Remote Resources

**Arquitetura do Sistema**

Instituto de Pesquisas Eldorado

Versão: 1.0

11/Outubro/2012

Índice

[Objetivos 3](#_Toc337737844)

[Visão geral 3](#_Toc337737845)

[Estruturação do código 3](#_Toc337737846)

[Tecnologias utilizadas 4](#_Toc337737847)

[Licença 4](#_Toc337737848)

Objetivos

O objetivo principal deste documento é dar uma visão geral da arquitetura da ferramenta proposta pelo Instituto Eldorado para a manipulação remota de dispositivos Android.

Trata-se de um documento simples, uma vez que o código da aplicação não é de extrema complexidade.

Visão geral

Em resumo, o Remote Resources está desenhado com um padrão arquitetural de **cliente** e **servidor**. O servidor é o computador no qual o dispositivo está fisicamente conectado e o cliente é qualquer computador que interage remotamente com este dispositivo.

A comunicação entre cliente e servidor se dá pela troca de **mensagens de controle** (transferência de comandos) **e dados** (transferência de informações).

O mesmo computador pode assumir os papéis de cliente e servidor. Neste caso, qualquer dispositivo físico conectado será tratado como um dispositivo local, reconhecido automaticamente pela ferramenta.

Estruturação do código

O código foi escrito em **Java SE**. Ele está organizado de acordo com uma visão cliente-servidor. Existe um pacote de classes específicas para o **servidor** (comunicação direta com o dispositivo), outro pacote específico para o **cliente** (obtenção de comandos do usuário e comunicação com o servidor) e um pacote comum ambos, o qual encapsula as **mensagens** trocadas entre cliente e servidor.

Além destes, existe outro pacote incluso no cliente que se destina a oferecer a interação entre o usuário e a aplicação. Esta camada de **interface** está apoiada em **Java Swing**, uma vez que por hora, trata-se de uma aplicação *desktop*.

Tecnologias utilizadas

As ações executadas pelo usuário na interface do sistema (presente no cliente), são transformadas em mensagens que são enviadas ao servidor. Uma vez recebidas, tais mensagens são decodificadas e interpretadas de acordo com seu tipo:

1. Comandos do sistema Android (toques, *slides*, obtenção de *screenshots*, etc.);
2. Utilização de teclas padrão (*home*, *back*, etc.);
3. Gravação de scripts;
4. Configuração de dispositivos;
5. Obtenção da lista de dispositivos cadastrados.

As mensagens dos tipos [3] à [5] são tratadas diretamente com Java. Já as mensagens dos tipos [1] e [2] são transformadas em eventos ADB e enviadas ao dispositivo por meio das APIs Chimpchat, DDMLIB, GuavaLib e SDKlib. Todas elas são JARs inclusivos no projeto.

Maiores detalhes podem ser encontrados nos comentários das classes e métodos ao longo do código.

Licença

Remote Resources é um projeto interno do Instituto de Pesquisas Eldorado.

As imagens dos ícones foram obtidas gratuitamente de [<http://www.doublejdesign.co.uk>].