

Este artigo contém perguntas feitas com frequência no [Fórum de Discussão de Java](#), traduzidas para o Português. Muita informação importante que pode ajudar a resolver problemas básicos e/ou frequentes no desenvolvimento Java.

Contents

- [1 Eu não consigo fazer o meu MIDlet funcionar: Por favor ME AJUDE! Urgente!](#)
- [2 O que eu preciso para criar o meu primeiro midlet?](#)
- [3 Eu preciso fazer o download do Nokia Development Suite X.X para J2ME. Onde posso encontrá-lo?](#)
- [4 Como posso transferir meu midlet para o celular?](#)
- [5 Como eu posso depurar o meu midlet?](#)
- [6 Por que existem tantos SDKs? Não posso simplesmente pegar qualquer um e testar meu midlet ?](#)
- [7 O que são SDKs de protótipo?](#)
- [8 O meu aparelho possui a API xyz?](#)
- [9 Acho que meu aparelho não tem a API xyz. É possível incluí-la nele ?](#)
- [10 Posso usar um midlet para ler e escrever arquivos no telefone?](#)
- [11 No meu telefone, tem como eu acessar os contatos/calendário/lista de tarefas ?](#)
- [12 Parece que meu midlet é muito grande e não cabe no meu telefone. Como posso fazer com o que os arquivos JAR fiquem menores ?](#)
- [13 Parece que meu midlet é muito grande e não cabe no meu telefone. Como posso fazer com o que os arquivos JAR fiquem menores ?](#)
- [14 Toda vez que o meu MIDlet tenta acessar a rede/contatos... existem essas caixas de diálogo incômodas. Como posso me livrar delas?](#)
- [15 Quais os tipos de permissões estão disponíveis para MIDlets assinados e não assinados?](#)
- [16 Quais certificados eu devo usar para assinar o meu MIDlet](#)
- [17 Posso usar um certificado e chave privada criados por mim para assinar meus midlets ?](#)
- [18 Eu tenho um MIDlet assinado e não consigo instalar no meu celular. Existe alguma forma de instalar esse MIDlet?](#)
- [19 Como eu leio o número do telefone a partir do midlet?](#)
- [20 Como eu posso ler o IMEI do telefone a partir de um midlet?](#)
- [21 A minha aplicação Java ME vai funcionar sem modificações em todas as plataformas?](#)
- [22 Posso usar a API "Mobile Media" para acessar conteúdo protegido por DRM ?](#)
- [23 Eu assinei o meu midlet com um certificado da Verisign/Thawte e não consigo instalá-lo no telefone. O que há de errado?](#)
- [24 Eu assinei o meu midlet e ainda estou recebendo os avisos de segurança quando tento acessar as APIs protegidas.](#)
- [25 Por que eu não consigo que meu midlet inicie automaticamente com PushRegist/ecry sem que um aviso de segurança seja exibido ao usuário, mesmo que a aplicação já tenha sido assinada ?](#)

Eu não consigo fazer o meu MIDlet funcionar: Por favor ME AJUDE! Urgente!

Primeiro verifique os Javadocs para a API específica, em questão (aqueles incluídos com todas as instalações de SDK). Existem também midlets de exemplo e documentação no site do [Forum Nokia](#). E você pode encontrar uma gama enorme de informações na [lista de discussão](#) e no [site sobre Java ME](#) da Sun. E finalmente, verifique também a [Biblioteca Técnica do Forum Nokia](#), que também pode ser baixada para o seu computador.

Se vocÊ continuar enfrentando problemas, por favor inclua as partes relevantes do seu cÃdigo-fonte em sua postagem, e tambÃm especifique com quais dispositivos vocÊs estÃ trabalhando. Se o problema tiver algo a ver com o seu computador de mesa, por favor adicione as informaÃões de Sistema Operacional do seu computador tambÃm. Em alguns casos a versÃo do *firmware* do dispositivo pode fornecer informaÃões valiosas para os leitores. A versÃo do *firmware* do aparelho pode ser obtida ao digitar-se *#0000# na tela principal.

O que eu preciso para criar o meu primeiro *midlet*?

VocÊ pode escrever o seu cÃdigo usando o seu editor de texto favorito ou vocÊ poderia usar um IDE (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento), como o Eclipse, Netbeans, ou JBuilder. [Nota: referÃncia para o Carbide.j foi removida porque a Nokia nÃo fornece mais essa ferramenta]

Quando vocÊ utiliza o *Wireless Toolkit* da Sun, NetBeans com *Mobility pack*, e outros ambientes Java para desenvolvimento de aplicaÃões mÃveis, vocÊ deveria lembrar que se vocÊ estÃ desenvolvendo o seu *midlet* para dispositivos Nokia, Ã aconselhÃvel testar a aplicaÃõo com emuladores da Nokia (bem como em dispositivos reais), considerando que emuladores genÃricos tÃm comportamento diferente dos emuladores da Nokia e obviamente, dos dispositivos Nokia.

Eu preciso fazer o download do *Nokia Development Suite X.X* para J2ME. Onde posso encontrÃ-lo?

O NDS foi substituído pelo Carbide.j, que por vez tambÃm foi removido do portfÃlio de ferramentas. Para o desenvolvimento de seus *midlets*, por favor utilize o Eclipse ME ou Netbeans com o *plugin Mobility Pack*.

Como posso transferir meu *midlet* para o celular?

Em telefones S60 e nos mais recentes S40: VocÊ pode enviar o JAR e o JAD via Bluetooth a partir do *File Browser*.

Em aparelhos S40 mais antigos, vocÊ precisarÃ utilizar o PC Suite. O PC Suite tambÃm funciona com aparelhos S60 e os S40 mais recentes.

Uma outra opÃõo seria armazenar os arquivos JAR e JAD em um servidor WEB e se conectar ao JAD atravÃs de um celular. Certifique-se que os tipos mime foram definidos corretamente no servidor, (?application/java-archive? para o jar e ?text/vnd.sun.j2me.app-descriptor? para jad).

Como eu posso depurar o meu *midlet*?

Todos os SDKs contÃm um emulador, que basicamente emula o ambiente Java do telefone e o conjunto de APIs incluídas naquele SDK no seu PC. Quando vocÊ executa seu *midlet* no emulador, vocÊ pode ver as informaÃões para diagnÃstico (como a saída de `printStackTrace()`) na aba console do IDE. VocÊ deve sempre testar suas aplicaÃões em um dispositivo real, porque algumas capacidades e característic

telefone não podem ser emuladas no PC. Também existem diferenças sutis nas implementações do telefone.

Por que existem tantos SDKs? Não posso simplesmente pegar qualquer um e testar meu *midlet* ?

Cada aparelho possui um conjunto diferente de APIs, e os SDKs trazem um conjunto de APIs (e também um emulador para testar *midlets*) compatível com um determinado tipo de aparelho. Os aparelhos da Nokia são baseados em plataformas (Series 40, S60, Series 80), edições (como 1ª, 2ª e 3ª), e *feature packs*. Os *feature packs* contêm funcionalidades extras para uma determinada edição. Cada tipo de aparelho é de uma determinada plataforma/edição/*feature pack*.

Por exemplo, se você estiver desenvolvendo uma aplicação para o N70, que é um aparelho S60 2ª Edição, *Feature Pack 3*, você deve usar esse SDK específico para desenvolver o seu *midlet*. Se você estiver usando um SDK mais antigo, algumas APIs poderão não estar presentes. Se você usar um SDK mais novo, pode ser que você use algumas funcionalidades do SDK que não estão presentes no N70. Importante também lembrar que o emulador do SDK 2ª Ed. FP3 é o que melhor emula o N70 dentre os emuladores disponíveis.

O que são SDKs de protótipo?

Um SDK de protótipo são destinados a desenvolvedores que queriam desenvolver aplicações para aparelhos ainda não disponíveis. Esse tipo de SDK contém APIs que ainda não foram implementadas em nenhum aparelho, mas que estarão disponíveis em futuros lançamentos. Com um SDK de protótipo, os desenvolvedores podem começar a criar aplicações antes que os aparelhos estejam disponíveis, de modo que o *midlet* possa ser disponibilizado no mercado com mais antecedência.

Os emuladores desses SDKs são baseados em uma especificação de aparelhos, e não em um aparelho real, como ocorre nos SDKs normais. Dessa forma, a implementação da interface gráfica e das novas APIs podem ser um pouco diferentes no aparelho real. Assim, é bom testar sempre a aplicação com os SDKs finais dos aparelhos reais, quando estiverem disponíveis.

O meu aparelho possui a API xyz?

Por favor, confira a especificação do aparelho [nesta página](#).

Para facilitar a sua vida, você pode acrescentar o nome do modelo no endereço acima, para acessar a especificação. Por exemplo, se quiser saber a especificação do Nokia 6131, basta acrescentar 6131 ao endereço, resultando [nesta página](#).

Se você quiser saber se uma determinada API está disponível em aparelhos de outros fabricantes, você pode também conferir o [J2ME Polish Device Database](#) ou [Tastephone MIDP Info](#).

Para a maioria das APIs mais novas, é possível também consultar o aparelho em tempo de execução para saber se uma determinada API existe. Isso pode ser feito usando o método `System.getProperty()`. Por favor, confira as *strings* de consulta corretas (exemplo: `microedition.pim.version` para a API PIM) neste [documento](#)

Uma exceção à essa dica é a MMAPI, que possui muitas opções extras. Por exemplo, nem todos os aparelhos podem usar a câmera. Você pode conferir como a MMAPI está disponível em vários aparelhos da Nokia a partir deste [documento](#).

Acho que meu aparelho não tem a API xyz. É possível incluí-la nele ?

Resposta curta: Não.

Resposta longa: Não, se a API puder ser implementada usando APIs Java já existentes no aparelho, então é possível incluir as classes requeridas no *midlet*. Isso fará com que o *midlet* fique maior, e se outro *midlet* precisar dessas classes elas terão que ser incluídas também nesse outro *midlet*, já que as classes de um *midlet* não são acessíveis a outro *midlet*. Importante também lembrar que em vários casos não é possível fazer isso porque essas APIs precisam que novas funcionalidades estejam presentes no aparelho.

Posso usar um *midlet* para ler e escrever arquivos no telefone?

Sim, se o seu telefone tiver a API FileConnection, que é um pacote opcional do JSR-75. Este outro [documento introdutório](#) pode ajudar nessa questão. Um *midlet* de exemplo também está incluído nesse arquivo.

No meu telefone, tem como eu acessar os contatos/calendário/lista de tarefas ?

Sim, se o seu telefone tiver a API PIM, que é um pacote opcional do JSR-45. Por favor, confira [este documento](#), que também contém um *midlet* de exemplo.

Parece que meu *midlet* é muito grande e não cabe no meu telefone. Como posso fazer com o que os arquivos JAR fiquem menores ?

O embaralhamento (*obfuscation*) substitui os nomes de classes e seus métodos por versões mais curtas (palavras com 1 caractere), indiretamente fazendo com que os arquivos das classes fiquem menores. Originalmente, o embaralhamento foi concebido como uma ferramenta para atrapalhar a engenharia reversa de um *midlet*. Um exemplo de ferramenta para embaralhamento é o Proguard.

Se a sua aplicação utiliza outros tipos de recurso (imagens, áudio, vídeo), certifique-se que esses arquivos estão otimizados para o uso em telefones celulares (ex: resolução de imagem, modo de cores, em configurações compatíveis com o aparelho).

É possível também eliminar alguns *bytes* a mais do arquivo JAR escolhendo nomes bem curtos para as pastas.

Parece que meu *midlet* é muito grande e não cabe no meu telefone. Como posso fazer com o que os arquivos JAR fiquem menores ?

O embaralhamento (*obfuscation*) substitui os nomes de classes e seus métodos por versões mais curtas (palavras com 1 caracter), indiretamente fazendo com que os arquivos das classes fiquem menores. Originalmente, o embaralhamento foi concebido como uma ferramenta para atrapalhar a engenharia reversa de um *midlet*. Um exemplo de ferramenta para embaralhamento é o Proguard.

Se a sua aplicação utiliza outros tipos de recurso (imagens, áudio, vídeo), certifique-se que esses arquivos estão otimizados para o uso em telefones celulares (ex: resolução de imagem, modo de cores, em configurações compatíveis com o aparelho).

É possível também eliminar alguns *bytes* a mais do arquivo JAR escolhendo nomes bem curtos para as pastas.

Toda vez que o meu MIDlet tenta acessar a rede/contatos... existem essas caixas de diálogo incômodas. Como posso me livrar delas?

O novo modelo de segurança introduzido no MIDP 2.0 protege o telefone e o usuário de aplicações maliciosas, através da restrição do acesso a APIs consideradas sensíveis. Em geral, é garantido ao MIDlet apenas o mínimo acesso as APIs, significando que um pedido de confirmação será feito para o usuário toda vez que essas APIs forem acessadas.

O usuário pode mudar algumas permissões padrão manualmente, através do Gerenciador de Aplicativos (os aparelhos S60 possuem um Gerenciador de Aplicativos separado; nos aparelhos Series 40, você pode configurar essas permissões através do menu *Application*).

O desenvolvedor também pode assinar o MIDlet com um certificado, de forma que possa ter menos restrições as APIs sensíveis. O certificado raiz correspondente precisa estar disponível no telefone; senão instalação irá falhar. O certificado *Java Verified* está disponível em um grande número de telefones de todos os fabricantes. O seu MIDlet será assinado com esse certificado depois de passar pelos testes do *Java Verified*. O critério do *Java Verified* está disponível no site do [Java Verified](#). Para mais informações sobre MIDlets assinados, veja o *Signed MIDlet Developer's Guide* [Signed MIDlet Developer's Guide](#).

Quais os tipos de permissões estão disponíveis para MIDlets assinados e não assinados?

O adendo da [especificação](#) da MIDP 2.0 lista as permissões padrão, assim como outras permissões para

Parece que meu midlet é muito grande e não cabe no meu telefone. Como posso fazer com o que os arquivos

MIDlets certificados de terceiros, e MIDlets não-certificados. Note que algumas operadoras garantem permissões aos MIDlets. Por exemplo a documentação *Java Signing Requirement* está disponível no [site de desenvolvedores da Cingular](#).

Quais certificados eu devo usar para assinar o meu MIDlet

Você pode assinar o seu MIDlet com qualquer certificado disponível nos telefones e permitindo a instalação da aplicação Java. (Você pode verificar quais certificados estão disponíveis no telefone através das configurações de segurança - assegure-se de que o certificado que você quer usar é destinado para assinar aplicações Java).

O certificado *Java Verified* está disponível em um grande número de telefones de todos os fabricantes. O seu MIDlet será assinado com esse certificado depois de passar pelos testes do *Java Verified*. O critério do *Java Verified* está disponível no site www.javaverified.com.

Posso usar um certificado e chave privada criados por mim para assinar meus *midlets* ?

Sim, você fazer isso (*self-signing*) se for executar a aplicação nos emuladores. Entretanto, esse procedimento não funciona em aparelhos Series 40, nem em aparelhos S60 3ª Edição.

Eu tenho um MIDlet assinado e não consigo instalar no meu celular. Existe alguma forma de instalar esse MIDlet?

Verifique se todas as permissões necessárias estão sendo requeridas no arquivo JAD. Você também pode tentar remover as linhas `MIDlet-Jar-RSA-SHA1:` e `MIDlet-Certificate-1-1:` do seu arquivo JAD. (O seu MIDlet assinado, agora vai ser tratado como um MIDlet não assinado). Se a instalação ainda falhar, verifique se o atributo de tamanho de arquivo JAR está de acordo com o valor descrito no arquivo JAD. Certifique-se de que todas as propriedades no arquivo JAD não estão vazias.

Como eu leio o número do telefone a partir do *midlet*?

Você teria que usar a [JSR 253 - Mobile Telephony API](#) que possibilita o uso de várias funções relacionadas a telefonia, então é possível que essa API irá prover uma forma de você saber qual é o número do telefone. Atualmente, nenhum telefone Nokia implementa a JSR 253.

Existem algumas alternativas:

Quando o usuário fizer o download da aplicação, um programa no lado do servidor, por exemplo, um servlet ou um script PHP que resgate o número do telefone do cabeçalho HTTP, e criar uma propriedade gerada dinamicamente no arquivo JAD. Assim, quando o seu *midlet* estiver rodando, ele pode verificar qual é o número do telefone usando o método `getAppProperty()` da class `MIDlet`. Essa abordagem só vai funcionar se a aplicação estiver acessando o lado do servidor usando um ponto de acesso WAP. Isso porque

nesse tipo de ponto de acesso, um cabeçalho HTTP contendo o MSISDN (número do telefone) é adicionado ao *request* do HTTP e seria redirecionado para a sua aplicação, para que você possa lê-la.

Se o telefone possuir uma implementação da *Wireless Messaging API*, você pode registrar uma conexão de SMS, usando o *PushRegistry*, com uma porta não registrada. Quando o usuário iniciar a aplicação, gere dados randômicos e envie para o número de telefone que o usuário digitar na caixa de texto. Se o número estiver correto, você receberá a mensagem em alguns segundos na conexão que você criou. Você pode verificar os dados para ver se é igual aos dados que você enviou. Se for, pode ler o número com `Message.getAddress()` e salvar com o RMS. Note que a configuração de segurança padrão para a *Wireless Messaging API* é "Perguntar sempre" e por isso você irá receber avisos de segurança se o *midlet* não estiver assinado e/ou as configurações de segurança não estiverem corretas.

Se você estiver usando um telefone S60, você pode escrever uma aplicação em Symbian C++ que lê o número do telefone e abre uma conexão socket em 127.0.0.1. O *midlet* então conecta com esse socket (local) e lê a informação. Existem alguns problemas com essa abordagem. A sua aplicação Symbian deve ser iniciado no quando o telefone é ligado, o método usado para ler o número do telefone não é portátil (se você estiver lendo o IMEI, e não o número do telefone, aí sim é portátil); terceiro, depende de uma aplicação C++, o que aumenta a complexidade. Existe um projeto chamado MIDP JNI que usa essa ponte para trazer funcionalidades antivas aos *midlets*, para ter também informações que estão disponíveis apenas para aplicações C++, como propriedades de rede e do telefone.

Como eu posso ler o IMEI do telefone a partir de um *midlet*?

Extraído do documento: [MIDP: System Properties \(with Example\) v1.2](#)

Você pode usar a propriedade de sistema `com.nokia.mid.imei`. Isso funciona apenas em telefones Série 40 3rd, exceto o Nokia 6270 e 6111. O *midlet* precisa estar assinado, senão o valor retornado será NULL. Para todos os outros modelos de Série 40, não é possível acessar o IMEI.

Alguns telefones S60, como o E61, também possuem essa propriedade de sistema. Na maioria dos telefones S60, você deverá usar a abordagem descrita acima.

A minha aplicação Java ME vai funcionar sem modificações em todas as plataformas?

A Nokia adota uma abordagem de plataforma nos seus produtos. Isso significa que se você escrever uma aplicação para uma dada plataforma, supõem-se que ela irá funcionar sem modificações em todos os outros telefones das mesma plataforma. Atualmente existem três plataformas: Série 40, S60 e Série 80.

Existem várias níveis de portabilidade, no entanto:

- Se você usar apenas recursos de MIDP 1.0 e componentes LCDUI (sem *Canvas*), sua aplicação vai rodar sem modificações em todos o telefones Nokia que rodem Java.
- Se você usar apenas recursos de MIDP 2.0 e componentes LCDUI (sem *Canvas*), sua aplicação vai rodar sem modificações em todos o telefones Nokia que tenham MIDP 2.0.

- Se vocÊ usar APIs opcionais, vocÊ terÃ que adaptar sua aplicaÃo de acordo com um dos casos abaixo:

DiferenÃas de cÃdigo fonte: Essas acontecem quando vocÊ estÃ usando uma certa API (como *FileConnection* ou *Bluetooth*) que estÃ implementada em um telefone ou plataforma, mas nÃo em outros. Por exemplo, a *Bluetooth API* (JSR 82) estÃ implementada no Nokia 6230i mas nÃo no Nokia 6101. Para resolver essas diferenÃas, vocÊ deverÃ gerar versÃes diferentes do seu *midlet* para cada telefone (ou plataforma), de acordo com a disponibilidade da API que vocÊ queira utilizar. Para tornar isso mais fÃcil, existem vÃrios pacotes de software que podem ajudÃ-lo a gerar vÃrias versÃes do seu cÃdigo fonte; dois dos mais populares macros pre-processadores: "Antenna and NetBeans Mobility Pack."

DiferenÃas de execuÃo (*runtime*): Essas acontecem quando vocÊ estÃ usando uma certa API que possui componentes de execuÃo (*runtime*) opcionais. Isso acontece principalmente na *Mobile Media API*, que tem um grande conjunto de recursos opcionais que podem ou nÃo estar implementados em um telefone, mas se esses estiverem implementados, nÃo hÃ mudanÃa no cÃdigo fonte, apenas no comportamento da execuÃo. Por exemplo, o 6230 e o 6681 possuem a *Mobile Media API*, portanto nÃo hÃ diferenÃas de cÃdigo fonte. No entanto, o 6230 nÃo pode fazer captura de imagens com a cÃmera do celular, nem gravar som, enquanto o 6681 Ã possÃvel realizar as duas operaÃes. Para que a sua aplicaÃo rode nos dois telefones, vocÊ deverÃ fazer verificaÃes em tempo de execuÃo, geralmente usando propriedades de sistema, para prover uma melhor experiÃncia de usuÃrio.

DiferenÃas de *hardware*: Essas geralmente se referem Ã resoluÃo e tamanho de tela. Por exemplo, o 6230i tem um *display* de 208x208, enquanto o E61 tem um *display* de 320x240, no formato paisagem. Com jÃ foi dito, se vocÊ estiver usando apenas classes LCDUI, vocÊ nÃo precisa se preocupar com isso, mas se vocÊ estiver usando grÃficos que dependem do tamanho da tela, vocÊ terÃ vÃrias versÃes do seu *midlet* ou entÃo irÃ fazer verificaÃes em tempo de execuÃo, para ajustar seu desenho de acordo com o tamanho da tela.

Posso usar a API "Mobile Media" para acessar conteÃdo protegido por DRM ?

Em aparelhos S60 3ª EdiÃo, Ã possÃvel fazer alguma coisa com DRM. VocÊ pode reproduzir um arquivo protegido por DRM com o seguinte trecho de cÃdigo:

```
Player p = Manager.createPlayer("file:///C:/Path/To/File.dcf");  
p.start();
```

Eu assinei o meu *midlet* com um certificado da Verisign/Thawte e nÃo consigo instalÃ-lo no telefone. O que hÃ de errado?

O certificado raiz para o seu certificado pode estar 1) nÃo presente no CA raiz do telefone 2) nÃo habilitado para autorizar e autenticar instalaÃo de software.

VocÊ pode saber se este Ã o caso da seguinte forma:

Em telefones SÃrie 40, vÃ em "Menu/Services/Settings/Security Settings/Authority certificates/Certificate list". Verifique se o certificado raiz estÃ lÃ e, se estiver, se estÃ habilitado para autenticar aplicaÃes,

escolhendo "Options/Select use". Nos telefones S40 mais antigos, os certificados estÃo debaixo do menu "Services".

Nos telefones S60, vÃa em "Tools/Settings/Security/Certificate management/Authority". Para verificar se um dado certificado estÃa habilitado para autenticar aplicaÃes, vÃa em "Options/Trust Settings".

Eu assinei o meu *midlet* e ainda estou recebendo os avisos de seguranÃa quando tento acessar as APIs protegidas.

Primeiro, vocÃe precisa verificar se a sua aplicaÃo possui as permissÃes apropriadas para as operaÃes que vocÃe quer executar.

Nos telefones SÃrie 40, vÃa em "Menu/Applications/Collection/Select Application/". Posicione o cursor no nome da sua aplicaÃo, vÃa em "Options/Application access", e configure as permissÃes desejadas.

Perceba que o acesso padrÃo nem sempre sÃo o acesso "mÃximo". O usuÃrio precisa alterar as opÃes manualmente.

Se as permissÃes estiverem corretamente configuradas e vocÃe ainda receber os avisos de seguranÃa, pode ser que o seu telefone tenha o firmware customizado para uma determinada operadora, que tem a opÃo de customizar a polÃtica de seguranÃa do telefone, nÃo permitindo, por exemplo, que um *midlet* conecte com a rede sem a confirmaÃo do usuÃrio. Este Ã um comportamento incomum, porque na maioria dos casos assim, a instalaÃo vai simplesmente falhar.

Nesse caso vocÃe deverÃ contactar a sua operadora e perguntar sobre os programas de desenvolvedores que eles possam ter, ou mesmo acordo de parcerias, normalmente disponÃvel apenas para empresas, e nÃo para pessoa fÃsica. Parceria com a operadora torna possÃvel assinar o seu *midlet* com o certificado deles, o que deixaria a sua aplicaÃo no domÃnio de seguranÃa da operadora, dando melhores configuraÃes de permissÃo.

Por que eu nÃo consigo que meu *midlet* inicie automaticamente com PushRegist/ecry sem que um aviso de seguranÃa seja exibido ao usuÃrio, mesmo que a aplicaÃo jÃ tenha sido assinada ?

Por favor, confira [este item da Forum Nokia Tech Library \(em InglÃs\)](#) para maiores detalhes.