

AFurbot – Migração do framework Furbot para Android

Aluno: Alexandre Rodrigues Coelho

Orientador: Mauro Marcelo Mattos

Roteiro da apresentação

- Objetivos do Trabalho
- Furbot
- Android
- Trabalhos Correlatos
- AFurbot
- Conclusão
- Extensões

Introdução

- Dificuldades de aprendizado em lógica de programação
- Sistema Android
- Migração do framework

Objetivos do Trabalho

- disponibilizar o ambiente Furbot para a plataforma Android;
- validar a aplicação através de um estudo de caso utilizando exercícios aplicados em sala de aula.

Furbot

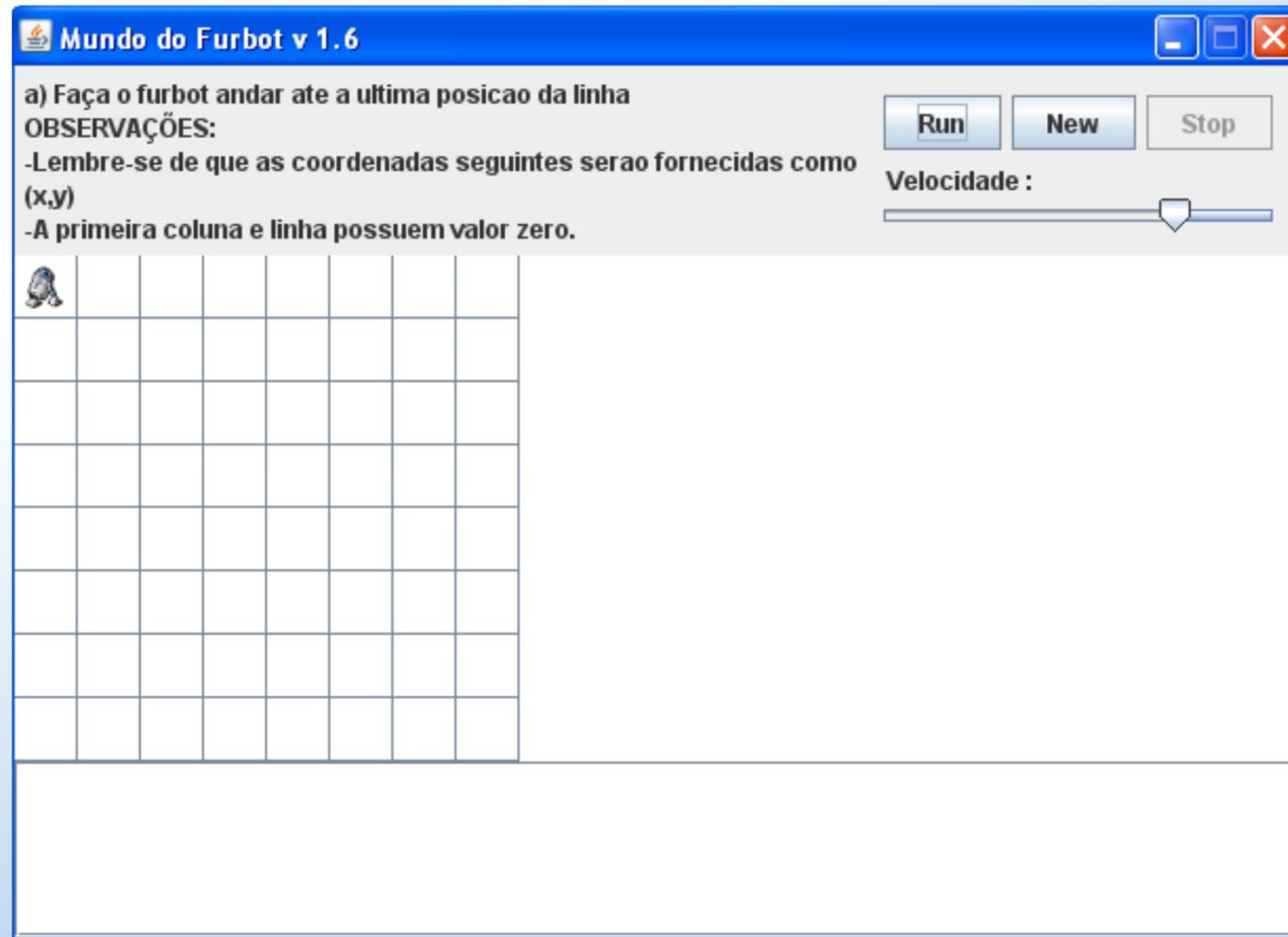
O que é ?

- Um framework para auxílio no ensino de algoritmos
- Jogo

Como funciona ?

- Interface 2D
- Exercícios em XML
- Elementos no mundo

Furbot - mundo



Furbot - XML

```
<furbot>
<enunciado>Exercicio 4. &lt;br&gt;
Faca o robo andar ate os extremos do mundo retornan
do a posicao inicial. &lt;br&gt;
Cada vez que o robo atingir um dos extremos, faca-o
informar que ele chegou ate aquela posicao. &lt;br&gt;
Lembre-se de que as coordenadas sempre serao forne
cidas como (x, y). &lt;br&gt;
A primeira coluna e linhas sao a de numero ZERO.</enunciado>
<mun
do>
  <qt
dadeLin>8</qt
dadeLin>
  <qt
dadeCol>8</qt
dadeCol>
  <expl
odir>false</expl
odir>
</mun
do>
<robo>
  <x>0</x>
  <y>0</y>
</robo>
</furbot>
```

Furbot – programando a inteligência

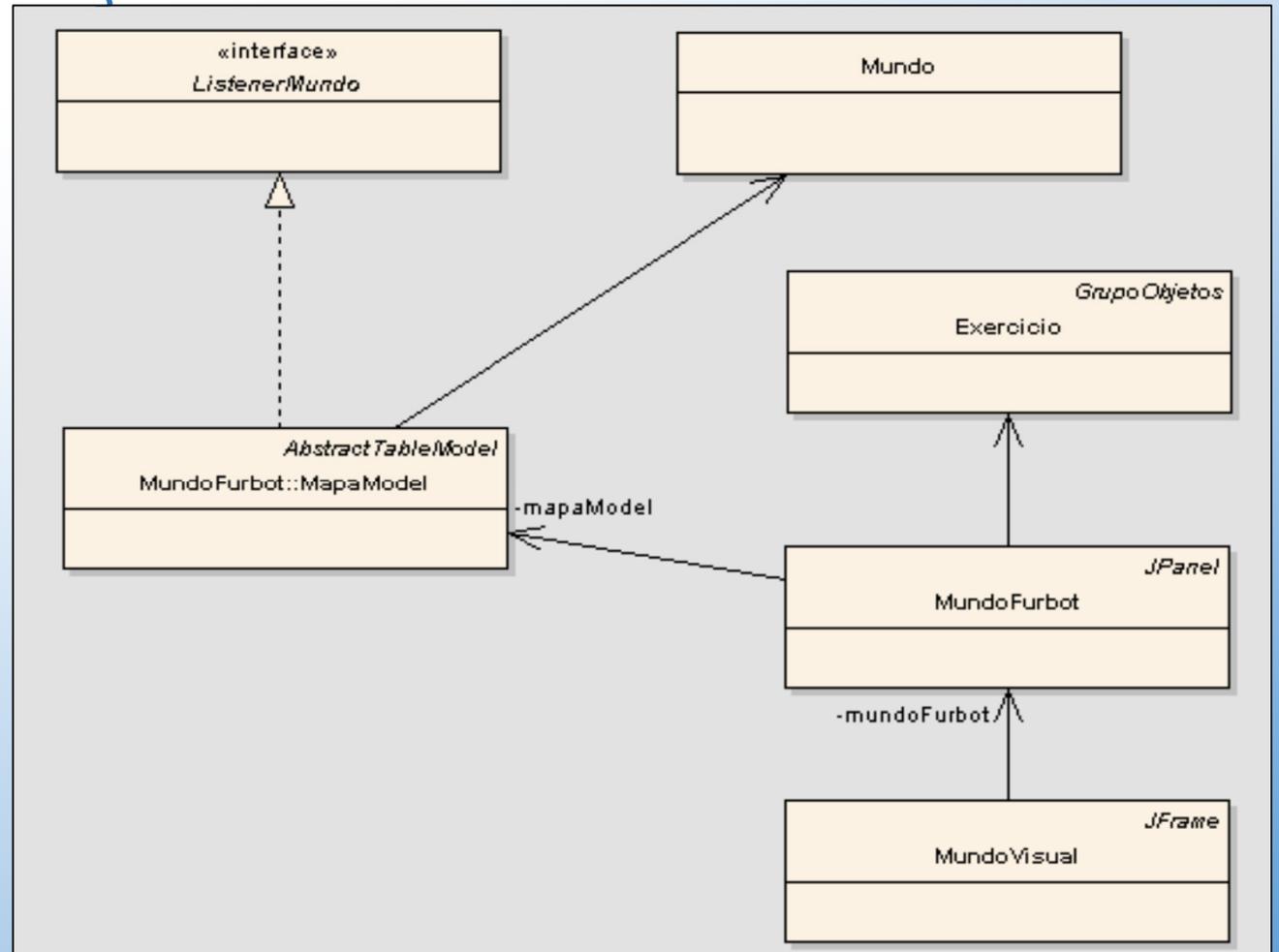
```
@Override
public void inteligencia() throws Exception {
    diga("exercicio4");
    while (!ehFim(DIREITA)) {
        andarDireita();
    };
    diga ("cheguei no canto superior direito");

    while (!ehFim(ABAIXO)) {
        andarAbaixo();
    };
    diga ("cheguei no canto inferior direito");

    while (!ehFim(ESQUERDA)) {
        andarEsquerda();
    };
    diga ("cheguei no canto inferior esquerdo");

    while (!ehFim(ACIMA)) {
        andarAcima();
    };
    diga ("retornei ao inicio");
}
```

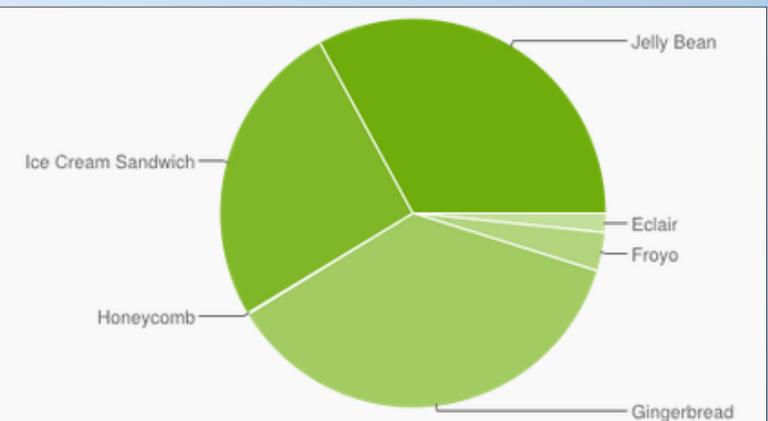
Furbot - especificação



Android

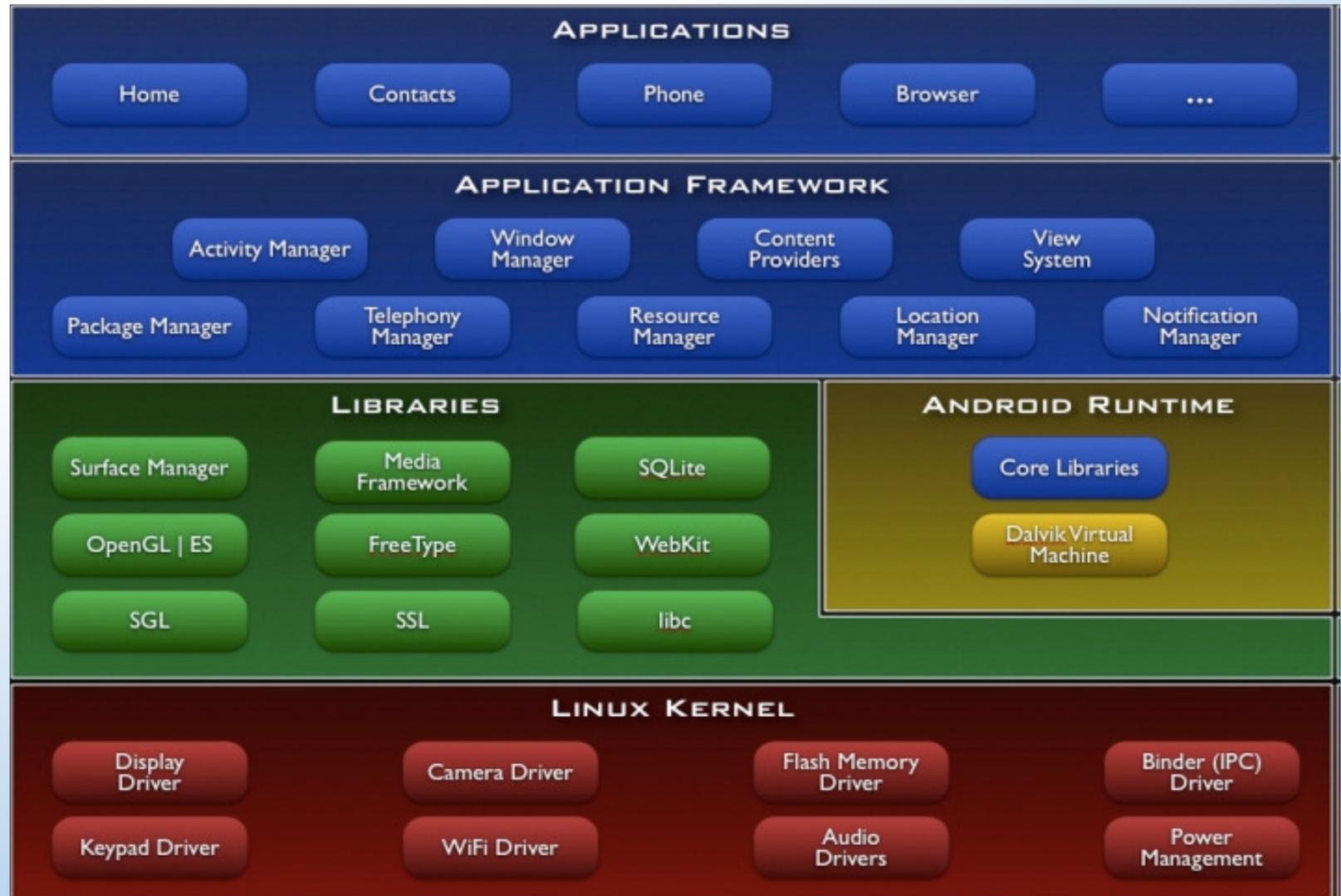
- Sistema Operacional
- Utilização em tablets, celulares e outros dispositivos.

Version	Codename	API	Distribution
1.6	Donut	4	0.1%
2.1	Eclair	7	1.5%
2.2	Froyo	8	3.2%
2.3 - 2.3.2	Gingerbread	9	0.1%
2.3.3 - 2.3.7		10	36.4%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.6%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	4.0%



*Data collected during a 14-day period ending on June 3, 2013.
Any versions with less than 0.1% distribution are not shown.*

Android



Trabalhos correlatos

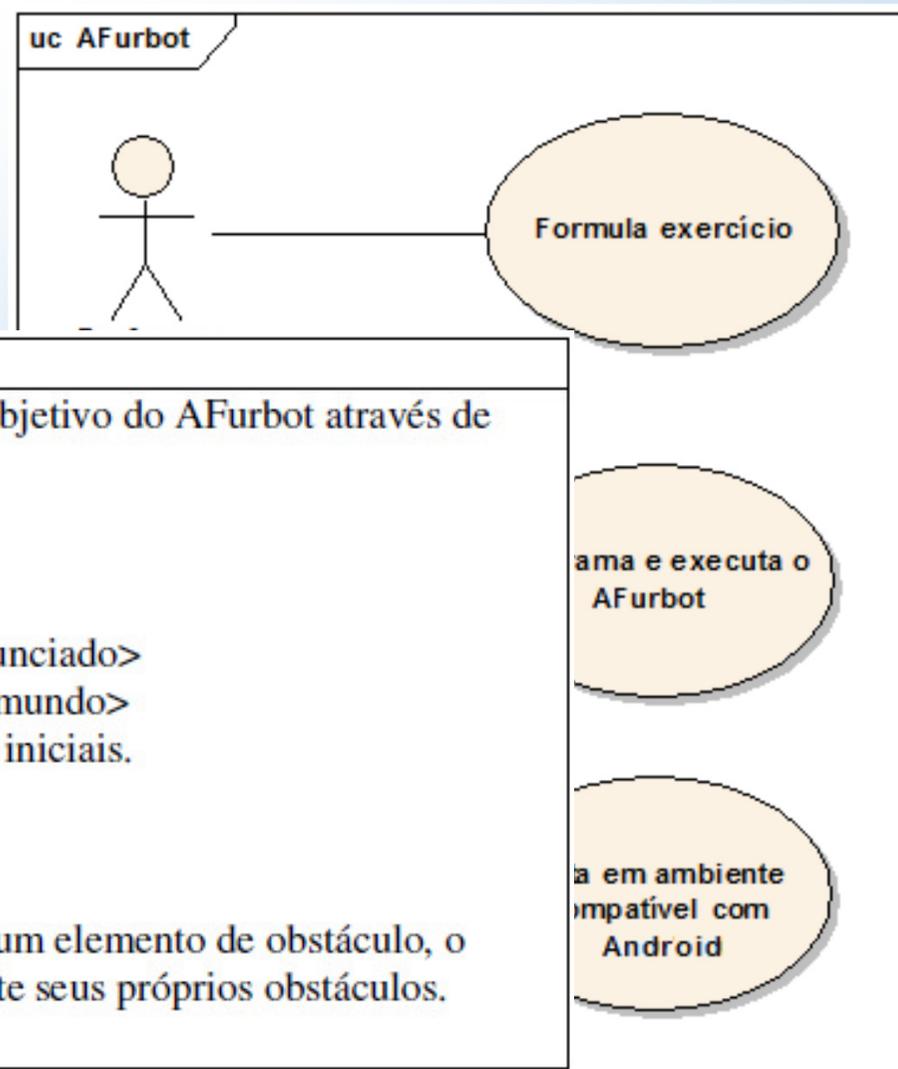
Trabalho	Ambiente de Desenvolvimento	Linguagem de programação	Interface Gráfica
Furbot	Eclipse / outros	Java	AWT
Mobile Furbot	Eclipse	Java ME	AWT
IFurbot	Xcode	Objective-C	Cocos2d
AFurbot	Eclipse	Java	XML

AFurbot - Requisitos

- Permitir a criação de jogos bidimensionais para Android
- Utilizar arquivos do tipo XML para armazenar propriedades para o AFurbot
- Desenvolver um framework similar
- Permitir executar aplicações desenvolvidas

Afurbot - Especificação

- Caso de uso



UC01 - Formula exercício

Caso de Uso: O professor de programação define o tema e objetivo do AFurbot através de um arquivo XML.

Cenários

Define tags do xml {Principal}.

1. O professor digita o objetivo do enunciado na tag <enunciado>
2. O professor define as propriedades de mundo na tag <mundo>
3. O professor inicializa o mundo com alguns obstáculos iniciais.
4. O professor disponibiliza o arquivo XML aos alunos.

Inicializa mundo vazio {Alternativo}.

No passo 3, caso o professor não tenha inicializado nenhum elemento de obstáculo, o professor pode optar para que o aluno gere dinamicamente seus próprios obstáculos.

AFurbot - Ferramentas

- Android Software Development (SDK)
- Eclipse
- Ambiente de testes – smartphone Galaxy SIII

AFurbot – Android (XML)

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.furb.AFurbot"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/r2d2"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="br.furb.AFurbot.Afurbot"
            android:label="@string/app_name"
            android:screenOrientation="portrait" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

AFurbot – Android (XML)

Activity_furbot.xml

```
<TextView
    android:id="@+id/textoinicio"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="top"
    android:text="@string/texto_inicio" />
<GridView
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:columnWidth="30dp"
    android:gravity="center"
    android:horizontalSpacing="3dp"
    android:numColumns="8"
    android:stretchMode="columnWidth"
    android:layout_below="@+id/textoinicio"
    android:verticalSpacing="3dp" />
<TextView
    android:id="@+id/textodiga"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@+id/button01"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="10dp"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/texto_diga" />
<Button
    android:id="@+id/button01"
    android:layout_alignParentBottom="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="run"
    android:text="@string/botao_run" />
```

AFurbot - implementação

Operacionalidade

- Leitura do XML do furbot
- Montar o Grid
- Posicionar os elementos no grid
- Executar inteligência

AFurbot - implementação

- XML - Furbot

```
private NodeList nodeList;
Document doc;

XML (Context context){
    try {
        InputStream url=context.getResources().openRawResource(R.raw.furbot);
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        DocumentBuilder db = dbf.newDocumentBuilder();
        doc = db.parse(new InputSource(url));
        doc.getDocumentElement().normalize();

        nodeList = doc.getElementsByTagName("furbot");
    } catch (Exception e) {
        System.out.println("XML Pasing Excpetion = " + e);
    }
}

public ArrayList<Integer> tesouros (){
    nodeList = doc.getElementsByTagName("tesouro");
    ArrayList<Integer> tesouro = new ArrayList<Integer>();
    for (int i = 0; i < this.nodeList.getLength(); i++){
        tesouro.add(Integer.parseInt(nodeList.item(i).getFirstChild().getNodeValue()));
    }
    return tesouro;
}
```

AFurbot - implementação

- Criação do Grid

```
public void cria_grid () {  
    gridView = (GridView) findViewById(R.id.gridview);  
    gridView.setAdapter(new ImageAdapter(this));  
    gridView.setNumColumns(col);  
}
```

```
public View getView(int position, View convertView, ViewGroup parent) {  
    ImageView imageView;  
    if (convertView == null) {  
        imageView = new ImageView(mContext);  
        imageView.setLayoutParams(new GridView.LayoutParams(80, 80));  
        imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);  
        imageView.setPadding(1,1,1,1);  
    } else {  
        imageView = (ImageView) convertView;  
        setmatriz(position, 8, 0);  
    }  
  
    imageView.setImageResource(img[position]);  
    return imageView;  
}
```

AFurbot - implementação

- Posicionar elementos no Grid

```
public void pos_matriz (int position, int tipo) throws InterruptedException{
    switch (tipo){
        case 1: im.img[position]=R.drawable.r2d2; break;
        case 2: break;
        case 3: im.img[position]=R.drawable.wall;break;
        case 101: im.img[position]=R.drawable.tesouro;break;
        case 201: im.img[position]=R.drawable.tesouro;break; //numeros
    }

    im.notifyDataSetChanged();
    gridView.setAdapter(im);
    gridView.invalidateViews();
    im.setmatriz(position,tam_mat,tipo);
}
```

AFurbot - implementação

- Gridview x Matriz

```
public int setpos(int x, int y, int tam){  
    return ((x*tam)+y);  
}  
  
public int getY(int position, int tammatriz){  
    return (position%tammatriz);  
}  
  
public int getX(int position, int tammatriz){  
    return (position/tammatriz);  
}
```

AFurbot - implementação

- Método inteligência

```
public void inteligencia (View view) throws InterruptedException {
//===== Contrua a inteligência do AFurbot abaixo

    diga("exercicio4");
    while (!ehFim(DIREITA)) {
        //anda até o canto superior direito
        andarDireita();
    };
    diga ("cheguei no canto superior direito");
    while (!ehFim(ABAIXO)) {
        //anda até o canto inferior direito
        andarAbaixo();
    };
    diga ("cheguei no canto inferior direito");
    while (!ehFim(ESQUERDA)) {
        //anda até o canto inferior esquerdo
        andarEsquerda();
    };
    diga ("cheguei no canto inferior esquerdo");
    while (!ehFim(ACIMA)) {
        //anda até o canto superior esquerdo
        andarAcima();
    };
    diga ("retornei ao inicio");

//===== Fim inteligencia AFurbot

    tam_list = stringList.size();
    executa (stringList.get(aux_tam));
}
```

AFurbot - implementação

- Handler

```
public void executa(final Direcao a) throws InterruptedException{
    handler.postDelayed(new Runnable() {
        public void run() {
            try {
                caminhar(a);
                aux_tam++;
                if (aux_tam < tam_list){
                    executa(stringList.get(aux_tam));
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }, 500);
    diga("Tesouros ---> "+Integer.toString(tesouros));
    System.out.println ("Tesoutos ---> "+Integer.toString(tesouros));
}
```

AFurbot - implementação

- Método caminhar

```
public void caminhagrid (final int position, final int anterior) {  
    im.img[anterior]=R.drawable.branco;  
    im.img[position]=R.drawable.r2d2;  
    im.notifyDataSetChanged();  
    gridview.setAdapter(im);  
}
```

```
private void caminhar(Direcao direcao) throws InterruptedException {  
    bordad = 0; bordab = 0; bordac = 0; bordae = 0;  
    switch (direcao) {  
        case ABAIXO:  
            if (im.getX(pos_ robo, tam_mat)==col-1){  
                bordab=1;                }else x++;  
            break;  
  
        case ACIMA:  
            if (im.getX(pos_ robo, tam_mat)==0){  
                bordac=1;                }else x--;  
            break;  
  
        case DIREITA:  
            if (im.getY(pos_ robo, tam_mat)==col-1){  
                bordad=1;                }else{ y++;  
            }  
            break;  
  
        case ESQUERDA:  
            if (im.getY(pos_ robo, tam_mat)==0){  
                bordae=1;                }else{ y--;  
            }  
            break;  
    }  
    pos_ant_ robo=pos_ robo;  
    pos_ robo=im.setpos(x, y, tam_mat);  
    caminhagrid(pos_ robo,pos_ant_ robo);  
}
```

AFurbot - implementação

- Tela AFurbot



Furbot x AFurbot

Mundo do Furbot v 1.6

Exercicio 4.
Faca o robo andar ate os extremos do mundo retornan do a posicao inicial.
Cada vez que o robo atingir um dos extremos, faca-o informar que ele chegou ate aquela posicao.
Lembre-se de que as coordenadas sempre serao forne cidas como (x, y).
A primeira coluna e linhas sao a de numero ZERO.

Run Stop New Me aj

Velocidade :



AFurbot

Exercicio 4.
Faca o robo andar ate os extremos do mundo retornando a posicao inicial. Cada vez que o robo atingir um dos extremos, faca-o informar que ele chegou ate aquela posicao.
Lembre-se de que as coordenadas sempre serao fornecidas como (x, y).
A primeira coluna e linhas sao a de numero ZERO.



Executar

Considerações finais

- Resultado satisfatório.
- Diferenças entre Furbot e AFurbot.

	Furbot	AFurbot
Elementos Gráficos	AWT	XML
Estrutura diretórios	/imagens	/res/drawable
Controle temporal	Thread	Handler

Conclusão

- Conversão do framework com êxito.
- Ferramentas utilizadas.
- Conversão desktop para mobile
- Limitações

Extensões

- jogos *multiplayer*
- efeitos visuais e áudio
- editor gráfico de mundo
- construir uma versão do Furbot com gráficos 3D

Obrigado