

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA

MARCO ANTONIO KELLER

**JOGO DE PÔQUER TEXAS HOLD'EM PARA ANDROID**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA - PR

2012

MARCO ANTONIO KELLER

**JOGO DE PÔQUER TEXAS HOLD'EM PARA ANDROID:**

Monografia de Especialização apresentada ao Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Especialista em Tecnologia Java” - Orientador: Prof. Paulo Bordin.

CURITIBA - PR

2012

## DEDICATÓRIA

Dedico este trabalho a Deus que me ajudou a conseguir terminar este trabalho. A minha esposa Jéssica que me incentivou e me ajudou a ter tempo para a realização do mesmo. Aos meus pais e minha irmã que me apoiaram a realizar esta especialização e ao meu primo Vinicius que ajudou na parte gráfica do jogo.

## RESUMO

Com a evolução dos celulares e o lançamento dos primeiros dispositivos com Android, tornou-se viável criar aplicativos que fossem suportados por diferentes aparelhos e por diversas marcas. Desde que o Android popularizou-se, foram desenvolvidas várias aplicações para o mesmo, incluindo jogos, onde estes tem uma grande percentagem. Pôquer está entre os jogos de cartas mais jogados no mundo, e a maioria dos jogos de Pôquer para Android são feitos para funcionar apenas com uma conexão de internet. Isto motivou o desenvolvimento deste trabalho que tem como objetivo demonstrar a criação de um jogo de Pôquer Texas Hold'em utilizando outra abordagem, o *Bluetooth*, como substituto à internet. O projeto visa também demonstrar a arquitetura do jogo, os conceitos sobre as tecnologias utilizadas, sua história e porquê foram utilizadas, e ainda a historia do Pôquer e o modo de jogar a modalidade Texas Hold'em. Como resultado foi desenvolvido um jogo de Pôquer *Multiplayer* que utiliza a tecnologia *Bluetooth*, o que possibilita um entretenimento mesmo sem uma conexão com internet.

**Palavras-chave:** Android, Pôquer, *Bluetooth*, *Multiplayer*.

## ABSTRACT

With the development of cell phones and the first Android devices being released, it made it possible to develop applications for different kinds of cell phone models and brands. Since Android became popular, many applications were developed, especially for games. As Poker is a very popular card game, Brazil included, and most of Android poker games are playable only on the internet, I was motivated to work on this project with the main purpose of showing the development process of a Poker Texas Hold'em Android Game, using Bluetooth as a substitute for the internet. This project also aims to introduce the game architecture, the Android and poker story, the main points of some technologies used to develop this software and why it was used. As a result, a Multiplayer Poker Game based on Android was developed with Bluetooth connection support which allows the user to play and enjoy it on travels or on places with no internet connection.

**Key-Words:** *Android, Poker, Bluetooth, Multiplayer.*

## LISTA DE ILUSTRAÇÕES

|  |    |
|--|----|
| Figura 1 - Ciclo de Vida do Android Fonte: Adaptado de ANDROID, 2012.....    | 14 |
| Figura 2: Estrutura de arquivos de <i>layout</i> do jogo "Smart Poker" ..... | 24 |
| Figura 3: Ilustração demonstrativa da tela " <i>Splash Screen</i> " .....    | 25 |
| Figura 4: Arquivo de layout XML da tela " <i>Splash Screen</i> " .....       | 25 |
| Figura 5: Ilustração demonstrativa da tela " <i>Main Lobby</i> " .....       | 26 |
| Figura 6: Ilustração demonstrativa da tela " <i>Create Table</i> " .....     | 26 |
| Figura 7: Ilustração demonstrativa da tela " <i>Scan Devices</i> " .....     | 27 |
| Figura 8: Ilustração demonstrativa da tela " <i>Poker Table</i> " .....      | 28 |
| Figura 9: Arquitetura de sincronização de estado do jogo "Smart Poker" ..... | 29 |
| Figura 10: Fluxo de processamento do servidor.....                           | 30 |
| Figura 11: Fluxo de processamento do cliente .....                           | 31 |
| Figura 12: Diagrama de caso de uso do jogo <i>Smart Poker</i> .....          | 31 |
| Figura 13: Tela de <i>splash screen</i> do jogo "Smart Poker" .....          | 37 |
| Figura 14: Tela <i>Main Lobby</i> do jogo "Smart Poker" .....                | 38 |
| Figura 15: Tela <i>Create Table</i> do jogo "Smart Poker" .....              | 38 |
| Figura 16: Tela <i>Scan Devices</i> do jogo "Smart Poker" .....              | 39 |
| Figura 17: Tela <i>Poker Table</i> do jogo "Smart Poker" .....               | 39 |

## LISTA DE ABREVIATURAS E SIGLAS

| <b>Sigla</b> | <b>Descrição</b>                      |
|--------------|---------------------------------------|
| APK          | Android Package File                  |
| DEX          | Dalvik Executable                     |
| DIP          | Density Independent Pixel             |
| EUA          | Estados Unidos da América             |
| ID           | Identificador                         |
| IDE          | Integrated Development Environment    |
| IFP          | International Federation of Poker     |
| IMSA         | International Mind Sports Association |
| JDK          | Java Development Kit                  |
| JVM          | Java Virtual Machine                  |
| OHA          | Open Handset Alliance                 |
| PC           | Personal Computer                     |
| PNG          | Portable Network Graphics             |
| PX           | Pixel                                 |
| SDK          | Software Development Kit              |
| SIG          | Bluetooth Special Interest Group      |
| TCP          | Transmission Control Protocol         |
| VM           | Virtual Machine                       |
| XML          | Extensible Markup Language            |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1: Descrição dos turnos do Pôquer .....                     | 19 |
| Tabela 2: Descrição das ações do jogo " <i>Smart Poker</i> " ..... | 19 |
| Tabela 3: Descrição de classificação de mão no Pôquer .....        | 20 |
| Tabela 4: Descrição das telas do jogo " <i>Smart Poker</i> " ..... | 23 |
| Tabela 5: Descrição do caso de uso "Criar Mesa" .....              | 32 |
| Tabela 6: Descrição do caso de uso "Juntar-se à mesa" .....        | 33 |

## SUMÁRIO

|   |    |
|---|----|
| 1. INTRODUÇÃO .....                             | 10 |
| 1.1 Objetivos Gerais .....                      | 10 |
| 1.2 Objetivos Específicos .....                 | 11 |
| 2. REFERENCIAL TEÓRICO .....                    | 12 |
| 2.1 Android.....                                | 12 |
| 2.1.2 Google Play .....                         | 12 |
| 2.1.3 Java no Android.....                      | 13 |
| 2.1.4 Activity .....                            | 13 |
| 2.1.4.1 Ciclo de Vida .....                     | 13 |
| 2.1.5 View .....                                | 15 |
| 2.1.6 Handler .....                             | 16 |
| 2.1.7 SlidingDrawer.....                        | 16 |
| 2.1.8 Bluetooth .....                           | 16 |
| 2.1.8.1 Bluetooth no Android .....              | 17 |
| 2.2 Poker.....                                  | 17 |
| 2.2.1 Regras do Texas Hold'em .....             | 18 |
| 2.2.3 Legislação do Poker no Brasil .....       | 21 |
| 3. METODOLOGIA E DESENVOLVIMENTO .....          | 23 |
| 3.1 Interface Gráfica .....                     | 24 |
| 3.2 Servidor e Cliente .....                    | 28 |
| 3.2.1 Fluxo Principal do Servidor.....          | 29 |
| 3.2.2 Fluxo Principal do Cliente.....           | 30 |
| 3.3 Casos de Uso .....                          | 31 |
| 3.4 Requisitos Funcionais e Não Funcionais..... | 34 |
| 3.4.1 Requisitos Não Funcionais .....           | 34 |
| 3.4.2 Requisitos Funcionais.....                | 34 |
| 4. RESULTADOS OBTIDOS .....                     | 36 |
| 5. CONSIDERAÇÕES FINAIS .....                   | 40 |
| REFERÊNCIAS .....                               | 41 |

# 1. INTRODUÇÃO

O avanço da tecnologia de dispositivos móveis possibilitou serem desenvolvidos jogos avançados, com gráficos 3D que necessitavam de uma maior capacidade de processamento, comuns em outras plataformas como *Playstation*, *Xbox* e *PC (Personal Computer)*. A plataforma Android, sendo uma tecnologia relativamente nova, veio preparada para utilizar o potencial desses novos dispositivos, os *smartphones* e os *tablets*.

No Android é utilizada a linguagem Java, uma linguagem que já é consagrada no mercado e que possui milhares de programadores espalhados ao redor do mundo. Isso colaborou para que muitos começassem a desenvolver aplicativos e jogos para Android.

A facilidade de aprendizado do Android, por utilizar uma linguagem conhecida, e o fato do Android possuir um meio unificado de distribuição de aplicativos (*Google Play*), ajudou com que o mercado de aplicativos para Android tivesse um crescimento exponencial ano a ano. Em maio de 2011, a quantidade de aplicativos no *Google Market (Google Play)* era de 200 mil, em Junho de 2012 esse número já chega a mais de 450 mil aplicativos (GOOGLE, 2012), onde 25% deles são jogos (INFOGRAFIC, 2012).

Este cenário demonstra um mercado grande e em expansão, em que existe uma enorme demanda por novos aplicativos. Com a possibilidade de criar um aplicativo que possa atingir não somente, o público brasileiro, como também qualquer pessoa que possua um dispositivo com Android, e sendo os jogos responsáveis por grande parte dos *downloads* (arquivos baixados) diariamente no *Google Play*, foi decidido desenvolver neste projeto um jogo de Pôquer *Texas Hold'em* para Android utilizando *bluetooth*.

O Pôquer *Texas Hold'em* atende os requisitos do projeto, por ser um jogo mundialmente conhecido e de continuo aumento de jogadores no Brasil. A decisão de utilizar *bluetooth*, foi devido a grande maioria dos jogos de Pôquer, encontrados no *Google Play*, serem voltados para internet ou não possuírem uma interface amigável e próxima dos jogos existentes para plataforma PC.

## 1.1 Objetivos Gerais

Desenvolver um jogo *multiplayer* de Pôquer *Texas Hold'em*, com uma interface próxima a de um jogo de Pôquer real ou de jogos *on-line* consagrados como "*poker stars*" (POKER, 2012), utilizando a plataforma Android, e como meio de comunicação a tecnologia *bluetooth*.

## 1.2 Objetivos Específicos

- Desenvolver uma arquitetura para comunicação via *bluetooth* em que permita sincronizar o estado do jogo entre os jogadores.
- Demonstrar uma arquitetura para implementação de um jogo de Pôquer *Texas Hold'em multiplayer*.
- Criar uma interface gráfica de uma mesa de Pôquer e seus componentes necessários para realizar um jogo de *Texas Hold'em*.
- Criar um menu de histórico de ações realizadas por cada jogador e estado do jogo.
- Criar uma tela de apresentação.
- Criar uma tela de menu e seleção de dispositivo móvel.
- Desenvolver uma arquitetura para verificar o vencedor de uma mão.

## 2. REFERENCIAL TEÓRICO

A seguir serão apresentadas as principais tecnologias e assuntos necessários para a melhor compreensão do desenvolvimento do jogo de Pôquer.

### 2.1 Android

Android é um sistema operacional *open source* para dispositivos móveis, baseado em Linux. Foi criado pela Android Inc. em Outubro de 2003 na cidade de Palo Alto, Califórnia, Estados Unidos, e tem como seus fundadores: Andy Rubin, Rich Miner, Nick Sears e Chris White (ANDROID, 2012).

Em Agosto de 2005, o Google comprou a Android Inc., dois anos depois a OHA (*Open Handset Alliance*) foi fundada com o intuito de evoluir a plataforma Android e investir na inovação das tecnologias de telefonia móvel (ANDROID, 2012).

Em 2007, mesmo ano que surgiu a OHA, foi lançada a primeira versão beta da plataforma Android, e em 23 de Setembro de 2008 o primeiro celular rodando o Android 1.0, o HTC Dream (G1) foi apresentado (ANDROID, 2012).

O sistema do Google rapidamente ganhou uma grande fatia de mercado e em 2011 já eram ativados mais celulares com Android do que o seu maior concorrente, o iOS da Apple.

#### 2.1.2 Google Play

Google *Play*, antigo Google *Market*, é uma ferramenta onde desenvolvedores, de aplicativos Android, podem enviar seus aplicativos e compartilhar com pessoas que tenham um dispositivo rodando Android.

São necessários três passos para publicar um aplicativo no Google *Play* (GOOGLE, 2012):

- Registrar-se no Google *Play*, concordar com o termo disponível no site e pagar uma taxa de cadastro.
- Enviar o aplicativo com extensão *.apk* (*Android Package File*), cujo tamanho máximo deve ser de 50 *Megabytes*.
- Enviar pelo menos duas imagens de captura de tela do aplicativo.
- Enviar um ícone em alta resolução do aplicativo.
- Publicar seu aplicativo.

### 2.1.3 Java no Android

Java é uma linguagem orientada a objetos, *open source*, desenvolvida pela Sun Microsystems na década de 90. Os códigos fontes Java (.java) são compilados em um arquivo com extensão .class chamado *bytecode*. A JVM (*Java Virtual Machine*) é responsável por interpretar os *bytecodes* Java.

Android utiliza a linguagem Java para desenvolvimento, porém não existe uma JVM no Android, existe a VM (*Virtual Machine*) Dalvik que utiliza arquivos .dex (*Dalvik Executable*) em vez dos .class do Java. Após criar o(s) arquivo(s) .java o JDK (*Java Development Kit*) é utilizado para compilá-lo(s) convertendo para .class e o Android converte os arquivos .class para .dex. Por fim para uma aplicação ser empacotada, distribuída e instalada é criado um arquivo com extensão .apk (*Android Package File*) (LECHETA, 2009).

### 2.1.4 Activity

É uma das principais classes do Android, geralmente uma *activity* será uma tela da aplicação, é nela em que será adicionada a interface gráfica e onde serão tratados os eventos gerados pelo usuário.

#### 2.1.4.1 Ciclo de Vida

O ciclo de vida de uma *activity* pode ser dividido em três ciclos (ANDROID, 2012):

- **Entire lifetime** inicia na chamada do método *onCreate()* e finaliza na chamada do método *onDestroy()*. Este ciclo engloba todo o período que uma *activity* existe, sendo que *onCreate()* é chamado quando uma *activity* é criada e *onDestroy()* quando é finalizada.
- **Visible lifetime** inicia na chamada do método *onStart()* e finaliza na chamada do método *onStop()*. Este ciclo ocorre durante o período em que o usuário pode ver a *activity* ou interagir com ela, porém não necessariamente a *activity* vai estar em primeiro plano na tela.
- **Foreground lifetime** inicia na chamada do método *onResume()* e finaliza na chamada do método *onPause()*. Durante este ciclo a *activity* se encontra visível e em foco.

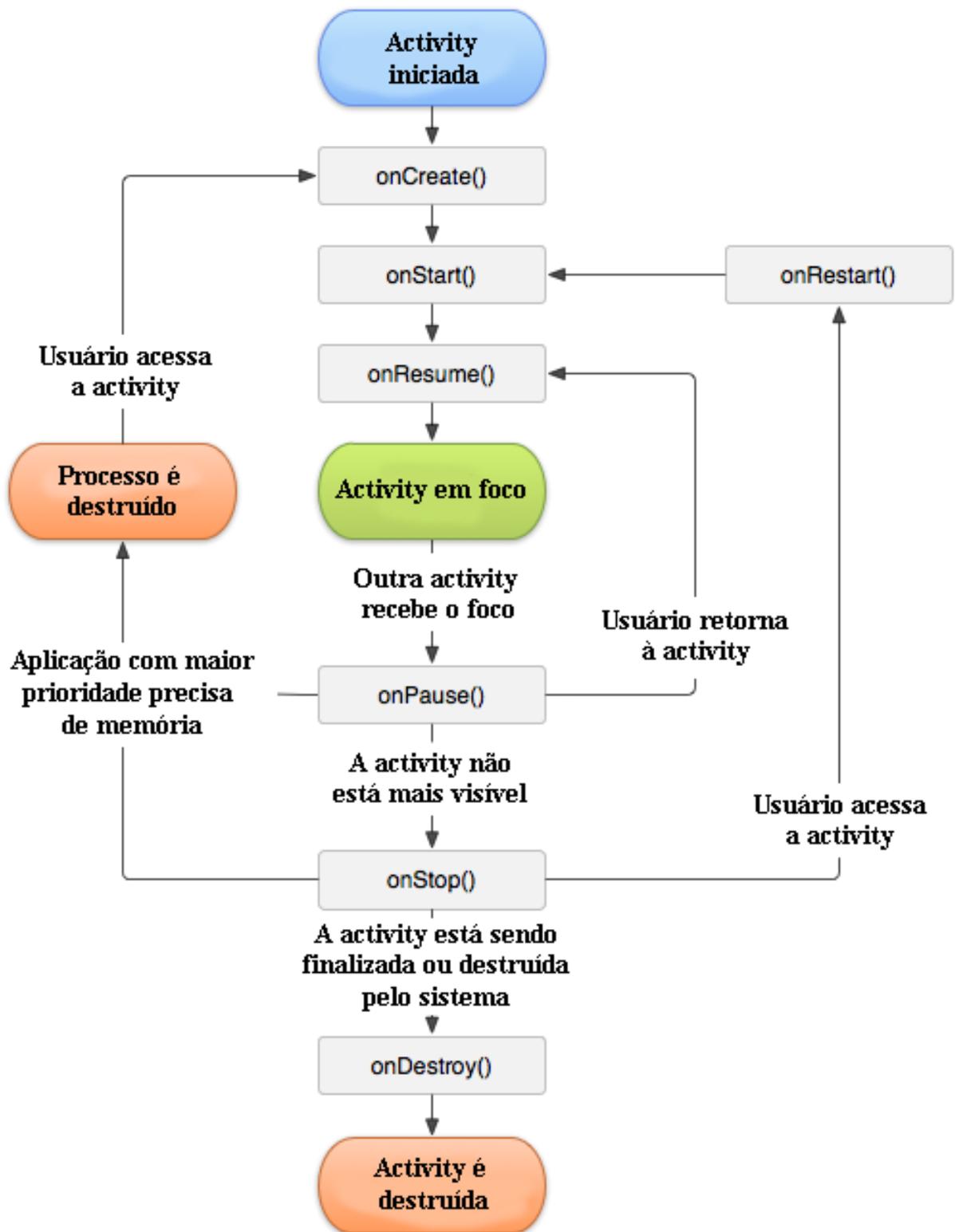


Figura 1 - Ciclo de Vida do Android  
 Fonte: Adaptado de ANDROID, 2012.

### 2.1.5 View

*View* é a classe responsável pela interface gráfica e por gerenciar os eventos criados pelo usuário. Também é a base para criar componentes interativos (*widgets*), como por exemplo: botões, campos de texto e etc.

Principais atributos definidos em uma *View* (LECHETA, 2009):

- *android:layout\_width* - Define o tamanho na horizontal. Seus possíveis valores são definir um valor numérico em *pixels* (px) ou em *Density Independent Pixel* (dip), *fill\_parent* que expande para o mesmo tamanho do componente a que pertence (o pai), e *wrap\_parent* para utilizar apenas o tamanho necessário.
- *android:layout\_height* - Define o tamanho na vertical. Seus valores são os mesmos do *android:layout\_width*.
- *android:orientation* - Define se os componentes filhos deve ser exibidos na horizontal ou na vertical.
- *android:id* - Especifica um id para a *View*.

Segue uma breve descrição de alguns *widgets* utilizados no projeto:

- *android.widget.Button* - Classe do Android que representa um botão, onde é possível o usuário clicar/pressionar e executar uma ação.
- *android.widget.EditText* - Classe para campos de texto onde o usuário pode digitar caracteres para serem recuperados posteriormente.
- *android.widget.TextView* - Classe para visualização de textos estáticos.
- *android.widget.ImageView* - Classe utilizada para mostrar imagens.
- *android.widget.ListView* - *View* que mostra uma lista de itens dispostos verticalmente e com *scroll* de rolagem.

A classe *ViewGroup*, também chamada de Gerenciadora de *Layout*, é uma subclasse de *View*. Ela é utilizada para organizar, agrupar e alinhar outras *Views*, além de ser possível conter outros *ViewsGroups*. Os três atributos mais comuns em gerenciadores de *layout* são: *android:layout\_width*, *android:layout\_height* e *android:orientation*.

São exemplos de *ViewGroups* (LECHETA, 2009):

- *LinearLayout* - Um dos *layouts* mais utilizados, com ele é possível adicionar elementos em uma única linha ou coluna, dependendo de como o atributo *android:orientation* foi definido.
- *RelativeLayout* - É um *layout* criado para permitir posicionar seus componentes em relação a outros componentes ou ao próprio *RelativeLayout*.

### 2.1.6 Handler

Handler é uma classe que permite enviar ou agendar uma mensagem para ser executada posteriormente. Cada Handler está associado exclusivamente com uma única *thread* e a fila de mensagens desta *thread*.

Outra utilidade de uma Handler é executar uma ação em uma *thread* diferente da que ela foi criada.

### 2.1.7 SlidingDrawer

É uma classe que herda de *android.view.ViewGroup*. Uma *SlidingDrawer* esconde um conteúdo da tela permitindo que seja exibido quando for explicitado no código fonte ou quando o usuário clicar sobre uma *View* filha de *SlidingDrawer*, que deve possuir um ID (identificador) igual a "*handle*".

O conteúdo deve ser criado também em uma *View* filha de *SlidingDrawer*, com um ID igual a "*content*".

### 2.1.8 Bluetooth

*Bluetooth* é uma tecnologia de rede sem fio, para curta distância, utilizada em diferentes segmentos como celulares, computadores, carros e etc..

A tecnologia *Bluetooth* foi criada em 1994 por um grupo de engenheiros da Ericsson. Em 1998 um grupo de companhias formou o *Bluetooth Special Interest Group* (SIG), responsável por continuar desenvolvendo a tecnologia, sendo aberto para novos membros que queiram contribuir (BLUETOOTH, 2012).

O alcance de sinal via *Bluetooth* depende muito da implementação de cada fabricante, uma distância mínima é definida pela especificação do *Bluetooth*, porém não existe limite. Assim cada fabricante pode programar um *range* (alcance) de acordo com sua necessidade.

Na especificação do *Bluetooth* foram definidas três classes de transmissores (BLUETOOTH, 2012):

- Classe 1: Usado mais em indústrias, alcance aproximado de cem metros.
- Classe 2: Mais comum entre os dispositivos, alcance de dez metros.
- Classe 3: Tem um alcance aproximado de um metro.

### 2.1.8.1 Bluetooth no Android

As bibliotecas *bluetooth* do Android estão dentro do pacote *android:bluetooth*, com elas é possível procurar dispositivos *bluetooth* ativos, estabelecer uma conexão, transferir dados, gerenciar conexões, além de realizar outras tarefas.

Existem quatro classes principais para realizar uma conexão com outro dispositivo *bluetooth* e começar a transferir dados entre eles, são elas:

- *BluetoothAdapter* é utilizada para representar o adaptador de *bluetooth* do aparelho. Esta interface permite habilitar o adaptador *bluetooth* para se tornar visível, instanciar a classe *BluetoothDevice* e criar um *BluetoothServerSocket*.
- *BluetoothDevice* representa o dispositivo *bluetooth*. Permite recuperar o nome, endereço, classe e estado do dispositivo (se está pareado, pareando ou não pareado).
- *BluetoothServerSocket* é usado no lado do servidor para ficar aguardando por uma conexão, funciona de maneira semelhante a um *socket* TCP. Quando uma conexão é aceita ele retorna uma instância de *BluetoothSocket* para gerenciar a conexão.
- *BluetoothSocket* gerencia uma conexão *bluetooth*, possui métodos para conectar, verificar se está conectado ou fechar uma conexão. Esta classe possui um *InputStream* e *OutputStream* que serão utilizados para ler ou enviar dados para outro dispositivo.

## 2.2 Poker

Segundo Leo Bello (BELLO, 2007), o Pôquer surgiu na Europa através de uma evolução de jogos de cartas e apostas. O baralho de cinquenta e duas cartas e quatro naipes foi

uma invenção francesa de 1480 e por volta do século XV, o Pôquer foi trazido para a América do Norte por navegadores e colonizadores. A origem da palavra “Pôquer” veio da forma como os americanos pronunciavam a palavra *poque* que era o nome do jogo na França (BELLO, 2007, p. 68).

O Pôquer se difundiu rapidamente nos EUA (Estados Unidos da América) e logo virou um jogo popular, fixando-se na cultura americana. Isto fez com que ao passar dos anos surgissem novas variantes do jogo, sendo uma delas o Pôquer *Texas Hold'em* que obviamente surgiu no Estado do Texas, EUA.

### 2.2.1 Regras do Texas Hold'em

No Pôquer *Texas Hold'em* cada jogador irá receber no início da mão apenas duas cartas e outras cinco cartas serão apresentadas na mesa, essas são chamadas de cartas comunitárias devido ao fato de poderem ser utilizadas por todas as pessoas na mesa.

O vencedor de uma mão de Pôquer é o jogador que, ao final de todas as rodadas de apostas, tiver o melhor jogo feito com cinco cartas. Para isso cada jogador poderá utilizar às sete cartas disponíveis (duas cartas da sua mão e mais as cinco cartas comunitárias), para formar o melhor jogo possível de cinco cartas, as duas piores cartas serão desconsideradas.

O jogo pode ser jogado com duas até dez pessoas, é definido inicialmente o *dealer* (distribuidor de cartas) que a partir do *flop* será o último a agir em uma mão. O jogo acontece em sentido horário.

As *blinds* são apostas obrigatórias que acontecem sempre no início de uma mão, são realizadas duas apostas de *blinds*: *big blind* e a *small blind* cujo valor é a metade da *big blind*. O jogador à esquerda do *dealer* é o *small blind*, em seguida, à esquerda da *small blind* está a *big blind*, este jogador irá apostar o valor da *big blind* que é pré-definido antes do início do jogo. No turno antes do *flop*, o último jogador a agir é o *big blind*. Isto acontece porque o *big blind* é o responsável por definir a aposta mínima (*pré-flop*) para um jogador continuar no jogo, após serem realizadas as apostas, o jogador na posição de *big blind* tem o direito então de aumentar a aposta ou apenas dar um *check*. Nos turnos seguintes, como serão iniciadas outras rodadas de apostas, a posição de último jogador a realizar uma ação passa a ser do *dealer*.

Ao final de cada rodada de apostas, as fichas que foram apostadas naquela rodada serão adicionadas em um pote da mesa, que será o prêmio para o vencedor daquela mão.

Os turnos de uma mão de Pôquer são divididos em quatro:

**Tabela 1: Descrição dos turnos do Pôquer**

| <b>Turno</b>    | <b>Descrição</b>   |
|-----------------|--|
| <i>Pré-Flop</i> | Turno em que são realizadas as primeiras apostas, ainda não foi apresentada nenhuma carta comunitária, e cada jogador já possui duas cartas que irão ficar até o final da mão. |
| <i>Flop</i>     | São viradas na mesa três cartas comunitárias, chamadas de <i>flop</i> , seguido de uma nova rodada de apostas.   |
| <i>Turn</i>     | É virada uma carta chamada de <i>turn</i> , seguido de uma nova rodada de apostas.   |
| <i>River</i>    | Última carta comunitária da mão é mostrada, seguido da última rodada de apostas.   |

Para cada rodada de apostas o jogador na vez terá as seguintes opções de ação:

**Tabela 2: Descrição das ações do jogo "Smart Poker"**

| <b>Ação</b>   | <b>Descrição</b>   |
|---------------|--|
| <i>Fold</i>   | Ação em que o jogador foge da mão, tendo que aguardar a próxima mão para jogar.  |
| <i>Check</i>  | Quando não existe uma aposta ou a maior aposta é a mesma do jogador a realizar a ação, ele pode apenas dar <i>check</i> , que quer dizer que irá manter-se no jogo, mas sem aumentar a aposta. |
| <i>Call</i>   | Quando o jogador apenas paga uma aposta realizada por um adversário.   |
| <i>Raise</i>  | O jogador aposta um valor maior do que outra aposta existente.   |
| <i>All In</i> | O jogador aposta todas as fichas que ele tem. Neste caso não é necessário ser igual ou maior a uma aposta realizada por um adversário.   |

No final da rodada de apostas do *river*, se sobrar apenas um jogador, os outros tiverem fugido da mão, o jogador restante será dado como vencedor daquela mão e irá somar as fichas

do pote da mesa às suas fichas. Caso sobrarem dois jogadores ou mais, será verificado quem possui o jogo mais alto para então atribuir as fichas ao vencedor.

A classificação dos jogos na ordem do mais baixo para o mais alto é apresentada na tabela 3 (BELLO, 2007):

**Tabela 3: Descrição de classificação de mão no Pôquer**

| <b>Jogo</b>                          | <b>Descrição</b>   | <b>Exemplo</b>                         |
|--------------------------------------|--|--|
| Carta mais alta ( <i>High Card</i> ) | Quando não é realizado nenhum jogo, o jogador que tiver a carta mais alta ganha, caso as cartas mais alta sejam iguais, é verificado a segunda carta mais alta e assim sucessivamente. | A♠ J♥ 2♦ 6♣ 8♠<br>Carta mais alta: A♠. |
| Par ( <i>Pair</i> )                  | Dentre as cinco cartas duas cartas iguais.   | A♠ J♥ 2♦ J♣ 8♠<br>Par: J♥ e J♣.        |
| Dois pares ( <i>Two Pair</i> )       | Dentre as cinco cartas existem um par de uma carta e outro de uma carta diferente.   | A♠ J♥ 2♦ J♣ 2♠<br>Pares: J♥ J♣ / 2♠ 2♦ |
| Trinca ( <i>Three of a Kind</i> )    | Dentre as cinco cartas existem três cartas iguais.   | A♠ J♥ 8♦ J♣ J♠<br>Trinca: J♥ J♣ J♠     |
| <i>Straight</i>                      | Cinco cartas formam uma sequência crescente sem serem todas do mesmo naipe.  | A♠K♣Q♥J♠10♦.                           |
| <i>Flush</i>                         | As cinco cartas são do mesmo naipe mas não formam uma sequência.   | A♠ 8♠ 2♠ 7♠ J♠                         |
| <i>Full House</i>                    | Dentre as cinco cartas existem uma trinca e um par.  | A♠ J♥ J♦ A♣ J♠                         |
| Quadra ( <i>Four of a Kind</i> )     | Dentre as cinco cartas existem quatro cartas iguais.   | A♠ A♥ A♦ A♣ J♠                         |
| <i>Straight Flush</i>                | É uma sequência, porém com as cinco cartas do mesmo naipe.   | 9♣ 8♣ 7♣ 6♣ 5♣                         |

|                             |   |                 |
|-----------------------------|---|-----------------|
| <i>Royal Straight Flush</i> | É um <i>straight flush</i> , mas começando pelo Ás e terminando no dez. | A♥ K♥ Q♥ J♥ 10♥ |
|-----------------------------|---|-----------------|

Para desempate em caso dos jogadores possuírem o mesmo jogo, como por exemplo, um par, o par mais alto será o vencedor, se for o mesmo par é usado a carta mais alta restante (*kicker*) para determinar o vencedor e se com as cinco cartas os jogadores possuírem o mesmo jogo, incluindo as cartas mais altas, então é dado como empate e as fichas são divididas.

### 2.2.3 Legislação do Poker no Brasil

O Pôquer *Texas Hold'em* é um jogo de cartas mundialmente conhecido e que por anos foi discriminado no Brasil por ser considerado um jogo de azar, que é proibido no Brasil. Porém com a ajuda da internet e da televisão, ouve um grande aumento de adeptos e começou uma discussão se o Pôquer realmente seria um jogo de azar ou de habilidade.

No Brasil não existe uma legislação específica para o Pôquer, ele era considerado um delito baseado no Artigo 50 do Decreto-Lei 3688/41, que rege as contravenções penais.

De acordo com o Artigo 50 do Decreto-Lei 3688/41, é jogo de azar: “o jogo em que o ganho e a perda dependem exclusiva ou principalmente da sorte”.

Isto mostra que caso seja comprovado que um jogo não depende exclusivamente ou principalmente da sorte, ele deve ser considerado legal. Esse fundamento colaborou para que as realizações de torneios de Pôquer deixassem de ser um delito. Um caso que contribuiu para isso foi um julgamento que aconteceu em Santa Catarina, onde a empresa Overbet, após ter sido negada, pelo Secretário de Segurança, o direito de realizar um evento de Pôquer *Texas Hold'em*, anexou um parecer jurídico como descreve Carlezzo (2011):

"A Overbet, no caso mencionado, anexou um parecer jurídico de um conhecido advogado criminalista onde ele analisa a legalidade do pôquer no país e traz argumentos importantes de defesa do pôquer como um jogo de habilidade e estratégia, por exemplo: a aceitação da Federação Internacional do Pôquer (IFP) pela Associação Internacional de Esportes Intelectuais (IMSA)." (CARLEZZO, 2011).

Após esse parecer, o juiz permitiu a realização do evento.

"O juiz entendeu que o pôquer não é proibido no Brasil já que é classificado como jogo de habilidade e estratégia e determinou que o Secretário de Segurança se abstivesse de impor obstáculos à realização do evento." (CARLEZZO, 2011).

Acrescentado a isso, Eduardo Mahon relata o posicionamento do Tribunal de Justiça de São Paulo sobre o Pôquer:

"o Tribunal de Justiça de São Paulo unanimemente posicionou-se da seguinte forma: "(...) assim decidem por já ter sido o pif-paf considerado jogo não punível, em reiterada jurisprudência do Egrégio Supremo Tribunal Federal. É certo que o pif-paf como o pôquer, não pode ser considerado como jogo de azar, pois não dependem exclusivamente de sorte, como o bacarat, a campista e outros jogos carteados. O ganho depende da habilidade e da observação dos parceiros". (AC – RT 228/499)" (MAHON, 2011).

A decisão do julgamento em Santa Catarina e outras que aconteceram subsequentes, permitem atualmente que sejam realizados torneios de Pôquer ao vivo e que seja feita publicidade sobre o jogo no Brasil.

Vale ressaltar que isto só é válido para jogos realizados ao vivo, isto porque o jogo através da internet ou dispositivos eletrônicos poderia ser considerado jogo de azar, visto que existe um software ou dispositivo eletrônico que pode vir a manipular o resultado do mesmo, excluindo assim a questão da habilidade.

### 3. METODOLOGIA E DESENVOLVIMENTO

O projeto demonstra o desenvolvimento de um jogo de Pôquer Texas Hold'em *multiplayer* (dois jogadores), utilizando a tecnologia *Bluetooth* como base para comunicação entre os aparelhos Android.

Como ambiente de desenvolvimento do jogo foi utilizado a IDE (*Integrated Development Environment*) Eclipse Indigo, o Android SDK (*Software Development Kit*) e o JDK 1.6.

Não foi utilizada nenhuma *engine* (ENGINE, 2012) para facilitar o desenvolvimento, apesar de existir a *engine* Mages (MAGES, 2012) que foi desenvolvida para simplificar a comunicação e sincronização de estado em jogos *multiplayer* com *bluetooth*, não foi encontrada documentação que auxiliasse o uso da mesma. Sendo assim foram utilizadas apenas classes nativas do Android.

O Pôquer é um jogo por turnos, onde cada jogador terá sua vez. Isto quer dizer que nunca os dois jogadores poderão fazer uma ação, que altere o estado do jogo, ao mesmo tempo, enquanto um jogador realiza uma ação o outro deve esperar sua vez. Sendo assim, o aplicativo irá se encarregar de gerenciar os turnos do jogo e apresentar as opções de ação para um jogador por vez.

O jogo consiste de cinco telas, de acordo com o descrito na tabela 4:

**Tabela 4: Descrição das telas do jogo "Smart Poker"**

| Nome da Tela            | Descrição   |
|-------------------------|---|
| 1. <i>Splash Screen</i> | Tela inicial do aplicativo tem a função de "tela de apresentação" do jogo. Serve apenas para visualização, automaticamente é enviado para a tela " <i>Main Lobby</i> ". |
| 2. <i>Main Lobby</i>    | Tela de menu do jogo, contém as opções de criar uma mesa ou entrar em uma já existente. É apresentada logo a seguir da " <i>Splash Screen</i> ".                        |
| 3. <i>Scan Devices</i>  | Tela acessada quando o usuário seleciona entrar em uma mesa já existente. Apresenta os dispositivos de <i>bluetooth</i> disponíveis.                                    |
| 4. <i>Create Table</i>  | Tela acessada pelo menu "Criar Mesa". Exibe os campos de texto necessários para criar a mesa e o botão "Criar".   |
| 5. <i>Poker Table</i>   | Tela principal do jogo, ela é acessada após ser estabelecida uma  |

|  |   |
|--|---|
|  | conexão de <i>bluetooth</i> entre os dois jogadores que tenham o jogo devidamente instalado. É nela que os usuários irão jogar. Ao fechar esta, a tela " <i>Main Lobby</i> " é apresentada. |
|--|---|

Apesar de existirem cinco telas, foram criadas seis *activities*, isto foi necessário porque duas delas tem a mesma função, ambas apresentam a tela "*Poker Table*", porém uma é usada pelo lado servidor e a outra pelo lado cliente.

### 3.1 Interface Gráfica

O *layout* do jogo foi todo criado em arquivos XML (*Extensible Markup Language*), apesar de o Android permitir criar as *Views* por meio das classes Java, não é considerada a melhor prática a ser utilizada, devido a não ser tão organizado e fácil de implementar *layouts* compatíveis com diferentes tamanhos de tela e de resoluções.

Todas as *activities*, exceto a "*Poker Table*" tem suporte a orientação de tela *portrait* (vertical) e *landscape* (horizontal). No caso da tela onde o jogo é realizado foi deixado o modo *landscape* fixo por questão de o espaço horizontal ser mais necessário neste jogo de Pôquer.

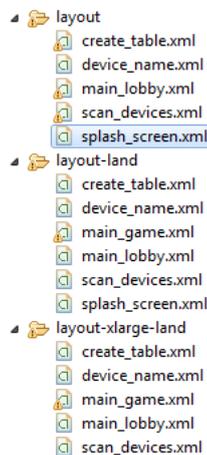


Figura 2: Estrutura de arquivos de *layout* do jogo "*Smart Poker*"

Ao iniciar o aplicativo o usuário visualizará a tela chamada "*Splash Screen*" que não possui nenhuma *view* de interação. Esta tela irá permanecer durante alguns segundos na tela e irá carregar automaticamente a tela de menu "*Main Lobby*".

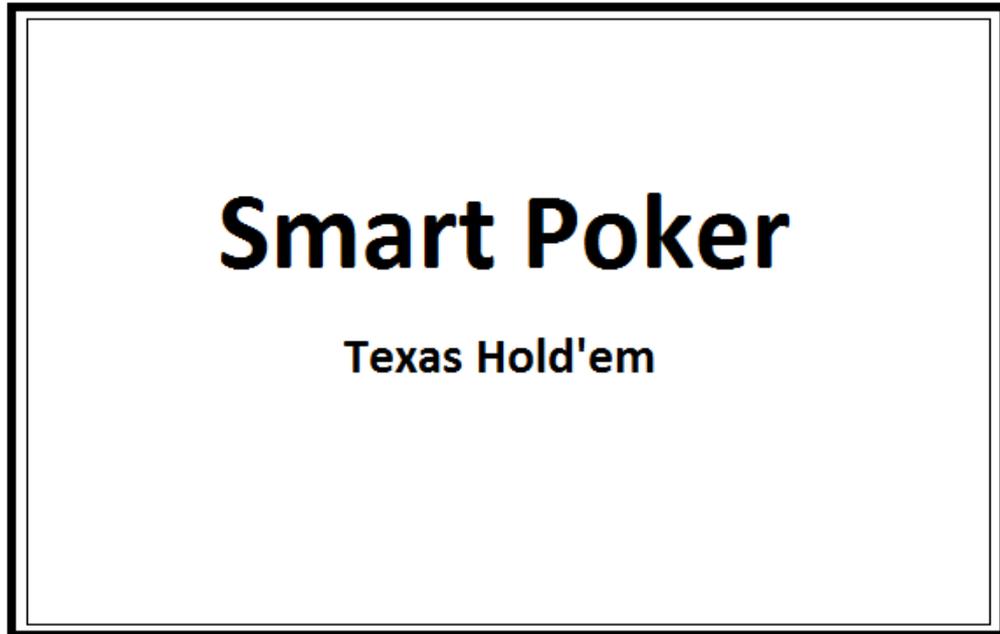


Figura 3: Ilustração demonstrativa da tela "Splash Screen"

O layout desta tela é muito simples visto que o objetivo é apenas mostrar, por alguns segundos, uma imagem de apresentação do jogo.

```
splash_screen.xml ✕
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="horizontal"
6   android:background="@drawable/splash_screen">
7 </LinearLayout>
```

Figura 4: Arquivo de layout XML da tela "Splash Screen"

Após carregar a tela de menu, o usuário terá as opções de criar uma mesa ou de juntar-se a uma mesa já existente, para isso será necessário informar o seu nome ou apelido em um campo de texto.

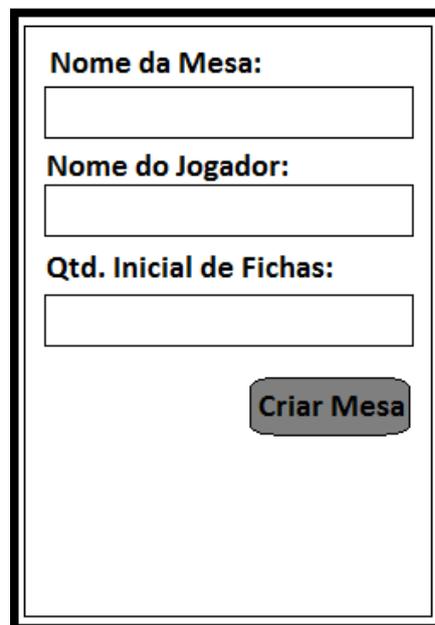
Foram utilizados três componentes principais para criar esse *layout*, são eles: *Button*, *TextView* e *EditText*.



The screenshot shows a mobile application interface for creating a table. At the top, there is a dark grey header with the text "Criar Mesa" in white. Below the header, the label "Nome:" is positioned to the left of a white text input field. To the right of the input field, there is a dark grey button with the text "Entrar" in white.

Figura 5: Ilustração demonstrativa da tela "Main Lobby"

O usuário que clicar em "Criar Mesa" será direcionado para uma *activity* "Create Table" que irá apresentar um campo de texto para o nome da mesa, o nome do jogador e a quantidade inicial de fichas para aquele jogo.



The screenshot shows a mobile application interface for creating a table. It features three text input fields stacked vertically. The first field is labeled "Nome da Mesa:", the second "Nome do Jogador:", and the third "Qtd. Inicial de Fichas:". Below the input fields, there is a dark grey button with the text "Criar Mesa" in white.

Figura 6: Ilustração demonstrativa da tela "Create Table"

Para entrar em uma mesa já existente, o usuário deverá preencher seu nome e clicar em "Entrar" e assim ser direcionado para a *activity* "Scan Devices" onde ele deverá selecionar, dentre uma lista de dispositivos *bluetooth*, o jogador que criou a mesa. Após ter sido realizada

a conexão com o servidor o jogo irá iniciar-se e o servidor se encarregará de fazer as configurações iniciais, distribuir as cartas, definir quem deve começar o turno e enviar um *status* atualizado para o cliente.

Foi utilizada a *widget ListView* para apresentar a lista de dispositivos.

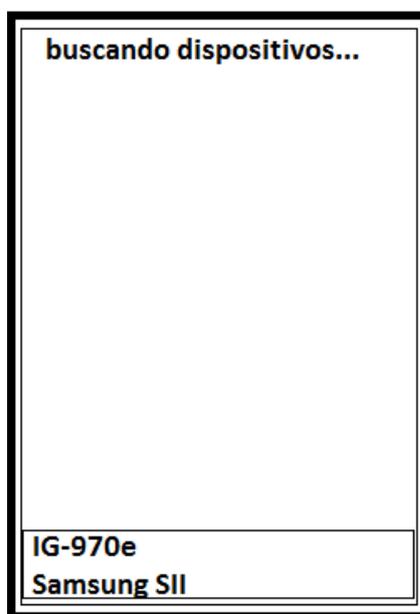


Figura 7: Ilustração demonstrativa da tela "Scan Devices"

No *layout* da mesa foi definida uma imagem em PNG de uma mesa de Pôquer como plano de fundo. As cartas dos jogadores ficam posicionadas nos cantos da mesa e as cartas da mesa no centro. Foi utilizada a classe *RelativeLayout* para posicionar corretamente os componentes da mesa.

Visto que alguns dispositivos possuem uma tela pequena, foi decidido usar todo o espaço da tela para visualizar o jogo. Os menus, botões de opção e histórico do jogo ficam escondidos e só aparecem na tela quando o usuário aciona um ícone específico ou quando o sistema se encarrega de mostrar. Logo que não são mais necessários eles voltam para o estado anterior. Essa abordagem foi possível utilizando a classe *SlidingDrawer*.

O *layout* da tela da mesa do jogo é demonstrado na figura 8:

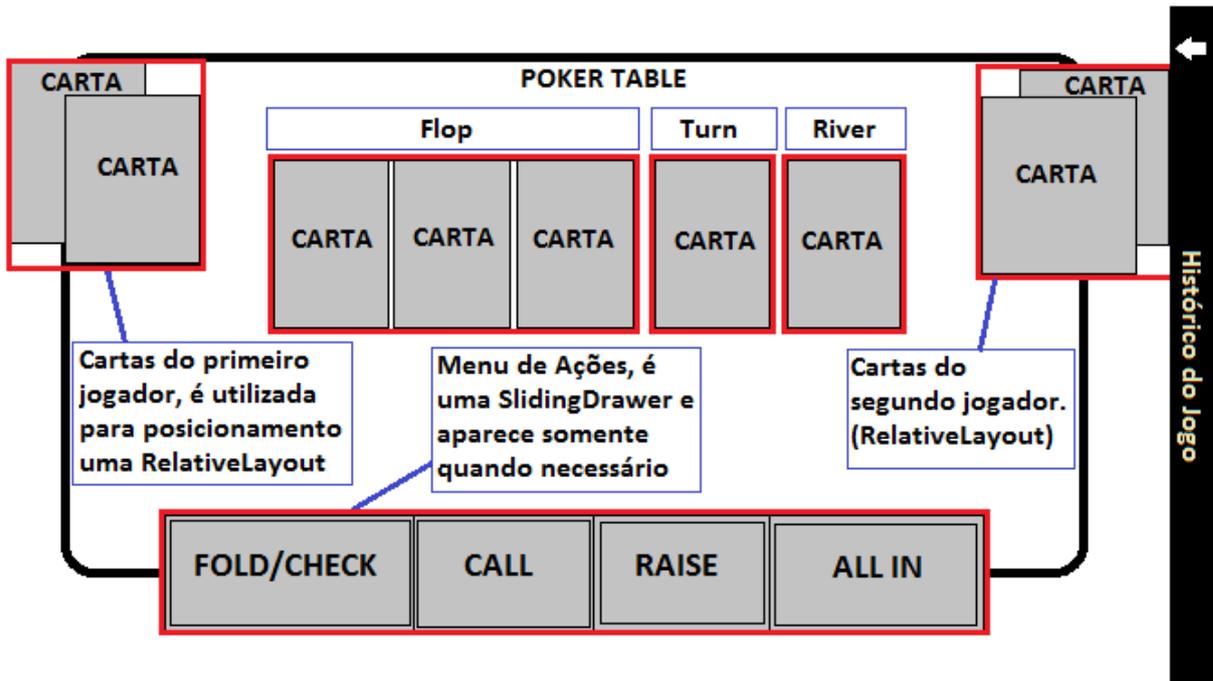


Figura 8: Ilustração demonstrativa da tela "Poker Table"

### 3.2 Servidor e Cliente

O servidor da aplicação sempre será quem criou a mesa e quem conectou a este será o cliente.

O estado do jogo é mantido pelo servidor, ele que irá realizar todo processamento necessário, distribuir as cartas, verificar o vencedor e etc. O *BluetoothSocket* do servidor, criado na conexão *bluetooth*, ficará em *loop* esperando uma mensagem do cliente, assim que receber uma mensagem, esta mensagem será lida e enviada para uma classe *MessageHandle* que irá processar e realizar as ações necessárias. Após o estado do jogo ter sido atualizado no servidor, o mesmo será responsável por atualizar a própria interface gráfica e enviar uma mensagem de resposta para o cliente. O servidor irá mandar uma nova mensagem sempre que o estado do jogo mudar, não importando se foi ele próprio, ou o cliente, o responsável pela mudança de estado.

Ao cliente é atribuído apenas receber a mensagem processada do estado do jogo e mostrar na tela para o usuário. Nenhum processamento lógico é feito no lado do cliente. É necessário apenas realizar a manutenção da parte gráfica e atualização de dados que vieram do servidor.

Quando o usuário cliente realizar uma ação, esta ação será convertida em uma mensagem e enviada ao servidor que processará a mesma. A figura 9 demonstra a arquitetura de sincronização de estado utilizada no jogo:

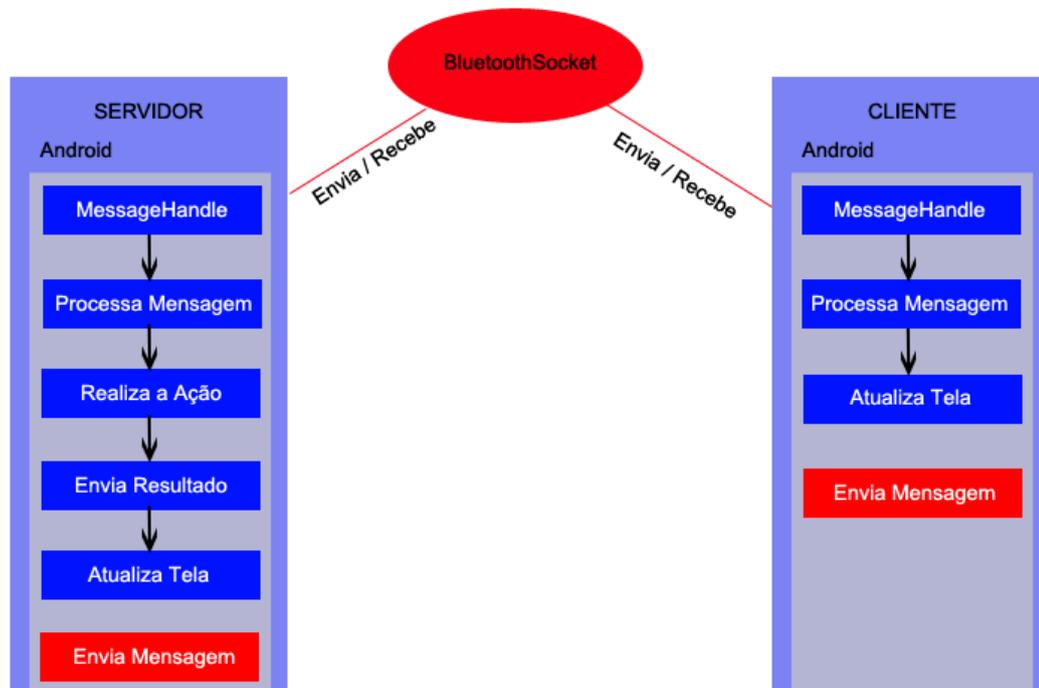


Figura 9: Arquitetura de sincronização de estado do jogo "Smart Poker"

### 3.2.1 Fluxo Principal do Servidor

O servidor inicia seu fluxo distribuindo as cartas dos jogadores e realizando as apostas obrigatórias (*Small Blind* e *Big Blind*), em seguida ele envia o estado do jogo atualizado para o cliente e aguarda uma resposta do mesmo. Após o estado do jogo estar sincronizado entre ambos, o servidor irá realizar algumas validações, verificando se existe outro turno de cartas, se é necessário trocar o turno e assim distribuir as cartas comunitárias, ou caso não seja necessário trocar de turno, o servidor irá apresentar as opções do jogador da vez.

Se a vez de jogar for do servidor, ele irá apresentar as ações possíveis, processar a ação realizada pelo jogador, atualizar a tela e enviar o estado para o cliente. Neste ponto o fluxo se repete, aguardando uma resposta do cliente.

Na vez do cliente jogar, o servidor irá mandar uma mensagem para o cliente, com as opções de ação. O servidor então processa a ação realizada pelo cliente e repete as mesmas validações anteriores.

O fluxo irá se repetir até que não exista outro turno e seja verificado qual jogador é o vencedor daquela rodada, processando em seguida as apostas e atribuindo as fichas ao ganhador.

A figura 10 demonstra o fluxo de execução do servidor.

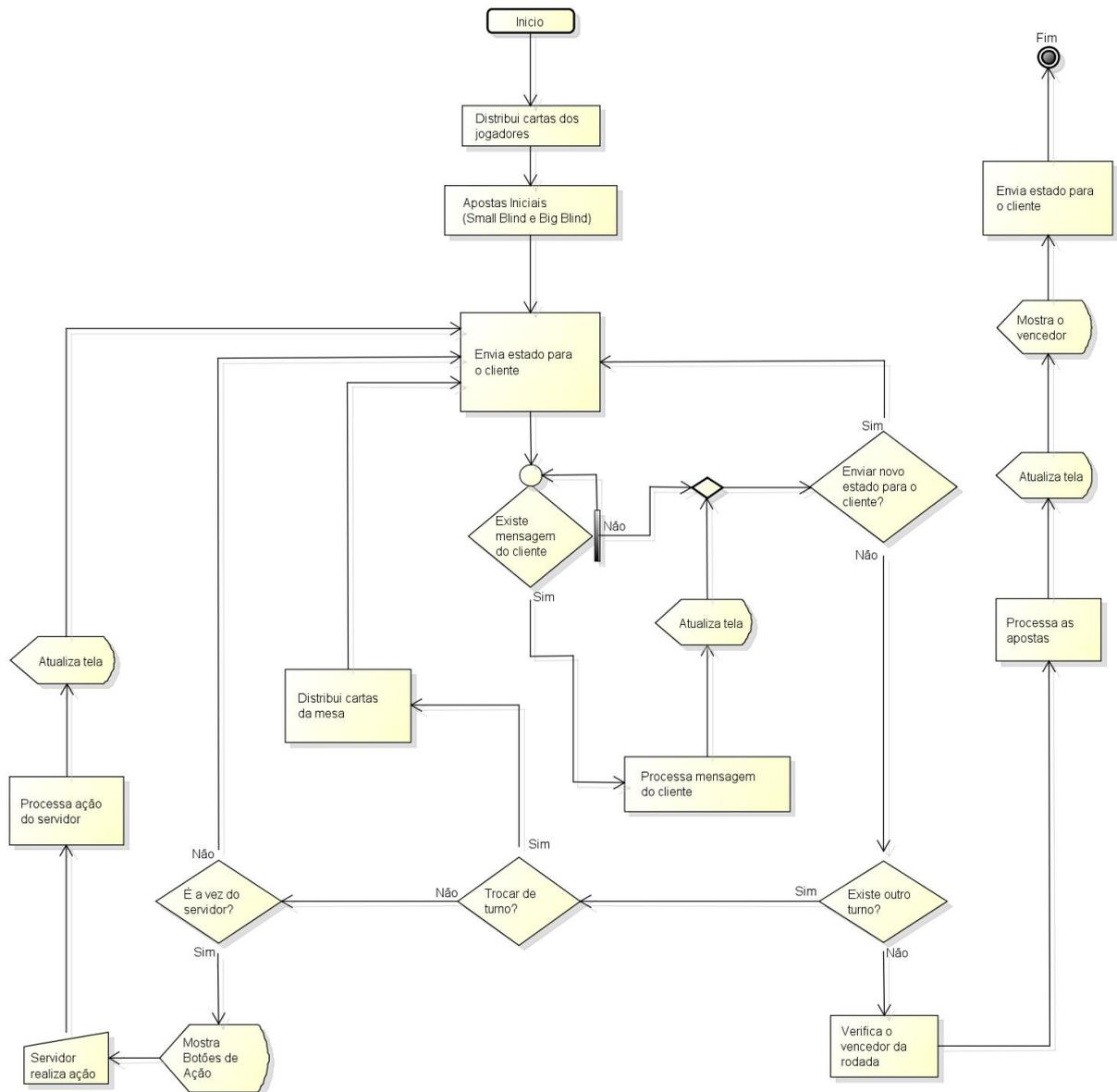


Figura 10: Fluxo de processamento do servidor

### 3.2.2 Fluxo Principal do Cliente

O fluxo principal do lado cliente começa aguardando por uma mensagem, do estado inicial do jogo, enviada pelo servidor. O cliente realizará o processamento dessa mensagem e irá atualizar a tela com os dados recebidos, caso seja a vez do cliente jogar, serão apresentadas as opções para o jogador, processar a ação realizada e enviar o estado para o servidor. O fluxo irá se repetir até o término da rodada, em que por fim o jogador irá visualizar o vencedor.

A figura 11 demonstra o fluxo de execução do cliente.

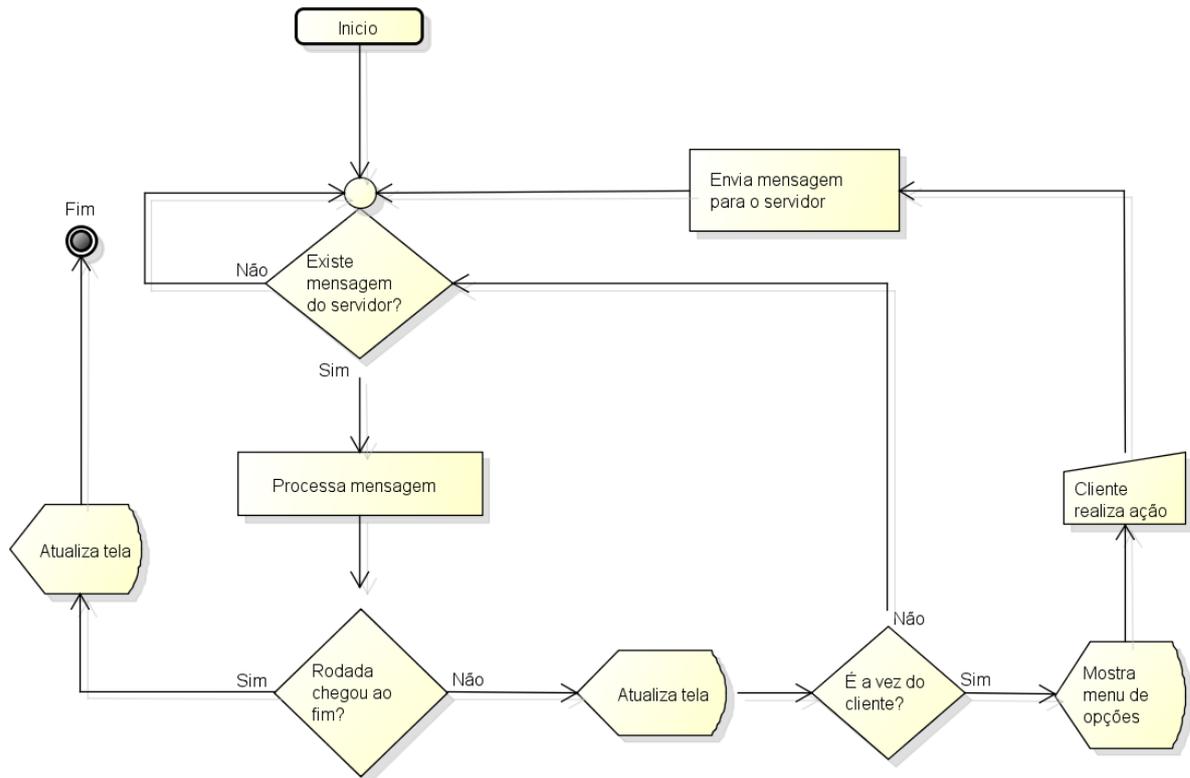


Figura 11: Fluxo de processamento do cliente

### 3.3 Casos de Uso

Um caso de uso é usado para descrever as interações de um usuário, ou ator, com o sistema. Descreve como o ator pode interagir com um conjunto de ações (casos de uso) que um sistema oferece (UML, 2012).

A figura 12 apresenta o diagrama de caso de uso do sistema "Smart Poker".

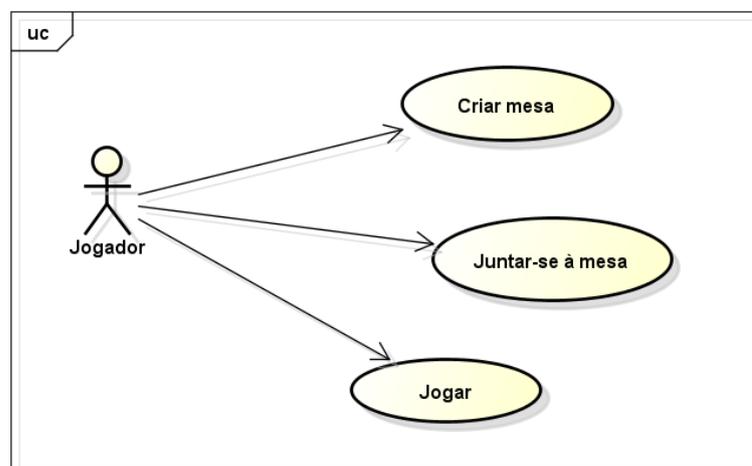


Figura 12: Diagrama de caso de uso do jogo Smart Poker

Tabela 5: Descrição do caso de uso “Criar Mesa”

|                            |   |
|----------------------------|---|
| <b>Nome do caso de uso</b> | Criar Mesa  |
| <b>Sumário</b>             | Jogador cria uma mesa no jogo <i>Smart Poker</i>  |
| <b>Ator primário</b>       | Jogador   |
| <b>Precondições</b>        | O jogador está com o aplicativo instalado no seu dispositivo Android e conexão de <i>Bluetooth</i> ativada  |
| <b>Fluxo principal</b>     | <ol style="list-style-type: none"> <li>1. Jogador inicia o jogo utilizando o ícone disponível na área de aplicativos do aparelho.</li> <li>2. O jogador visualiza a tela de apresentação e esperar o sistema carregar a próxima tela.</li> <li>3. Jogador visualiza tela “<i>Main Lobby</i>”.</li> <li>4. Na tela <i>Main Lobby</i>, jogador seleciona opção “Criar Mesa”.</li> <li>5. Na tela “<i>Create Table</i>” jogador preenche os campos de nome da mesa, nome do jogador e quantidade iniciais de fichas.</li> <li>6. Jogador clica no botão “Criar Mesa”.</li> <li>7. Jogador visualiza <i>pop-up</i> de “Aguardando outros jogadores conectarem...”.</li> </ol> |
| <b>Fluxos alternativos</b> | <ol style="list-style-type: none"> <li>1. Sistema não consegue se conectar a outro aparelho, a tela apresenta mensagem de erro e volta para o passo três do fluxo principal.</li> <li>2. Jogador clica em cancelar, espera por outros jogadores é finalizado e tela volta para passo três do fluxo principal.</li> </ol>  |

Tabela 6: Descrição do caso de uso “Juntar-se à mesa”

|                            |  |
|----------------------------|--|
| <b>Nome do caso de uso</b> | Juntar-se à Mesa   |
| <b>Sumário</b>             | Jogador entra em uma mesa no jogo <i>Smart Poker</i>   |
| <b>Ator primário</b>       | Jogador  |
| <b>Precondições</b>        | O jogador está com o aplicativo instalado no seu dispositivo Android e conexão de <i>Bluetooth</i> ativada   |
| <b>Fluxo principal</b>     | <ol style="list-style-type: none"> <li>1. Jogador inicia o jogo utilizando o ícone disponível na área de aplicativos do aparelho.</li> <li>2. O jogador visualiza a tela de apresentação e esperar o sistema carregar a próxima tela.</li> <li>3. Jogador visualiza tela “<i>Main Lobby</i>”.</li> <li>4. Na tela <i>Main Lobby</i>, jogador preenche seu nome no campo “Nome:” e seleciona a opção “Juntar-se à Mesa”.</li> <li>5. Na tela “<i>Scan Devices</i>” o jogador seleciona o dispositivo <i>Bluetooth</i> que criou a mesa e no qual irá se conectar.</li> <li>6. Jogador visualiza a tela “<i>Poker Table</i>” e o jogo é iniciado.</li> </ol> |
| <b>Fluxos alternativos</b> | <ol style="list-style-type: none"> <li>3. Sistema não consegue se conectar a outro aparelho, a tela apresenta mensagem de erro e volta para o passo três do fluxo principal.</li> <li>4. Jogador clica em voltar, e a tela volta para passo três do fluxo principal.</li> </ol>  |

### 3.4 Requisitos Funcionais e Não Funcionais

A seguir serão apresentados os requisitos funcionais e não funcionais do jogo "*Smart Poker*".

#### 3.4.1 Requisitos Não Funcionais

- Dispositivo com sistema operacional Android versão 2.3 (SDK 9) ou superior.
- Dispositivo com tela de quatro ou de 10.1 polegadas.
- Mínimo de 5(cinco) *Megabytes* de espaço livre para instalação.
- Tela *Touch screen*.
- Dispositivo com *Bluetooth*.
- Todas as imagens do jogo estão no formato PNG.

#### 3.4.2 Requisitos Funcionais

- O jogo deve ter suporte à conexão *Bluetooth*.
- O jogo deve ter suporte a modo *multiplayer* com dois jogadores.
- O jogo deve distribuir as cartas para os jogadores de maneira aleatória e sem repetir cartas em uma mesma rodada.
- Ao final da rodada o jogo deve apresentar o vencedor e atribuir as fichas que estavam no(s) pote(s) das apostas para o mesmo.
- O jogo deve suportar consulta as jogadas e ao estado do jogo em cada turno de uma rodada.
- O jogo deve apresentar as opções para o jogador de pagar uma aposta (*CALL*), fugir de uma aposta ou não apostar nada (*FOLD / CHECK*), aumentar uma aposta (*RAISE*) e apostar todas suas fichas (*ALL IN*).
- Não existe um tempo máximo para um jogador realizar uma ação no jogo.
- O jogo deve apresentar uma tela de menu com opções de criar uma nova mesa ou entrar em uma mesa criada por outro jogador.
- O menu para criar uma nova mesa deverá ter três campos: nome da mesa; nome do jogador e quantidade inicial de fichas.
- Para entrar em uma mesa, o jogador deverá preencher um campo de nome.
- O jogo não permitirá entrar em uma mesa com o jogo já iniciado.

- O jogo só termina quando algum jogador perder todas suas fichas ou desistir do jogo.
- Para iniciar um novo jogo é necessário sair da mesa e criar uma nova.
- O jogo se iniciará quando for estabelecida uma conexão *Bluetooth* entre os dois jogadores.
- Em caso de empate as fichas serão divididas pelo número de ganhadores.
- O jogo deve seguir as regras do Pôquer *Texas Hold'em No Limit*.

## 4. RESULTADOS OBTIDOS

Foi realizada uma análise da dinâmica do jogo de Pôquer *Texas Hold'em*, suas regras e particularidades para então entender quais passos seguir e os componentes e bibliotecas do Android que seriam necessários para a programação do aplicativo.

O desenvolvimento do protótipo foi dividido em cinco partes, primeiramente foi realizada a lógica geral sobre um jogo de Pôquer, que envolve a distribuição das cartas, embaralhar, verificar apostas, turnos e validar a mão vencedora.

Pode-se dizer que a parte de maior dificuldade foi validar a mão vencedora, devido à complexidade dos passos que isso engloba. Foi necessário primeiramente, validar qual o jogo que o usuário fez, em seguida validar se aquele jogo era maior ou menor do que dos seus adversários, caso fosse o mesmo jogo teria que ser validado então qual dos jogos tinham cartas maiores, por exemplo se dois jogadores tivessem um par, validar qual par era de carta maior, ainda sim poderiam ter um par da mesma carta, seria necessário então checar carta por carta, verificando a maior carta (*kicker*) dos jogadores até chegar a um desempate e se caso continuasse empatado seriam então divididas as fichas.

A segunda parte foi a interface gráfica do jogo, primeiramente foi decidido criar uma mesa alinhada a esquerda da tela e no topo, deixando um espaço livre no rodapé e no canto direito, que seriam utilizados para mostrar as opções de ação e *status* do jogo, respectivamente. Essa abordagem gerou um problema comum com relação a desenvolver para dispositivos móveis, espaço. Não sobrou espaço suficiente na tela (telas maiores como de *tablets* não teriam problema), para dispor todos os componentes de uma mesa de Pôquer, fichas do usuário, nome, cartas do jogador, as cinco cartas da mesa, mais botões e etc. Então foi realizada uma pesquisa sobre componentes do Android, e o componente *SlidingDrawer* solucionou o problema de espaço permitindo que os botões de ação e *status* do jogo aparecessem somente quando necessários.

A terceira parte do desenvolvimento, foi criar as telas precedentes ao jogo, tela de menu, de criar uma mesa ou entrar em uma mesa já criada e a *splash screen*.

A quarta etapa foi desenvolver o lado cliente do jogo, este possuía poucas classes e uma *activity*, o objetivo do lado cliente é atualizar a tela com dados recebidos do servidor e tratar e enviar as ações para o servidor, onde iria ser feito o processamento.

A quinta e última etapa foi criar a arquitetura de manter o estado do jogo utilizando *bluetooth*. Foram utilizadas as classes padrões do Android para conexão com *bluetooth*, onde o cliente e o servidor tinham uma instância de *BluetoothSocket* que é responsável por enviar e receber mensagens. Cada mensagem enviada é uma grande *string* que é composta por vários dados em posições determinadas previamente, onde cada posição detém um tipo de dado específico, como por exemplo: cartas do jogador receptor, número de fichas atualizadas e etc. Os dados foram separados por ";", quando o servidor ou cliente verifica que uma nova mensagem chegou é feito o tratamento dessa mensagem, criando atributos para serem utilizados na atualização do *status* do jogo.

A dificuldade encontrada nesta etapa foi realizar testes, para o mesmo foi necessário utilizar dois aparelhos com Android, já que o emulador de Android não consegue emular dois aparelhos *bluetooth* conectados.

O jogo contém cinco telas e foi testado utilizando um tablet Samsung Galaxy e um celular Galaxy SII.

O resultado de cada tela é apresentado a seguir, a figura 13 mostra a tela "*Splash Screen*" que é apresentada quando o usuário clicar no ícone do jogo:



Figura 13: Tela de *splash screen* do jogo "*Smart Poker*"

A figura 14 mostra a tela "*Main Lobby*", onde é apresentado o menu com um botão para criar uma mesa ou um campo de texto para digitar o nome do jogador e o botão "*Join Table*" que entra em uma mesa já existente.

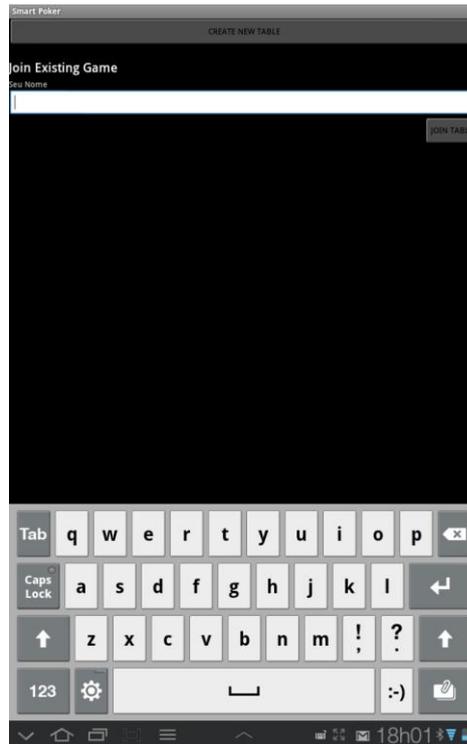


Figura 14: Tela *Main Lobby* do jogo "Smart Poker"

A figura 16 mostra a tela "*Create Table*", em que são apresentados campos de texto para o usuário digitar o nome ou apelido do seu jogador na mesa, um campo para digitar o nome da mesa e outro para o número inicial de fichas.

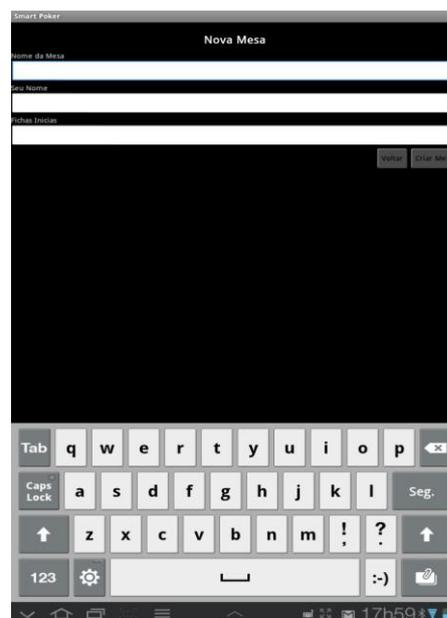


Figura 15: Tela *Create Table* do jogo "Smart Poker"

A figura 16 apresenta a tela "Scan Devices", em que o jogador que clicou em "Join Table" (entrar em uma mesa), irá selecionar o dispositivo *bluetooth* do usuário que criou a mesa.

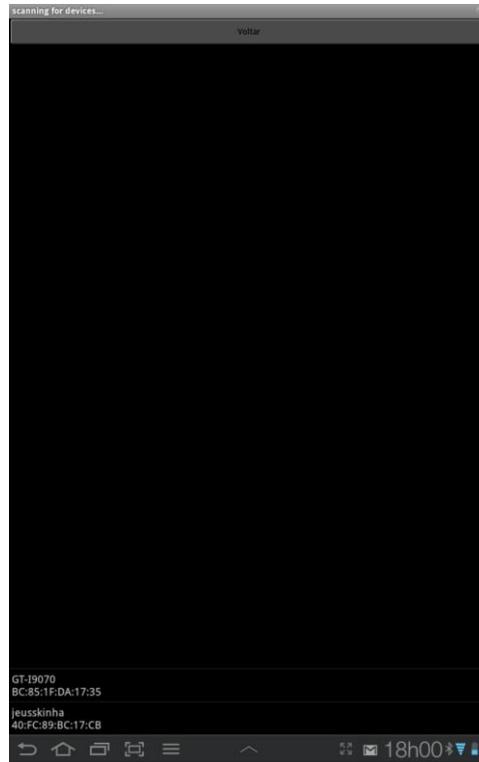


Figura 16: Tela *Scan Devices* do jogo "Smart Poker"

A figura 17 apresenta a tela "Poker Table", que é o local em que o jogo de Pôquer é realizado.



Figura 17: Tela *Poker Table* do jogo "Smart Poker"

## 5. CONSIDERAÇÕES FINAIS

A pesquisa realizada neste trabalho sobre o Android e o Pôquer *Texas Hold'em* demonstrou que a plataforma Android é robusta o bastante para permitir o desenvolvimento de um jogo de Pôquer com uma boa jogabilidade e uma interface gráfica de qualidade, como encontrada em uma plataforma PC.

Além disso, foi possível desenvolver uma arquitetura de sincronização de estado do jogo, utilizando *bluetooth*, eficaz e com um tempo de resposta rápido.

Para o jogo ser compatível com dispositivos de poucas polegadas, foram realizadas adaptações no tamanho das letras e naipes das cartas, foi aumentado o tamanho das margens dos botões, com o intuito de facilitar o clique com o toque, e também alterado o fundo do histórico do jogo para utilizar uma cor translúcida, permitindo assim o jogador visualizar suas cartas e a mesa quando o histórico estivesse em primeiro plano.

A utilização de *bluetooth* no jogo, mostrou-se de uma grande utilidade, possibilitando que duas pessoas joguem um jogo de Pôquer sem necessidade de uma conexão 3G ou *wi-fi*.

Com os conceitos apresentados neste projeto foi desenvolvido com sucesso um jogo de Pôquer *multiplayer*, utilizando *bluetooth* e com uma interface amigável, que permite rodar o aplicativo mesmo em dispositivos com uma tela de poucas polegadas, sem perder a qualidade gráfica ou ter sido necessário alterar a disposição dos componentes.

Possíveis trabalhos futuros para o projeto estão inclusos: aumentar a capacidade de jogadores de dois para seis jogadores; melhorar a interface gráfica de menus e telas iniciais e criar um *ranking* (classificação) das vitórias de cada jogador.

## REFERÊNCIAS

LECHETA, Ricardo R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. São Paulo: Novatec, 2009.

ANDROID, Story disponível em: <<http://www.xcubelabs.com/the-android-story.php>>. Acesso em: 18 de maio 2012.

GOOGLE, Play disponível em: <<http://support.google.com/googleplay/android-developer>>. Acesso em: 17 de maio 2012.

BLUETOOTH, How it Works disponível em: <<http://www.bluetooth.com/Pages/Basics.aspx>>. Acesso em: 12 de maio 2012.

ANDROID, Activities disponível em: <<http://developer.android.com/guide/components/activities.html>>. Acesso em: 05 de maio 2012.

BELLO, Leo. **Aprendendo a Jogar Poker: Princípios, Técnica & Prática**. São Paulo: Novera, 2007.

INFOGRAFIC, Android disponível em: <<http://www.androidauthority.com/evolution-of-the-android-market-infographic-35881>>. Acesso em: 05 de maio 2012.

CARLEZZO, Eduardo. **Poker: regulamentação do jogo no Brasil**. Carlezzo Advogados, São Paulo, 3 maio 2011. Disponível em: <<http://www.carlezzo.com.br/pt/ler-publicacao.php?id=11>>. Acesso em: 13 de maio 2012.

MAHON, Eduardo. Poker não deve ser considerado contravenção. Dr Otávio de Queiroga blog, 23 jan 2011. Disponível em: <<http://otaviodequeiroga.blogspot.com.br/2011/01/poker-nao-deve-ser-considerado.html>>. Acesso em: 23 de maio 2012.

POKER, Poker Stars disponível em: <[www.pokerstars.com/br/](http://www.pokerstars.com/br/)>. Acesso em: 13 de junho 2012.

UML, Use Case Diagrams. Disponível em: <<http://www.uml-diagrams.org/use-case-diagrams.html>> Acesso em: 06 de junho 2012.

MAGES, Engine. Disponível em: <<http://code.google.com/p/mages/>> Acesso em: 10 de abril 2012.

ENGINE, Software Engine. Disponível em: <<http://www.wisegeek.com/what-is-a-software-engine.htm>> Acesso em: 08 de junho 2012.