

Recentemente uma nova geração de dispositivos com suporte a ambos Flash Lite e Java ME tem tido uma ampla distribuição no mercado, assim abrindo as portas para aplicações híbridas. Focado nesse diferencial de negócio é que surgiu o Jarpa, criado para permitir que os desenvolvedores de ambas as tecnologias possam desfrutar dessa combinação.

Criado por [Felipe Andrade](#) e [Alessandro Pace](#), Jarpa é uma solução open source sobre os termos da licença de software [Apache](#).

## Contents

- [1 Como funciona o Jarpa](#)
  - ◆ [1.1 JSR 75](#)
  - ◆ [1.2 Integrando Flash Lite com Java ME](#)
    - ◇ [1.2.1 Copiando arquivos Flash Lite para o dispositivo](#)
    - ◇ [1.2.2 Efetuando o download de conteúdo Flash Lite](#)
    - ◇ [1.2.3 Extendendo a tecnologia Flash Lite](#)
- [2 Dispositivos Compatíveis](#)
- [3 Download](#)
- [4 Autor](#)

## Como funciona o Jarpa

O Jarpa é uma combinação de características e APIs disponibilizadas na tecnologia Java ME que nos permite o gerenciamento e abertura de arquivos no dispositivo. A JSR 75 foi utilizada para copiar os recursos residentes dentro do JAR, no caso o arquivo SWF, para uma pasta no sistema de arquivos do dispositivo e a função *platformRequest* utilizada para o download e abertura do arquivo.

## JSR 75

A JSR 75 é composta por duas APIs opcionais, que são a *FileConnection* e a *PIM*, ela oferece acesso ao sistema de arquivos do dispositivo e também o gerenciamento de informações pessoais.

## Integrando Flash Lite com Java ME

Com o decorrer do desenvolvimento foram descobertas novas formas de utilizar conteúdo Flash Lite através de aplicações Java ME, são elas:

- Copiar o arquivo SWF residente no JAR para o sistema de arquivos do dispositivo de forma permanente
- Efetuar o download e abertura de conteúdo Flash Lite através de uma determinada URL
- Estender a tecnologia Flash Lite através das APIs existentes no Java ME

### Copiando arquivos Flash Lite para o dispositivo

```
// copia os recursos do arquivo residentes no jar
this._incomingData = this.getClass().getResourceAsStream(this._fromFolder);

// abre uma conexao com o sistema de arquivos, podem existir problemas de assinatura em alguns dispositivos
FileConnection fContent = (FileConnection)Connector.open(this._contentFolder);

// caso a pasta não existir, cria.
if(!fContent.exists()){
    fContent.mkdir();
}
fContent.close();

FileConnection fConn = (FileConnection)Connector.open(this._toFolder);
// se o arquivo não existir, cria.
if (!fConn.exists()) {
    fConn.create();

    int ch = 0;
    DataOutputStream dataStorage = fConn.openDataOutputStream();

    // copia os dados no arquivo recém criado
    while ((ch = this._incomingData.read()) > -1) {
        dataStorage.write(ch);
    }

    dataStorage.flush();
    dataStorage.close();

    if(this._incomingData != null)
        this._incomingData.close();
}

fConn.close();
```

### Efetando o download de conteúdo Flash Lite

```
try {
    // download do swf através de uma url
    platformRequest('http://www.i2tecnologia.com.br/jarpa/jarpa_app.swf');
} catch (Exception ex) {
    ex.printStackTrace();
}
```

### Extendendo a tecnologia Flash Lite

Utilizando um servidor socket local é possível integrar o Flash Lite como front-end para aplicações Java ME. Alguns esforços vem sendo produzidos neste sentido, como o caso de um aplicativo que utiliza a JSR 179 (Location API) do Java ME e o Flash Lite como front-end, mas o código fonte ainda não está sendo disponibilizado.



Abaixo podemos ver como criar uma conexão socket com Java ME:

```
// Cria um socket do servidor na porta 9100
ServerSocketConnection scn = (ServerSocketConnection)
Connector.open(?socket://:9100?);

// Espera por uma conexão
SocketConnection sc = (SocketConnection) scn.acceptAndOpen();
```

## Dispositivos Compatíveis

Muitos dispositivos possuem restrições quanto ao uso de certas APIs. Notamos com o desenvolvimento do Jarpa que tanto a JSR 75 como a função *platformRequest* fazem parte deste conjunto. A lista de dispositivos abaixo provê suporte a no mínimo uma das três possibilidades apresentadas neste artigo, caso você tenha testado com sucesso o aplicativo e o seu dispositivo não esteja listado abaixo, por favor utilize a guia de comentários para nos ajudar a incrementar nossa lista.

Nokia 3250 S60 3rd Edition  
 Nokia N95 - 1 S60 3rd Edition, Feature Pack 1  
 Nokia 6110 Navigator S60 3rd Edition, Feature Pack 1  
 Nokia 6630 S60 2nd Edition, Feature Pack 2  
 Nokia 7500 Prism S40 5th Edition  
 Nokia 7390 S40 3rd Edition, Feature Pack 2  
 Nokia 6301 S40 3rd Edition, Feature Pack 2

## Download

[code.google.com](http://code.google.com)

## **Autor**

FelipeAndrade 09:32, 14 December 2007