracteriza como um agrupamento de pessoas que possuem algum tipo de relação.

Muitos são os exemplos de comunidades virtuais que podem ser formadas pelo agrupamento de membros de organizações dos mais diversos setores de atividade, em espaços online: intranet corporativa, contextos de trabalho cooperativo, sistemas de educação à distância, entre outros. Por essa razão, se fez necessário classificar as comunidades virtuais de acordo com a sua finalidade, sendo um esforço de classificação apresentado a seguir (KOLLOCK, 1997):

- **de aprendizado:** como a comunidade dos estudantes e professores de um curso de graduação;
- **de prática:** como a formada pelos colaboradores de uma organização; ou

- **de lazer:** como a dos amantes da pesca, colecionadores de selos, etc.

Comunidades virtuais se originam das redes sociais, como sendo, grupos de interesse criados a partir da grande rede social, uma vez que o espaço *online* é público onde cada um pode expressar-se livremente, muitas vezes são encontrados conteúdos redundantes, valores nulos, não essenciais e que não possuem significância para todos, surgindo assim a poluição de dados (GONÇALVES, 2008).

Essa poluição deu origem às comunidades virtuais, pois nessas comunidades, (que estão inseridas dentro das redes sociais através de grupos e páginas específicas), limitam o conteúdo a ser expresso pelos participantes a um só assunto, sendo assim evita que o participante se depare com conteúdos que não lhe interessem, proporcionando uma melhor utilização e significância da rede social, fortalecendo também a relação entre os membros do grupo.

## 2.3 DISPOSITIVOS MÓVEIS

Atualmente é notório o crescimento de *smartphones* no mercado, mais e mais pessoas vêm aderindo ao uso desse dispositivo, portanto, trata-se de um item que já foi inserido na rotina das pessoas, segundo a analista Mary Meeker, em um estudo foi concluído que no Brasil existiam, no ano de 2013, 70 milhões de *smartphones* (PESQUISA, 2013).

### 2.3.1 Plataformas

Pode-se concluir que esse mercado tende a aumentar a cada dia, assim atualmente existem diversas plataformas no uso dos *smartphones*, sendo essas plataformas o meio que permite com que o usuário do dispositivo possa interagir com os aplicativos e itens de hardware nele existentes. As principais plataformas com base no estudo de MASI (2011) são:

- **Android:** Plataforma desenvolvida pela Google em parceria com outras empresas de software e é de código aberto. Segundo a opinião de Gargenta

(2011), o Android está revolucionando o espaço das tecnologias móveis. Pela primeira vez, há uma plataforma verdadeiramente aberta que separa o hardware do software que está rodando no aparelho. Foi criada pensando-se na integração com os serviços oferecidos pela Google, tem custo reduzido dos aparelhos que a utilizam em relação à maioria das plataformas.

- **IOs:** É uma plataforma desenvolvida pela Apple em 2007, só funciona em aparelhos produzidos pela própria Apple, é muito presente no mercado. Por ser exclusivo, possui alto desempenho do hardware e do software. Segundo Masi (2011), é a plataforma que mais cresce no ponto de vista de desenvolvedores e isso se deve aos consumidores que em sua maioria tem um poder aquisitivo maior, efetuando mais compras dos aplicativos do que nas demais plataformas.

- **BlackBerry:** Lançado inicialmente para Pagers, em 2002 passou a ser utilizado em *smartphones*. É uma plataforma de grande presença no mercado corporativo devido aos padrões de segurança que possui, foi desenvolvida pela empresa *Research in Motion* (RIM).

- **Windows Phone:** Desenvolvida pela Microsoft, é voltada para o mercado de consumidores, ela é diferente das demais pelo seu design Metro. Atualmente empresas como HTC e Nokia a utilizam.

- **Symbian OS:** Segundo FELIZZOLA (2012) o Symbian é exclusivo da Nokia, surgiu a partir de outro sistema antigo para agenda dos anos 90, roda somente em aparelhos Nokia com processador AMD. É uma plataforma bastante fragmentada, pois existem várias ramificações em produtos da Nokia. A Nokia já não lança mais dispositivos com essa plataforma, porém, prestará atualizações até o ano de 2016.

### 2.3.2 Desenvolvimento Mobile

Para desenvolvimento mobile, Ambros (2013) apresenta três tipos diferentes de aplicações: aplicações nativas, aplicações Web e aplicações híbridas

- Aplicações Nativas: São desenvolvidas a partir da linguagem da plataforma, sendo assim tem a possibilidade de acessar os recursos de hardware presentes no

aparelho, para serem utilizados eles precisam ser colocados nas lojas de aplicativos da plataforma, como *Play Store* do Android, sendo assim, devem ser instaladas no aparelho para serem utilizadas. Uma desvantagem grande desse método é o fato de ter que desenvolver cada aplicação dependendo da plataforma em que será utilizada.

- Aplicações Web: São desenvolvidas e disponibilizadas na rede Web, para acessá-la basta acessar o *link* através do celular, esse método é bastante utilizado pois não requer instalação do aplicativo no dispositivo e todas as plataformas podem acessá-la. Uma desvantagem é o fato de que não se pode acessar os itens de hardware específicos, como a câmera, por exemplo.

- Aplicações Híbridas: É a mistura da nativa com a Web, é um aplicativo que após desenvolvido deve ser instalado no dispositivo porém, a partir do código desenvolvido pode-se exportá-lo para a plataforma desejada, sendo assim não é necessário desenvolver novamente a aplicação para cada plataforma. Mas deve-se como na aplicação nativa, disponibilizar nas lojas de aplicativos.

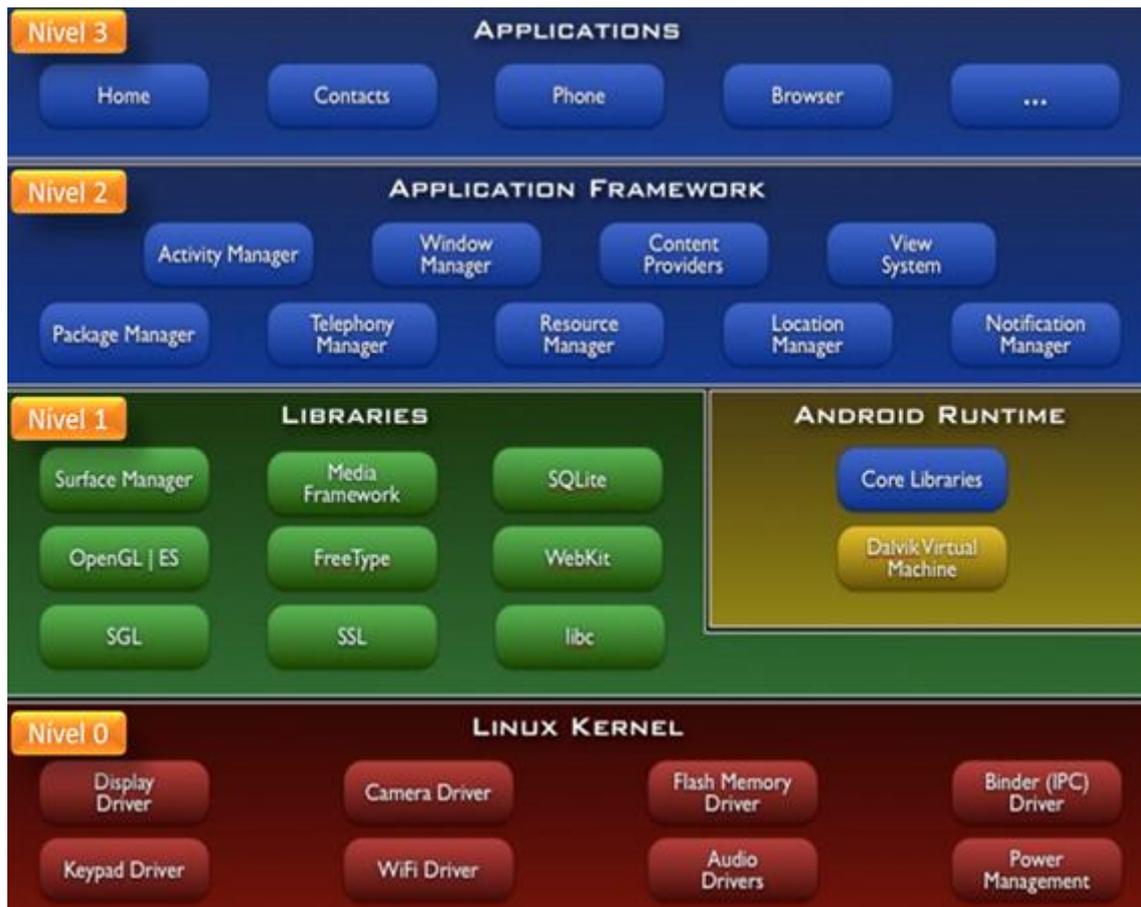
### 2.3.3 Desenvolvimento para Android

Esse trabalho se utilizará da plataforma Android para seu desenvolvimento, sendo assim, abaixo serão apresentados mais detalhes sobre o funcionamento de uma aplicação dentro dessa plataforma.

O Android é uma plataforma de código aberto, criado a partir da licença Apache License versão 2.0, isso torna essa plataforma utilizável em qualquer aparelho *smartphone*, essa licença favoreceu muito o seu crescimento.

Para o desenvolvimento de aplicações para essa plataforma o Google disponibilizou um kit de ferramentas chamado Android SDK, com ele qualquer programador está apto a desenvolver para Android.

Segundo o estudo de Ogliari e Brito (2014, p.3) a infraestrutura de software do Android “consiste de uma pilha que engloba um sistema operacional Linux, conjunto de bibliotecas, API chamada *Android Runtime*, aplicações preexistentes no Android e aplicações diversas.” Isso pode ser visualizado na figura a seguir (Figura 1):



**Figura 1- Arquitetura em camadas Android.**  
**Fonte: GADELHA (2011).**

- Nível 0: A comunicação entre o software e o hardware é feita pelo sistema operacional, que também controla o acesso à rede, gerencia a memória, gerencia os processos, entre outros. O *kernel* Linux é de versão 2.6 e 3.0 para versões acima da API Level 11.
- Nível 1: São bibliotecas C/C++, vários componentes se utilizam delas, o acesso de dá através do *framework* da aplicação criada. O *runtime* se concentra nas bibliotecas básicas do Android.
- Nível 2: Ogliari e Brito (2014, p.4) trazem o nível *Application Framework*

como:

... são bibliotecas de mais alto nível, responsáveis por funções específicas no aparelho, como, por exemplo, o gerenciamento de janelas, recursos dos dispositivos móveis, notificações etc. Essas são as bibliotecas costumeiramente acessadas pelos desenvolvedores Android para interagir com o dispositivo móvel.

- Nível 3: No nível 3 estão os aplicativos nativos que todo *smartphone* e celular possui, que podem se comunicar entre eles e com as demais aplicações.

#### 2.3.4 Exemplos de aplicações *Mobile* existentes

O mercado de aplicações para celular é vasto, tanto em tipo como em funcionalidade. Entre os 10 aplicativos gratuitos mais instalados, segundo Play Store (2014), estão:

- *Facebook*: Aplicativo móvel da rede social facebook, tem todas as funções que o acesso via web oferece, é feito especialmente para móvel, com funcionalidades específicas para a versão para celular. (FACEBOOK, 2014)
- *What's App*: Aplicativo exclusivo para *smartphones*, permite realizar a troca de mensagens de áudio, vídeo, texto e imagens, é amplamente utilizado. (WHAT'S APP, 2014)
- *Instagram*: Aplicativo de compartilhamento de imagens, cada pessoa tem uma conta e a partir do aplicativo é possível postar as fotos em sua conta, um diferencial do aplicativo é a possibilidade de se adicionar diferentes tipos de filtros e efeitos em uma imagem. (INSTAGRAM, 2014)
- *Perguntados*: É um aplicativo do tipo Quiz, de perguntas e respostas sobre seis categorias: Arte, Ciência, Esporte, Entretenimento, Geográfico e História. É possível compartilhar os pontos e competir com outras pessoas. (PERGUNTADOS, 2014)

#### 2.4 TRABALHOS RELACIONADOS

O mercado apresenta alguns aplicativos já desenvolvidos, pensados para integrar a prática esportiva e a tecnologia, alguns desses são:

- *Eu Atleta*: Aplicativo desenvolvido pela Globo, focado em corridas de rua, que permite a marcação de locais para a prática do esporte, possui aplicativo

*mobile* e *web*. Como diferencial possui o mapa de calor, que marca os locais mais utilizados para corrida, e a parte *web* que contém muitas informações e dicas de saúde (EU ATLETA, 2014).

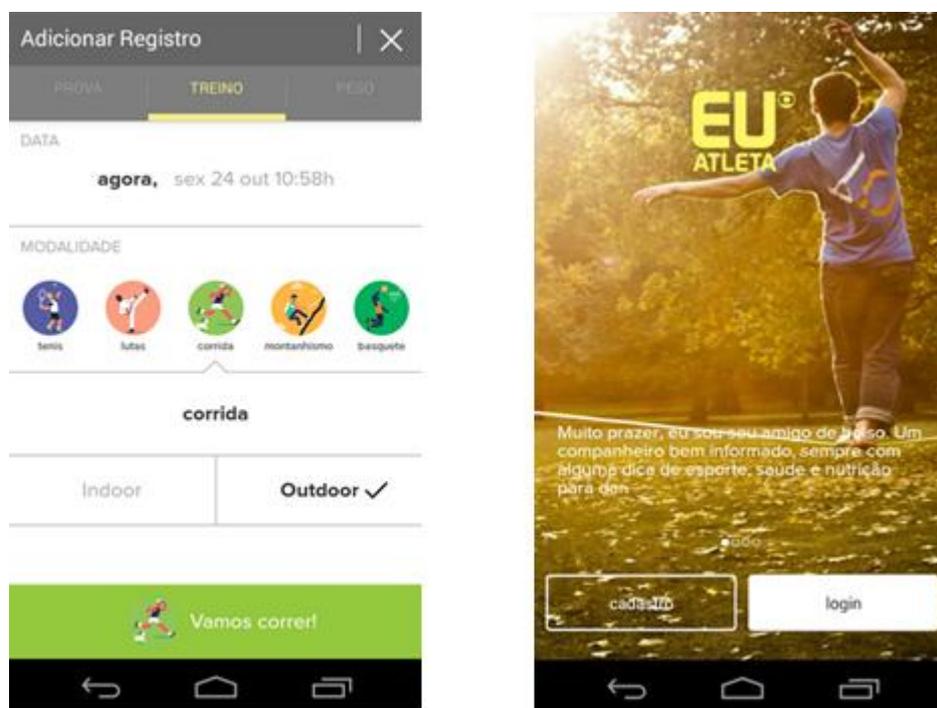


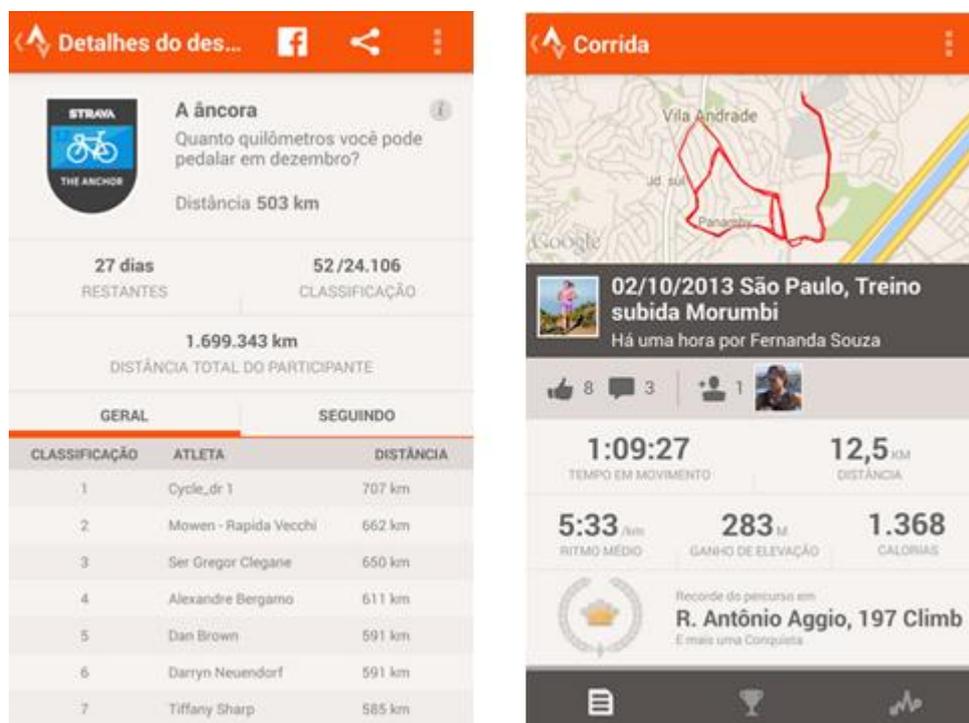
Figura 2 - Telas do Aplicativo Eu Atleta.  
Fonte: Eu Atleta (2014).

- Peladeiro.com – Site Web que organiza partidas de futebol, com locais e *rankings*, é um gerenciador bastante utilizado, com mais de 700 mil usuários. Cada usuário se cadastra e a partir disso pode participar de partidas organizadas e criar uma partida de futebol. Apresenta *rankings* e comparações de melhores equipes e atletas (PELADEIRO, 2014).



**Figura 3 - Tela do Site Peladeiro.com.**  
**Fonte: PELAIDEIRO(2014).**

- Strava: Aplicativo voltado para ciclistas e corredores, possui interface Web e dois aplicativos, um para ciclismo e um para corridas, ele mostra locais, permite adicionar amigos e marca o trajeto percorrido pelo GPS, possui também sistema de desafios (STRAVA, 2014).



**Figura 4 - Tela do Aplicativo Strava.**  
**Fonte: STRAVA (2014)**

- Nike Running: Aplicativo e interface Web voltado para corrida de rua desenvolvido pela Nike, que permite fazer amigos e marcar o trajeto

percorrido via GPS. Possui um histórico de corridas e acumula quanto já foi percorrido pelo usuário (NIKE, 2014).



Figura 5 – Tela do Aplicativo Nike Runnig.  
Fonte: NIKE (2014).

### 3 MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e o método utilizados para a realização deste trabalho. Os materiais estão relacionados às tecnologias e ferramentas utilizadas e o método apresenta a sequência das principais atividades realizadas.

#### 3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e as tecnologias que foram utilizadas para modelar e implementar o sistema.

Ferramenta / Tecnologia	Versão	Referência	Finalidade
Visual Paradigm Community	11.1	<a href="http://www.visual-paradigm.com/">http://www.visual-paradigm.com/</a>	Documentação da modelagem baseada na UML.
Android Development Toolkit	23.0.2	<a href="http://developer.android.com/sdk/eclipse-adt.html">http://developer.android.com/sdk/eclipse-adt.html</a>	Plugin Android para Eclipse
Java	7	<a href="http://www.java.com/pt_BR/">http://www.java.com/pt_BR/</a>	Linguagem de programação.
Android SDK	23.0.2	<a href="http://developer.android.com/sdk/index.html">http://developer.android.com/sdk/index.html</a>	Ferramenta de Desenvolvimento Android
Eclipse Plataforma	4.2.1	<a href="http://www.eclipse.org/plataform">http://www.eclipse.org/plataform</a>	Ambiente de desenvolvimento.
SQLite	3.8.6	<a href="http://www.sqlite.org/">http://www.sqlite.org/</a>	Banco de dados.
MySQL	5.5.15	<a href="http://www.mysql.com">http://www.mysql.com</a>	Servidor Banco de Dados

**Quadro 1 - Ferramentas e tecnologias utilizadas**

Fonte: Elaborado pelo autor

#### 3.2 MÉTODO

A modelagem e a implementação do aplicativo *mobile* para interação na prática de esportes têm como base o modelo sequencial linear descrito em Pressman (2008), complementado pelo processo unificado (WAZLAWICK et al., 2006). O modelo sequencial foi utilizado para denominar os processos (ou fases). O processo unificado auxiliou na definição dos ciclos iterativos de modelagem e implementação.

O Quadro 2 apresenta os processos (fluxos de trabalho) e as iterações desenvolvidas.

Iterações \ Processos	1ª iteração	2ª iteração	3ª iteração
Requisitos	Definição dos requisitos básicos.	Complemento dos requisitos.	Revisão dos requisitos.
Análise e projeto	Modelagem de casos de uso, do modelo conceitual e do diagrama entidade e relacionamento preliminar	Expansão dos principais casos de uso. Ajustes e atributos e campos Revisão de diagramas de classes e entidade e relacionamento.	Modelagem do restante das funcionalidades do sistema.
Implementação	Estudo das diferentes versões das plataformas Android, visando identificar a melhor forma para implementação do sistema.	Da interface padrão dos cadastros.	
Testes		Das telas e cadastros desenvolvidas pelo usuário.	

**Quadro 2 - Iterações definidas**

Fonte: Elaborado pelo autor

A seguir estão descritas as etapas (identificadas como processos no Quadro 2) definidas para o desenvolvimento do aplicativo e as principais atividades de cada uma dessas etapas.

#### **a) Levantamento de requisitos**

O levantamento dos requisitos iniciou com um estudo sobre os conceitos acerca de redes sociais, comunidades virtuais, dispositivos móveis e suas características, além de trabalhos relacionados às redes sociais e comunidades virtuais, o que trouxe uma visão geral sobre o tema e ofereceu o embasamento necessário para a proposta e desenvolvimento do sistema. Dessa visão foram extraídos os requisitos principais e outros foram identificados posteriormente. Os requisitos foram organizados em funcionais e não funcionais.

Os requisitos foram complementados à medida que as iterações ocorriam. Várias alterações ocorreram como forma de definir mais adequadamente as funcionalidades do sistema.

#### **b) Análise e projeto do sistema**

Com base nos requisitos foram definidos os casos de uso do sistema. Esses casos de uso foram documentados gerando informações para a definição do modelo conceitual e modelo de entidades e relacionamento. Posteriormente, os cadastros a

serem implementados foram definidos e elaborados/complementados os diagramas de classes e de entidades e relacionamentos, sendo definidas as tabelas e dos seus campos, tipo e tamanho de dados. Nessa fase também foi realizada a prototipação das telas para o sistema fazendo uso desse recurso na ferramenta Visual Paradigm.

### **c) Implementação**

A implementação foi realizada utilizando a ferramenta *Eclipse Plataform*. Como trabalho de estágio foram implementadas as funcionalidades de cadastros simples visando exemplificar o uso da tecnologia *Android*. Em termos de interface, o objetivo da segunda iteração foi experimentar e testar a melhor forma de compor os formulários e de disponibilizar as informações na tela.

### **d) Testes**

Os testes foram informais e realizados à medida que as iterações ocorriam. Esses testes incluíram verificação do código (teste de unidade), da forma de interação com o aplicativo (interface) e da integração de suas funcionalidades.

## 4 RESULTADOS E DISCUSSÕES

Este capítulo apresenta o resultado deste trabalho que tratou da modelagem de um sistema para uma comunidade virtual de pratica de esportes voltada a aparelhos *mobile*.

### 4.1 DESCRIÇÃO DO SISTEMA

O aplicativo será implementado de forma nativa, para permitir que o usuário tenha acesso à parte dos dados mesmo sem ter uma conexão com a Internet, a forma nativa também foi optada levando-se em conta o conhecimento que já se possui para implementação desse tipo de aplicação.

O aplicativo NoPodium tem o objetivo de possibilitar a interação entre pessoas que tem o esporte como lazer, assim uma pessoa poderá ingressar nessa comunidade virtual para se relacionar com outros praticantes de esportes e ter conhecimento sobre os eventos esportivos que lhe interessam, tanto para participar como para assistir.

O aplicativo deve permitir o ingresso na rede após o usuário cadastrar seus dados como: Nome, Data de Nascimento, E-mail, Senha para Acesso, Telefone, Cidade/UF e esportes que pratica a partir de uma lista pré-definida. Ao se cadastrar, o usuário tem acesso ao perfil onde ele pode inserir uma imagem de perfil e postar textos, *links* e imagens.

O cadastro de eventos esportivos também poderá ser feito pelo usuário, onde deve ser informado o título, local, data e horário, esporte, número máximo de participantes, contato para informações, observações, entre outros. Além disso, o aplicativo permitirá informar também se o evento é público ou privado. Após o cadastro, poderá ser feito o convite aos amigos para participarem do evento.

A visualização dos eventos também é permitida, organizados por cidade e esportes, o usuário poderá ver informações desses eventos esportivos como lista de

participantes, endereço, data e horário, isso se o evento estiver definido como público.

O aplicativo NoPodium também permitirá realizar pesquisas por pessoas que estão na rede, então o permite o pedido de amizade para o participante.

Também contará com um cadastro de locais, permitindo ao usuário cadastrar e visualizar os locais no momento da inclusão do evento.

O aplicativo terá a funcionalidade de troca de mensagens, onde um usuário poderá enviar uma mensagem para outro usuário que esteja em sua rede de amigos, essas mensagens só poderão ser visualizadas pelos usuários da conversa.

Possuirá duas formas de funcionamento, elas mudam conforme o usuário está ou não conectado à Internet.

Quando conectado à Internet o usuário tem acesso a todas as funções do aplicativo, tais como: visualizar os amigos e seus respectivos textos postados, visualizar os eventos criados em todo o Brasil, ver as pessoas que utilizam o aplicativo, postar texto, interagir com os eventos, etc.

Sem conexão à Internet o usuário, através de uma lista, pode visualizar os eventos que estão para ocorrer em seu município, porém não pode confirmar sua participação, nem fazer comentários ou interagir com o mesmo.

## 4.2 MODELAGEM DO SISTEMA

O diagrama de atividades da Figura 6 apresenta uma visão geral do sistema proposto, destacando suas funcionalidades.

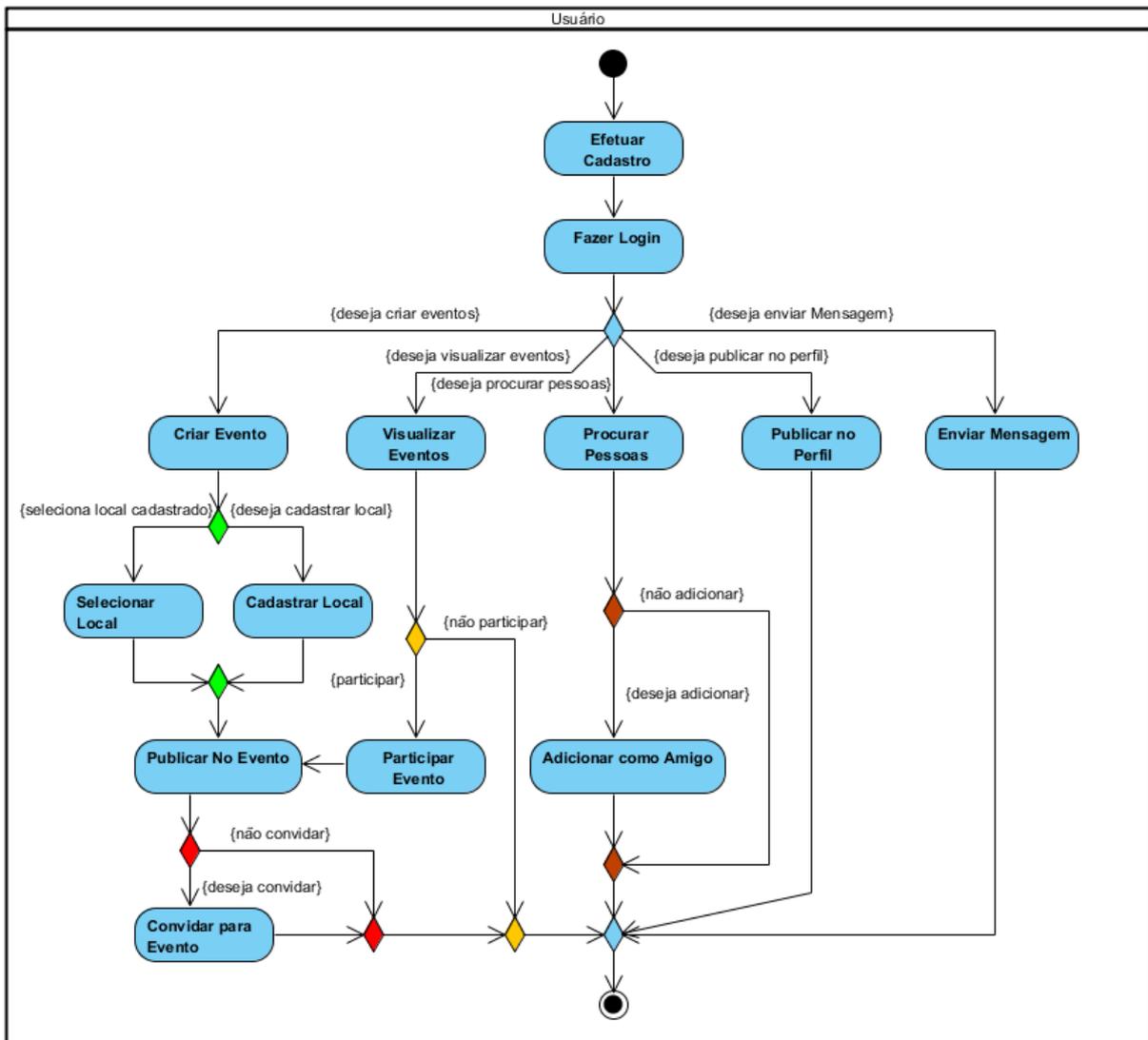


Figura 6 - Visão geral do sistema  
 Fonte: Elaborado pelo autor

Com base na identificação dos requisitos de usuário e visão geral do sistema apresentada anteriormente, foram identificados os requisitos funcionais e não funcionais para o sistema proposto, conforme os Quadros 3 a 13.

<b>F1 Cadastrar Usuário</b>		<b>Oculto ( )</b>		
<b>Descrição:</b> O aplicativo deve permitir o cadastro de um novo usuário através dos dados informados. Além disso, deve ser permitida a alteração, exclusão e visualização desse cadastro.				
<b>Requisitos Não-Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>

<b>NF 1.1 Inclusão de foto</b>	O formulário de cadastro do usuário deve disponibilizar em sua interface a possibilidade de inserir uma foto atual a partir da galeria de fotos do celular ou tirar uma foto na hora do cadastro através da câmera.	Interface	( )	( x )
<b>NF 1.2 Seleção dos esportes que Pratica.</b>	No cadastro o aplicativo deve fornecer uma lista para o usuário selecionar os esportes que pratica informando também se preferir o ano de início da pratica.	Interface	( x )	( )

**Quadro 3 - Requisito Funcional - Cadastrar Usuário**  
**Fonte: Elaborado pelo autor**

<b>F2 Visualização do Perfil</b>		<b>Oculto ( )</b>		
<b>Descrição:</b> O aplicativo deve disponibilizar a visualização dos perfis e postagens entre amigos relacionados.				
<b>Requisitos Não-Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>

**Quadro 4 - Requisito Funcional – Visualização do perfil**  
**Fonte: Elaborado pelo autor**

<b>F3 Criar Evento</b>		<b>Oculto ( )</b>		
<b>Descrição:</b> O aplicativo deve permitir que o usuário crie um evento esportivo, o qual poderá ser público ou privado, e convide os amigos de sua rede para o evento, permitindo a alteração e exclusão do evento.				
<b>Requisitos Não-Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
<b>NF 3.1 Seleção do Esporte</b>	Ao cadastrar um evento o aplicativo deve disponibilizar em sua interface uma lista para o usuário selecionar a modalidade esportiva do evento.	Interface	( )	( x )
<b>NF 3.2 Selecionar Local</b>	No cadastro de evento deve-se permitir ao usuário selecionar um local onde o evento será realizado.	Interface	( )	( x )

**Quadro 5 - Requisito Funcional - Cadastrar Evento**  
**Fonte: Elaborado pelo autor**

<b>F4 Convidar para Evento</b>		<b>Oculto ( )</b>		
<b>Descrição:</b> O usuário participante do evento poderá convidar outros usuários para o evento				

que estiverem relacionados a ele na rede social.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente

**Quadro 6 - Requisito Funcional – Convidar para Evento**  
 Fonte: Elaborado pelo autor

F5 Visualização dos Eventos		Oculto ( )		
<b>Descrição:</b> Um evento pode ser visualizado por todos os usuários da rede social caso esse esteja com a privacidade pública.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 5.1 Visualizar eventos disponíveis em sua cidade sem conexão com a internet	O aplicativo deve permitir ao usuário visualizar eventos em sua cidade mesmo sem possuir conexão com a Internet.	Regra de Negócio	( x )	( )

**Quadro 7 - Requisito Funcional – Visualização dos Eventos**  
 Fonte: Elaborado pelo autor

F6 Confirmar Participação em Eventos		Oculto ( )		
<b>Descrição:</b> O aplicativo deve permitir a visualização e confirmação em um evento que o usuário foi convidado.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente

**Quadro 8 - Confirmar Participação em Eventos**  
 Fonte: Elaborado pelo autor

F7 Cadastrar Locais		Oculto ( )		
<b>Descrição:</b> O aplicativo deve permitir ao usuário cadastrar e selecionar locais em um cadastro de evento através do endereço e título do local.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente

**Quadro 9 - Requisito Funcional – Cadastrar Locais**  
 Fonte: Elaborado pelo autor

F8 Solicitar Amizade		Oculto ( )		
<b>Descrição:</b> O aplicativo deve permitir ao usuário localizar e fazer um pedido de amizade com outros usuários na rede, permitindo que esses relacionados troquem mensagens entre si.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente

**Quadro 10 - Requisito Funcional – Solicitar amizade**  
 Fonte: Elaborado pelo autor

F9 Trocar mensagens entre os usuários		Oculto ( )		
<b>Descrição:</b> Os usuários poderão trocar mensagens entre si no qual somente eles poderão ter acesso as mesmas.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF 9.1 Notificação das Mensagens	O aplicativo deve notificar o usuário quando houver uma	Interface	( x )	( )

	nova mensagem em sua caixa de entrada. Com um ícone na barra de tarefas do dispositivo.			
--	---	--	--	--

**Quadro 11 - Requisito Funcional – Troca de Mensagens entre os usuários**

Fonte: Elaborado pelo autor

<b>F10 Localizar outros Usuários</b>		<b>Oculto ( )</b>		
<b>Descrição:</b> O aplicativo apresentará uma tela permitindo que o usuário localize outros usuários, filtrando a busca por cidade.				
<b>Requisitos Não-Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>

**Quadro 12 - Requisito Funcional – Localizar outros usuários**

Fonte: Elaborado pelo autor

<b>F11 Publicar Conteúdo</b>		<b>Oculto ( )</b>		
<b>Descrição:</b> O aplicativo deve permitir ao usuário publicar em seu perfil e em eventos que participará ou participou, possibilitando a exclusão dos mesmos.				
<b>Requisitos Não-Funcionais</b>				
<b>Nome</b>	<b>Restrição</b>	<b>Categoria</b>	<b>Desejável</b>	<b>Permanente</b>
<b>NF11.1 Conteúdo de Publicação</b>	O aplicativo deve permitir ao usuário publicar texto, imagens ou links em seu perfil e em eventos no qual é participante.	Regra de Negócio	( )	( x )

**Quadro 13 - Requisito Funcional – Publicar Conteúdo**

Fonte: Elaborado pelo autor

A Figura 7 apresenta o modelo de casos de uso para o sistema proposto, o qual apresenta as funcionalidades completas, ou seja, aquelas que além de possuir algum tipo de interação com um ator, podem ser realizadas de forma isolada.

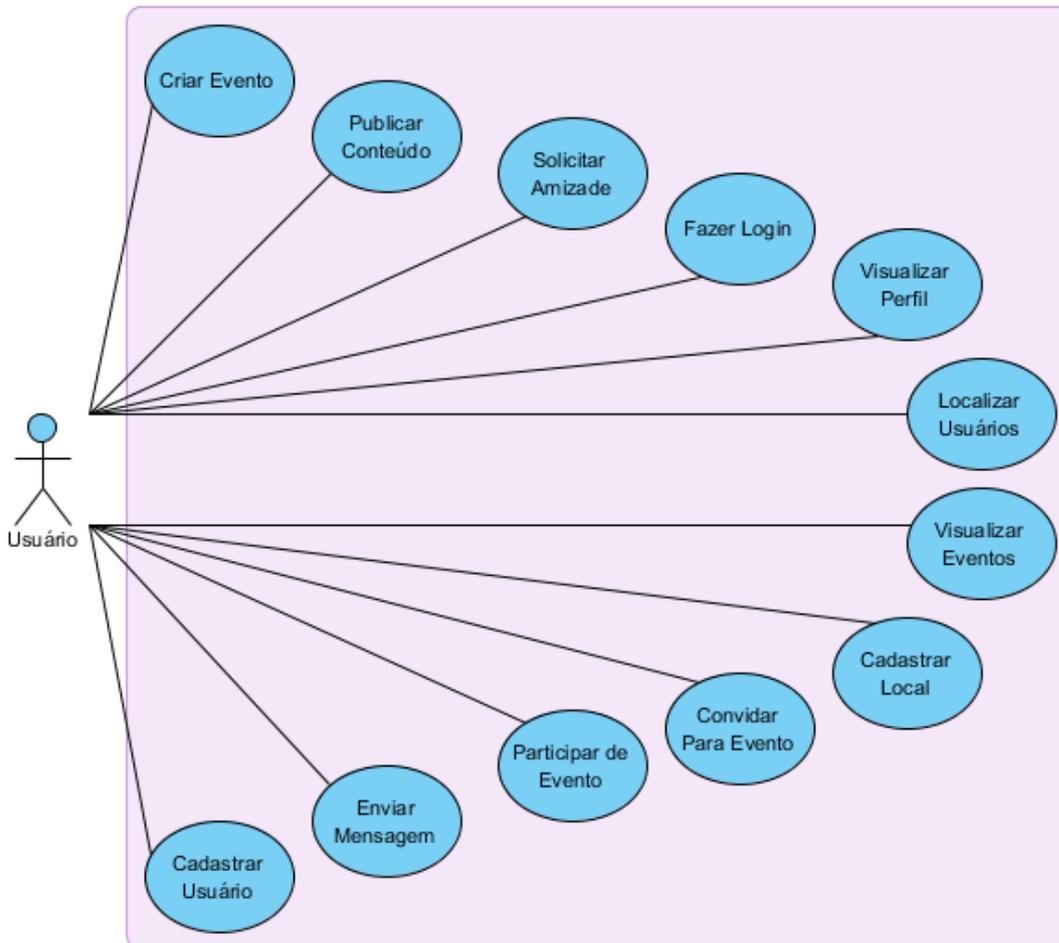


Figura 7 - Diagrama de casos de uso  
 Fonte: Elaborado pelo autor

Os Quadros 14 a 17 apresentam três casos de uso expandido para o sistema proposto, onde mostram seus atores, pré-condições, pós-condições, os requisitos relacionados, sequência típica de eventos e as possíveis exceções e variantes que podem ocorrer.

<b>Caso de Uso: Solicitar Amizade</b>
<b>Atores: Usuário</b>
<b>Precondições: O usuário deverá estar cadastrado na rede social, bem como o usuário a quem deseja fazer o pedido de amizade. Ambos deverão estar conectados à Internet.</b>
<b>Pós-condições: Usuários ficam relacionados permitindo a interação entre eles.</b>
<b>Requisitos Correlacionados: RF1 – Cadastrar Usuários, RF8 – Solicitar Amizade, RF10 – Localizar Outros Usuários.</b>
<b>Sequência típica de eventos (Fluxo Principal):</b>

<p>Esse caso de uso inicia quando:</p> <ol style="list-style-type: none"> <li>1. [IN] O usuário localiza outro usuário na rede.</li> <li>2. [IN] O usuário faz um pedido de amizade.</li> <li>3. [OUT] O sistema aprova o pedido de amizade.</li> <li>4. [OUT] O sistema permite que o usuário convidado passe a ser visível para o usuário solicitante.</li> </ol>
<p><b>Tratamento de Exceções e Variantes:</b>  <b>Exceção 1a: Usuário não localiza outro usuário.</b>          [IN] O usuário informa outros dados no filtro.          Retorna ao passo 1.  <b>Exceção 3a: Usuário não aprova o pedido de amizade.</b>          [OUT] O caso de uso é finalizado.</p>

Quadro 14 - Caso de Uso Expandido – Solicitar Amizade

Fonte: Elaborado pelo autor

<p><b>Caso de Uso: Criar Evento</b></p>
<p><b>Atores: Usuário.</b></p>
<p><b>Precondições: O usuário deverá possuir um perfil na rede social e estar conectado à Internet.</b></p>
<p><b>Pós-condições: Usuário cria um evento permitindo convidar seus amigos para o mesmo.</b></p>
<p><b>Requisitos Correlacionados: RF1 – Cadastrar Usuário, RF2 – Criar Evento</b></p>
<p><b>Sequência típica de eventos (Fluxo Principal):</b></p> <p>Esse caso de uso inicia quando:</p> <ol style="list-style-type: none"> <li>1. [IN] O usuário clica sobre criar evento.</li> <li>2. [OUT] O sistema disponibiliza os campos para informar os dados do evento.</li> <li>3. [IN] O usuário informa os dados do evento.</li> <li>4. [IN] O usuário seleciona um local.</li> <li>5. [OUT] O sistema cria o evento e permite o convite para amigos relacionados.</li> </ol>
<p><b>Tratamento de Exceções e Variantes:</b>  <b>Exceção 5a: Campos não completos.</b>          [IN] O cliente informa todos os dados necessários.          [OUT] Retorna ao passo 3.</p>

Quadro 15 - Caso de Uso Expandido – Criar Evento

Fonte: Elaborado pelo autor

<p><b>Caso de Uso: Publicar Conteúdo</b></p>
<p><b>Atores: Usuário.</b></p>
<p><b>Precondições: O usuário deverá possuir um perfil na rede social e estar conectado à Internet.</b></p>
<p><b>Pós-condições: Usuário publica um conteúdo visível para seus amigos em seu perfil.</b></p>

<b>Requisitos Correlacionados:</b> RF1 – Cadastrar Usuário, RF11 – Publicar Conteúdo.
<p><b>Sequência típica de eventos (Fluxo Principal):</b></p> <p>Esse caso de uso inicia quando:</p> <ol style="list-style-type: none"> <li>1. [IN] O usuário entra em seu perfil e escreve um conteúdo ou insere uma imagem.</li> <li>2. [IN] O usuário clica sobre o botão “Postar”</li> <li>3. [OUT] O sistema publica o conteúdo.</li> <li>4. [OUT] O sistema disponibiliza o conteúdo para os usuários relacionados.</li> </ol>
<p><b>Tratamento de Exceções e Variantes:</b></p> <p><b>Exceção 1a: Formato de conteúdo inválido ou excede o tamanho.</b>  [IN] O cliente informa outro conteúdo.  [OUT] Retorna ao passo 1.</p>

**Quadro 16 - Caso de Uso Expandido – Publicar Conteúdo**

Fonte: Elaborado pelo autor

<b>Caso de Uso:</b> Participar de Evento
<b>Atores:</b> Usuário
<b>Precondições:</b> O usuário deverá estar cadastrado na rede social e ter eventos disponíveis para ele participar.
<b>Pós-condições:</b> Usuário é relacionado como participante do evento e pode publicar na página do evento.
<b>Requisitos Correlacionados:</b> RF1 – Cadastrar Usuários, RF4 – Convidar para Evento, RF5 – Visualização dos Eventos, RF6 – Confirmar Participação em Eventos.
<p><b>Sequência típica de eventos (Fluxo Principal):</b></p> <p>Esse caso de uso inicia quando:</p> <ol style="list-style-type: none"> <li>1. [IN] O usuário visualiza um evento na rede.</li> <li>2. [IN] O usuário confirma sua participação.</li> <li>3. [OUT] O sistema relaciona o usuário ao evento.</li> <li>4. [OUT] O sistema permite que o usuário participante possa convidar outros para o evento e publicar na página do evento.</li> </ol>
<p><b>Tratamento de Exceções e Variantes:</b></p> <p><b>Exceção 1a: Usuário não encontra um evento disponível.</b>  [OUT] O caso de uso é finalizado.</p>

**Quadro 17 - Caso de Uso Expandido – Participar do Evento**

Fonte: Elaborado pelo autor

A Figura 8 apresenta o modelo conceitual para o sistema proposto, o qual apresenta as classes de análise identificadas para o sistema.



Os quadros abaixo de 18 a 30, apresentam os mapeamentos de entidade para o sistema proposto, mostram os atributos das tabelas do sistema.

### Amizade

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodUsuario1	Inteiro	Não	Não	Sim	
CodUsuario2	Inteiro	Não	Não	Sim	
DataAmizade	Data	Não	Não	Não	

Quadro 18 - Mapeamento entidade – Amizade

Fonte: Elaborado pelo autor

### Mensagem

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Codigo	Inteiro	Não	Sim	Não	
CodRemetente	Inteiro	Não	Não	Sim	
CodDestinatario	Inteiro	Não	Não	Sim	
Mensagem	Texto	Não	Não	Não	

Quadro 19 - Mapeamento entidade – Mensagem

Fonte: Elaborado pelo autor

### UsuarioPublicacao

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Codigo	Inteiro	Não	Sim	Não	
Publicacao	Texto	Não	Não	Não	
CodUsuario	Inteiro	Não	Não	Sim	
CodAutor	Inteiro	Não	Não	Sim	

Quadro 20 - Mapeamento entidade – UsuarioPublicacao

Fonte: Elaborado pelo autor

### Usuario

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Codigo	Inteiro	Não	Sim	Não	
Nome	Texto	Não	Não	Não	
Sobrenome	Texto	Não	Não	Não	
DataNascimento	Data	Sim	Não	Não	
Sexo	Char	Sim	Não	Não	

Email	Texto	Não	Não	Não	
Senha	Texto Longo	Não	Não	Não	
CodCidade	Inteiro	Não	Não	Sim	
Imagem	Blob	Sim	Não	Não	

**Quadro 21 - Mapeamento entidade – Usuario**

Fonte: Elaborado pelo autor

### UsuarioEsporte

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Chave primária</b>	<b>Chave estrangeira</b>	<b>Observações</b>
Codigo	Inteiro	Não	Sim	Não	
CodUsuario	Inteiro	Não	Não	Sim	
CodEsporte	Inteiro	Não	Não	Sim	
AnoInicio	Inteiro	Não	Não	Não	

**Quadro 22 - Mapeamento entidade – UsuarioEsporte**

Fonte: Elaborado pelo autor

### Cidade

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Chave primária</b>	<b>Chave estrangeira</b>	<b>Observações</b>
Codigo	Inteiro	Não	Sim	Não	
Descricao	Texto	Não	Não	Não	
CodUf	Inteiro	Não	Não	Sim	

**Quadro 23 - Mapeamento entidade – Cidade**

Fonte: Elaborado pelo autor

### Uf

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Chave primária</b>	<b>Chave estrangeira</b>	<b>Observações</b>
Codigo	Inteiro	Não	Sim	Não	
Descricao	Texto	Não	Não	Não	
Sigla	Texto	Não	Não	Não	

**Quadro 24 - Mapeamento entidade – Uf**

Fonte: Elaborado pelo autor

### Esporte

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Chave primária</b>	<b>Chave estrangeira</b>	<b>Observações</b>
Codigo	Inteiro	Não	Sim	Não	
Descricao	Texto	Não	Não	Não	

**Quadro 25 - Mapeamento entidade – Esporte**

Fonte: Elaborado pelo autor

### EventoPublicacao

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Codigo	Inteiro	Não	Sim	Não	
Publicacao	Texto	Não	Não	Não	
CodParticipante	Inteiro	Não	Não	Sim	
CodEvento	Inteiro	Não	Não	Sim	
CodUsuario	Inteiro	Não	Não	Sim	

Quadro 26 - Mapeamento entidade – EventoPublicacao

Fonte: Elaborado pelo autor

### ParticipanteEvento

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Codigo	Inteiro	Não	Sim	Não	
CodEvento	Inteiro	Não	Não	Sim	
CodUsuario	Inteiro	Não	Não	Sim	
DataConfirmacao	Data Hora	Não	Não	Não	
Confirmado	Inteiro	Não	Não	Não	

Quadro 27 - Mapeamento entidade – ParticipanteEvento

Fonte: Elaborado pelo autor

### Evento

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Codigo	Inteiro	Não	Sim	Não	
Título	Texto	Não	Não	Não	
DataHora	Data Hora	Não	Não	Não	
NumeroParticipantes	Inteiro	Sim	Não	Não	
Contato	Texto	Sim	Não	Não	
Privacidade	Inteiro	Não	Não	Não	
Descricao	Texto	Sim	Não	Não	
CodUsuarioAdministrador	Inteiro	Não	Não	Sim	
CodEsporte	Inteiro	Não	Não	Sim	
CodLocal	Inteiro	Não	Não	Sim	

Quadro 28 - Mapeamento entidade – Evento

Fonte: Elaborado pelo autor

### Local

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
-------	------	------	----------------	-------------------	-------------

Codigo	Inteiro	Não	Sim	Não	
Titulo	Texto	Não	Não	Não	
Logradouro	Texto	Não	Não	Não	
Complemento	Texto	Sim	Não	Não	
Cep	Texto	Sim	Não	Não	
Observação	Texto	Não	Não	Não	
CodCidade	Inteiro	Não	Não	Sim	

**Quadro 29 - Mapeamento entidade – Local**

Fonte: Elaborado pelo autor

### Imagem

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CaminhoImagem	Texto	Não	Não	Não	
CodEvento	Inteiro	Sim	Não	Sim	
CodPublicacao	Inteiro	Sim	Não	Sim	

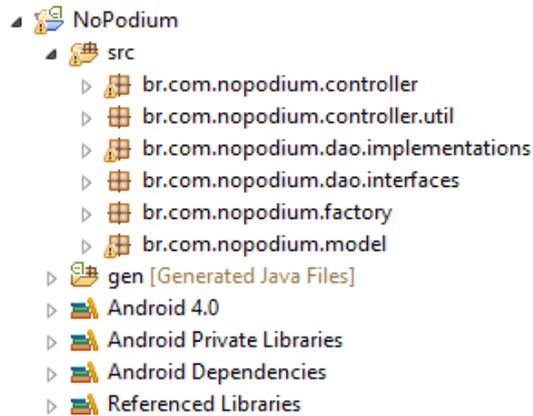
**Quadro 30 - Mapeamento entidade - Imagem**

Fonte: Elaborado pelo autor

## 4.3 IMPLEMENTAÇÃO DO APLICATIVO

A Figura 10 mostra como o projeto foi estruturado dentro da IDE Eclipse, utilizou-se o *Model View Controller* (MVC), onde cada pacote possui as classes Java com diferentes funções, no pacote controller está a interação com a tela, essas classes são as que recuperam os dados da tela e a processam.

O pacote util contém uma classe de utilidade para se usar no programa, como funções de formatação de data. O pacote implementations contém a lógica de recuperação dos dados para inserção no banco, o pacote interfaces se encontra as classes que contém as funções que conectarão as classes de implementations com a classe controller. No pacote factory estão os arquivos de configuração de conexão com o banco e por fim a classe *model* contém os objetos que serão utilizados no aplicativo.



**Figura 10 - Estruturação na IDE**  
**Fonte: Elaborado pelo autor**

A seguir são apresentadas as telas desenvolvidas para o aplicativo móvel proposto e partes do código.

A Figura 11 mostra a tela de *login* do aplicativo, onde o usuário tem a opção de fazer o *login*, recuperar sua senha, utilizar o aplicativo de forma *offline* (que apresenta os eventos da cidade do usuário) ou criar uma nova conta.



**Figura 11 – Tela de Login**  
**Fonte: Elaborado pelo autor**

A listagem 1 apresenta o código usado na tela de *login* para persistir o código do usuário que será utilizado nas funções dentro do aplicativo.

```
1. SharedPreferences settings = getSharedPreferences(PREFS_NP, 0);
2. SharedPreferences.Editor editor = settings.edit();
3. editor.putInt("codigoUsuario", codigoUsuario);
4. editor.commit();
```

**Listagem 1 - Código para persistir o código do usuário na aplicação.**

O *SharedPreferences* é uma classe utilizada no Android que permite que se armazene dentro dele dados de tipo primitivo, utilizando o formato chave-valor. É uma forma de armazenamento rápida que pode ser compartilhada dentro da aplicação e também com outras aplicações, caso necessário, geralmente se armazena configurações e preferências do usuário nesse tipo de arquivo.

Nesse caso, na primeira linha foi criado o arquivo do tipo *SharedPreferences* com o nome *PREFES\_NP*, essa variável foi criada no cabeçalho como uma constante do tipo *String*, o parâmetro 0 indica que é do tipo *private* que pode ser lido somente pela aplicação, na linha 2 é aberto o arquivo criado para editá-lo, na linha 3 foi incluso no arquivo criado o código do usuário que está armazenado na variável 'codigoUsuario', salvo com a chave do mesmo nome, então na 4 linha é salva a edição feita no arquivo criado.

Assim pode-se recuperar o código do usuário dentro da aplicação utilizando o código da listagem 2 que está a seguir:

```
1. SharedPreferences settings =
    getSharedPreferences(LoginActivity.PREFS_NP, 0);
2. codigoUsuario = settings.getInt("codigoUsuario", 0);
```

**Listagem 2 - Código para recupera o código do usuário dentro da aplicação.**

Na primeira linha é recuperado o arquivo do tipo *SharedPreferences* que está localizado com constante na tela 'LoginActivity', na segunda a variável 'codigoUsuario' recebe o valor da chave 'codigoUsuario', 0 indica que é o valor padrão caso não encontre nenhum valor na chave.

A Figura 12 apresenta o cadastro de usuário, esse cadastro são os dados preenchidos antes do usuário ingressar na rede, portanto é o cadastro das informações básicas.

**Figura 12 - Formulário de cadastro de usuário**  
**Fonte: Elaborado pelo autor**

A Listagem 3 apresenta o código utilizado para implementação da função para recuperar uma imagem salva no aparelho celular do usuário e utilizá-la como imagem de perfil do usuário.

```

public void btSelecionarImagemOnClick(View v) {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(
        Intent.createChooser(intent, "Selecione uma imagem"), 1);
}

protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == 1 && resultCode == RESULT_OK && null != data) {

        Uri selectedImage = data.getData();
        String[] filePathColumn = { MediaStore.Images.Media.DATA };

        Cursor cursor = getContentResolver().query(selectedImage,
            filePathColumn, null, null, null);
        cursor.moveToFirst();

        int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
        String picturePath = cursor.getString(columnIndex);
        cursor.close();
    }
}

```

```
        imgImagemUsuario.setImageBitmap(BitmapFactory.decodeFile  
(picturePath));  
        pathImagem = picturePath;  
    }  
}
```

### Listagem 3 - Código para recuperar uma imagem do aparelho

O método *btSelecionarImagemOnClick* está associado ao botão no cadastro do usuário para selecionar a imagem, quando o botão é clicado, um novo objeto *intent* é instanciado. Esse objeto é usado para executar a chamada de algum recurso, seja ele interno ou externo, nesse objeto é declarado o tipo de busca (ex. imagem) e a ação de busca de conteúdo no aparelho.

Então o aplicativo aguarda pela seleção da imagem, uma vez selecionada a imagem é enviada para o método *onActivityResult*, esse por sua vez faz uma validação para ver se ele selecionou uma imagem, caso afirmativo, essa imagem é valorizada no componente *ImageView* que está na tela de cadastro do usuário e o caminho da imagem é salvo na variável *pathImagem*. Então o usuário pode finalizar seu cadastro.

A Figura 13 apresenta a tela inicial do sistema, aqui o usuário escolhe que operação deseja realizar dentro do aplicativo.



**Figura 13 - Tela Inicial**  
Fonte: Elaborado pelo autor

Por padrão, quando se cria uma tela dentro da aplicação Android, é criado automaticamente um arquivo XML e um arquivo Java, no arquivo XML ficam armazenados os componentes da interface gráfica do aplicativo e no arquivo Java fica o código-fonte da tela, onde está localizada a lógica do aplicativo. Essa tela é criada a partir de um arquivo XML, onde contém um objeto visual *ListView* que é usado para fazer listagens dentro do Android, este apresentado na Listagem 4.

```
<ListView
    android:id="@+id/lvPrincipal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:entries="@array/opcoes_tela_principal"/>
```

**Listagem 4 - Código XML criando uma ListView**

Assim como na maioria das linguagens de desenvolvimento atuais, há uma facilidade considerável em inserir e modificar os elementos visuais, porém uma das diferenças do Android é que em cada componente deve-se implementar os atributos *layout\_width* e *layout\_height*.

Esses atributos definem a largura e a altura do componente, podem-se inserir números ou funções do Android como *fill\_parent* e *wrap\_content*. A primeira opção *fill\_parent* define que o componente terá o tamanho total da tela, por exemplo, caso seja colocado no *layout\_width* o componente irá ocupar a largura total da tela.

Já o *wrap\_content*, permite que o tamanho do componente seja o tamanho mínimo possível, ou seja, o tamanho do componente em si. Por exemplo, se esse atributo estiver no *layout\_width*, ele vai ocupar somente a largura que o componente precisa para ser exibido.

Assim, a lista criada ocupa todo o espaço da tela tanto em largura e altura. O último atributo especifica que a lista de entrada que o componente vai exibir é o array 'opcoes\_tela\_principal', este com uma lista de valores, conforme Listagem 5.

```
<resources>
    <string-array name="opcoes_tela_principal">
        <item>Meu Perfil</item>
        <item>Meus Eventos</item>
        <item>Minhas Amizades</item>
        <item>Pesquisar Eventos</item>
        <item>Pesquisar Pessoas</item>
        <item>Novo Evento</item>
        <item>Mensagens</item>
        <item>Configurações</item>
    </string-array>
```

```
</resources>
```

#### Listagem 5 - Código XML com os Itens do Array.

Este array possui os itens de texto que são exibidos na tela principal do aplicativo, com o mesmo nome do atributo *android:entries* da lista criada, assim a lista é apresentada na tela.

A Figura 14 apresenta o cadastro do evento, com os principais dados para o cadastro, nessa tela o usuário seleciona também se o evento será público ou privado. Essa opção permite que o evento seja visualizado somente pelos convidados do evento ou por todos os participantes da comunidade virtual, para essa alteração foi usado o componente *ToggleButton* é um botão que possui um texto que muda dependendo do estado e uma indicação colorida para indicar se está ativo ou não.

A imagem mostra a interface de usuário para o cadastro de um novo evento. O formulário contém os seguintes campos e elementos:

- Título do Evento:** Campo de texto.
- Data e Hora:** Campo de texto com máscara `dd/mm/aaaa hh:mm`.
- Número de Participantes:** Campo de texto.
- Contato:** Campo de texto.
- Privacidade:** Botão *ToggleButton* atualmente configurado para "Privado".
- Descrição:** Área de texto para a descrição do evento.
- Esporte:** Campo de texto com botão "Selecionar".
- Estado:** Campo de texto com botão "Selecionar".
- Cidade:** Campo de texto com botão "Selecionar".
- Local:** Campo de texto com botão "Selecionar".
- Finalizar Cadastro:** Botão de ação principal.

Figura 14 - Formulário de cadastro de evento  
Fonte: Elaborado pelo autor

A Listagem 6 apresenta parte do código XML da tela de cadastro do evento, incluindo a declaração do *ToggleButton*.

```

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Privacidade:" />

    <ToggleButton
        android:id="@+id/etTooglePrivacidade"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="Público"
        android:textOff="Privado"/>
</LinearLayout>

```

**Listagem 6 - Código XML parte visual cadastro de evento**

Nesse código XML é apresentado o componente *LinearLayout*, esse componente tem o papel de organizar os componentes na tela, ele faz isso através dos atributo *android:orientation*, que quando colocado como *horizontal* dispõe os objetos horizontalmente na tela um abaixo do outro, já a opção *vertical* dispõe os objetos verticalmente um ao lado do outro.

O *ToggleButton* está configurado para quando o botão estiver pressionado aparecer o texto 'Público' e quando está livre aparecer o texto 'Privado'. Essa função permitira ao usuário cadastrar o evento que pode estar visível ou não para outros usuários.

A Figura 15 apresenta a tela do perfil do usuário, nela estão contidas as informações do usuário, a opção de envio de mensagem ao usuário e a opção de visualizar e publicar no perfil do usuário.



**Figura 15 - Tela das Informações do Perfil**  
**Fonte: Elaborado pelo autor**

Para essa tela foram utilizados os recursos do componente *TabHost*, no qual define-se as tabs, para mudança do conteúdo da tela, abaixo na listagem 7 é possível visualizar o seu funcionamento.

```
TabHost tabHost = (TabHost) findViewById(android.R.id.tabhost);

TabSpec tab1 = tabHost.newTabSpec("Perfil");
TabSpec tab2 = tabHost.newTabSpec("Mensagem");
TabSpec tab3 = tabHost.newTabSpec("Publicações");

tab1.setIndicator("Perfil");
tab1.setContent(new Intent(this, TabPerfilActivity.class));

tab2.setIndicator("Mensagem");
tab2.setContent(new Intent(this, TabEscreverMensagemActivity.class));

tab3.setIndicator("Publicações");
tab3.setContent(new Intent(this, PublicarPerfil.class));

tabHost.addTab(tab1);
tabHost.addTab(tab2);
tabHost.addTab(tab3);
```

**Listagem 7 - Código Java criação do Tab.**

Na primeira linha da aplicação é instanciado o componente *TabHost*, no qual recebe um componente criado no XML para ter interação com a tela, após são especificadas as telas que serão criadas Perfil, Mensagem e Publicações, assim,

cada uma dessas tabs especificadas recebe o indicador e qual tela deverá mostrar ao ser clicada, então as tabs são adicionadas no componente *TabHost* criado finalizando a criação das tabs.

Após ser adicionada as tabs, o conteúdo das mesmas já está disponível para exibição, a primeira que aparece é a tab 'Perfil' que mostra alguns dados do usuário selecionado e os esportes que ele pratica, para essa tela foi usado um componente chamado '*ImageView*' que exibe a imagem do usuário, a lógica para isso acontecer pode ser vista na Listagem 8.

```
if (usuarioSelecionado.getImagem() != null) {
    int blobLenght;
    byte[] imgByte = null;
    try {
        blobLenght = (int)usuarioSelecionado.getImagem().length();
        imgByte = usuarioSelecionado.getImagem().getBytes(1, blobLenght);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    ByteArrayInputStream is = new ByteArrayInputStream(imgByte);
    Bitmap picBitmap = BitmapFactory.decodeStream(is);

    imgUsuarioPerfil.setImageBitmap(picBitmap);
}
```

**Listagem 8 - Código Java para Recuperar Imagem do Banco.**

A imagem de perfil no banco de dados foi guardada como tipo Blob, que é a abreviação para objeto binário extenso, a imagem é armazenada em forma binária dentro do banco, e para exibí-la, foi feito o seguinte processo, na primeira linha há uma validação para se certificar de que a imagem não está vazia, após foram criadas as variáveis para fazer a conversão do binário, foi declarada uma variável inteira que recebe o tamanho do binário, e uma variável do tipo *byte* que recebe os *bytes* do *blob* para se converter, a conversão de *blob* para *byte* é feito nas linhas seguintes, o *byte* é passado para uma variável do tipo *ByteArrayInputStream* para ser decodificada em *Bitmap* para finalmente ser colocada no *ImageView* e ser exibida.

A lista de esportes foi montada a partir de uma lista de Strings, lista essa feita a partir da lista de objetos que foi recuperada do banco de dados. Foi instanciado um *Adapter* que é quem faz a mediação do conteúdo do banco e os itens na tela, como se fosse um adaptador entre a lista e os conteúdos do banco, conforme apresentado na Listagem 9.

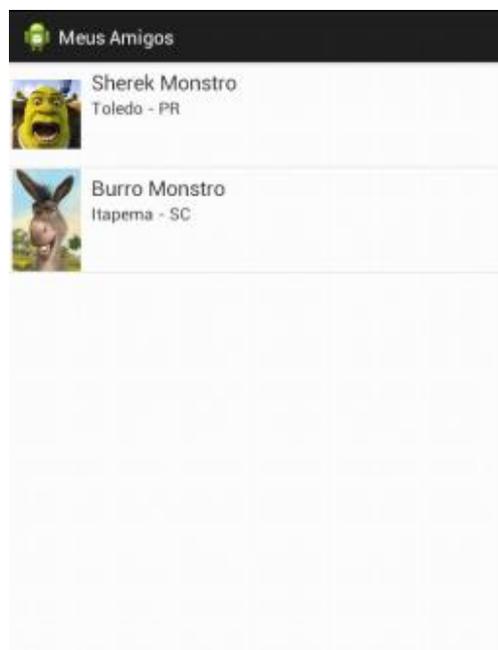
```
aaUEsportes = new ArrayAdapter<String>(TabPerfilActivity.this,  
android.R.layout.simple_list_item_1, uEsportes);  
  
listaPerfil.setAdapter(aaUEsportes);
```

#### Listagem 9 - Código Java para Exibir os itens da Lista.

Na primeira linha a variável `aaUEsportes`, do tipo `ArrayAdapter` de `String`, que foi previamente criada, recebe os atributos do adaptador, primeiramente a classe em que o adaptador vai ser utilizado é declarada, o segundo atributo indica como os itens do `array` vão ser mostrados na tela.

O Android já traz alguns padrões de tipo de lista para se utilizar, nesse caso foi utilizado o tipo padrão, onde será exibido um item por linha, o terceiro atributo é a lista de esportes previamente carregada no banco, assim o adaptador foi criado. Por fim, na última linha da listagem a lista recebeu o adaptador 'aaUEsportes' fazendo com que os itens fossem listados.

Além dos Adapters que o Android traz como modelo, é possível criar um Adapter personalizado para ser usado, um Adapter personalizado foi usado para criar a lista mostrada na Figura 16.



**Figura 16 - Lista de Amigos**  
Fonte: Elaborado pelo autor

Para criar essa lista com o Adapter personalizado, inicialmente foi criado através do XML como os registros devem ser colocados na lista, como pode ser visto na listagem 10.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <ImageView
        android:id="@+id/imgUsuarioLista"
        android:layout_width="80dp"
        android:layout_height="100dp"
        android:paddingRight="10dp"
        android:paddingLeft="5dp"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:paddingTop="5dp">

        <TextView
            android:id="@+id/tvNomeUsuario"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize="20sp"/>

        <TextView
            android:id="@+id/tvCidadeUsuario"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textSize="16sp"/>
    </LinearLayout>
</LinearLayout>
```

**Listagem 10 - Código XML para exibir os itens na lista.**

No XML foi criado um componente do tipo *LinearLayout* na horizontal que contém os demais componentes da tela, dentro dele foi criado o *ImageView* e na sequência outro *LinearLayout* na vertical para dispor os dois *TextViews* criados na sequência que, conforme seu Id sugere, o primeiro irá armazenar o nome do usuário e o segundo a cidade do usuário.

Após isso, deve-se criar uma classe que estende da classe *ArrayAdapter* para que possa fazer a mediação entre o *layout* criado no XML e os dados do banco, como pode-se ver na Listagem 11.

```
public AdapterUsuarios(Context context, int resource, List<Usuario>
objects) {
    super(context, resource, objects);
    this.inflater = LayoutInflater.from(context);
    this.resourceId = resource;
}
```

```

@Override
public View getView(int position, View convertView, ViewGroup parent){
    Usuario usuario = getItem(position);
    convertView = inflater.inflate(resourceId, parent, false);

    ImageView imgUsuario = (ImageView)
convertView.findViewById(R.id.imgUsuarioLista);
    TextView tvNomeUsuario = (TextView)
convertView.findViewById(R.id.tvNomeUsuario);
    TextView tvCidadeUsuario = (TextView)
convertView.findViewById(R.id.tvCidadeUsuario);

```

**Listagem 11 - Código Java para Povoar os Dados na Lista.**

Primeiramente quando se chama o Adapter o método construtor recebe o local onde foi codificado o Adapter, o código XML com os componentes e a lista que será exibida, nesse caso, a lista de usuários. O segundo método, o *getView* é responsável por inserir os dados do objeto na lista, para isso ele instancia um objeto que será utilizado para recuperar os dados um de cada vez, a variável *convertView*, que é o objeto responsável pelos componentes visuais, utilizando o método *inflate*, faz com que o XML seja recuperado e passado para a variável *convertView*, a partir daí, ele recupera os objetos de imagem e texto e faz as variáveis dos componentes locais receberem os mesmos, após ele recupera os itens da lista e valoriza os componentes, ao final de tudo ele retorna a variável *convertView*, como mostra a Listagem 12.

```

return convertView;

```

**Listagem 12 - Código de retorno.**

A listagem de eventos também segue o mesmo padrão, utilizando o Adapter, ele lista os eventos, a diferença para a listagem de usuários é de que a listagem de eventos não utiliza imagem, somente textos, como pode-se ver na Figura 17.

Listagem de Eventos	
Futebol Praça	
Futebol	
Pato Branco	10/12/2014 15:34
Corrida 5Km Centro	
Corrida de Rua	
Pato Branco	10/12/2014 20:40
Corrida 10Km Interior	
Corrida de Rua	
Pato Branco	15/01/2015 10:25

**Figura 17 - Listagem de Eventos**  
**Fonte: Elaborado pelo autor**

Para a listagem de eventos o usuário tem a opção de filtrar o resultado, sendo obrigatório o usuário selecionar de qual cidade ele deseja listar os eventos, a tela para pesquisa de eventos é mostrada abaixo na Figura 18.



**Figura 18 - Tela de Pesquisa de Evento**  
**Fonte: Elaborado pelo autor**

Nessa tela, as opções para seleção de estado, cidade e esporte são feitas em uma lista. Após a seleção nessa lista, essa retorna o id da seleção e o texto, e coloca o texto no campo adequado. A lógica para isso está na Listagem 13 que mostra a seleção do estado.

```

public void btSelecionarEstadoOnClick(View v) {
    Intent iUf = new Intent( this, ListaUfs.class);
    startActivityForResult(iUf, 2);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1 && data != null) {
        codigoEsporte = data.getIntExtra("codEsporte", 0);
        String selUf = data.getStringExtra("selEsporte");
        etEsporteEvento.setText(String.valueOf(selUf));
    } else if (requestCode == 2 && data != null) {
        codigoUf = data.getIntExtra("codUf", 0);
        String selUf = data.getStringExtra("selUf");
        etEstadoEvento.setText(String.valueOf(selUf));
        Intent iCidade = new Intent( this, ListaCidades.class);
        iCidade.putExtra("codigoUf", codigoUf);
        startActivityForResult(iCidade, 3);
    } else if (requestCode == 3 && data != null) {
        codigoCidade = data.getIntExtra("codCidade", 0);
        String selCidade = data.getStringExtra("selCidade");
    }
}

```

```
        etCidadeEvento.setText (String.valueOf(selCidade));  
    }  
}
```

**Listagem 13 – Código para Recuperar o Código e Descrição do Estado.**

O método `btSelecionarEstadoOnClick` é executado quando o usuário pressiona o botão para selecionar o estado, ele declara uma nova *Intent* que é o método usado para a navegação de telas, a tela que ele abre é a *activity* 'ListaUfs', ele aguarda pelo retorno da *activity* e seu resultado através do método *startActivityForResult*, quando ele retorna o método *onActivityResult* é chamado, sendo que o *requestCode* é que define qual foi a tela que retornou, nesse caso o *requestCode* é 2, quando ele entra no método ele insere o código da UF a partir dos dados recebidos, assim como o componente 'etEstadoEvento' - que é onde fica o texto do estado - recebe o estado, após isso ele instancia uma nova *intent* e chama a tela para seleção da cidade.

Depois de preencher os campos para a pesquisa e clicar sobre o botão pesquisar, a tela de listagem vista anteriormente é apresentada, na tela de listagem o usuário tem a opção de visualizar os dados do evento desejado, a tela com os dados do evento é mostrada na Figura 19.



**Figura 19 - Tela de Exibição do Evento**  
**Fonte: Elaborado pelo autor**

Nessa tela, o usuário tem acesso aos dados do evento, aqui também ele pode confirmar sua participação no evento, assim ele entra para a lista de participantes e pode publicar no evento, e também o evento ficará salvo na lista dos eventos do usuário.

## 5 CONSIDERAÇÕES FINAIS

O trabalho realizado teve por objetivo realizar uma pesquisa acerca de redes sociais, comunidades virtuais, prática de esportes e o desenvolvimento de aplicativos móveis, visando o desenvolvimento de um aplicativo móvel, baseado no conceito de redes sociais. Com isso, foi possível realizar a especificação dos requisitos necessários, bem como a modelagem e a implementação das telas e funções para um aplicativo baseado em comunidade virtual.

A partir das pesquisas concluiu-se que o aplicativo a ser desenvolvido trata-se uma rede social, podendo ser classificada como uma comunidade virtual de prática de esportes.

Na pesquisa das plataformas para desenvolvimento móvel foi aprofundado o estudo na plataforma Android, plataforma escolhida para o desenvolvimento do aplicativo, e que utiliza a tecnologia Java e vem se destacando no mercado Mobile. Na modelagem foram levantados requisitos e, a partir desses, elaborados modelos a fim de facilitar o entendimento das funcionalidades do aplicativo, bem como da sua arquitetura inicial de classes e organização dos dados envolvidos. Por fim o aplicativo foi implementado, com suas funções e a interface do usuário.

De maneira geral a pesquisa e as tecnologias utilizadas nesse trabalho foram de grande valia, uma vez que são atuais, pois a plataforma móvel é uma plataforma que está cada vez mais no mercado, assim como a tecnologia Android, tendência essa que deve continuar crescendo nos próximos anos. Por isso, atualmente existem muitos fóruns na internet que disponibilizam conteúdos e resolução de dúvidas sobre o desenvolvimento Android, e que foram consultados durante o desenvolvimento desse aplicativo sanando as dificuldades na implementação.

O trabalho realizado atingiu seus objetivos, a partir da resolução do problema proposto. Acredita-se que, o aplicativo NoPodium ao ser disponibilizado na loja de aplicativos da Google para uso, pode tornar-se um grande canal de comunicação entre os praticantes e admiradores do esporte e também um segundo passo que poderia ser tomado é ser melhorado com mais funções como o uso de GPS para marcar os locais, a implementação de compartilhamento de imagens pelo aplicativo, a interação com as principais redes sociais e também ser desenvolvido em outras plataformas móveis.

## REFERÊNCIAS

AMBROS, Luisa. **Diferença entre Aplicativos Nativos, Híbridos e Mobile Web Apps**. Luisaambros.com. 24 de Setembro de 2013. Disponível em: <<http://luisaambros.com/blog/diferenca-entre-aplicativos-nativos-hibridos-e-mobile-web-apps/>>. Acesso em 23 de Maio 2014.

CASSIANO, Adriele Machado. **Ativismo a partir das redes sociais**. 2011. Artigo científico - Curso de Pós-Graduação em Mídia, Informação e Cultura, USP, São Paulo.

CUMMINGS, Jonathon N; BUTLER, Brian; KRAUT, Robert. **The Quality of online Social**. 2002. Relationships. Communications of the ACM, Volume 45, Number 7, pp. 103-108.

EU ATLETA. **Globo**. Disponível em: <<http://globoesporte.globo.com/eu-atleta/>>. Acesso em 16 de Maio 2014.

FACEBOOK. **Play Store**. 16 de Maio de 2014. Disponível em: <<https://play.google.com/store/apps/details?id=com.facebook.katana>>. Acesso em 16 de Maio 2014.

FELIZZOLA, Humberto Bernardes. **Desenvolvimento de Aplicativo Mobile para Pesquisa de Informações sobre Transportes Públicos**. 2012. Trabalho de Graduação, Universidade Federal do Rio Grande do Sul, Porto Alegre.

GADELHA, Reginaldo. **Desenvolvendo para Android: Arquitetura Android**. Tiselvagem.com.br. 20 de Outubro de 2011. Disponível em: <<http://www.tiselvagem.com.br/geral/desenvolvendo-para-android-arquitetura-android/>>. Acesso em 30 de Maio 2014.

GARGENTA, Marko. **Larning Android**. Sebastopol: O'Reilly, 2011.

GONÇALVES, Alysson Pires. **Rede Social na UM: Um estudo de Caso**. 2008. Dissertação – Mestrado em Tecnologias e Sistemas de Informação, Universidade do Minho, Guimarães.

GRONER, Loiane. **Slides da Palestra Iniciando com desenvolvimento híbrido de aplicações mobile com HTML5 (UFG)**. Loiane.com. 14 de Novembro de 2013. Disponível em: <<http://www.loiane.com/2013/11/slides-da-palestra-iniciando-com-desenvolvimento-hibrido-de-aplicacoes-mobile-com-html5-ufg/>>. Acesso em 30 de Maio 2014.

HAETINGER, D. **Fatores Relevantes à Formação e Manutenção de Comunidades Virtuais Facilitadores da Aprendizagem**. CINTED-UFRGS, Novas Tecnologias na Educação, V.3, Nº 1, Maio, 2005.

IHAA. **Benefícios da prática do esporte para a nossa saúde**. 26 de Dezembro de 2011. Disponível em: <<http://ihaa.com.br/beneficios-da-pratica-do-esporte-para-a-nossa-saude/>>. Acesso em 16 de Maio 2014.

INSTAGRAM. **Play Store**. 21 de Maio de 2014. Disponível em: <<https://play.google.com/store/apps/details?id=com.instagram.android>>. Acesso em 16 de Maio 2014.

KOLLOCK, P. **Design Principles for Online Communities**, in: *The Internet and Society: Harvard Conference Proceedings*, O'Reilly and Associates, Cambridge, pp. 1997.

MASI, Breno. **A Batalha das Plataformas Mobile**. Slide Share. 25 de Setembro de 2011. Disponível em: <<http://pt.slideshare.net/MacMasi/a-batalha-das-plataformas>>. Acesso em 28 de Maio de 2014.

MINARELLI, José Augusto. **Networking: como utilizar a rede de relacionamentos na sua vida e na sua carreira**. São Paulo. Editora Gente: 2001.

NIKE. **Play Store**. 15 de Abril de 2014. Disponível em: <<https://play.google.com/store/apps/details?id=com.nike.plusgps>>. Acesso em 23 de Maio 2014.

OGLIARI, Ricardo da Silva; BRITO, Robison Cris. **Android - Do Básico ao Avançado**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2014.

PELADEIRO. **Peladeiro.com**. Disponível em: <<https://www.peladeiro.com.br/>> Acesso em 16 de Maio 2014.

PEREIRA, Sara; PEREIRA, Luis; PINTO, Manuel. **Internet e Redes Sociais**. Braga: Diário do Minho, 2011.

PERGUNTADOS. **Play Store**. 12 de Maio de 2014. Disponível em: <<https://play.google.com/store/apps/details?id=com.etermax.preguntados.lite>>. Acesso em 16 de Maio 2014.

PESQUISA: Brasil é o quarto país do mundo em número de smartphones. **Exame**. 29 de Maio de 2013. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/brasil-e-o-quarto-pais-do-mundo-em-numero-de-smartphones/>>. Acesso em 16 de Maio 2014.

PESQUISA: Facebook é a rede social mais utilizada no Brasil. **TERRA**. 17 de Julho de 2013. Disponível em: <<http://tecnologia.terra.com.br/internet/pesquisa-facebook-e-a-rede-social-mais-utilizada-no-brasil,5256aaa450def310VgnVCM4000009bcceb0aRCRD.html>>. Acesso em 13 de Abril 2014.

PLAY STORE, Aplicativos Gratuitos mais baixados. **Play Store**. Disponível em: <[https://play.google.com/store/apps/collection/topselling\\_free](https://play.google.com/store/apps/collection/topselling_free)>. Acesso em 16 de Maio 2014.

PREECE, J. **Online communities: designing usability, supporting sociability**. Chichester: Wiley, pp. 464, 2000.

RECUERO, Raquel. **Comunidades virtuais em redes sociais na internet: uma proposta de estudo**. Ecompos, Internet, v.4, Porto Alegre: Sulina, 2009. (Coleção Cibercultura).

STRAVA. **Strava**. Disponível em: <<http://www.strava.com/>> Acesso em 16 de Maio 2014.

SYMBIAN OS. **Wikipedia**. Dezembro de 2013. Disponível em: <[http://pt.wikipedia.org/wiki/Symbian\\_OS](http://pt.wikipedia.org/wiki/Symbian_OS)>. Acesso em 29 de Maio de 2014.

WHAT'S APP. **Play Store**. 06 de Maio de 2014. Disponível em: <<https://play.google.com/store/apps/details?id=com.whatsapp>>. Acesso em 16 de Maio 2014.

WAZLAWICK, Raul. **Análise e projeto de sistemas de informação orientados a objetos**. 2. ed. Rio de Janeiro: Elsevier, 2011.

WHO. **Issues in health development** Geneva: WHO/HPR/HEP/98.1., 1998.