

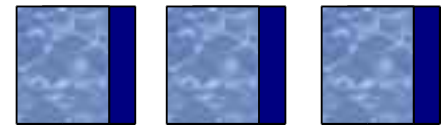
Game API

Simplicidade e poder nos jogos para celulares



Agenda

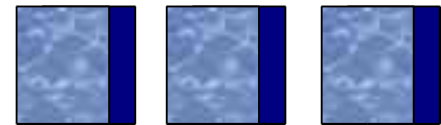
- ⇒ Por que usar a Game API?
- ⇒ Elementos da Game API
- ⇒ Demonstração
- ⇒ Conclusão
- ⇒ Mais informações



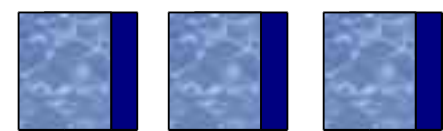
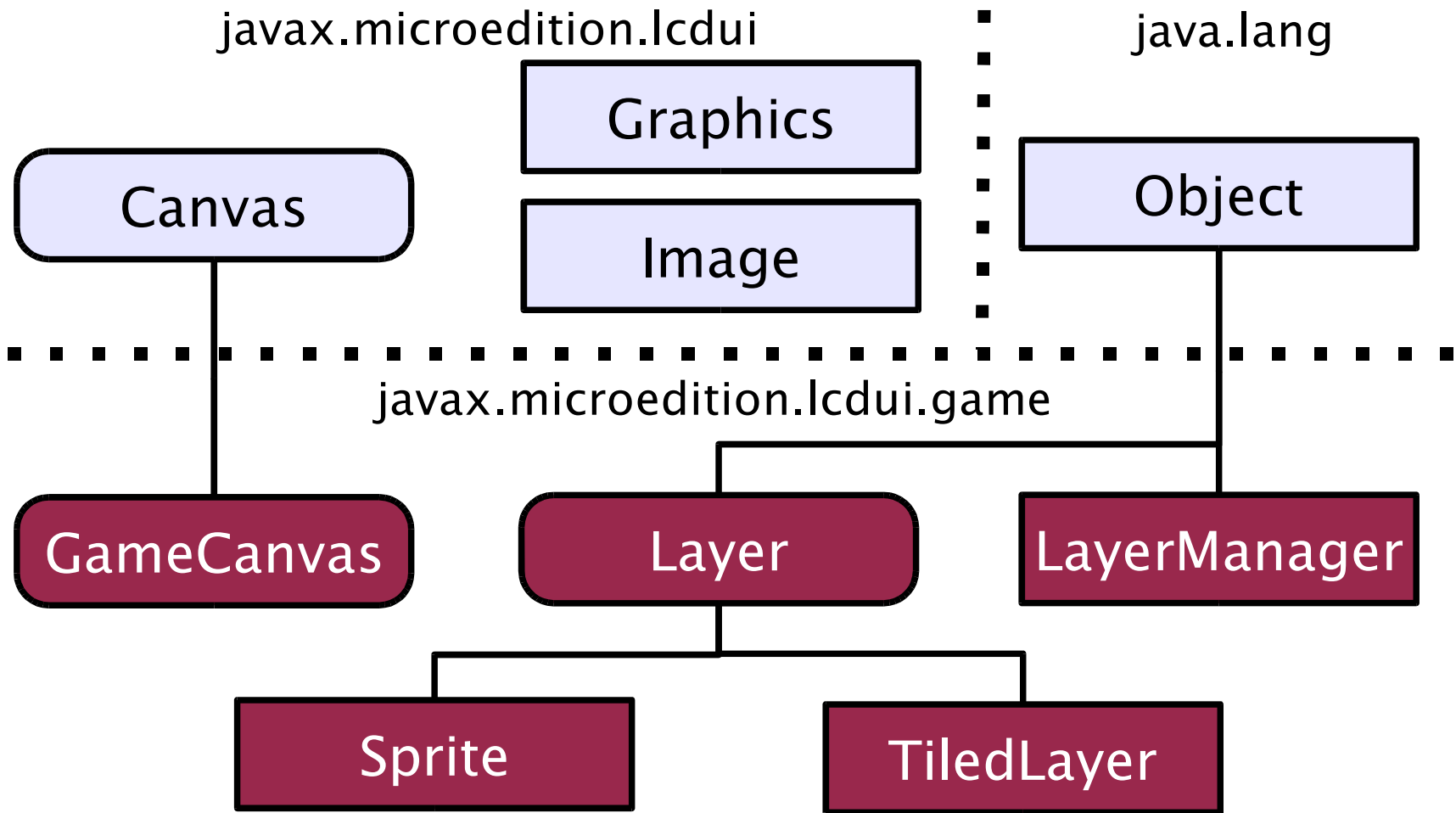
Por que usar a Game API?

*“Remove as barreiras entre sua **imaginação** e seu **jogo**”*

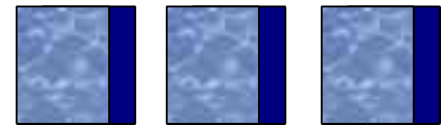
- ⇒ Simplifica o desenvolvimento
- ⇒ Reduz o tamanho da aplicação
- ⇒ Aumenta a performance
- ⇒ Padronização



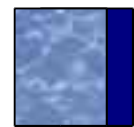
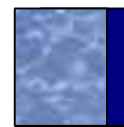
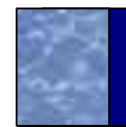
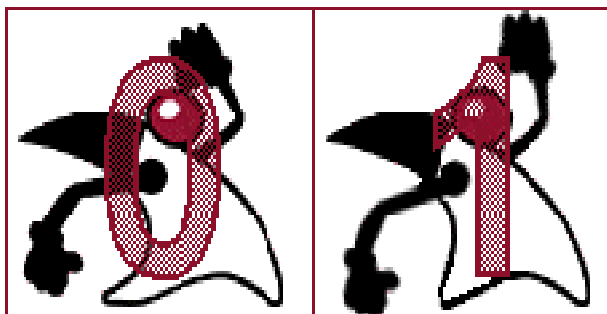
Elementos da Game API



Demonstração



Sprite



Sprite – Construtor

```
import java.io.IOException;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.game.Sprite;

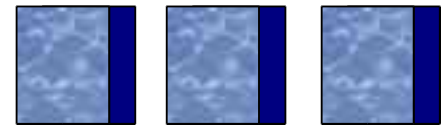
public class DukeSprite extends Sprite {
    private char direcao;

    public DukeSprite() throws IOException {
        super(Image.createImage("/duke.png"), 40, 36);
        direcao = 'd';
        setPosition(20, 20);
        defineReferencePixel(20,18);
    }
}
```



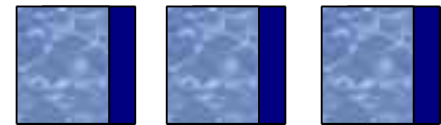
Sprite – Transform

```
public void setDirecao(char c) {  
    direcao = c;  
    switch (direcao) {  
        case 'c':  
            setTransform(TRANS_ROT270);  
            break;  
        case 'b':  
            setTransform(TRANS_ROT90);  
            break;  
        case 'e':  
            setTransform(TRANS_MIRROR);  
            break;  
        case 'd':  
            setTransform(TRANS_NONE);  
            break;  
    }  
}
```

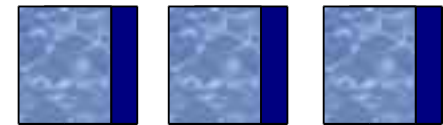
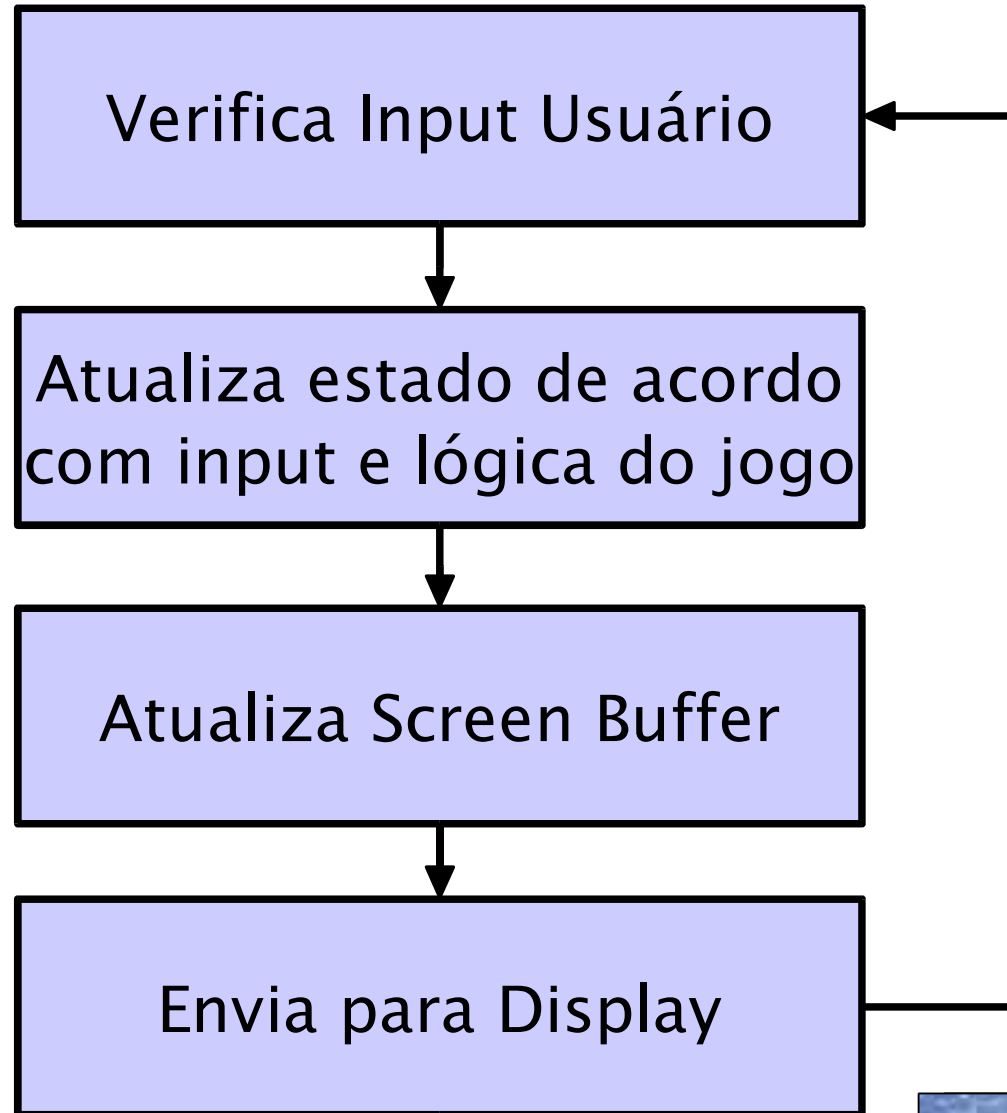


Sprite – Move

```
public void mover(int passos) {  
    switch (direcao) {  
        case 'c':  
            move(0, -passos);  
            break;  
        case 'b':  
            move(0, passos);  
            break;  
        case 'e':  
            move(-passos, 0);  
            break;  
        case 'd':  
            move(passos, 0);  
            break;  
    }  
    this.nextFrame();  
}
```



Fluxo na GameCanvas

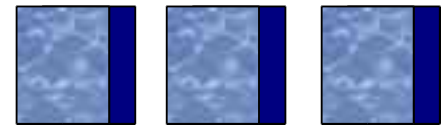


GameCanvas – Construtor

```
public class M4JCanvas extends GameCanvas
implements Runnable {

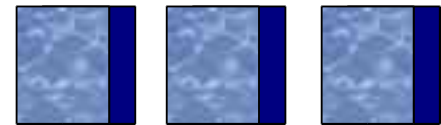
    private final LayerManager layerManager;
    private final DukeSprite duke;
    private boolean executando;

    public M4JCanvas() throws IOException {
        super(true);
        duke = new DukeSprite();
        layerManager = new LayerManager();
        layerManager.append(duke);
    }
}
```

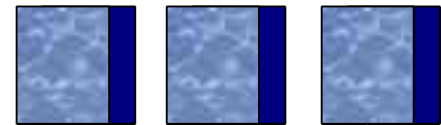
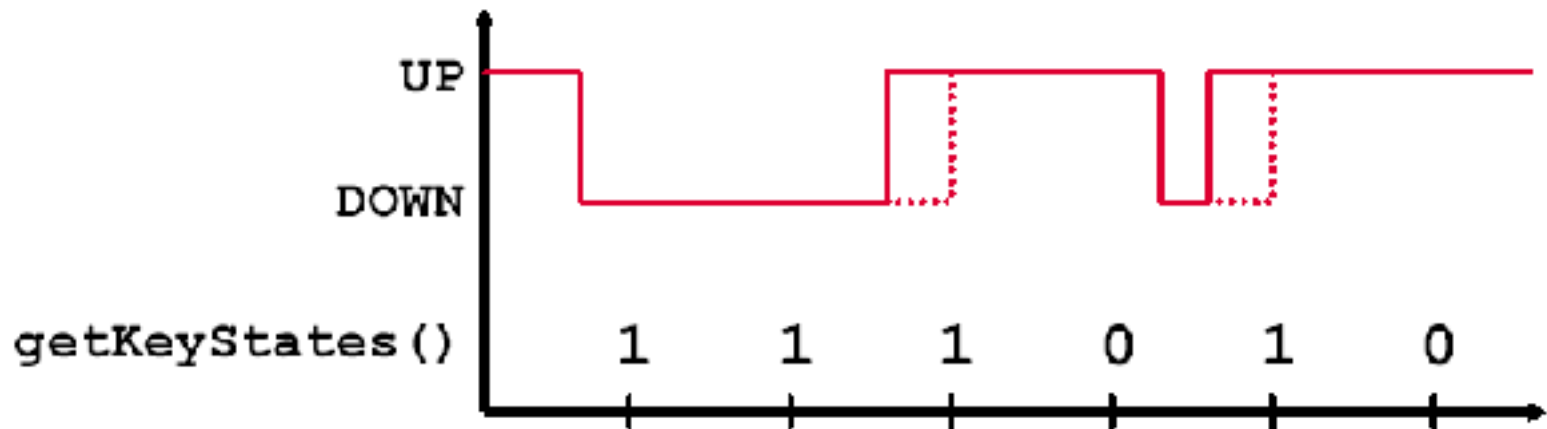


GameCanvas – Input

```
private void movimenta() {
    int keyStates = getKeyStates();
    if ((keyStates & LEFT_PRESSED) != 0) {
        duke.setDirecao('e');
        duke.mover(2);
    } else if ((keyStates & RIGHT_PRESSED) != 0) {
        duke.setDirecao('d');
        duke.mover(2);
    } else if ((keyStates & UP_PRESSED) != 0) {
        duke.setDirecao('c');
        duke.mover(2);
    } else if ((keyStates & DOWN_PRESSED) != 0) {
        duke.setDirecao('b');
        duke.mover(2);
    }
}
```

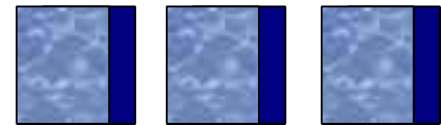


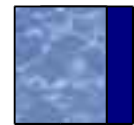
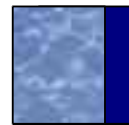
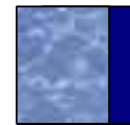
Estados das teclas



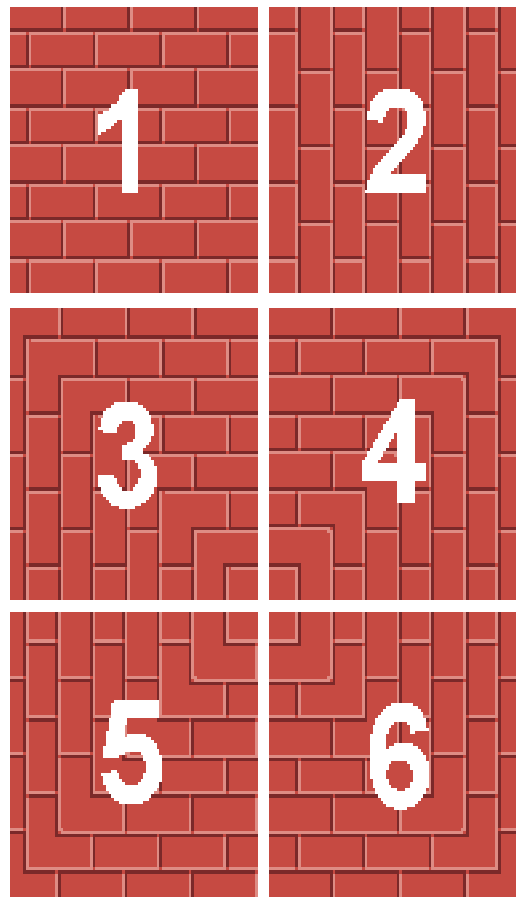
GameCanvas – Run

```
public void run() {  
    Graphics g = getGraphics();  
    int largura = getWidth();  
    int altura = getHeight();  
    while (executando) {  
        movimenta();  
        g.setColor(0xffffffff);  
        g.fillRect(0, 0, largura, altura);  
        layerManager.paint(g, 0, 0);  
        flushGraphics();  
        try {  
            Thread.sleep(100);  
        } catch (InterruptedException ie) {}  
    }  
}
```

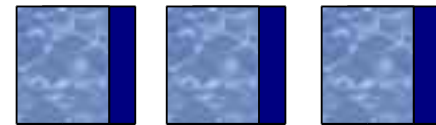




TiledLayer

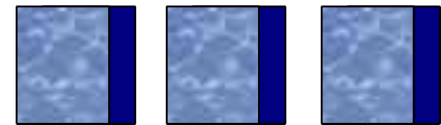


3	1	1	1	1	1	1	1	1	4
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	2
5	1	1	1	1	1	1	1	1	6



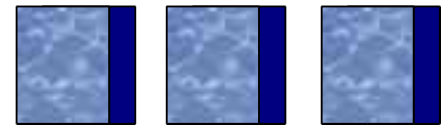
Tiled Layer

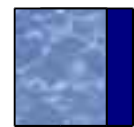
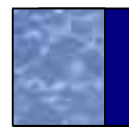
```
private TiledLayer criaParede() throws IOException {
    Image imagem = Image.createImage("/tijolo.png");
    TiledLayer layer = new TiledLayer(10, 10, imagem, 16,
16);
    final int[] mapa = {
        3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4,
        2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
        2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
        ...
        2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
        2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
        5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 6,
    };
    for (int i = 0; i < mapa.length; i++) {
        int coluna = i % 10;
        int linha = (i - coluna) / 10;
        layer.setCell(coluna, linha, mapa[i]);
    }
    return layer;
}
```



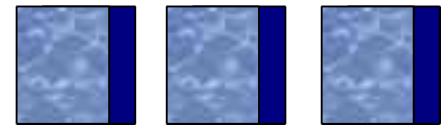
Colisão

```
private void verificaColisaoParede() {  
    if (duke.collidesWith(parede, false)) {  
        duke.mover(-2);  
    }  
}
```

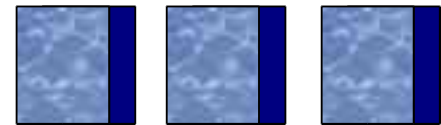




Conclusão



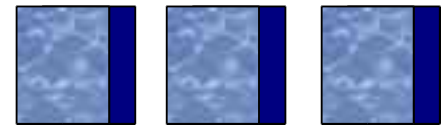
Desenvolvimento de jogos para celulares



Características do meio

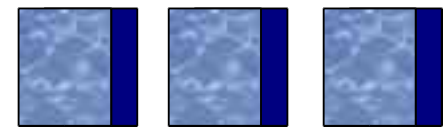
- ⇒ Processamento limitado
 - similar a 2ª geração de arcades ou PCs de 1990
- ⇒ Memória limitada
 - entre 128K e 500K, alguns chegando a 4Mb
- ⇒ Display e input limitados

- ⇒ Grande base instalada
- ⇒ Envio e recebimento de dados digitais
 - habilidade de compartilhar informações é uma oportunidade para interação entre jogadores



Desenvolvimento de jogos

	<i>Tradicional</i>	<i>Celulares</i>
Equipe	12 a 30 pessoas	3 a 5 pessoas
Orçamento	US\$ 1.5 a 5 milhões	< US\$ 100.000
Ciclo Desenv.	2 a 3 anos	alguns meses
Padrões	Domínio de empresas	Padrões abertos
Distribuição	Lojas	Download



Pontos Fortes

⇒ Mercado potencial

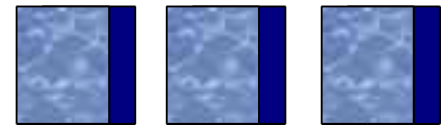
- Mais de 1 bilhão de celulares no mundo
- Maior que PCs, GameBoy ou Playstation

⇒ Portabilidade

- Jogar em qualquer lugar, a qualquer hora

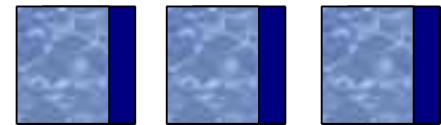
⇒ Rede

- Possibilidade de jogos multiplayer



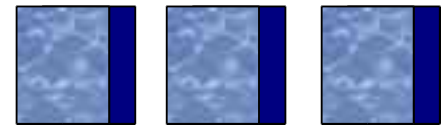
Limitações

- ⇒ Tamanho da tela
 - Resolução aumenta, mas tamanho da tela não
 - Formatos diferentes
- ⇒ Suporte a cor e som
 - Diferentes profundidades de cores
 - Suporte a MIDI recente
- ⇒ Tamanho da aplicação
 - Pouca memória
- ⇒ Alta latência
 - Medida em segundos em rede OTA



Dicas

- ⇒ Tempos de jogo curtos (< 5 minutos)
 - Capacidade de interromper e salvar jogadas
- ⇒ Evite a latência
- ⇒ Use a rede
 - multiplayer ou ranking
- ⇒ Jogos pequenos
- ⇒ Suporte a múltiplos aparelhos



Mais informações

Artigos

- ⇒ <http://wireless.java.sun.com/midp/articles/game/>
- ⇒ <http://wireless.java.sun.com/blueprints/articles/game/>
- ⇒ http://www.forum.nokia.com/main/1,6566,050_20,00.html
- ⇒ <http://www.microjava.com/>

Comunidade

- ⇒ <http://www.guj.com.br/forum/>
- ⇒ j2me-list@soujava.dev.java.net

Contato

- ⇒ vanessasabino@yahoo.com.br
- ⇒ <http://www.alemdojava.cjb.net/> ⇐ apresentação e código fonte

