

Introdução ao SuperWABA

Por Rogério Pereira

Nos últimos anos a computação móvel tem ganhado cada vez mais importância no mundo da informática. Atualmente, dispositivos como o celular tem se aproximado cada vez mais das funcionalidades de um Pocket PC e vice-versa.

Porém, uma coisa ainda não mudou: a variedade de sistemas operacionais para cada dispositivo, o que deixa a realidade das aplicações multiplataforma um pouco distante.

Felizmente, surgiram projetos bastante interessantes neste sentido. Um deles é o Waba, mantido pela Wabasoft. Waba é uma Máquina Virtual (VM do inglês Virtual Machine) para aplicações móveis e que está presente atualmente para as plataformas DOS, Linux, Windows CE, Gameboy e PalmOS.

Contudo, o Waba ainda tinha algumas deficiências, tais como o suporte somente a dois tons de cores (preto e branco) e a falta de suporte para threads entre outros recursos. Foi então que surgiu o SuperWaba, criado por Guilherme Campos Hazan. O SuperWaba é uma versão melhorada do Waba em muitos aspectos, e roda em dispositivos móveis com os sistemas operacionais WindowsCE, PalmOS e Symbian OS.

Este artigo fará uma introdução ao desenvolvimento para dispositivos móveis utilizando o SuperWaba, abordando desde suas origens, as ferramentas disponíveis para o desenvolvimento de aplicações nessa plataforma, até os detalhes de uma simples aplicação escrita com o mesmo.

História do SuperWaba

O SuperWaba começou no início de 2000, quando Guilherme Campos Hazan (Guich) percebeu que o Waba (linguagem, máquina virtual e SDK para PalmOS e Windows CE) era simples demais para criar uma aplicação financeira. Rapidamente ele começou a adicionar alguns métodos, classes, como também criou um teclado virtual para o Waba. Desejando compartilhar seu trabalho com o resto do mundo, ele criou o Waba 1.0G (G = Guich).

A intenção era que as mudanças feitas no 1.0G fossem repassadas para o Rick Wild (criador do Waba) incluir na distribuição padrão do Waba, mas isso nunca aconteceu.

Então, Guich continuou aperfeiçoando o Waba 1.0G, e na próxima versão o nome foi mudado para SuperWaba. Uma das mais importantes mudanças no SuperWaba 1.1 e 1.2 foi o suporte para janelas popup. Na versão 1.2 Dave Slaughter portou a versão PalmOS do SuperWaba para o WindowsCE. Até este ponto, o SuperWaba era apenas uma outra versão do Waba com exceção de algumas classes e métodos nativos aperfeiçoados.

O real desafio surgiu no início de 2001: adicionar suporte para tons de cinza (pois Waba só possuía suporte a dois tons de cores: preto e branco) em todos os dispositivos com PalmOS, iniciando com PalmProfessional e PalmOS 2.0. Mas para colocar suporte à tons de cinza, Guich teve que criar algumas funções do assembler 68k (a Palm não tinha lançado suporte a tons de cinza até a versão PalmOS 3.5). O suporte para cores também foi adicionado rapidamente no SuperWaba. Após centenas de otimizações de alto nível, o SuperWaba ficou agora 2x mais rápido do que o Waba. A partir de outubro de 2001 Guich deixou seu antigo emprego para dedicar seu tempo para construir um SuperWaba cada vez melhor. Renato Rocha Ribeiro também entrou no time para ajudar na estrutura de negócios do SuperWaba. No início de 2003, Marcos Guirland Nosowad entrou no time para produzir produtos de alta qualidade e aperfeiçoamentos voltados para a plataforma SuperWaba. Em setembro de 2003, Pierre G. Richard, também se juntou ao time. Marcos e Pierre irão brevemente oferecer serviços de consultoria, treinamento e suporte nos USA e na França, respectivamente. O próprio Guilherme oferece esses mesmos serviços aqui no Brasil.

O que é SuperWaba? Quais as suas características?

O SuperWaba é uma plataforma que possui uma implementação própria de linguagem (com sintaxe semelhante ao Java), máquina virtual, formato de arquivos de classes e um conjunto de classes. Além disso, ele permite o uso de ferramentas Java no desenvolvimento.

Dentre as diversas características, podemos notar o suporte para SQL, leitor de código de barras, protocolo GPS, protocolo HTTP, tratamento de imagens JPEG, PNG, GIF e PNG, e suporte para compressão de dados.

Podemos comparar o SuperWaba ao .NET Compact Framework em recursos (exceto linguagem), porém o SuperWaba é ainda mais portátil, rodando em várias plataformas como comentado na seção anterior.

Somente podemos compará-lo com o J2ME em termos de sintaxe da linguagem utilizada, pois enquanto o J2ME usa a linguagem Java o SuperWaba possui uma linguagem de sintaxe similar ao Java, porém não podemos comparar os dois em termos de recursos e portabilidade, pois o SuperWaba roda em dispositivos em que o J2ME não possui máquina virtual, sem considerar que o SuperWaba possui muito mais recursos que o J2ME.

Ferramentas para desenvolvimento em SuperWaba

Após conhecermos melhor o que seria essa nova tecnologia e suas principais características, passaremos a abordar como desenvolver uma aplicação em SuperWaba. Para isso, inicialmente iremos descrever algumas ferramentas disponíveis para desenvolvimento de aplicações em SuperWaba.

O SuperWaba dispõe de algumas ferramentas de desenvolvimento bastante produtivas. Entre elas vale ressaltar:

- SuperWaba IDE, um plugin do Eclipse que facilita (e muito) o desenvolvimento de aplicações usando o SuperWaba;
- Tauschke MobileCreator, é outra excelente opção, em conjunto com o uiCreator o desenvolvedor consegue obter alta produtividade no desenvolvimento de aplicações SuperWaba. Para usuários do Windows eu recomendo esta incrível IDE.

O Eclipse e o SuperWaba IDE são gratuitos, porém pelo fato do Eclipse ser uma IDE de múltiplos propósitos ele se torna um pouco mais complicado para utilizar do que o Tauschke MobileCreator. Mesmo assim, muitos desenvolvedores utilizam este conjunto para desenvolvimento de aplicações para rodar na VM SuperWaba.

Por outro lado temos o Tauschke Mobile Creator como uma excelente opção para desenvolvimento de aplicações SuperWaba, ele é completamente voltado para este fim, possui uma ferramenta específica para criação de janelas. Entretanto, sua versão profissional é paga, somente a versão pessoal é gratuita e não pode ser utilizada para fins comerciais.

Preparando o ambiente de desenvolvimento de uma aplicação em SuperWaba

Neste artigo estarei abordando o desenvolvimento de nossa aplicação exemplo usando o Eclipse em conjunto com o plugin SuperWaba IDE descrito anteriormente.

Vale a pena ressaltar que você deverá ter o SUN JDK 1.3 ou 1.4 instalado em seu computador. Caso você não conheça o Eclipse, ele é um IDE multiplataforma e multilinguagem mantido pela IBM, e é baseado no conceito de plugins, possuindo inúmeros plugins com as mais variadas funcionalidades. Você poderá baixar o Eclipse neste site: www.eclipse.org. Após isso, siga alguns passos:

Passo 1

Após feito o download, descompacte em uma pasta de sua preferência. Após a descompactação você verá uma pasta chamada *eclipse*, e dentro dela existe um arquivo executável (eclipse.exe para Windows ou eclipse para Unix), execute-o para iniciar a IDE.

Passo 2

Agora vamos baixar o SuperWaba. Você irá encontrá-lo no endereço <http://www.superwaba.com.br>. Porém, para realizar o download do SDK você terá que se registrar.

Após o registro faça o login, então você verá na página um item escrito [Software Development Kit], clique em full para baixar o SDK no caso do windows ou em tar.gz no caso do SDK para Linux. Surgirá uma mensagem de confirmação, clique em OK e baixe o arquivo.

Passo 3

Crie uma pasta chamada superwaba, pode ser em qualquer local que você achar conveniente, e descompacte o conteúdo deste arquivo dentro desta pasta. Adicione o caminho pasta a pasta bin dentro da variável de ambiente PATH do seu sistema operacional.

Neste momento já temos o Eclipse rodando e o SuperWaba em uma pasta do sistema de arquivos. Porém ainda falta o plugin SuperWaba IDE.

Passo 4

Para baixar esse plugin, vamos usar um recurso do Eclipse chamado Update Site. Dentro do Eclipse siga esses passos:

- Selecione Help Software Updates Find and Install (**Figura 1**).

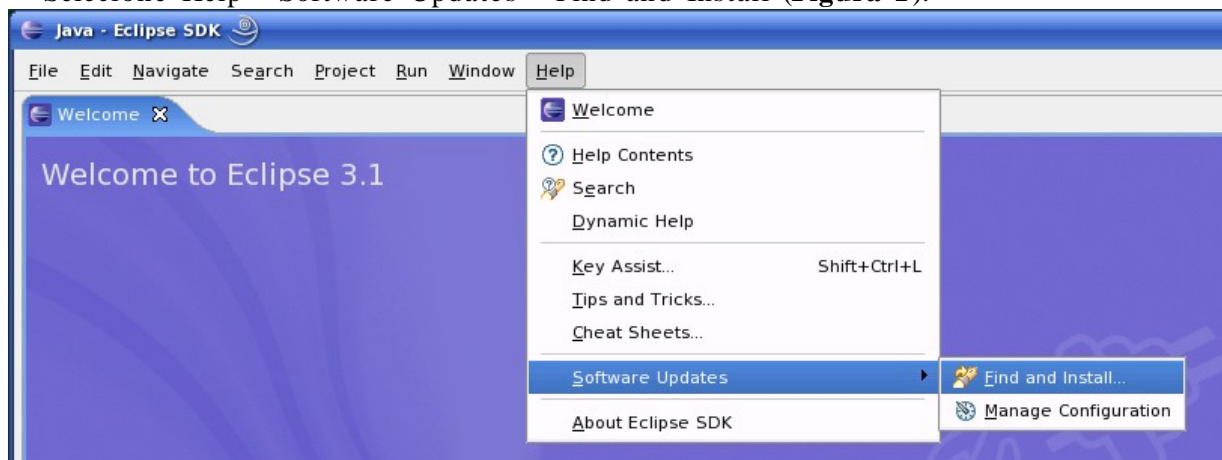


Figura 1. Menu de Instalação.

- Selecione "Search for new features to install...".

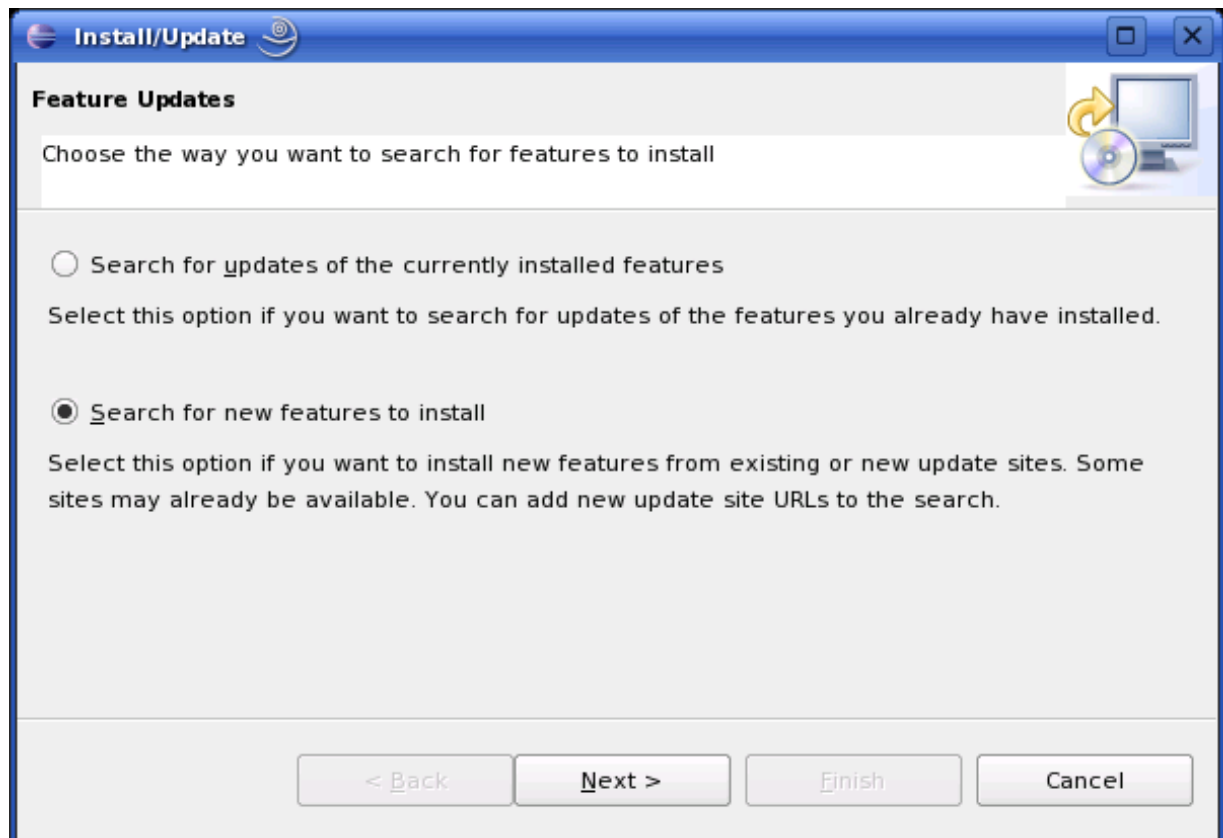


Figura 2. Instalando uma nova funcionalidade no Eclipse.

- Selecione "New remote site..." (**Figura 3**).

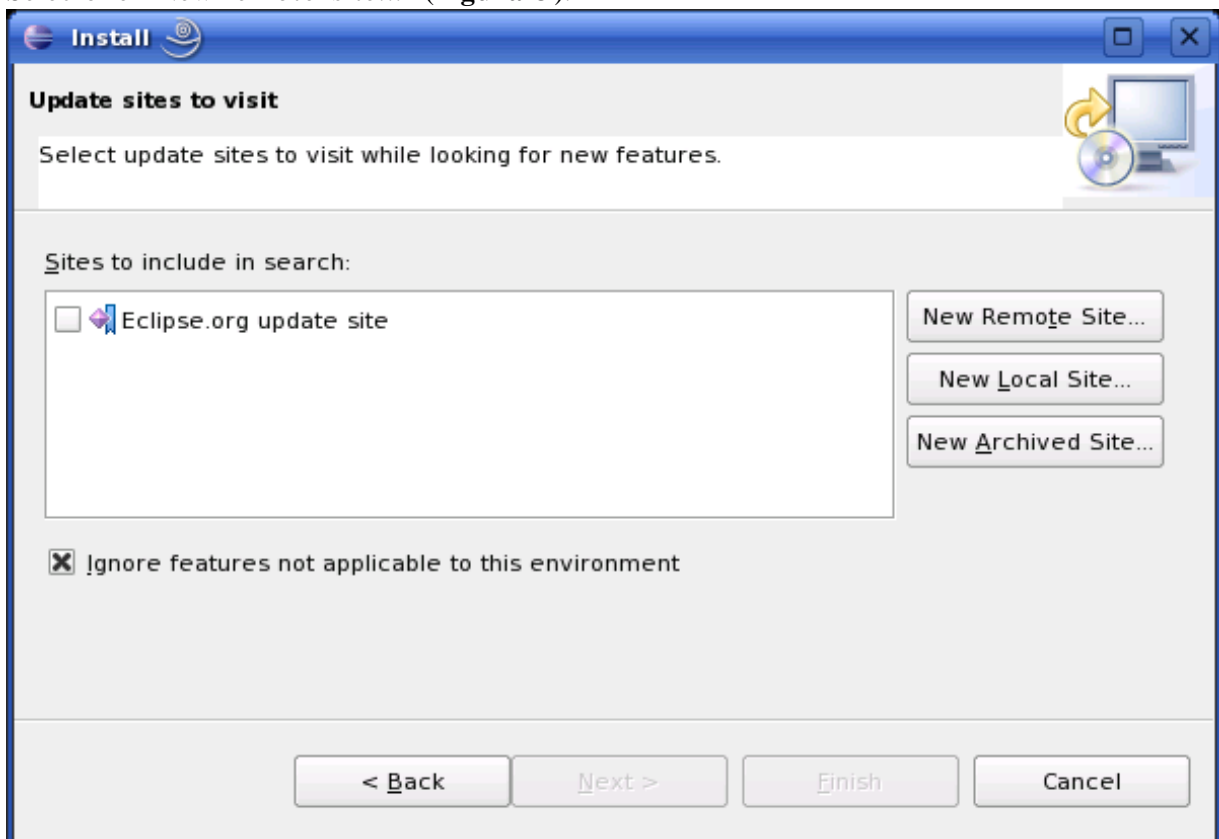


Figura 3. Instalando funcionalidades a partir de um servidor remoto.

- Digite no campo Name o valor "SuperWaba IDE". Este é o identificador do site que contém as atualizações do plugin SuperWaba IDE (**Figura 4**).
- Digite no campo URL o valor <http://superwaba-ide.sourceforge.net/update>. Neste campo você indica o endereço do site onde estão as atualizações do plugin SuperWaba IDE (**Figura 4**).

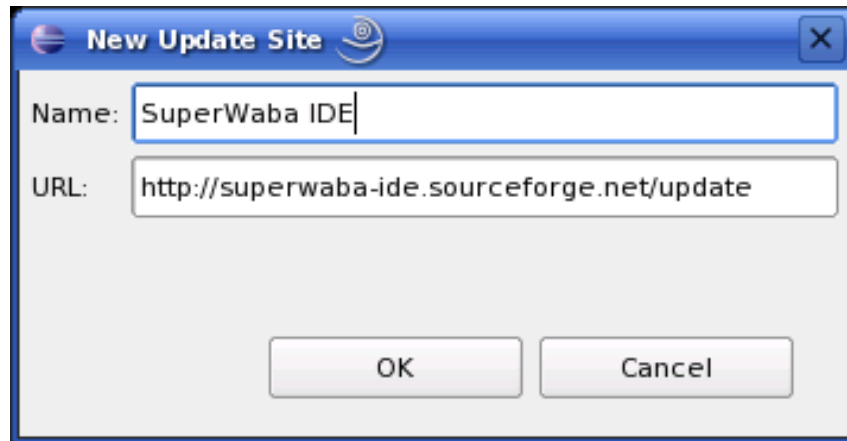


Figura 4. Informando a localização do instalador do plugin.

- Clique em OK, você vai ver "SuperWaba IDE" marcado no campo [Features to install], então clique em [Install]. A partir deste campo, simplesmente aceite os termos do contrato, e no momento certo clique no botão [Install All]. Ao final, o Eclipse irá perguntar se você deseja reiniciá-lo (**Figura 5**). Confirme, e o Eclipse irá reiniciar.

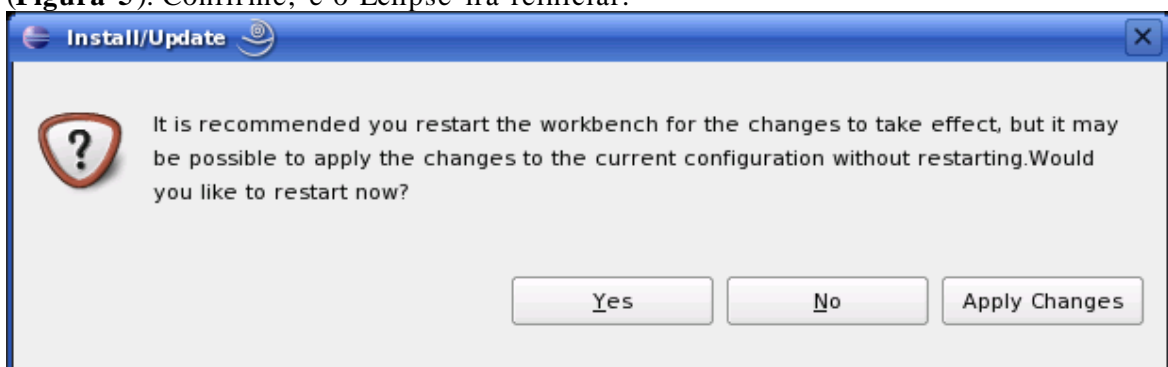


Figura 5. Finalizando a Instalação do SuperWaba no Eclipse.

Com isso, temos o nosso ambiente de desenvolvimento configurado. O passo seguinte é o desenvolvimento da aplicação. Neste artigo iremos desenvolver uma aplicação com uma barra de menus e que exiba caixas de diálogo, a intenção é apresentar as funcionalidades mais básicas de uma aplicação escrita com o SuperWaba.

Criando a sua primeira aplicação usando o SuperWaba

Antes de tudo, devemos configurar o plugin SuperWaba IDE, ou seja, configurar o plugin para que seja possível trabalhar com o Eclipse no desenvolvimento de aplicações utilizando o SuperWaba SDK.

Para isso, selecione Window > Preferences (**Figura 6**). Em seguida surgirá uma nova janela, onde você deve clicar em [SuperWaba] no menu do lado esquerdo. Então, surgirá no lado direito a tela apresentada na **Figura 7**.

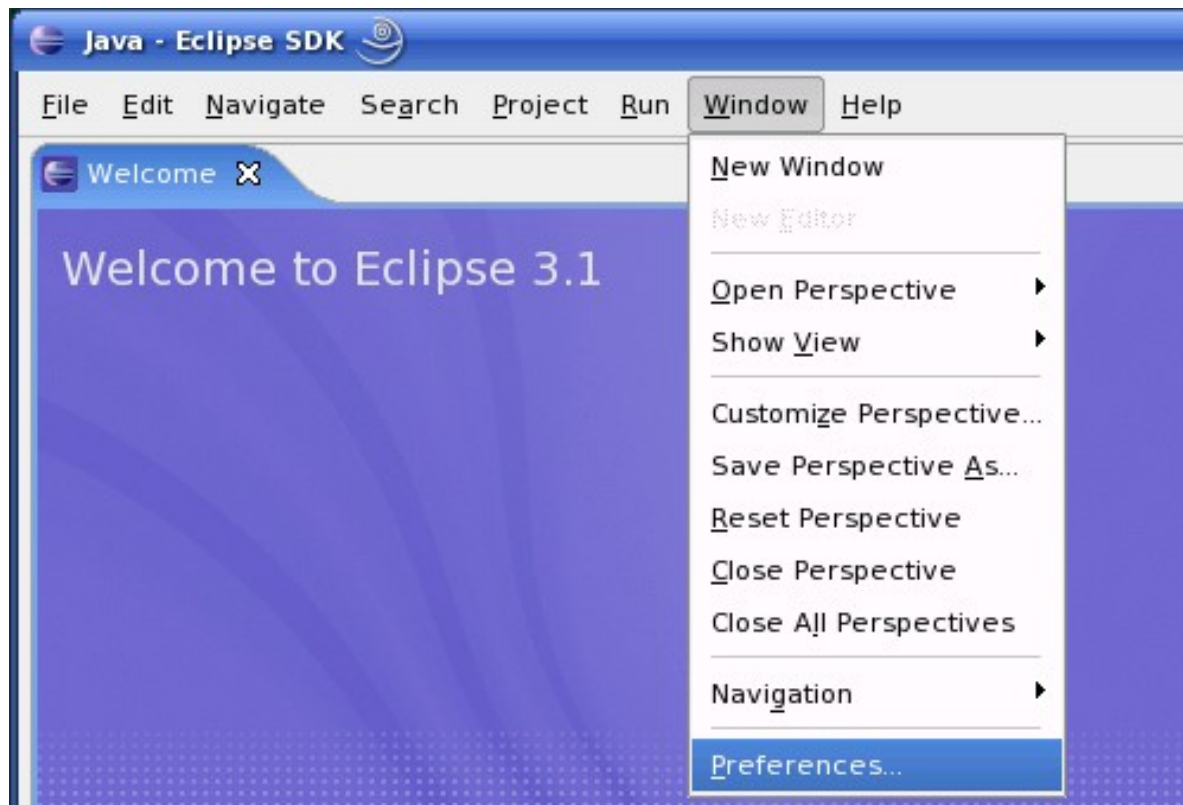


Figura 6. Menu preferências do Eclipse.

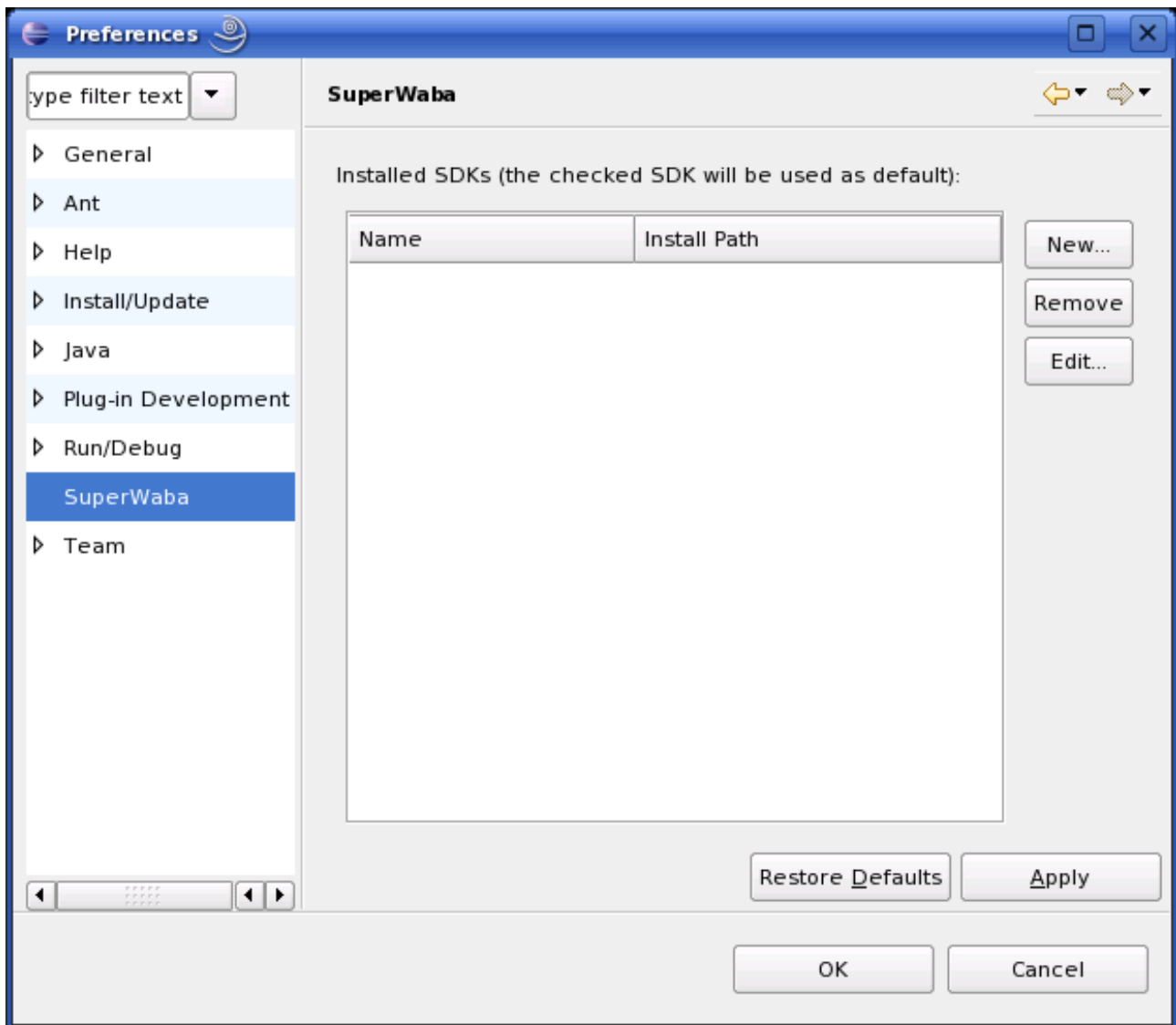


Figura 7. Tela de configuração do plugin SuperWaba IDE.

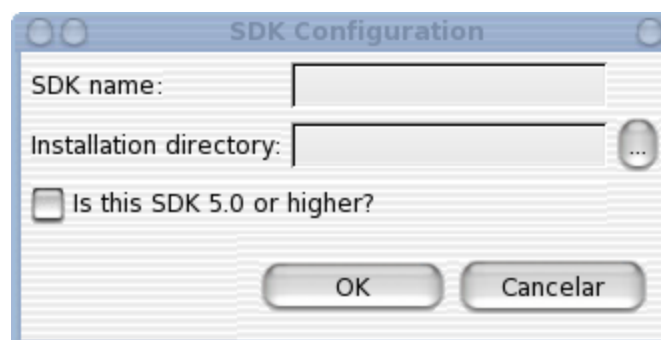


Figura 8. Configurando o SDK SuperWaba para funcionar com o plugin SuperWaba IDE.

Clique no botão New.... Em seguida surgirá uma janela pedindo o nome do SDK e o caminho onde está instalado o SuperWaba (aquele caminho onde você descompactou o arquivo do SDK). Após informar estes valores marque também a opção [Is this SDK 5.0 or higher?] conforme a **Figura 8**. Clique em Apply e depois em OK. Agora o SuperWaba IDE está pronto para ser usado no Eclipse, este procedimento só é necessário uma vez.

Para criar sua aplicação, selecione File New Project e selecione SuperWaba Project como na **Figura 9**.

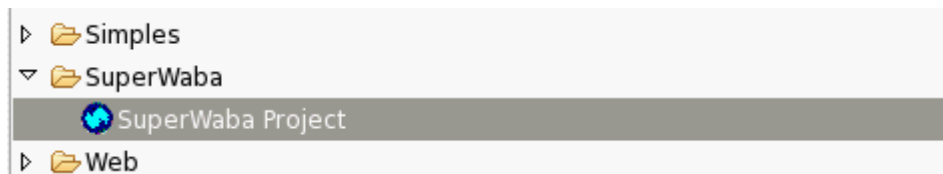


Figura 9. Selecionando o tipo do projeto a ser criado.

Logo em seguida, clique no botão Next. Então surgirá uma tela pedindo o nome do projeto. Forneça o nome e clique em Finish.

Agora crie a classe principal de sua aplicação SuperWaba, para isto selecione File New Class. Nesta tela informe o nome da classe no campo [Name] e no campo [Superclass] digite waba.ui.MainWindow. Você deve informar esse valor para criar o código da janela principal da aplicação, desmarque a opção que indica a criação do método main e então clique em OK. O código será exibido no editor de código do Eclipse conforme apresentado na **Listagem 1**.

Listagem 1. Código gerado pelo Eclipse quando criamos uma classe que herda de MainWindow.

```
import waba.ui.MainWindow;

public class Cinemaz extends MainWindow {

}
```

Vamos adicionar então algum código a esta classe conforme apresentado na **Listagem 2**. Neste código criaremos uma janela, nesta janela teremos um item de menu com dois submenus. Cada um vai exibir uma janela popup com uma mensagem para o usuário. Os comentários no próprio código ajudam a entendê-lo.

Listagem 2. Nossa classe Cinemaz aprimorada para a nossa aplicação de exemplo.

```
//importando os pacotes necessários para a execução da aplicação

import waba.fx.*;
import waba.sys.*;
import waba.util.*;
import waba.ui.*;
import waba.ui.MainWindow;

public class Cinemaz extends MainWindow
{

    //declarando a variável que representa o menu principal da aplicação
    private MenuBar menu;
    //declarando a variável que armazena o nome da plataforma em que a aplicação está rodando
    static String plataforma = Settings.platform;

    //Método construtor da classe principal da aplicação, indispensável!
    public Cinemaz(){
        //Passando para o método construtor da classe MainWindow
        //o título da aplicação e o formato da tela principal da aplicação (aba no topo da janela).
        super ("Cinemaz", TAB_ONLY_BORDER);
        //Indica que usará o dobro do tamanho normal do buffer para executar algumas
        //tarefas tais como atualização do conteúdo das janelas a aplicação
        setDoubleBuffer(true);

        //verifica se a plataforma na qual a aplicação está rodando é PalmOS,
        //caso seja ajusta o visual dos controles da tela para o visual
        //dos controles desta plataforma.
        if (plataforma.compareTo("PalmOS") == 0)
            Settings.setPalmOSStyle(true);
    }

    //Este método é executado na inicialização da aplicação
    public void onStart(){
        Vm.interceptSystemKeys(Vm.SK_ALL|Vm.SK_LAUNCH);
    }
}
```

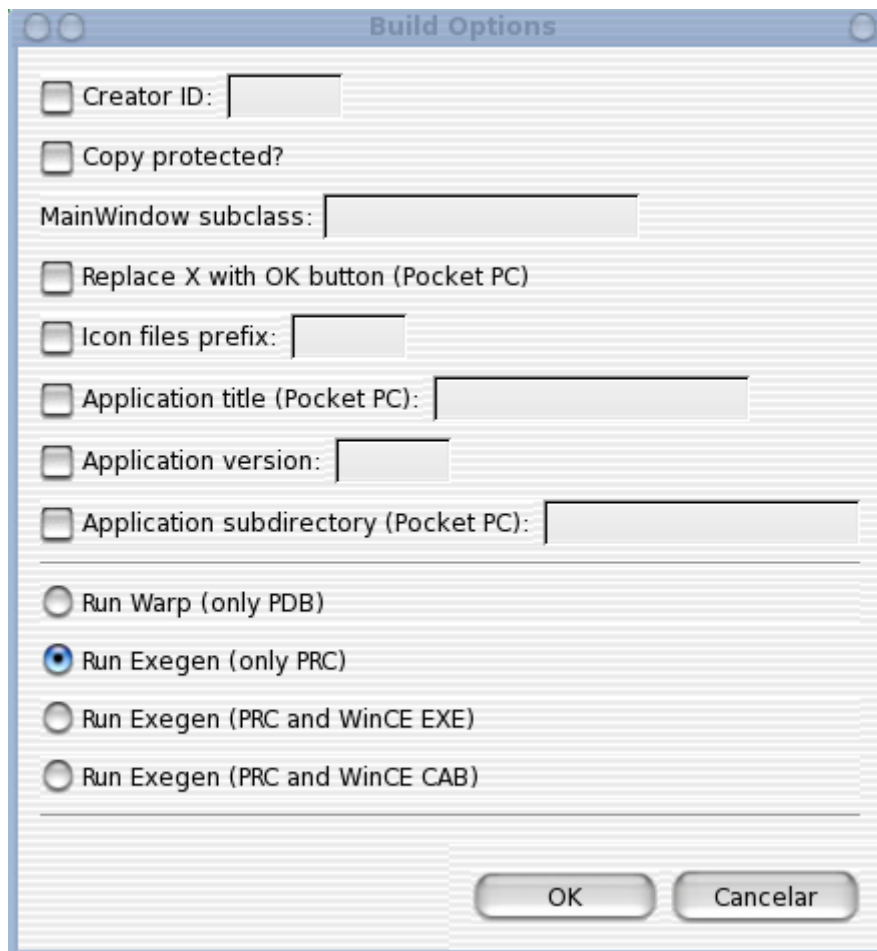



Figura 11. Configurando a geração da aplicação.

Marque a opção [Creator ID:] e informe o valor 2345 (pode ser qualquer valor desde que seja para testes, para distribuição definitiva esse código só pode ser obtido através da própria Palm). No campo [MainWindow subclass] (este campo é onde indicamos a classe principal da aplicação, ou seja, a classe por onde a aplicação vai iniciar sua execução) informe Cinemaz que é o nome da classe criada no código da **Listagem 2**. Vale ressaltar que neste caso a classe Cinemaz não pertence a um package (pacote). Caso sua classe derivada de MainWindow pertença a um package, favor informar o nome completo da classe (package + nome da classe).

Por fim selecione [Run Exegen (only PRC)] para compilar a aplicação (este campo indica que será gerada sua aplicação SuperWaba para ser instalada no dispositivo). As outras duas opções [Run Exegen (PRC and WinCE EXE)] e [Run Exegen (PRC and WinCE CAB)], indicam a geração do executável para Windows EXE e/ou a geração do instalador para Windows CE, respectivamente.

Os outros campos têm o objetivo de estabelecer características específicas tais como título da aplicação, subdiretório onde será instalado os binários (WindowsCE), ícone da aplicação, versão, proteção contra cópia e etc.

Testando nossa aplicação

O teste pode ser efetuado usando o POSE (Palm OS Emulator) que pode ser baixado no endereço http://www.palmbrasil.com.br/cgi-bin/sc-pe/dl.pl?PalmOS_Emulator. Porém, o emulador não é o suficiente, você precisa também da ROM. A ROM é um arquivo contendo uma imagem do PalmOS para um determinado modelo de Palmtop. Para obter uma ROM, faça o

download do arquivo zip contendo a ROM do Palm m505 neste endereço <http://www.palmbrasil.com.br/programas/pgms/rom- palmos40- en- m505.zip>.

Após o download do emulador e da ROM, você deverá instalar a ROM no emulador. Para isso, execute o emulador que está no arquivo executável pose.exe (windows) ou pose (linux). Surgirá, então, uma tela para iniciar uma nova sessão. Clique na caixa de seleção [ROM File:] e selecione [Other...]. Surgirá uma caixa de diálogo na qual você vai localizar o arquivo da ROM. Ao encontrar o arquivo, selecione-o e clique em OK (ver **Figura 12**). Após este procedimento surgirá o nome do arquivo da ROM selecionado na caixa de seleção [ROM File:]. Agora é só clicar em OK (ver **Figura 13**).

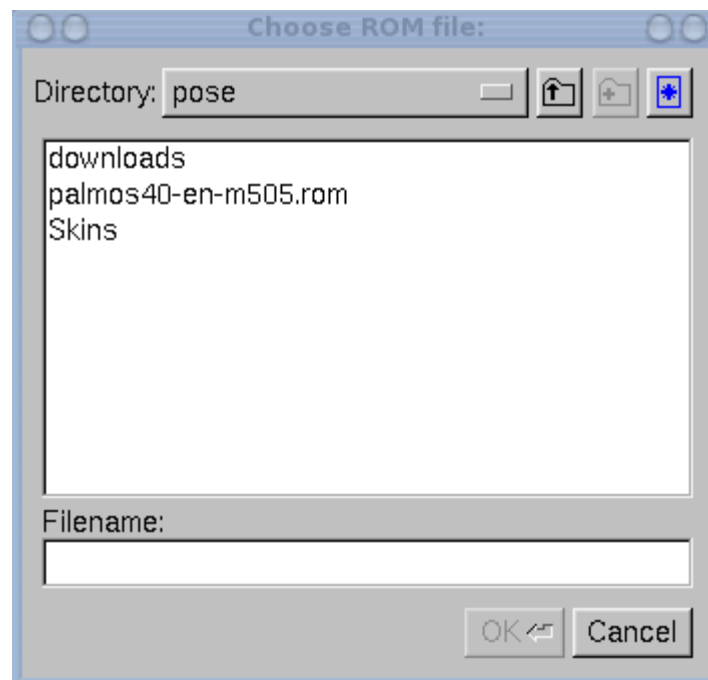


Figura 12. Localizando e instalando a ROM no emulador.

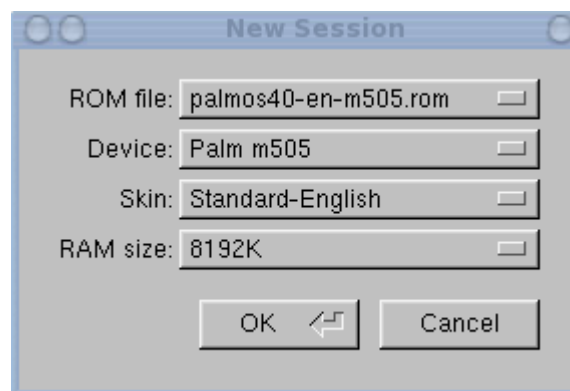


Figura 13. Tela de configuração de nova sessão do emulador.

Com a ROM instalada, agora podemos iniciar uma sessão do emulador. Clique em OK para dar início a esta sessão.

Um procedimento importante a ser considerado são alguns ajustes que devemos fazer no mecanismo de debug do emulador. Para isso, selecione Settings Debugging... e desmarque as cinco primeiras opções da coluna esquerda e três primeiras da coluna esquerda e então clique em OK (ver **Figura 14**). As opções indicadas são somente uma sugestão. Esse procedimento é necessário porque uma vez que o SuperWaba faz uso de instruções especiais escritas no assembler 68k, essas mesmas instruções disparam uma mensagem de aviso a cada vez que são executadas, o que torna o teste no emulador um pouco inviável.

Somente desmarcando essas opções você evitará essas mensagens e poderá testar sua aplicação normalmente.

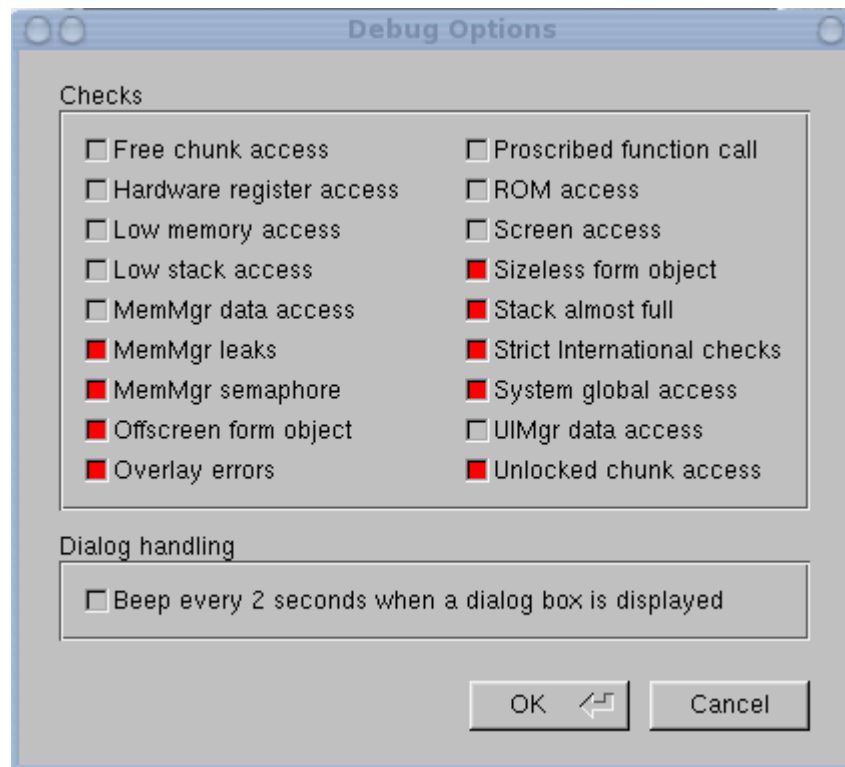


Figura 14. Configurações de debug do emulador.

Vamos agora instalar o VM SuperWaba no emulador. A instalação da VM SuperWaba no emulador é essencial para a execução de aplicações escritas com o uso do SuperWaba SDK. Para isso, clique no emulador com o botão esquerdo e selecione Install Application / Database Other... (**Figura 15**). Localize e selecione os arquivos Superwaba.prc e SWNatives.prc que estão dentro de \$superwaba\lib\vm\palm\68k\PalmOS3_or_4_with_8bits_screen.

Faça o mesmo procedimento para instalar os arquivos contendo os pacotes do SuperWaba (XplatUI.pdb, XplatUtil.pdb, XplatSql.pdb, XplatIoHttp.pdb, XplatIo.pdb) que estão na pasta \$superwaba\lib\vm\xplat. Esses arquivos contêm as classes para criar janelas, botões, menus, tabelas, conexões http, entre outros. Ao concluir este passo você verá na tela no emulador o ícone do SuperWaba como apresentado na **Figura 16**.



Quit	Alt+q	
New...	Alt+n	
Open		▶
Close	Alt+w	
Save	Alt+s	
Save As...		
Save Bound Emulator...		
Save Screen...	Alt+m	
Session Info...		
Install Application/Database	Other...	Alt+i
Export Database...	1 /home/rogerio/projetos/java/cinemaz/dist/cinemaz.pdb	
HotSync	Alt- 2 /home/rogerio/projetos/java/cinemaz/dist/cinemaz.prc	
Reset...	Alt- 3 /home/rogerio/projetos/java/unitsPda/unitsPda/dist/unitsPda.prc	
Transfer ROM...	Alt- 4 /home/rogerio/projetos/java/unitsPda/unitsPda/dist/unitsPda.pdb	
Gremlins	5 /opt/superwaba/lib/vm/xplat/XPlatUtil.pdb	
Settings	6 /opt/superwaba/lib/vm/xplat/XPlatUI.pdb	
About Palm OS(R) Emulator	7 /opt/superwaba/lib/vm/xplat/XPlatSql.pdb	
	8 /opt/superwaba/lib/vm/xplat/XPlatIoHttp.pdb	
	9 /opt/superwaba/lib/vm/xplat/XPlatIo.pdb	

Figura 15. Instalando programas no emulador.

Figura 16. Ícone do VM Superwaba já instalado no emulador.

Esta é a confirmação de que a VM do SuperWaba foi devidamente instalada. Sendo assim, para instalar nossa aplicação basta selecionarmos Install Application / Database Other... (**Figura 15**), localizar a pasta onde está o código fonte de nossa aplicação de teste e dentro da pasta *dist*, selecionar os arquivos de extensão .pdb e .prc e clicar em OK para instalar (**Figura 17**). Depois disso, basta clicar duas vezes no ícone da aplicação e pronto!

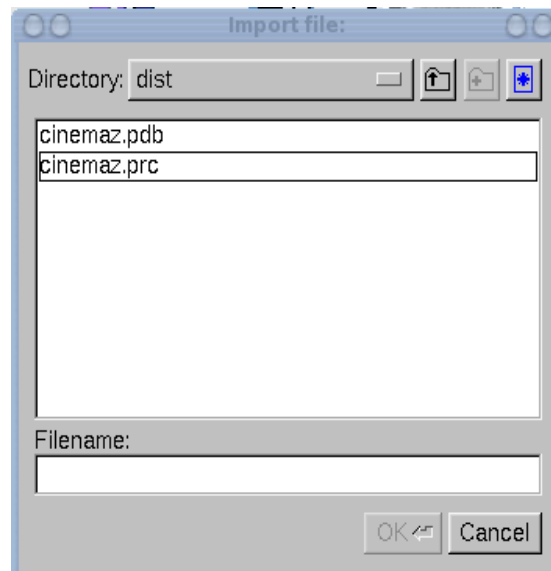


Figura 17. Selecionando os arquivos da aplicação para instalar no emulador.

As **Figuras 18 e 19** mostram a nossa aplicação em execução.



Figura 18. O menu principal da aplicação.



Figura 19. Uma caixa de diálogo fornecendo detalhes da aplicação.

Conclusões

Como você pode observar, é muito simples desenvolver aplicações para dispositivos portáteis utilizando SuperWaba. Basta o conhecimento da linguagem Java e o uso de uma boa IDE como o Eclipse.

Nos próximos artigos estaremos abordando outros recursos do SuperWaba tais como: requisições HTTP, controles visuais (edit, button e etc) e uso do SQL para manipulação de tabelas utilizando o PDBDriver. Até lá!



Rogério Araújo é desenvolvedor especialista em tecnologias WEB, Mobile e Desktop. Atualmente trabalha no Tribunal de Justiça de Goiás utilizando a linguagem PHP e Java no desenvolvimento de aplicações WEB. É colaborador da Web Mobile e do portal linhadecodigo.com.br, membro ativo dos grupos DevGoiás e GOJava. Pode ser contatado em rogerio.araujo@gmail.com.