

**FACULDADE DE TECNOLOGIA DA PARAIBA  
SISTEMAS PARA INTERNET  
ITAMAR GOMES DE LIMA**

**PROPOSTA DE UM PROTÓTIPO ANDROID PARA GERENCIAR  
CAMPEONATO DE SURF**

**Cabedelo/PB**

**2017**

**ITAMAR GOMES DE LIMA**

**PROPOSTA DE UM PROTÓTIPO ANDROID PARA GERENCIAR  
CAMPEONATO DE SURF**

Trabalho de conclusão do Curso de Sistemas para Internet, da Faculdade de Tecnologia da Paraíba, em cumprimento às exigências para a obtenção de grau de tecnólogo em sistemas para internet.

Orientador: Prof. Gláucio Bezerra Rocha

**Cabedelo/PB**

**2017**

## RESUMO

Os surfistas amadores realizam disputas de baterias sem ter nenhuma tecnologia que possa ser usada para aplicar as regras do esporte, como o registro das notas das ondas surfadas e o tempo que cada bateria terá. Diante deste contexto foi implementado um protótipo, destinado a gerenciar campeonato de surf amador. O mesmo será desenvolvido para ser usado nos dispositivos, que utiliza a tecnologia o sistema operacional Android. Para ser criado este aplicativo utilizará como linguagem de programação Java e para a criação do banco de dados foi usado o banco de dados SQLite. Já como ferramenta de programação foi usado o Android Studio que é uma IDE fornecida pela Google. Para ajudar a entender as funcionalidades foi criado caso de uso e diagrama de classe da aplicação e a ferramenta que foi usada é o Umbrello para criar tanto o caso de uso como o diagrama de classe. A aplicação terá imagens, que serão usadas como ícones da aplicação e imagem no botão, para criar estas imagens utilizou as seguintes ferramentas gráficas o Gimp e o Inkscape. O aplicativo tinha as seguintes funcionalidades cadastrar o nome do campeonato, inserir categoria no campeonato existente, inserir atleta na categoria escolhida, cadastrar atleta, selecionar atleta que participará da bateria, inserir nota da onda surfada, somar as notas da onda surfada, mostra quanto o atleta que esteja perdendo precisa tirar para ganhar a bateria, estabelecer o tempo da bateria ou seja terá um cronômetro que permitirá iniciar o tempo, pausar e zera o cronômetro. Esta aplicação utilizou fragments que proporciona para aplicação código limpo e reutilizável. Durante o desenvolvimento o projeto teve que ser reiniciado devido a falta de um bom planejamento da interface gráfica. Para o aplicativo poder evoluir mais se vê necessário implementar um webservice para que a já compartilhamento de informações, independente da tecnologia que esteja sendo usada.

**Palavras-chave:** Android. Surf. Fragments

## LISTA DE ILUSTRAÇÕES

Figura 1: Ferramenta de desenvolvimento .....	11
Figura 2: Área de trabalho do Umbrello .....	13
Figura 3: Área de trabalho GIMP .....	14
Figura 4: Tela inicial do Inkscape .....	15
Figura 5: elipse .....	18
Figura 6: Ator .....	18
Figura 7: Generalização .....	19
Figura 8: Include .....	19
Figura 9: Caso de uso da aplicação .....	20
Figura 10: Diagrama de classe da aplicação .....	21
Figura 11: Ciclo de vida da Activity .....	22
Figura 12: Tela inicial da aplicação .....	23
Figura 13: código do imagebutton .....	23
Figura 14: código do que exibirá o fragment .....	24
Figura 15: view do fragment criar competição .....	25
Figura 16: AlertDialog para selecionar categoria .....	26
Figura 17: view do fragmentListCampeo .....	27
Figura 18: view exibe categorias que irá inseri atleta .....	28
Figura 19: view do cadastro do atleta .....	29
Figura 20: view referente a lista de atletas inserido no campeonato .....	30
Figura 21: view exibindo um AlertDialog .....	31
Figura 22: view das categorias do campeonato .....	32
Figura 23: view da lista de atleta que poderá participar da bateria .....	33
Figura 24: view da bateria exibindo os três fragments .....	34

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	4
1.1 APRESENTAÇÃO DO PROBLEMA	7
1.2 OBJETIVO GERAL	7
1.3 OBJETIVOS ESPECÍFICOS	7
1.4 METODOLOGIA DA PESQUISA	7
1.5 JUSTIFICATIVA	8
<b>2 HISTÓRIA DO SURF</b>	9
<b>3 TECNOLOGIAS E FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO</b>	10
3.1 ANDROID	10
3.2 ANDROID STUDIO	10
3.3 JAVA	12
3.4 BANCO DE DADOS	12
3.5 UMBRELLO	13
3.6 FERRAMENTAS GRÁFICAS	13
<b>4 METODOLOGIAS ÁGEIS</b>	15
4.1 REQUISITOS DO PROGRAMA	16
4.2 REQUISITOS FUNCIONAIS	17
4.3 REQUISITOS NÃO FUNCIONAIS	17
<b>5 MODELAGEM DO SISTEMA EMPREGANDO UML</b>	17
5.1 DIAGRAMA DE CASO DE USO	18
5.2 DIAGRAMA DE CLASSE	20
<b>6 DESENVOLVIMENTO</b>	21
<b>7 CONCLUSÃO</b>	35
<b>REFERÊNCIA</b>	36

## 1 INTRODUÇÃO

Surf arte de deslizar sobre as ondas, o esporte que permite ao praticante se conectar com a natureza, possibilitando ao mesmo melhor qualidade de vida. O surf é um esporte muito antigo que começou a ser difundido pelo mundo pelo príncipe havaiano Duke Paoa Kahanamoku.

O surf surgiu no Brasil na década de 1940 no Rio de Janeiro, mas foi por volta de 1980 que o mercado do surf surgiu no país, incentivado pela conquista de uma etapa do mundial de surfe realizada no Rio de Janeiro. O primeiro título mundial do Brasil chega na década de noventa com a conquista do mundial amador de surf e o campeonato de divisão de acesso ao mundial de surf o World Qualifying Series (WQS).

Com mais de 7 mil Km de extensão o litoral brasileiro possui ótimas praias para a prática do surf. Estando entre os melhores países para a prática do esporte, recentemente o surf brasileiro vive o seu melhor momento da história, com a conquista do primeiro campeonato da liga mundial de surf World Surf League (WSL), com Gabriel Medina em 2014 e logo em seguida com Adriano de Souza, “O mineirinho”, em 2015.

A conquista de dois mundiais aumentou a procura pelo esporte proporcionando assim a ‘febre do surf’: praias lotadas por surfistas querendo aprender a surfar e realização de disputas amadoras entre esses praticantes do esporte. Diante desse contexto surgiu a necessidade de desenvolver um aplicativo que permite criar campeonato e incluir as categorias e os atletas os quais formaram as baterias e possibilitaram registrar as notas obtida por cada onda surfada e cronometrar o tempo de cada bateria.

No desenvolvimento foram usadas tecnologias como: Java que foi a linguagem de programação e SQLite como banco de dados. Outras ferramentas serão usadas: Android Studio, como IDE de desenvolvimento; Umbrello, que se destina a criar modelagem do sistema e para tratar as imagens serão usado o editores gráficos Gimp e Inkscape.

Para desenvolver um software com qualidade, rapidez e que atenda as necessidades do usuário se adotará o processo de metodologias ágeis Scrum. O Scrum é um framework destinado tanto a pequenos e grandes projetos além de ser

flexível e ágil que possibilita desenvolvimento incremental e iterativo e adapta-se constantemente às mudanças dos software (BISSI, 2007).

## 1.1 APRESENTAÇÃO DO PROBLEMA

Atletas praticantes do surf têm como prática realizar baterias para ver quem surfa melhor. Mas as baterias realizadas pelos desportistas não têm como aplicar os critérios de julgamento das ondas surfadas, assim como determinar quem está na frente ou seja ganhando a disputa e informar qual a nota do atleta, que está perdendo precisa tirar para passar a liderar a bateria. Estas normas são estabelecidas no Brasil pela CBS e mundialmente quem determina é a WSL.

## 1.2 OBJETIVO GERAL

Desenvolver um aplicativo para smartphone, destinado ao sistema operacional Android, para organizar as competições amadoras de Surf.

## 1.3 OBJETIVOS ESPECÍFICOS

- Realizar pesquisa bibliográfica sobre o tema
- Levantar requisitos funcionais e não funcionais do aplicativo
- Desenvolver diagrama de caso de uso e diagrama de classe
- Implementar formulário de manutenção de campeonato, bateria, atleta, categorias, regras da categorias, o vínculo do campeonato com categorias e notas dos atletas da baterias.

## 1.4 METODOLOGIA DA PESQUISA

A metodologia usada para o desenvolvimento da aplicação adotou-se técnica de coleta de dados ou seja documentação indireta. A qual destina-se a o uso de fonte de pesquisa como livros, sites, monografias, artigos e tcc.

Os livros aplicados na composição do software foram livros sobre programação de java, Android e livro sobre modelagem UML. Na construção do

banco de dados utilizou-se consulta a sites que falam sobre o banco de dados SQLite.

Vídeos sobre programação e campeonato de surf foram utilizados para esclarecer dúvidas relacionado a programação e entender de como as competição de surf funciona, assim como o cálculo das notas e regras estabelecidas nas disputas das baterias. Livros e sites sobre a história do surf serviram de fonte de consulta assim como o surgimento do mesmo no Brasil.

Estes materiais utilizados contribuem para que se desenvolva-se com qualidade, segurança e rapidez.

## 1.5 JUSTIFICATIVA

Observando a não existência de um aplicativo desenvolvido para plataforma Android, iniciou-se a construção desse software para smartphone, o qual se dispusera a gerenciar competições para surfistas amadores. Contribuindo para melhoria das disputas realizadas por estes atletas amadores nas praias brasileiras.

## 2 HISTÓRIA DO SURF

Evidências constata que o surf tenha surgido por volta de 1500 anos no triângulo polinésio situado no Oceano Pacífico. Existem historiadores que discordam, afirmando que o esporte citado originou-se no Peru. Mas há evidências que sustentam a primeira versão (SOUZA, 2013).

Em 1778 James Cook foi apresentado ao esporte começou a difundir pela Europa. O surf chegou a ser proibido por missionários europeus em 1821 por considerarem a atividade do esporte imoral e passou a ser coibido. A popularização do surf ocorreu um século depois com Duke Kahanamoku em 1912, quando ganhou uma medalha de ouro na natação nas olimpíadas de Estocolmo natação e apresentou ao mundo o surf (SOUZA, 2013).

A Califórnia foi o país que o surf se popularizou de tal forma que é conhecida por ter a maior população de surfista do globo. Isto se deve a Duke Kahanamoku que em 1915, realizou diversas exibições do esporte. Duke Kahanamoku foi batizado como pai do surf (SOUZA, 2013).

As primeiras pranchas de surf, chamadas de tábuas havaianas chegaram ao Brasil na Segunda Guerra Mundial, trazidas por turistas e soldados americanos, devido a uma base naval americana que encontrava-se instalada no Rio de Janeiro. Osmar foi considerado o primeiro surfista brasileiro, o mesmo com mais dois amigos conseguiram fazer uma prancha de surf que pesava 80kg. Logo surgiram as pranchas de vidro que chegaram ao Brasil em 1964 (SOUZA, 2013).

O Brasil é um país com um grande litoral com muitas ondas este cenário possibilitou ao surf o crescimento muito rápido. Na década de 1980 o esporte começou a lucrar com competições, o surf ganha mercado e logo surgiu as pranchas de fibra de vidro, mais leves e ágeis nas ondas proporcionando ao surfista realizar manobras mais ousadas, com isto surgiu uma geração de surfista nas praias(SOUZA, 2013).

O esporte que antes era marginalizado agora passa a ser visto pela sociedade como uma atividade séria e profissional. Logo é realizado o primeiro circuito brasileiro de surf profissional e na mesma época é fundada a Associação Brasileira de Surf (ABRASP) (GUTEMBERG,1989) apud (PEREIRA & ARMBRUST, 2010). Atualmente o surf brasileiro está entre as potências mundiais do surf.

### 3 TECNOLOGIAS E FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO

As tecnologias e ferramentas aplicada na construção do programa. As tecnologias operadas será o sistema operacional Android, linguagem de programação Java e banco de dados SQLite, e as ferramentas empregada será o Android Studio como IDE de desenvolvimento, na modelagem do sistema será empregado o Umbrello e o Gimp e o Inkscape como ferramentas gráficas..

#### 3.1 ANDROID

Em 2005 a Google adquiriu a empresa Android Inc. Empresa responsável por criar inicialmente a plataforma Android. A Open Handset Alliance, é um conjunto de empresa, gigantes do mercado mobile que ajudaram na criação da plataforma. O motivo pelo qual levou a plataforma Android obter um grande sucesso no mercado de dispositivos móveis se deve a o mesmo ser de código aberto, fornecendo acesso a todos os recursos do hardware e software aos aplicativos desenvolvido por outras empresas além de disponibilizar design moderno (OGLIARI ; BRITO, 2014). “O sistema operacional Android tem como base o kernel do linux, que é responsável pelo gerenciamento de processos, drives, memória e energia” (GLAUBER, 2015, p.17).

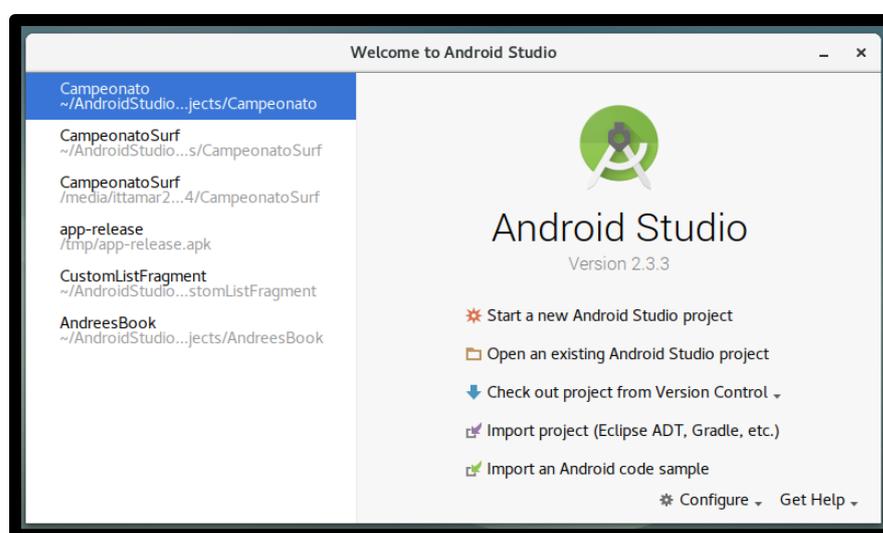
Um diferencial do Android é ser a primeira plataforma de código a berto do mercado mobile. Isto possibilita outros fabricantes usarem o sistema operacional em seus produtos, sem terem de pagar por isso além de permitir a customização do código sem que seja obrigado a compartilhar estas mudanças realizadas no código. Essa idéia permite atrair pequenas empresa que não dispõe de muitos recurso (LECHETA, 2015). “O Android é código aberto e distribuído sob licença Apache 2.0, o que quer dizer que você tem acesso aos códigos-fonte e também pode contribuir como projeto” (MONTEIRO, 2012, p.5).

#### 3.2 ANDROID STUDIO

Para realizar a criação do aplicativo será utilizado uma IDE, a qual servirá para melhora a produtividade na criação do aplicativo. A IDE será o Android Studio que oferece diversos recursos além de disponibilizar de um Editor visual arraste e

solte, sistema moderno baseado em Gradle, Diversos templates. Tem uma ótima integração com SDK e sofre atualizações com frequência. “O Android Studio foi anunciado no Google I/O 2013 e é baseado no IntelliJ IDEA da JetBrains” (LECHETA, 2015, p.44). Existe a possibilidade de se utiliza o Eclipse para desenvolver para a plataforma Android. Mas o Android Studio é a IDE oficial para desenvolvimento para Android (LECHETA, 2015). Na figura 1, está página de boas vindas do Android Studio

Figura 1: Ferramenta de desenvolvimento



Fonte: Próprio Autor, 2017

“SDK é o software utilizado para desenvolver aplicações no Android, que tem um emulador para simular o dispositivo, ferramentas utilitárias e uma API completa para a linguagem Java, com todas as classes necessárias para desenvolver as aplicações”(LECHETA, 2015, p.42). Os Requisito do Sistema Utilizado no desenvolvimento são:

- Debian GNU/Linux jessie 64-bit
- 8 GB memória RAM
- Java 1.7
- Processador Intel Core i3 4 geração

Estas configurações permitirão trabalhar de forma tranquila sem ter problema com o desempenho do pc, já que o emulador do Android Studio requer bastante desempenho da máquina. O sistema operacional utilizado para trabalhar na criação da aplicação será o GNU/Debian 8.

### 3.3 JAVA

Esta linguagem surgiu no mercado em 1995, desenvolvida pela Sun Microsystems. O criador desta linguagem chama-se James Gosling, o primeiro nome de batismo da linguagem foi Oak, mas como já existia um código, nomeado com este nome teria que colocar outro nome. Mas um certo dia uma equipe da Sun estava em uma cafeteria e observou o nome de origem de um café importado que vinha de uma cidade chamada Java daí surgiu o nome de uns do sistema mais operado no mundo(DEITEL, h. ; DEITEL, p., 2010).

Java é a linguagem empregada na programação dos aplicativos android. Por ser umas das linguagem mais utilizadas no mundo. Além de código aberto, gratuita, orientada a objetos e ser conhecida pela comunidade da programação(DEITEL, A.; DEITEL, h. ; DEITEL, p., 2015).

### 3.4 BANCO DE DADOS

Com os usuários armazenando cada vez mais dados nos smartphone. Os fabricantes dos aparelhos começaram a proporcionar meios de armazenamento. Então a plataforma Android desenvolveu seus métodos de armazenamento local (OGLIARI ; BRITO, 2014).

As formas de persistência de dados empregue na plataforma Android são Sqlite,SharedPreferences, em forma de arquivo mas existem outros métodos de persistência, mas para este projeto serão utilizado as formas citadas anteriormente.

O SQLite é um servidor que já vem instalado nos smartphone com sistema operacional Android. O SQLite permite transações, sintaxe sql e possui recursos de banco de dados relacional, além de ser de código aberto. (Cordeiro, Guardando Dados com SQLite, 2014).

O SharedPreferences tem sua capacidade de armazenamento muito limitada. A forma que o SharedPreferences usa para guardar os dados é chave e valor, o seu uso é mais empregado para armazenar login(LECHETA, 2015).

O banco utilizado na persistência dos dados da aplicação será o SQLite. por ser um banco de dados leve e integrado com a plataforma Android. Uma das características desse banco é permitir a finalização de uma operação realizada,

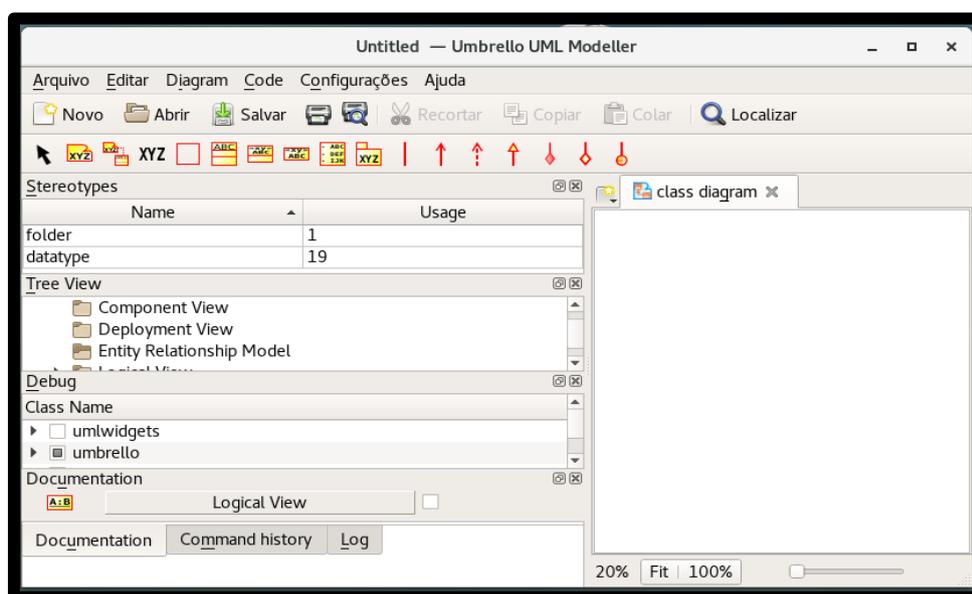
apenas se não existe operações pendentes, caso existam ou alguma venha a falhar poderá ser desfeito as outras operações já realizada(GLAUBER, 2015).

As característica marcantes no SQLite são suporte a transações, ou seja não depender totalmente do sistema operacional, é um banco interno além de ser usado localmente sem necessidade de se comunicar via protocolo TCP/IP (GLAUBER, 2015).

### 3.5 UMBRELLO

É uma ferramenta livre ou seja é de graça o usuário não tem custo algum no uso dessa ferramenta, faz parte do projeto KDE e foi inicialmente batizado de UML Modeller. Quem teve a ideia inicial desse projeto foi Paul Hensgen, mas atualmente este projeto é mantido por um grupo de programadores espalhado pelo mundo (SANTOS, CINDRA, & SOUZA, 2009). Exemplo da ferramenta na figura 2.

Figura 2: Área de trabalho do Umbrello



Fonte: Próprio Autor, 2017

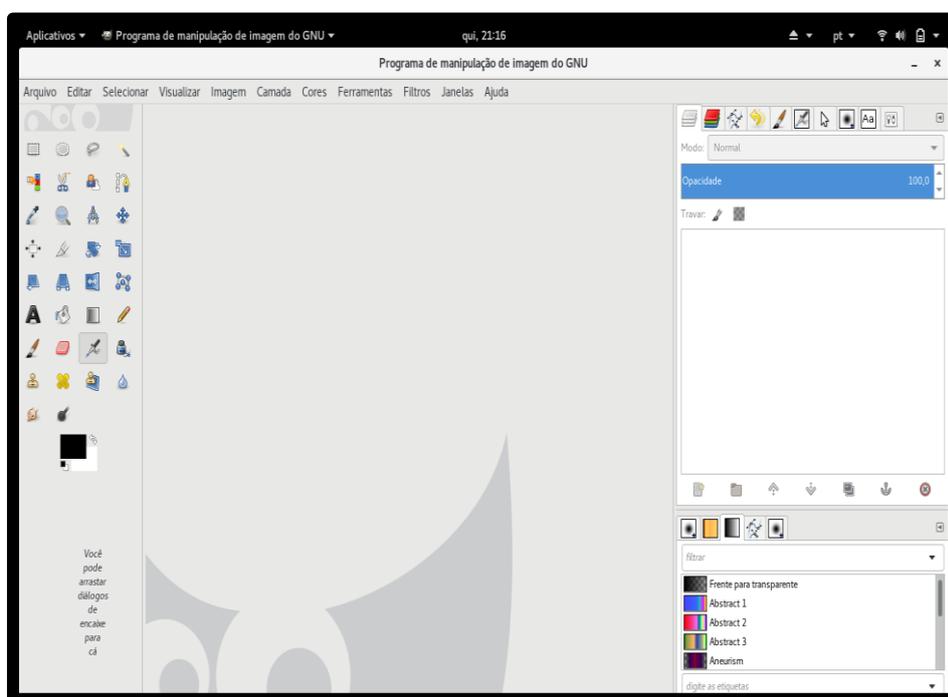
Esta ferramenta destina-se a modelagem de sistema possibilitando a criação de diagramas de caso de uso, classe, sequência e atividade. Esta ferramenta possibilita a conversão do diagrama em código Java, Python e JavaScript. Oferece a funcionalidade de exportar arquivos no padrão XML e consegue oferecer engenharia reversa de classe (CINDRA, BARCELOS, & LISBÔA, 2011).

### 3.6 FERRAMENTAS GRÁFICAS

As ferramentas gráficas usadas serão o GIMP e o Inkscape. O GIMP foi criado em 1985 por Spencer Kimball e Peter Mattis e atualmente é mantido por um grupo de pesquisadores universitários, a palavra GIMP quer dizer “General Public License e Image Manipulation Program”. É uma ferramenta multiplataforma e de código aberto ou seja pode ser utilizada de graça, é uma ótima ferramenta para edição de imagem, esta ferramenta será usado para criar os ícones e background da aplicação (SILVA, 2013).

Esta ferramenta apresentou-se ser muito útil no desenvolvimento por ser muito fácil de usar, e conter muito conteúdo na internet sobre sua utilização. A interface gráfica do GIMP é formada por caixa de ferramentas e janela de imagem, existe outras caixa de diálogo que pode ser exibida conforme a necessidade do usuário, na figura 3 está a área de trabalho do GIMP.

Figura 3: Área de trabalho GIMP



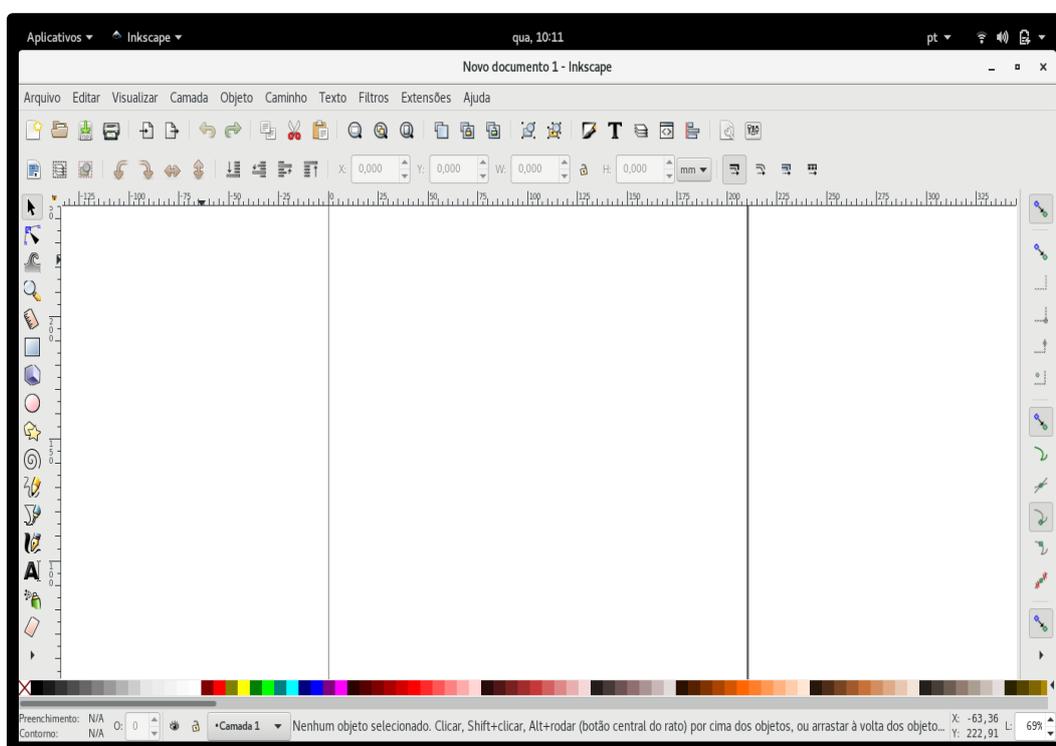
Fonte: Próprio Autor, 2017

Já o Inkscape é um software livre de edição gráfica ou seja não paga nada para usar, tem o código a berto o qual pode ser modificado ou estudado. O mesmo

surgiu de um projeto chamado de Sodipodi e teve como idealizadores Mentalguy, Nathan Hurst, Bryce Harrington e Ted Gould.

O mesmo possui muitas funcionalidades disponíveis. Esta ferramenta de desenho vetorial possui funcionalidades semelhantes ao Illustrator, CorelDraw, Xara X, entre outras aplicações do mercado na figura 4 é exibido a tela inicial do Inkscape (NOVAIS, 2012). Apresentou muito intuitivo e existe muito material na internet ensinando como usar lá, podendo ser usada tanto por usuários experientes e iniciantes

Figura 4: Tela inicial do Inkscape



Fonte: Próprio Autor, 2017

Estas duas ferramentas possuem um desempenho muito satisfatório quando usadas para manipular imagens. Estas duas ferramentas serão usadas em conjunto em conjunto para se chegar ao produto final. Lembrando que não se paga nada para usá-las ou seja proporciona economia ao desenvolver um projeto, possibilitando realizar orçamento mais atraente para possíveis clientes. Mas estes softwares sofriam ainda com falta de conhecimento dos usuários que preferiam usar softwares piratas.

## 4 METODOLOGIAS ÁGEIS

A produção de software é auxiliada por conjunto de atividades que caracteriza a metodologia de desenvolvimento de software. A condução do processo reflete a forma de atividade que resulta em um produto (SOMMERVILLE, 2003) apud (KOSCIANKI & SOARES, 2006).

O desenvolvimento ágil é de suma importância para o produto, em desenvolvimento, devido a constante mudança do mercado. Se na criação do software o mesmo demora a ser lançado poderá a vir a ser cancelado antes de estarem completos e terem seus custos aumentados, por não atender mais as necessidades do cliente além de estar obsoleto. Por isto se dá a utilização do Scrum, um método para agilizar o desenvolvimento, existem outros como XP e Kanban, mas para este projeto o escolhido foi o Scrum (SOMMERVILLE, 2011).

Scrum é uma ferramenta que pode ser utilizada por pequenas, médias e grandes equipes de desenvolvimento. Esta ferramenta concede ganhos na elaboração dos programas como redução de riscos, aumento de produtividade, entrega de pequenas partes do software ao cliente e visibilidade do trabalho, podendo todos verem o que já foi concluído e o que falta (SABBAGH, 2013).

Apesar de ser apenas um desenvolvedor na construção deste trabalho esta ferramenta, contribui para agilizar o desenvolvimento. O mesmo irá representar todos os papéis exigidos pelo Scrum que são o Scrum Master, Product Owner e equipe.

### 4.1 REQUISITOS DO PROGRAMA

Os requisitos do programa permitem que tanto desenvolvedores como usuários tenham a mesma visão do problema a ser resolvido ou seja é o desenvolvimento do software aplicado a compreensão do problema (BEZERRA, 2007).

“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento” (SOMMERVILLE, 2013, p.57). Quando o escopo do software for determinado então os requisitos serão estabelecidos. Quanto maior for o entendimento dos requisitos do sistema solicitado,

melhor será o resultado final do produto. “Os requisitos são as características que definem os critérios de aceitação de um produto. A engenharia tem por objetivo colocar nos produtos as características que são requisitos” (FILHO, 2000, p. 13).

#### 4.2 REQUISITOS FUNCIONAIS

“São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações”(SOMMERVILLE, 2013, p.59). Muitos problemas na engenharia de software é causado pela imprecisão na especificação dos requisitos (SOMMERVILLE, 2011).

Os requisitos funcionais do aplicativo a ser desenvolvido são: realizar cadastro de campeonato, inseri categorias no campeonato selecionado, inseri atleta na categoria selecionada, selecionar atleta que participará da bateria, acessar informação sobre a faixa etária da categoria, insere background, desenvolver ícone do aplicativo e calcular as notas das ondas surfadas.

#### 4.3 REQUISITOS NÃO FUNCIONAIS

Estes requisitos estabelece a qualidade do software como usabilidade, performace, segurança e estabilidade da aplicação. As funções que não são especificada pelo sistema dizem respeito aos requisitos não funcionais (SOMMERVILLE, 2003).

“São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas” (SOMMERVILLE, 2013, p.59).

Os requisitos não funcionais tornam-se mais importantes que os requisitos funcionais devido a alguns requisitos representar o sistema como um todo e não características individuais do sistema. Se algum requisito não funcional não for executado o sistema corre o risco de se tornar completamente inútil, já se deixar de cumprir com o requisito funcional individual poderá desgastar o sistema (SOMMERVILLE, 2003)

## 5 MODELAGEM DO SISTEMA EMPREGANDO UML

No desenvolvimento desse sistema, será aplicado a linguagem de modelagem UML. Esta linguagem de modelar facilita o mapeamento de seus modelos para a linguagem de programação. Propicia a geração de código a partir do UML criado e cria a documentação (MELO, 2010).

### 5.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso é utilizado para exibir ao usuário final as funcionalidades do sistema desenvolvido e suas especificações de requisitos. O mesmo é representado por uma elipse exemplo na figura 5 logo abaixo. A forma de relação do caso de uso se dá através de três formas que são: generalização, inclusão e extensão (SILVA e VIDEIRA, 2001).

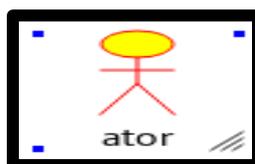
Figura 5: elipse



Fonte: Próprio Autor, 2017

“Atores são entidades externas que interagem com o sistema durante a execução de um caso de uso. Normalmente são pessoas, mas podem também ser um hardware ou outro sistema” (SEABRA, 2013, p.80). Exemplo na figura 6.

Figura 6: Ator



Fonte: Próprio Autor, 2017

Generalização tem sua representação gráfica em forma de seta. Determina que o filho herde o comportamento e semântica do pai. Mas o filho pode introduzir

nos comportamento que não são pertinente ao pai (SILVA, 2001; VIDEIRA, 2001). Exemplo na figura 7.

Figura 7: Generalização



Fonte: Próprio Autor, 2017

Inclusão “ É representado por uma relação de dependência (seta tracejada) com o estereótipo <<include>>” (SILVA, 2001, p.149; VIDEIRA, 2001, p.149). Disponibiliza comportamentos pertinentes a outro caso de uso. O caso é classificado como inclusor e incluído. O inclusor inclui o comportamento ; O incluso tem seu comportamento determinado por outros (BEZERRA, 2007).

Extensão “Estabelece uma relação em que um dos casos de uso tem seu comportamento estendido através do comportamento definido em outro caso” (SILVA, 2007, p.108), na figura 8 esta o exemplo.

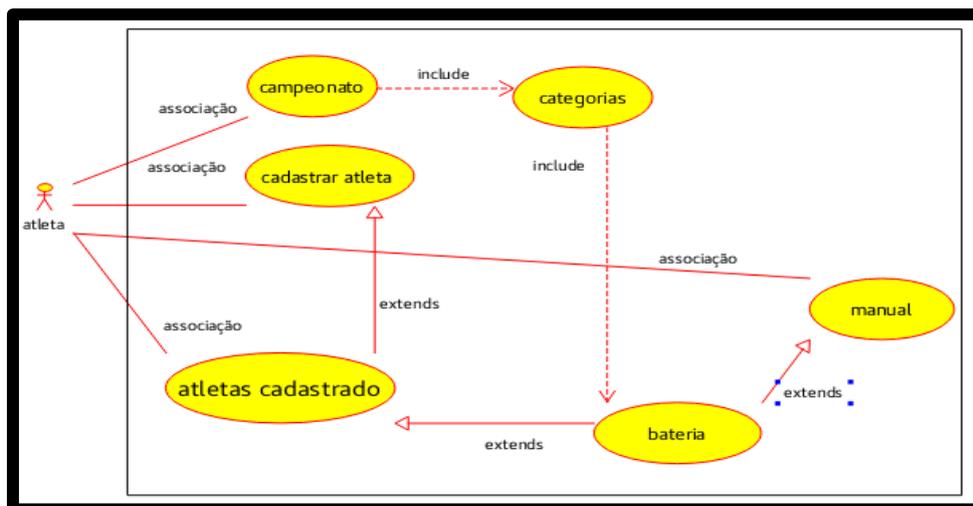
Figura 8: Include



Fonte: Próprio Autor, 2017

O atleta, criará campeonato em seguida incluirá as categorias determinada pelo mesmo, em seguida determinará as baterias das categorias. A quantidade das baterias irá variar em relação a quantidade de participantes inserido na categoria. No relacionamento cadastrar atletas os mesmo serão inserido. O atleta poderá visualizar os atletas cadastrado mesmo antes de criar campeonato caso já estejam cadastrado. Os atletas cadastrado poderão ser inserido na bateria conforme desejado. Em manual o usuário poderá consultar as regras empregada no julgamento das ondas. Este manual está conforme as regras confederação brasileira de surf (CBS). Exemplo do diagrama de caso de uso da aplicação na figura 9.

Figura 9: Caso de uso da aplicação



Fonte: Próprio Autor, 2017

O diagrama de caso de uso possibilita melhor entendimento de como o usuário irá interagir com a aplicação.

## 5.2 DIAGRAMA DE CLASSE

Este diagrama possibilita ter uma visão do software a nível de programação. “O diagrama de classes é utilizado na construção do modelo de classes desde o nível de análise até o nível de especificação” (BEZERRA, 2007, p. 112). O diagrama de classe é usado para representar os elementos de um programa orientado a objeto de forma mais clara e é um dos treze modelos gráficos da linguagem UML. O mesmo permite modelar os métodos e atributos da classe que está sendo modelada (SILVA, 2007).

A classe é representada por uma “caixa” que possui até três compartimentos. O primeiro compartimento é inserido o nome da classe, este nome tem que ser no singular e tem sua letra inicial em maiúscula. No segundo compartimento é inserido os atributos que são as características do objeto. No terceiro são inseridos os métodos do objeto que serão as ações do mesmo (BEZERRA, 2007). “Um diagrama de classes ilustra um conjunto de classes, interfaces, colaborações e respectivas relações, em geral de dependência, generalização e de associação” (SILVA e VIDEIRA, 2001, p. 186).

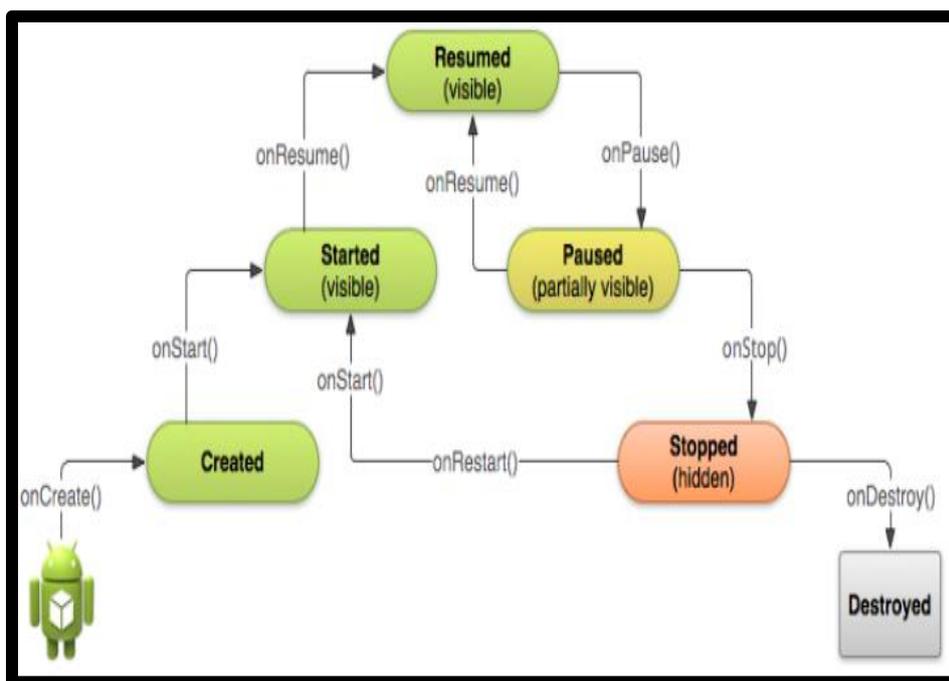
O diagrama de classe da aplicação será exibido na figura 10, o qual possui o nome das classes existentes na aplicação seus métodos e atributos, assim como os



## 6 DESENVOLVIMENTO

A aplicação utiliza de diversos componentes do Android como a activity. Este componente é uma interface gráfica que permite ao usuário interagir com a aplicação. Um conjunto de funcionalidade representa diversas activity (MONTEIRO, 2012). “A Activity é um componente de aplicação com um ciclo de vida específico. Quando o usuário acessa a aplicação, navega pelas opções, sai ou retorna para a mesma, as atividades que a compõem passam por uma série de estados do ciclo de vida” (MONTEIRO, 2012, p. 47). Activity “É responsável por controlar os eventos da tela e definir qual View será responsável por desenhar a interface gráfica do usuário” (LECHETA, 2015, p. 96). Exemplo do ciclo de vida da activity na figura 11.

Figura 11: Ciclo de vida da Activity

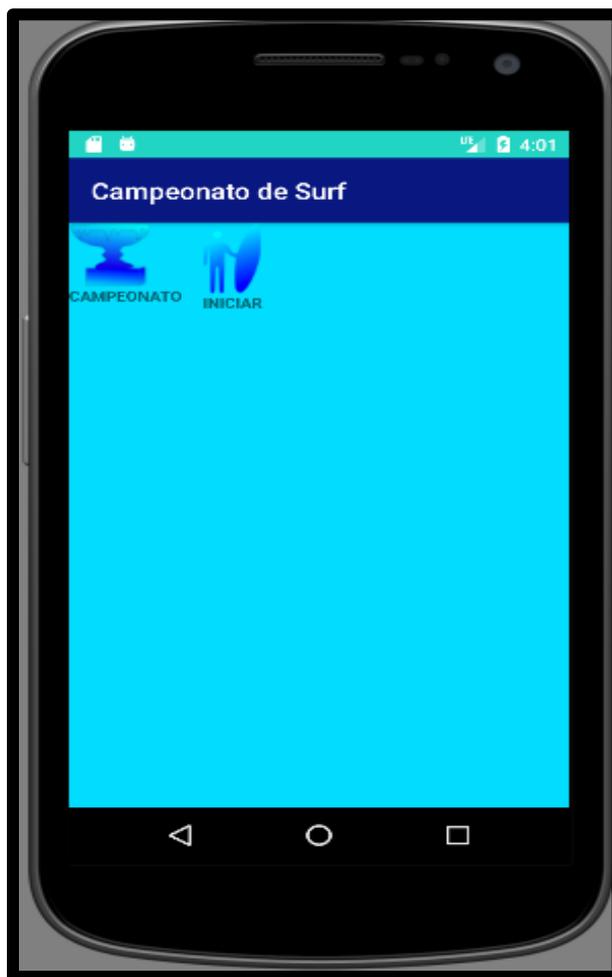


Fonte: (Monteiro, 2012, p. 48)

“Activity é uma classe que herda da classe android.app.Activity ou de alguma subclasse desta, a qual representa uma tela da aplicação e é responsável por tratar os eventos gerados nessa tela” (LECHETA, 2015, p. 96).

Activity main da aplicação a apresenta dois ImageButton que permite usar uma imagem na tela para desenhar um botão e dois TextView que é usado para exibir um texto exemplo na figura 12.

Figura 12: Tela inicial da aplicação



Fonte: Próprio Autor, 2017

O código do ImageButton localiza-se na activity. O ImageButton que tem um ícone de um troféu chamará a activity campeonato e se for clicado o ImageButton que tem um ícone de um surfista acionará ou chamará activity iniciar campeonato exemplo do código na figura 13.

Figura 13: código do ImageButton

```
ImageButton = (ImageButton) findViewById(R.id.image3720);  
ImageButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent intent = new Intent(MainActivity.this, ActivityCampeonato.class);  
        startActivity(intent);  
    }  
});
```

Fonte: Próprio Autor, 2017

Na Activity campeonato a mesma usa o recurso fragments. “Fragments são componentes modulares de interface gráfica, com um ciclo de vida próprio, criados para facilitar o desenvolvimento de aplicativos com layouts ajustáveis a diferentes tamanho de tela” (MONTEIRO, 2012, p. 261).

A activity campeonato gerencia os fragments, adicionando e removendo em tempo de execução. Na figura 14’ está o código de criação do fragments da activity campeonato.

Figura 14: código do que exibirá o fragment

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_campeonato);

    if (savedInstanceState == null) {
        FragmentCriarCompeticao fcc2 = new FragmentCriarCompeticao();
        FragmentManager manager = getSupportFragmentManager();
        FragmentTransaction ft = manager.beginTransaction();
        ft.replace(R.id.frameFormCriaCompet, fcc2);
        // ft.addToBackStack(null);
        ft.commit();
    }
}
```

Fonte: Próprio Autor, 2017

Quando a activity campeonato for chamada a view do fragments será exibido exemplo da tela na figura 15. A view do fragment, que é responsável por criar o campeonato, permitirá ao usuário inserir o nome da competição, o local que irá acontecer a competição, selecionar a data, a categoria que será inserida no campeonato criado, além de poder inserir os atletas na categoria desejada.

Para melhorar a interação do usuário com o aplicativo será implementado um menu, que terá como funcionalidade permitir ao usuário ao clicar no menu retorne para activity inicial, este menu será representado por três pontinhos na activity campeonato, ao clicar nos mesmos é exibido a opção sair. “Os arquivos de menu devem ficar dentro do diretório res/menu” (MONTEIRO, 2012, p. 104).

Figura 15: view do fragment criar competição



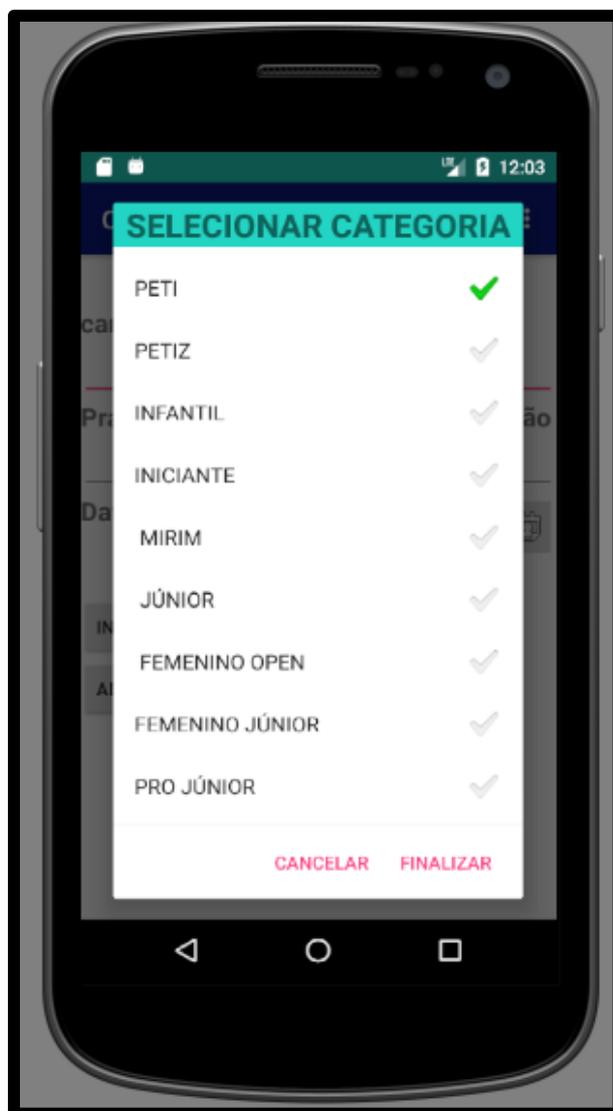
Fonte: Próprio Autor, 2017

A figura 15 possui TextView, EditText, Button e ImageButton. O fragment possui seus próprios comportamentos e gerencia os mesmo ou seja o comportamento do Button e ImageButton será gerenciado pelo fragments, criar competição.

Pensando na interação do usuário com a aplicação será implementado um alertDialog no botão inserir categoria exemplo do alertDialog na figura 16. Possibilitando ao utente interagir com a aplicação. Este alertDialog possibilitará, usuário selecionar a categoria que o mesmo deseja que tenha no campeonato. Para finalizar e fechar o alertDialog será necessário clicar na opção finalizar. Se escolher clicar em cancelar o alertDialog será fechado, mas sem inseri nenhuma categoria no

campeonato. “Quando o alertDialog está sendo executado, o usuário não pode interagir com o aplicativo” (DEITEL, DEITEL e DEITEL, 2015, p. 100).

Figura 16: AlertDialog para selecionar categoria



Fonte: Próprio Autor, 2017

Na view do fragmentListCampeo o mesmo exibe um ListView com adapter customizado. que possibilita ao desenvolvedor criar lista de elementos conforme sua necessidade. O ListView é um dos componentes disponibilizado pelo o Android que permite exibir uma lista de informações (GLAUBER, 2015). Para usar um adapter customizado, deve-se criar uma classe que herde de BaseAdapter (GLAUBER, 2015).

A lista customizada que será exibida terá o nome do campeonato, praia onde será realizada a competição e a data do evento. Esta lista ao ser clicada exibe um

AlertDialog com as seguintes opções selecionar categoria e excluir campeonato. Na opção selecionar categoria será exibido uma lista de categoria existente no campeonato. Já na opção excluir campeonato o mesmo será excluído do banco de dados e não será mais exibido na lista de campeonatos criados, exemplo na figura 17.

Figura 17: view do fragmentListCampeo



Fonte: Próprio Autor, 2017

Quando o usuário clicar em selecionar categoria o mesmo será direcionado para o fragmento, que será responsável por exibir as categorias existentes no campeonato selecionado anteriormente, este fragmento utiliza de ListView customizada, o qual exibirá apenas o nome da categoria. Esta lista possui um alertDialog que será exibido, quando uma categoria for clicada. Este alertDialog

possui as seguintes opções inseri atleta, exhibe atletas inserido, excluir atletas, botão não e sim. Quando o usuário clicar em inseri atleta, exibira uma activity que solicitará os dados do participante. Já se a opção escolhida for exhibi atletas inserido, exibirá uma ListView com os taletas inserido na categoria clicada. Mas se o utente clicar em excluir será excluido a categoria selecionada. Ao clicar no botão não o alertDialog será fechado e o botão sim está sem funcionalidade, exemplo na figura 18.

Figura 18: view exhibe categorias que irá inseri atleta



Fonte: Próprio Autor, 2017

Na activity responsável por cadastrar os atleta formulário onde o usuário irá inserir os dados do atleta que participará do campeonato está activity possui em sua view dois EditText, dois RadioButton e um Button. Os EditText servirão para inserir o nome e idade, já no RadioButton será selecionado o estilo do surfista e o Button

salvar os dados do atleta no banco de dados. Na figura 19 é exibido a view dessa activity.

Figura 19: view do cadastro do atleta



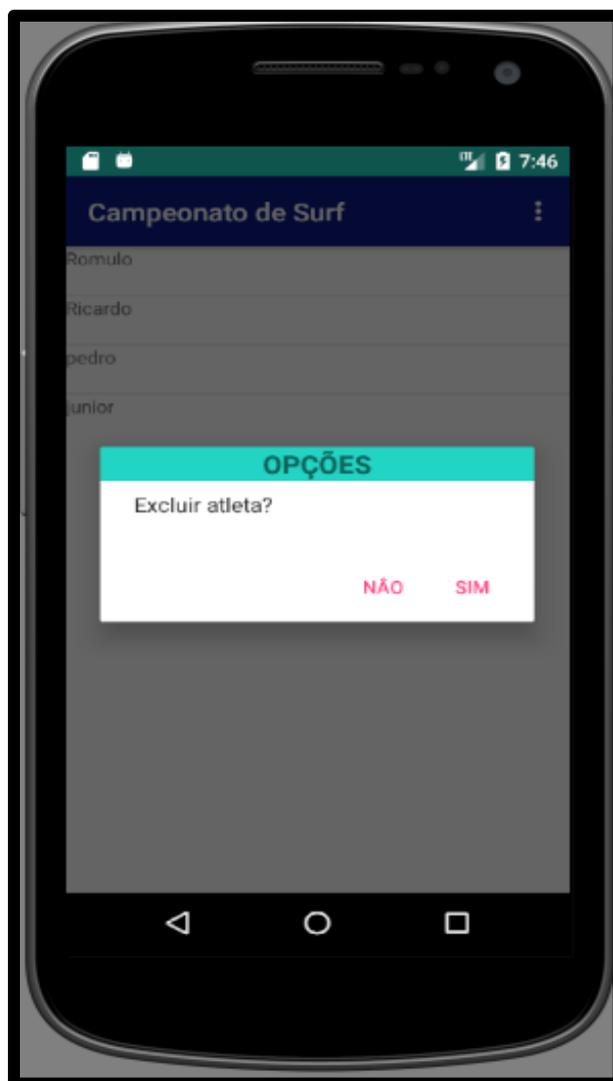
Fonte: Próprio Autor, 2017

O fragmento que exibe a lista de atleta, cadastrado no campeonato, possui uma ListView. A ListView disponibiliza um método que trata o click, que é `onItemClickListener`, para implementar este método, basta passar um objeto que implemente a interface `OnItemClickListener`, quando o item da lista for clicado esse método será chamado (GLAUBER, 2015).

Quando o `onItemClickListener`, detectar que algum item da lista foi clicado será exibido um `AlertDialog` que mostrará uma mensagem, que ficará esperando a confirmação do cliente, caso seja escolhido sim, o atleta será excluído do banco de dados caso

seja escolhido não, o atleta não será excluído do banco de dados eo alertDialog será fechado. Exemplo dessa view na figura 20.

Figura 20: view referente a lista de atletas inserido no campeonato



Fonte: Próprio Autor, 2017

Para iniciar o campeonato o usuário terá que acionar o `ImageButton`, que tem como imagem um `person` segurando a prancha de surf. Pressionado será exibido uma `activity` que terá mais de um `fragment`. Os `fragments` que será exibido na `activity` serão `ListaDeCampeonato`, `FragmentoCategoriasInserida` e `FragmentListaAtleta`.

No `fragment ListaDeCampeonato` o mesmo compartilha a mesma `view` com o `fragmentListCampeo` ou seja o mesmo `layout` possibilitando assim o reuso. Mas as ações que os `fragmentos` executa são diferentes, este `fragment` disponibiliza um `AlertDialog` que exibe uma mensagem de texto, se for respondido `sim`, é chamado

outro fragment e caso seja escolhido não, permanece na activity atual exemplo na figura 21.

Figura 21: view exibindo um AlertDialog



Fonte: Próprio Autor, 2017

No FragmentoCategoriasInserida, será exibido uma lista de categorias existente no campeonato selecionado. O mesmo ocorre com este fragment o reaproveitamento da view ou seja uma view para dois fragments. O fragment que compartilhara com este e o fragment FragmentListarCategorialInseridaCampeo. Apesar de compartilharem a mesma view mas tem funcionalidades diferentes como o AlertDialog que exibe uma mensagem e fica esperando uma confirmação do cliente se confirmar com sim, vai para uma lista de atletas inserida na categoria.

Caso pressione em não ficará na mesma view que se encontra, exemplo na figura 22.

Figura 22: view das categorias do campeonato

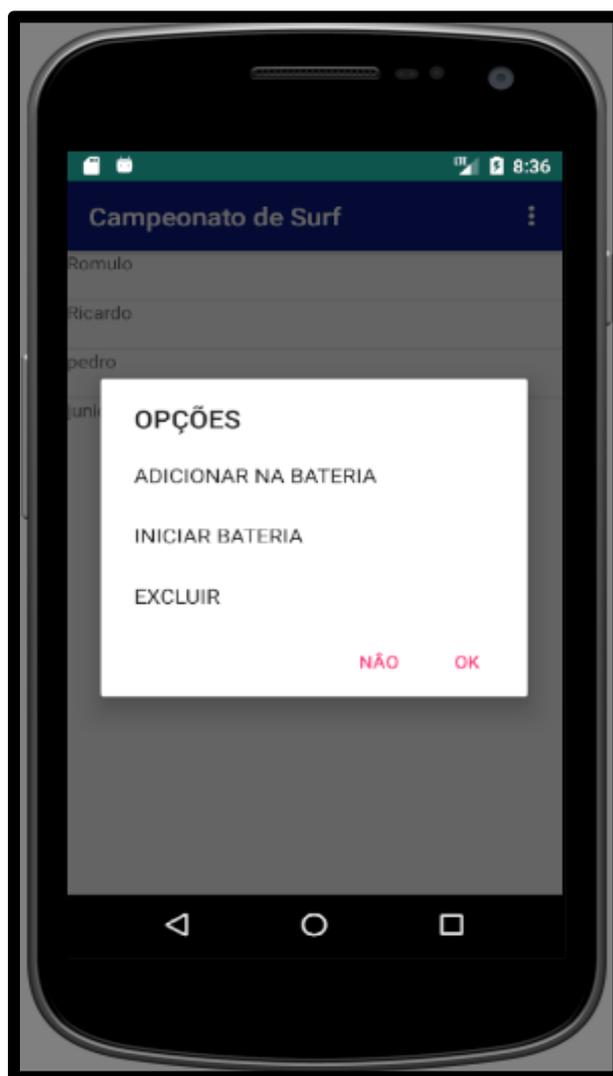


Fonte: Próprio Autor, 2017

No fragment que exibe os atletas da categoria selecionada, acontece o mesmo procedimento reaproveitamento de view ou seja uma view que está sendo usado por dois fragments, mas funcionalidades distintas. Este fragment disponibiliza de um AlertDialog que exibe uma lista com as seguintes opções adicionar na bateria, iniciar bateria e excluir. Na opção adicionar na bateria, será adicionado o atleta na bateria, a medida que for inserindo o atleta será exibido um Toast, informado que o atleta foi inserido na bateria, caso queira inseri mais de quatro participante na bateria, é exibido um Toast, informando que a bateria está completa. Quando o

usuário selecionar a opção iniciar bateria, o mesmo só sera redirecionado para a activityBateria, se a quantidade de atletas for maior que dois. Já na opção excluir, permitir excluir o participante do banco de dados. Ao clicar em ok, o alertDialog será fechado, o mesmo acontece na opção não. Exemplo na figura 23.

Figura 23: view da lista de atleta que poderá participar da bateria



Fonte: Próprio Autor, 2017

Na activityBateria figura 24 utilizará os seguintes fragmentos ao mesmo tempo ou seja os fragmentos dividiram a activityBateria, mas as funcionalidades serão independente, cada fragmento controlará suas funcionalidades.

Os fragments usados serão BateriaFragment, exibirá uma lista customizada de atletas. Quando o atleta desta lista for clicado exibirá um AlertDialog que solicitará que seja inserido a nota da onda surfada por este a atleta. Já no fragment

NotaFragment, o mesmo pegar o valor inserido da nota e exibirá este valor, os valores exibido serão a maior nota, segunda maior nota, a soma dessas notas e a diferença em relação às notas dos outros atletas. O mesmo se aplica para os fragments NotaFragment2,

Figura 24: view da bateria exibindo os três fragments



Fonte: Próprio Autor, 2017

NotaFragment3, NotaFragment4. O fragment FragmentCronometro exibirá o cronômetro que servirá para o utente controlar o tem que terá a bateria este fragments terá três botão iniciar, pausar e parar.

## 7 CONCLUSÃO

Ao término dessa pesquisa pode observar que o projeto possibilitará ao usuário criar o campeonato assim como inserir as categorias e os atletas e gerenciar o tempo assim como atribuir as notas das ondas.

Foram encontradas algumas dificuldades durante o desenvolvimento da aplicação, pois a falta de planejamento no início do projeto das funcionalidades assim como a organização do código levou a um retrabalho. Durante a implementação notou-se de como comentar o código ajuda a resolver bugs.

Por fim este aplicativo desenvolvido poderá ser aprimorado com a integração de um webservice ou a utilização da plataforma Firebase da google.

## REFERÊNCIAS

- ALEXANDRINO, R. WQS: Adriano de Souza é vice na Califórnia. **blogs.oglobo.globo.com**, 27 mar. 2011. Disponível em: <<http://blogs.oglobo.globo.com/radicais/post/wqs-adriano-de-souza-vice-na-california-371340.html>>. Acesso em: 17 set. 2017.
- ALVES, M. **Como Escrever Teses e Monografias Um roteiro passo a passo**. Rio de Janeiro: Elsevier, 2007.
- BEZERRA, E. **PRINCÍPIOS DE ANÁLISE E PROJETO DE SISTEMAS COM UML**. 2. ed. Rio de Janeiro: Editora Campus/Elsevier, 2007.
- BISSI, W. **SCRUM - METODOLOGIA DE DESENVOLVIMENTO ÁGIL**. *BVS-Vet*, 1-4, 2007.
- CINDRA, J. D. S.; BARCELOS, M. R. D. S.; LISBÔA, J. C. UMA PESQUISA SOBRE FERRAMENTAS CASE PARA ENGENHARIA REVERSA ESTÁTICA. **PERSPECTIVASonline**, Blumenau, 1, jun. 2011.
- CORDEIRO, F. (17 de 07 de 2014). **Guardando Dados com SQLite**. Disponível em: <<http://www.androidpro.com.br/sqlite/>>. Acesso em 06 de 09 de 2017.
- CORDEIRO, F. Android Aprendiz. **androidpro**, 2017. Disponível em: <[materiais.androidpro.com.br/android-aprendiz](http://materiais.androidpro.com.br/android-aprendiz)>. Acesso em: 06 set. 2017.
- Desconhecido. (nd de nd de nd). **Capítulo 1. Introdução**. Disponível em <[https://docs.kde.org/stable4/pt\\_BR/kdesdk/umbrello/introduction.html](https://docs.kde.org/stable4/pt_BR/kdesdk/umbrello/introduction.html)>. Acesso em 18 de 09 de 2017.
- FILHO, W. D. P. P. **Engenharia de Software: fundamentos, métodos e padrões**. 3. ed. [S.l.]: Indaial, 2000.
- FURLAN, J. **MODELAGEM DE OBJETOS ATRAVÉS DA UML - the Unified Modeling Language**. São Paulo: MAKROS Books, 1998.
- GLAUBER, Nelson. **DOMINANDO O ANDROID DO BÁSICO AO AVANÇADO**. São Paulo: Novatec Editora Ltda, 2015.
- RICO de Souza conta sua trajetória do Surf no Brasil. **tvbrasil**, 11 jan. 2017. Disponível em: <<http://tvbrasil.ebc.com.br/estudiomovel/episodio/rico-de-souza-Conta-sua-trajetoria-do-surf-no-brasil>>. Acesso em: 28 ago. 2017.
- GUEDES, G. T. A. **UML2 UMA ABORDAGEM PRÁTICA**. São Paulo: Novatec Editora Ltda, 2009, 2011.
- KOSCIANKI, A. ; SOARES, M. Dos S. **Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. 2. ed. São Paulo: Novatec, 2006.

LECHETA, R. **ANDROID Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 4. ed. São Paulo: Novatec Editora Ltda, 2015.

LEITE, Edinho. (s.d.). **Gabriel Medina e a maior exposição do surf no Brasil na história**. Disponível em: <[http://espn.uol.com.br/noticia/467114\\_gabriel-medina-e-a-maior-exposicao-do-surf-no-brasil-na-historia](http://espn.uol.com.br/noticia/467114_gabriel-medina-e-a-maior-exposicao-do-surf-no-brasil-na-historia)>. Acesso em 27 de 08 de 2017

LIBARDI, P. L.; BARBOSA, V. **Métodos Ágeis**. p. 35, 2010.

MELO, A. C. **Desenvolvendo aplicações com UML 2.2**. 3. ed. Rio de Janeiro: Brasport, 2010.

MONTEIRO, J. B. **GOOGLE ANDROID CRIE APLICAÇÕES PARA CELULARES E TABLETS**. São Paulo: Casa do Código, 2012.

NOVAIS, C. (07 de Março de 2012). **Inkscape: uma ferramenta completa para desenho vetorial**. disponível em:< <http://ubuntued.info/inkscape-uma-ferramenta-completa-para-desenho-vetorial>>. Acesso em 2017 de Novembro de 01.

OGLIO, M. D. **APLICATIVO ANDROID PARA O AMBIENTE UNIVANTES VIRTUAL univantes**, 2013. Disponível em: <[https://univantes.br/bdu/bitstream/10737/382/1/Marcel Dall'Oglio.pdf](https://univantes.br/bdu/bitstream/10737/382/1/Marcel%20Dall'Oglio.pdf)>. Acesso em: 15 set. 2017.

OLIVEIRA, M. M. **Como Fazer projetos, relatórios, monografias, dissertação e teses**. Rio de Janeiro: CAMPUS, 2008.

DEITEL, P.; DEITEL, H.; DEITEL, A. **Android para programadores: uma abordagem baseada em aplicativos**. 2. ed. Porto Alegre: Bookman, 2015.

DEITEL, P.; DEITEL, H. **Java COMO PROGRAMAR**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

PEREIRA, D. W.; ARMBRUST, I. **Pedagogia de AVENTURA**. 1. ed. Jundiaí, São Paulo, 2010.

Rafael, A. **Scrum Gestão Ágil para Projetos de sucesso**. São Paulo: Casa do Código, 2013.

SANTOS, É. C; CINDRA, J. S; SOUZA, J. K. (20 de março de 2009). **Google acadêmico**. Disponível em: <<http://bd.centro.iff.edu.br/bitstream/123456789/321/3/Documento.pdf>>. Acesso em 18 de 10 de 2017

SEABRA, J. M. **UML Uma ferramenta para o Design de Software**. Rio de Janeiro: Editora CIÊNCIA MODERNA, 2013.

SILVA, A. M; VIDEIRA, C. A. **UML, Metodologias e Ferramentas CASE** (Vol. 1). Lisboa: Centro Atlântico, Lda, 2001.

SILVA, C. F. (14 de fevereiro de 2013). **O GIMP COMO FERRAMENTA NO AUXÍLIO DA INCLUSÃO DIGITAL**. Disponível em: <<http://bdm.unb.br/handle/10483/4479>>. Acesso em 26 de outubro de 2017.

Silva, J. P. (2011). Revista Brasileira de Informática na Educação.

SILVA, O. R; Robison, B. **Android do Básico ao Avançado**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2014.

SILVA, R. P. **UML2 em Modelagem Orientada a Objetos**. .2 ed. Florianópolis: Editora Visual Books, 2007.

SOARES, M. S. (s.d.). **Metodologias Ágeis Extreme Programming** e. *Revista Eletrônica de Sistemas de Informação*, pp. 1-8.

SOMMERVILLE, I. **Engenharia de Software**. .6 ed. São Paulo: Pearson, 2003.

SOMMERVILLE, I. **engenharia de SOFTWARE**. .9 ed. São Paulo: Pearson, 2011.

SOUSA, A. G. (2012). <http://bdm.unb.br>, Disponível em:<[http://bdm.unb.br/bitstream/10483/5719/1/2012\\_AndrezaGalvaodeSousa.pdf](http://bdm.unb.br/bitstream/10483/5719/1/2012_AndrezaGalvaodeSousa.pdf)>. Acesso em 26 de Outubro de 2017.

SOUZA, J. **Atividades e esporte de aventura para profissionais de Educação Física**. São Paulo: Phorte editora, 2013.