

Desenvolvimento de um jogo multiplayer para celular

Vilson Trupel

Orientador: Prof. Dr. Mauro Marcelo Mattos

Sumário

- ▶ **Introdução**
- ▶ **Objetivos do trabalho**
- ▶ **Fundamentação teórica**
- ▶ **Desenvolvimento do trabalho**
- ▶ **Resultados**
- ▶ **Conclusão**

Introdução

- ▶ Mercado promissor e crescente.
- ▶ Alto número de vendas de aparelhos celulares.
- ▶ Aprimoramento contínuo de tecnologias de desenvolvimento e hardware.
- ▶ Existência inexpressiva de jogos *multiplayer* para celulares se comparado a PCs.
- ▶ Superar desafios tecnológicos como a alta latência nas redes e a mudança de paradigmas dos jogadores.

Ano	Faturamento
2007	US\$ 3,9 bilhões
2008	US\$ 4,5 bilhões
2009	crescimento de 10,2%

Objetivos do trabalho

- ▶ Desenvolvimento de um cliente do jogo Batalha Naval para jogo *multiplayer* no celular.
- ▶ Desenvolvimento de uma aplicação servidora que rode em um PC.
- ▶ Utilização da plataforma J2ME.

Fundamentação teórica

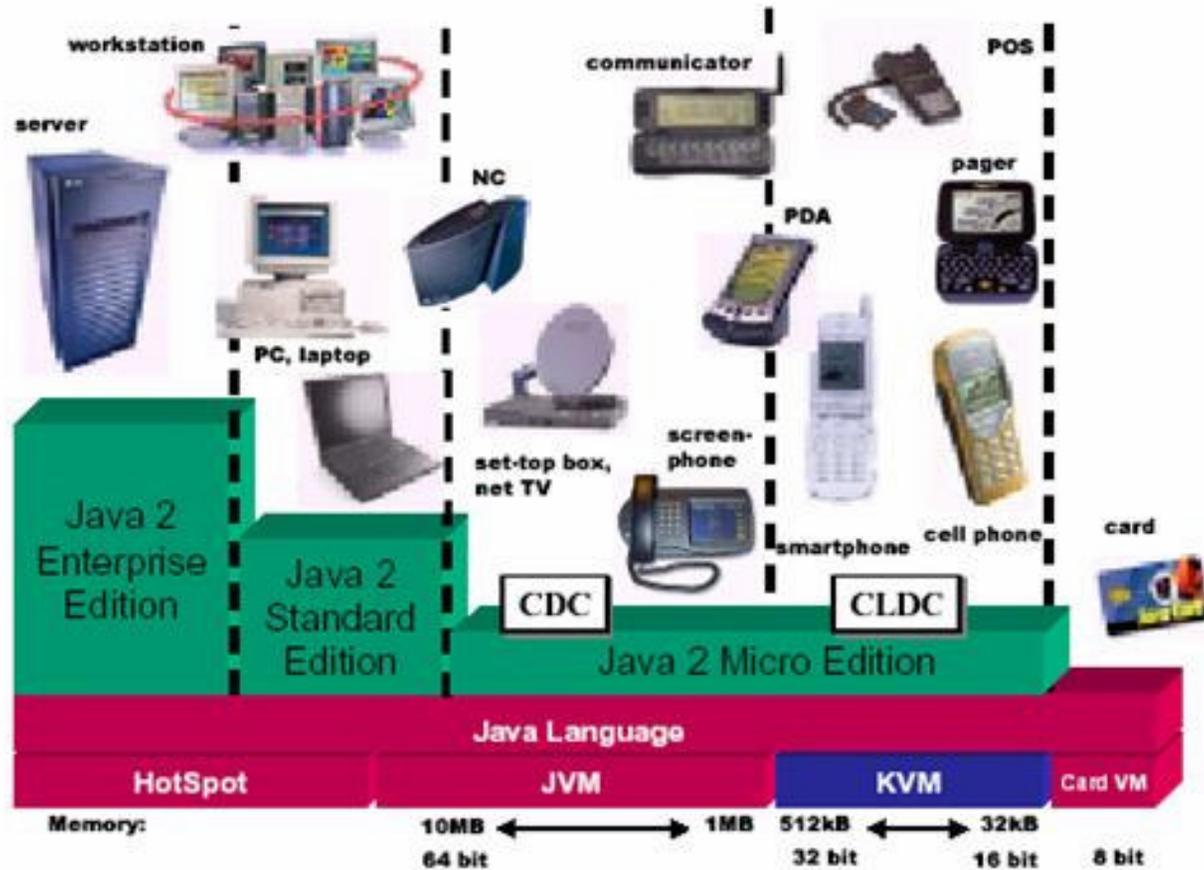
- ▶ Tecnologia JAVA.
- ▶ Plataforma J2ME.
- ▶ Arquitetura de servidores.
- ▶ Jogos *multiplayer*.
- ▶ Trabalhos correlatos.

Tecnologia JAVA

► Dividida em três plataformas:

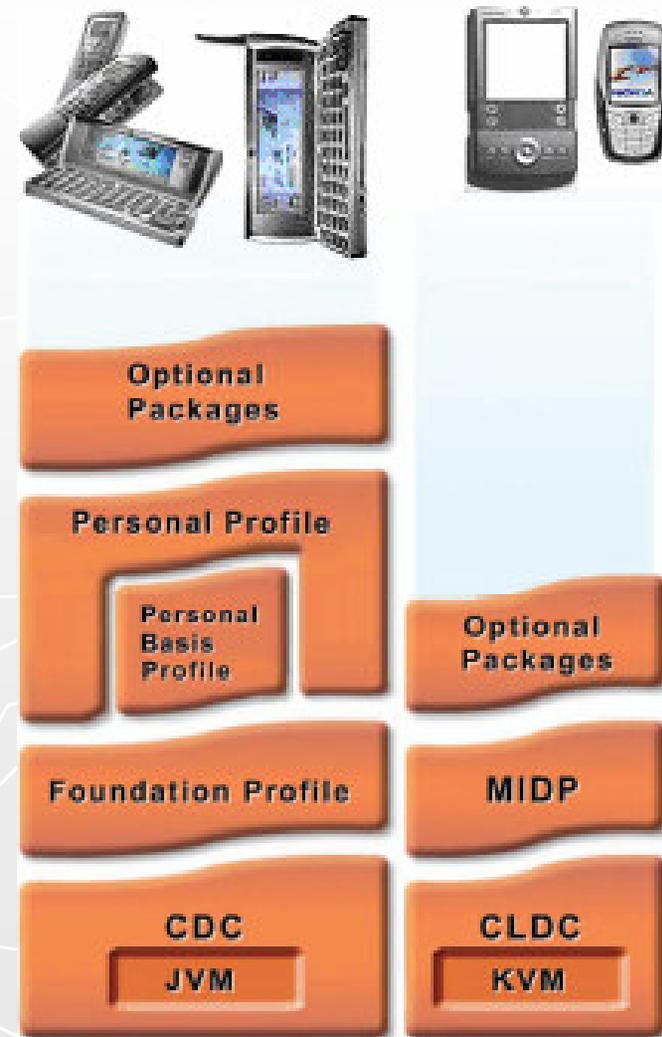
- J2SE – destinada a máquinas pessoais e estações de trabalho.
- J2EE – destinada a aplicativos baseados no servidor.
- J2ME – destinada a dispositivos com memória, vídeo e processamento limitados.

Tecnologia JAVA



Plataforma J2ME

- ▶ Os dispositivos que suportam esta tecnologia variam desde *paggers* e celulares até grandes aparelhos domésticos como televisores digitais, refrigeradores e automóveis.
- ▶ Dividida em configurações e perfis.



Configurações

- ▶ Conceito introduzido pela SUN, para suportar uma ampla variedade de produtos que se encaixam dentro do escopo do J2ME.
- ▶ A linha divisória imaginária de uma configuração é, de modo geral, baseada na memória, no vídeo e na conectividade de rede e no poder de processamento do dispositivo
- ▶ Há basicamente, duas configurações: CDC e CLDC.

Configurações

▶ *Connected Device Configuration (CDC)*

- Voltada para dispositivos com no mínimo 512KB de memória, 256KB de memória dinâmica além de conectividade de rede e largura de banda persistente e alta.

▶ *Connected Limited Device Configuration (CLDC)*

- Voltada para dispositivos com 128KB de memória, 32KB de memória dinâmica, baixo poder de processamento, conectividade de rede geralmente sem fio e largura de banda baixa.

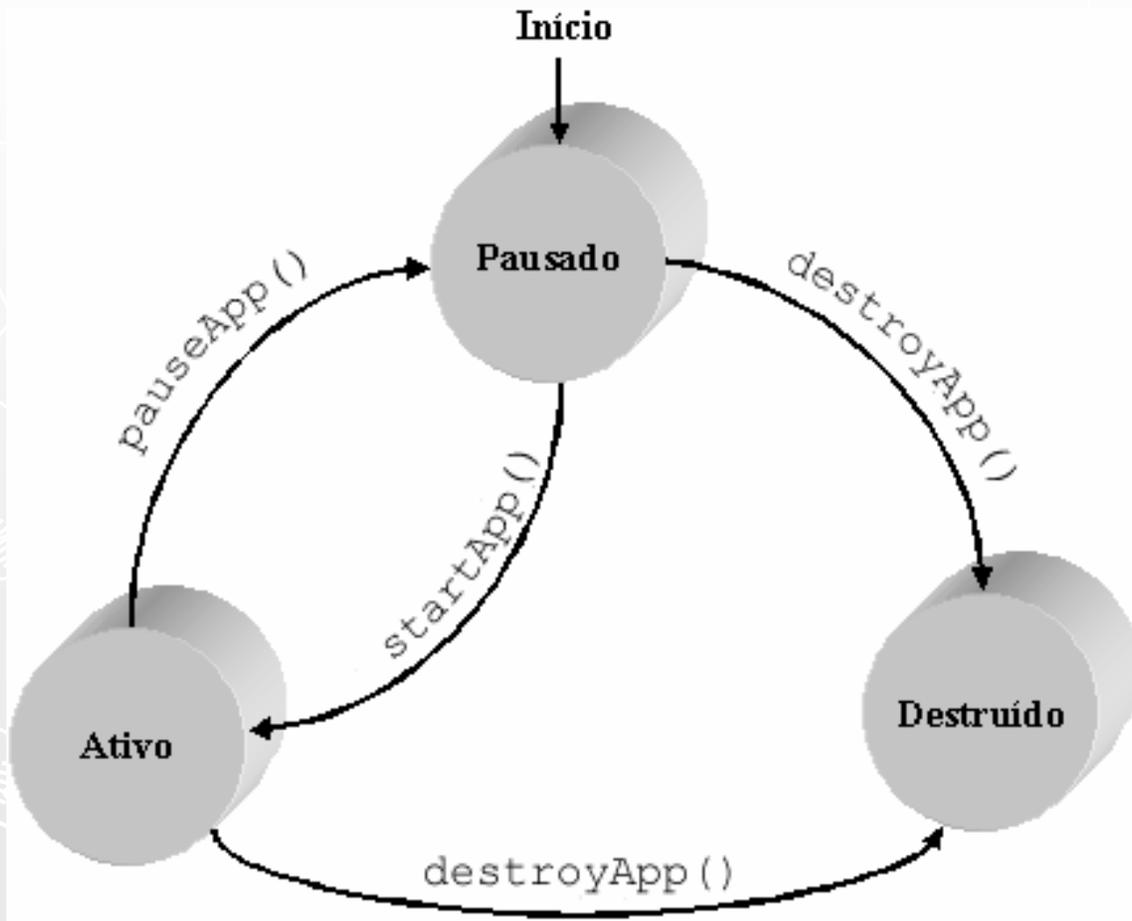
Perfil MIDP

- ▶ Atende a demandas específicas de uma certa família de dispositivos.
- ▶ O perfil MIDP foi o primeiro perfil definido para a plataforma J2ME, tendo sido lançado sobre a configuração CLDC.

MIDLET

- ▶ É uma aplicação que implementa o perfil MIDP e roda sobre uma máquina virtual *Kilobyte Virtual Machine* (KVM) proposta pela configuração CLDC.
- ▶ Uma aplicação MIDP deve conter no mínimo uma classe MIDLET.
- ▶ Compreende três estados, onde cada um representa o estado da aplicação: ativo, pausado e destruído.

Estados de uma MIDLET



Exemplo de uma MIDLET

```
import javax.microedition.midlet.*;
public class MidletBasica extends MIDlet {
    public void MidletBasica() {
        System.out.println("Construtor");
    }
    public void startApp() {
        System.out.println("Início");
    }
    public void pauseApp() {
        System.out.println("Pausa");
    }
    public void destroyApp(boolean unconditional) {
        System.out.println("Fim");
    }
}
```

Arquitetura de servidores

- ▶ Existem algumas arquiteturas padrão para suportar jogos *multiplayer*.
 - *peer-to-peer unicast*: não empregam boa escalabilidade. Cada mudança, tem que ser comunicada as N-1 estações;
 - *broadcast e multicast*: basicamente ainda uma arquitetura *peer-to-peer* com a diferença que somente envia-se uma mensagem de atualização;
 - *cliente/servidor*: separa as aplicações clientes de aplicações servidoras sendo interligadas entre si, geralmente utilizando uma rede de computadores.

Jogos multiplayer

- ▶ Manter um estado global consistente entre os jogadores.
- ▶ Pontos importantes a serem analisados:
 - latência fim-a-fim
 - largura de banda
 - consistência de estados
- ▶ Preocupar-se com:
 - jogabilidade/interatividade
 - interface (tamanho do teclado e *display*)
 - mudança no perfil dos jogadores

Trabalhos correlatos

TRABALHOS CORRELATOS				
	<i>Multiplayer</i>	MMO	Plataforma	Arquitetura
Batalha Naval	Sim	Servidor: Sim Jogo: Não	Celular	Cliente/Servidor
Cellmons	Sim	Não	Celular	Centralizado
Galaxy Navigator	Não	Sim	Celular	Distribuído
Tetris	<i>Não</i>	Não	Celular	Centralizado

Desenvolvimento do trabalho

- 
- ▶ Requisitos
 - ▶ Especificação
 - ▶ Implementação

Desenvolvimento do trabalho

► Requisitos:

- o jogador deverá informar seu nome e sua preferência pelo jogo individual (contra outro jogador) ou em duplas (Requisitos Funcionais – RF).
- o servidor deverá monitorar uma porta específica para o registro de novos jogadores (RF).
- o servidor deverá monitorar uma porta para cada jogador que já esteja registrado (RF).
- o servidor deverá associar os jogadores aos adversários conforme a preferência escolhida por cada um (RF).
- o servidor não deverá implementar o conceito de *ranking* (RF).

Desenvolvimento do trabalho

► Requisitos:

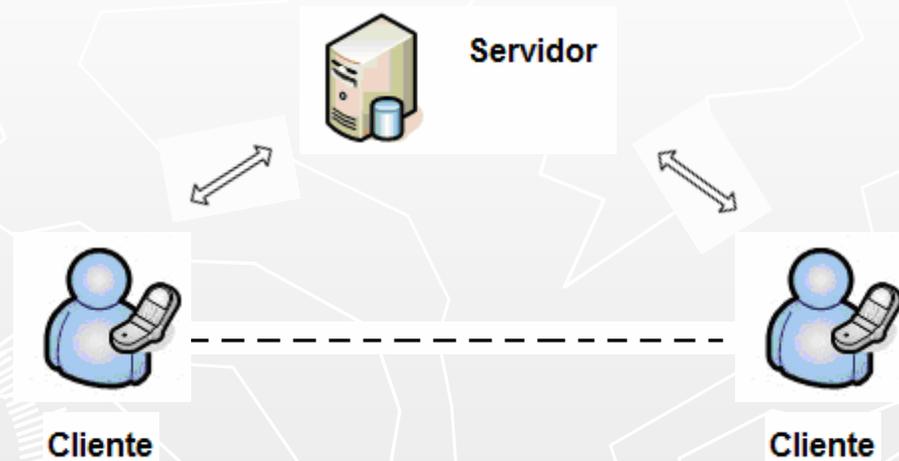
- o jogador deverá submeter seus dados ao servidor a cada nova rodada do jogo (RF).
- o jogo deverá permitir uma interação entre o jogador e seu avatar (RF).
- o servidor do jogo deverá ser implementado na linguagem de programação Java (Requisito Não Funcional – RNF).
- o jogo deverá ser implementado na plataforma J2ME (RNF).
- deverá ser utilizado no desenvolvimento do jogo a configuração CLDC e o perfil MIDP 2.0 (RNF).
- a comunicação entre o servidor e o cliente do jogo, deverá ser feito utilizando *socket* (RNF).

Desenvolvimento do trabalho – Especificação

- ▶ Especificação desenvolvida utilizando Enterprise Architect.
- ▶ Programação Orientada a Objeto.

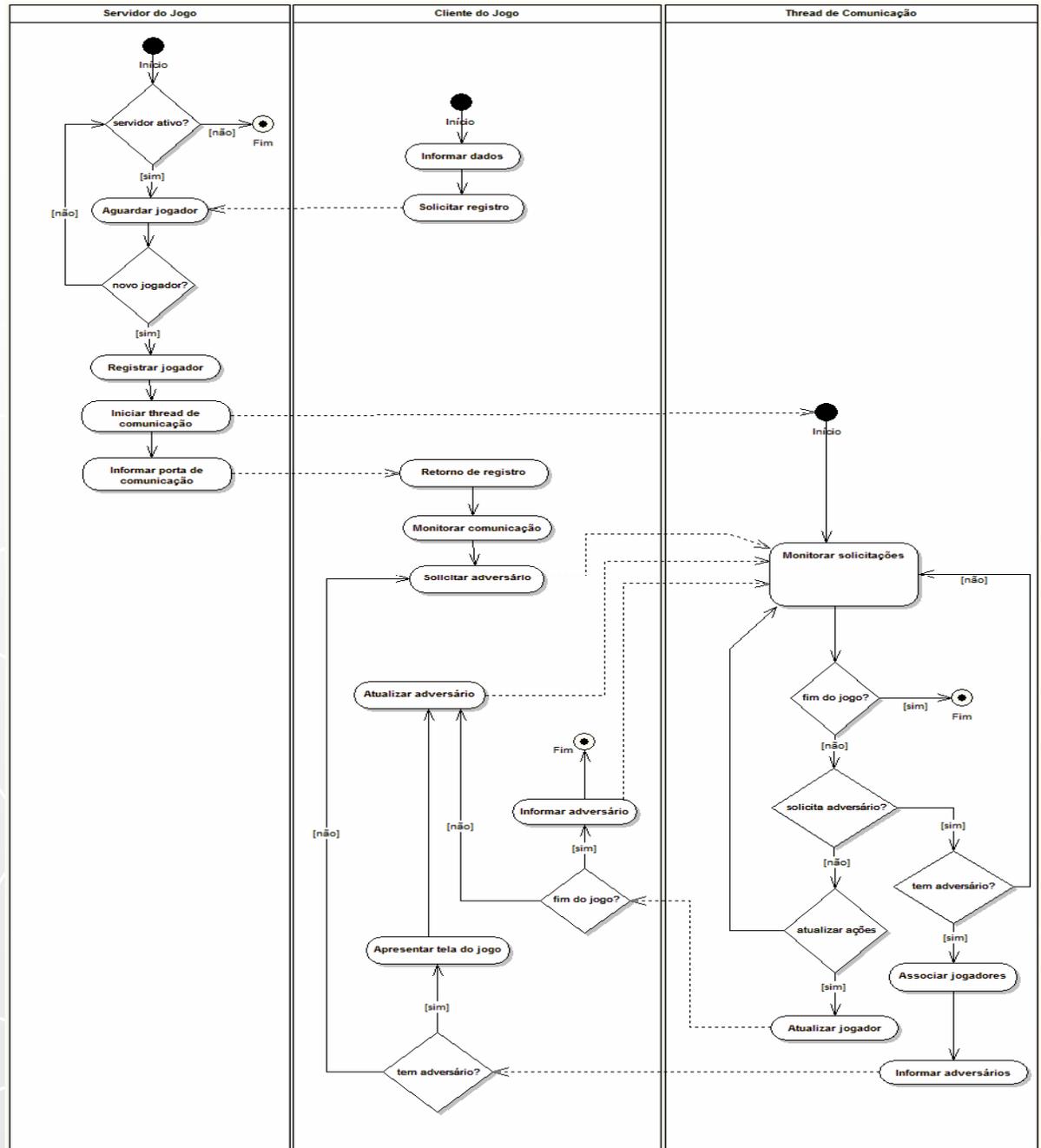
Desenvolvimento do trabalho – Especificação

Entendendo o problema:



Desenvolvimento do trabalho – Especificação

Diagrama de Atividade



Desenvolvimento do trabalho – Especificação

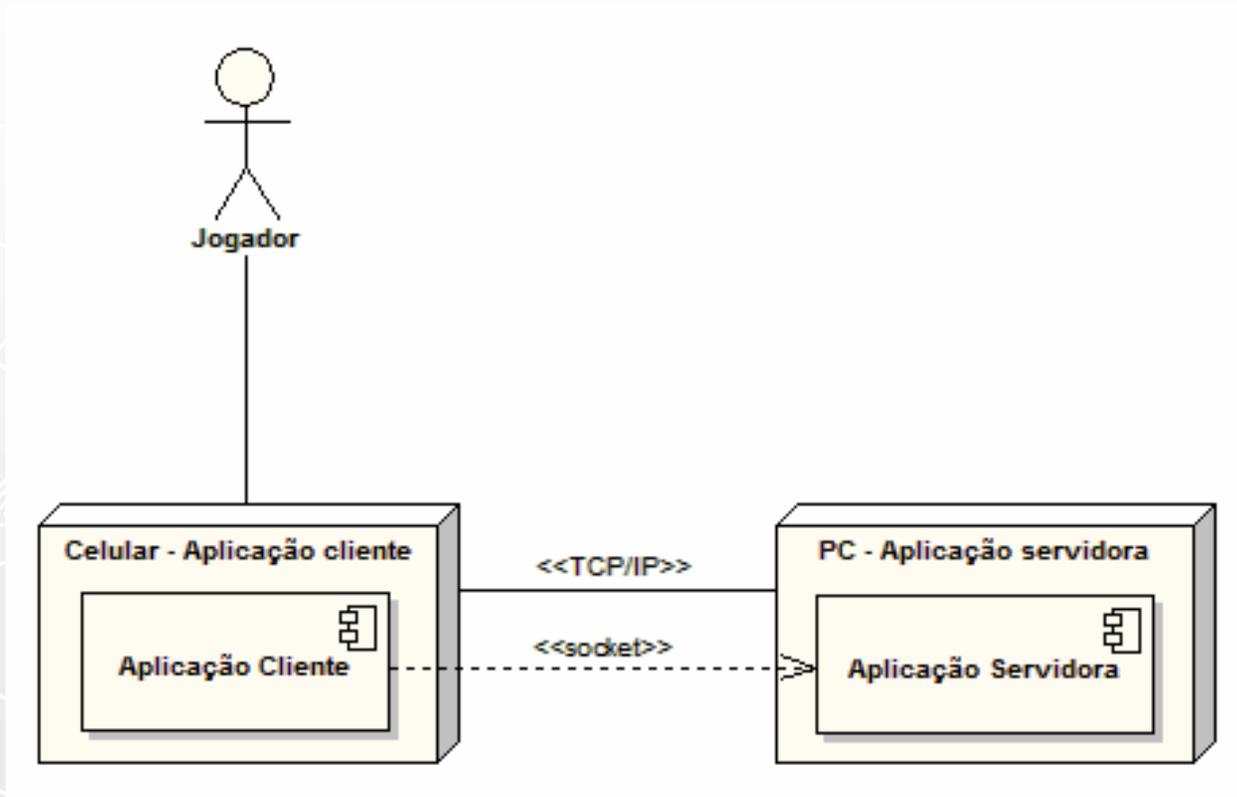
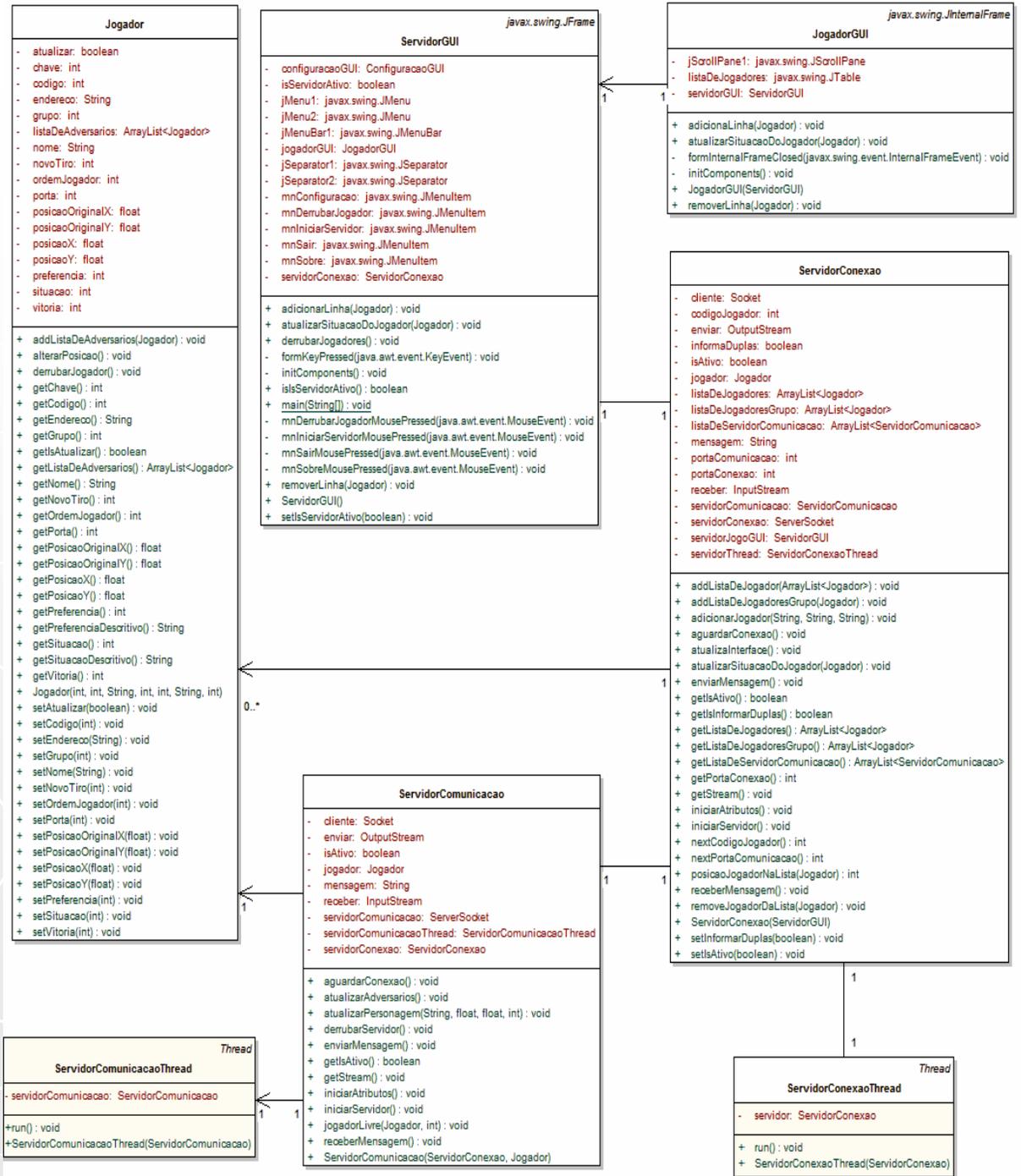


Diagrama de Disposição

Desenvolvimento do trabalho – Especificação

Diagrama de classe do servidor



Desenvolvimento do trabalho – Especificação

Diagrama de classe do cliente



Desenvolvimento do trabalho – Implementação

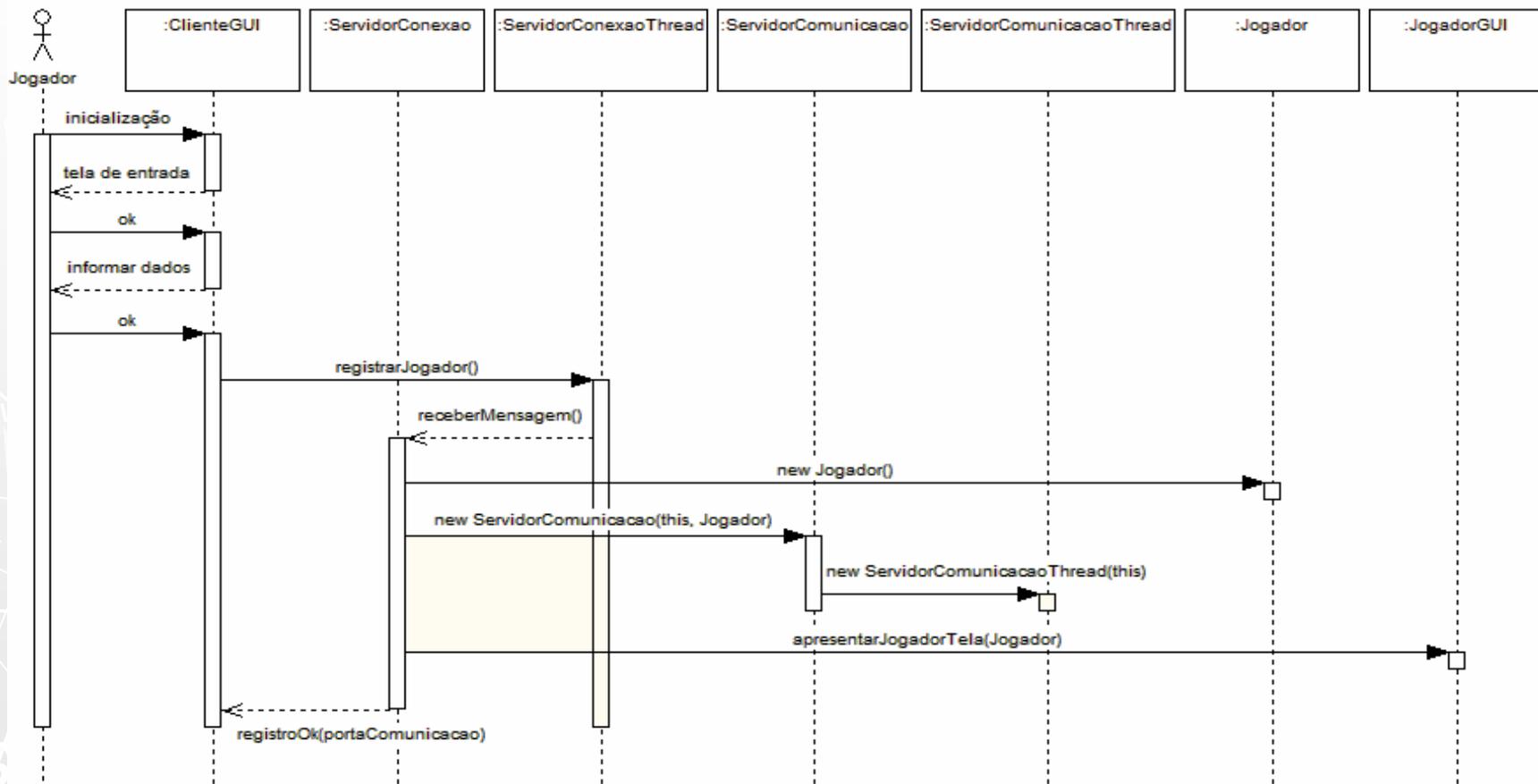


Diagrama de Sequencia

Desenvolvimento do trabalho – Implementação

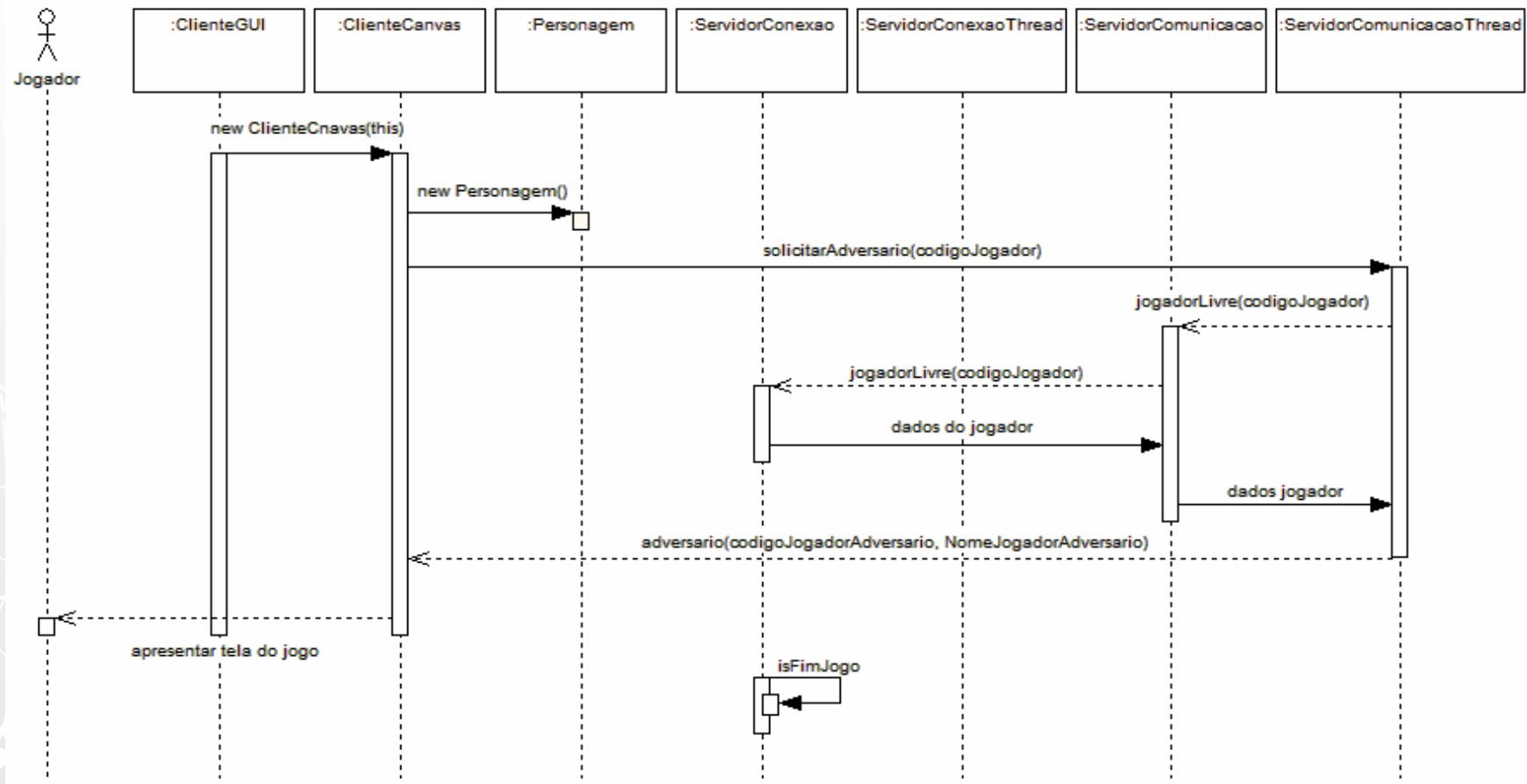


Diagrama de Sequencia

Desenvolvimento do trabalho – Implementação

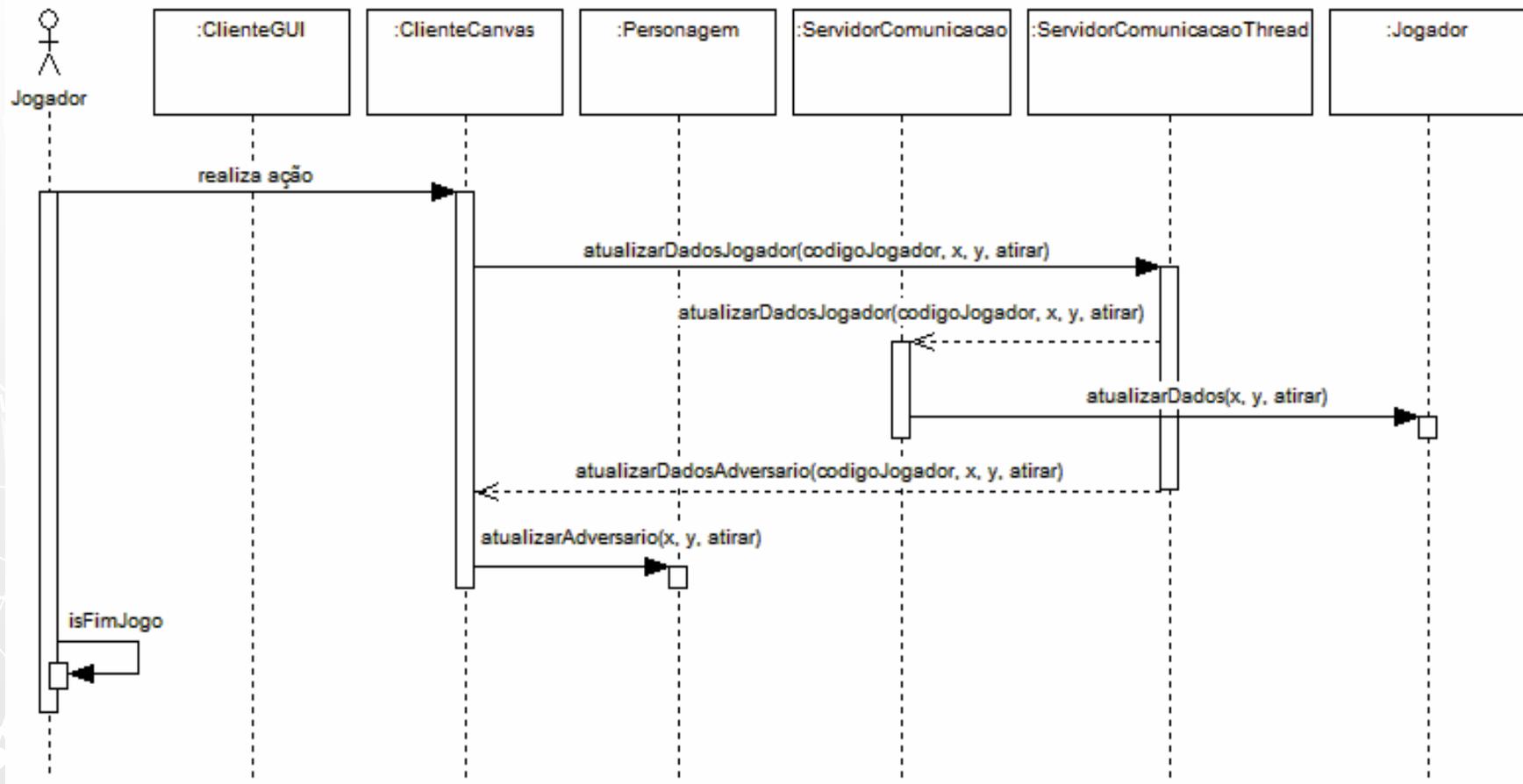


Diagrama de Sequencia

Desenvolvimento do trabalho – Implementação

▶ Netbeans 6.1

▶ Java JDK 1.6

▶ MIDP 2.1

▶ CLDC 1.1

Conclusão

- ▶ Resultados obtidos satisfatórios.
- ▶ O desenvolvimento de jogos *multiplayer* é viável e promissor apesar de poucos jogos nesta linha.
- ▶ O perfil MIDP 2.0 agregado a configuração CLDC, torna possível o desenvolvimento de jogos mais complexos e interativos.
- ▶ Dificuldades:
 - Conexão entre a aplicação cliente e o servidor utilizando *socket*.

Extensões

- ▶ o desenvolvimento de uma base de dados para registrar os jogadores implementando um controle de *login* para identificar os jogadores.
- ▶ desenvolver um *ranking* para deixar disponível para consulta aos jogadores, aqueles que tem o maior número de pontos e também uma consulta individual no *ranking*.
- ▶ pode ser implementado uma opção para que o jogador possa jogar *sigle-player* sem a necessidade da conexão com o servidor do jogo.