Python para Maemo

Osvaldo Santana Neto
osantana@triveos.com

Baseado nas apresentações de:
Lauro Moura <lauro.moura@openbossa.org>
Raul Fernandes <raul@embedded.ufcg.edu.br>

Instituto Nokia de Tecnologia

- Instituição sem fins lucrativos, fundada pela Nokia em 2002
- Investimentos da lei de informática
- Escritórios em Manaus, Brasília e Recife
- Pesquisa e Desenvolvimento em Mecânica e Produtos Eletrônicos, Software, Multimídia, Telecomunicações e em Operações e Logística

INSTITUTO NOKIA DE TECNOLOGIA

Osvaldo Santana Neto

- Sócio fundador da Triveos Tecnologia Ltda.
- Trabalhou para as empresas: Conectiva (Mandriva), Instituto Nokia de Tecnologia, Objective Solutions, Haxent, ...
- Responsável pela criação do projeto Python para Maemo no INdT em 2005
- http://www.pythonologia.org/

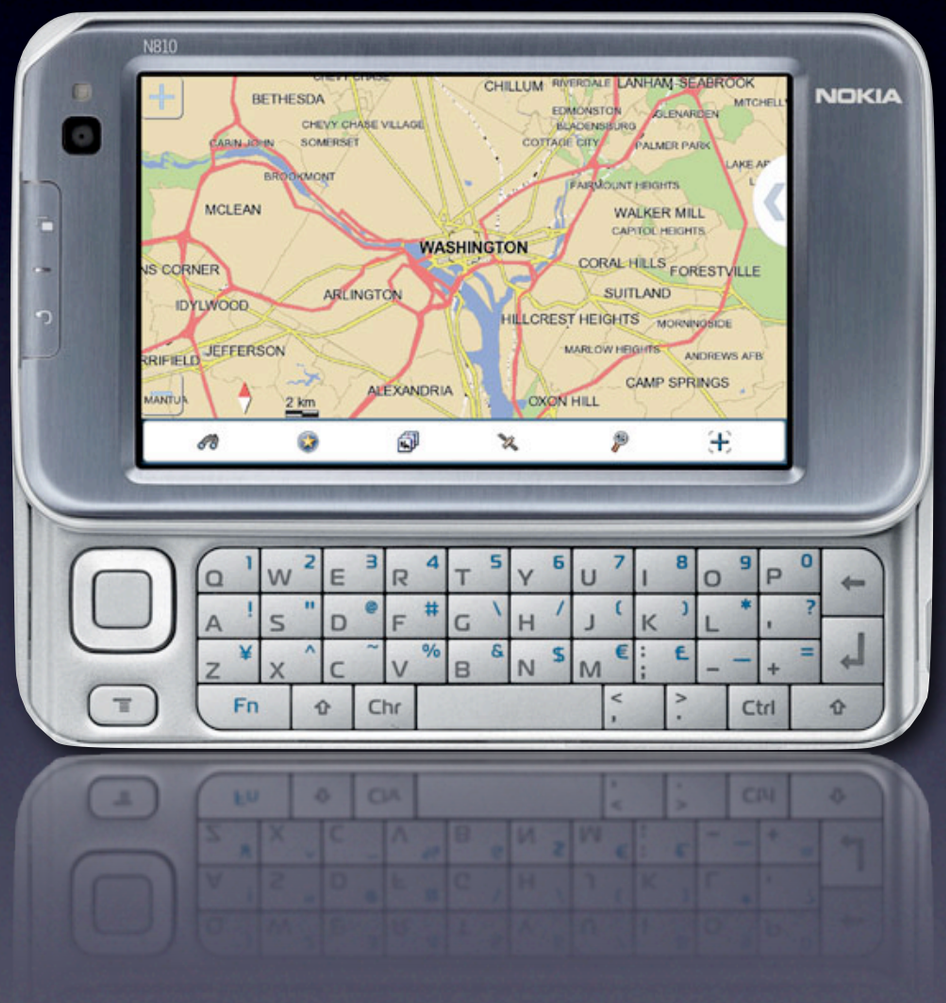
Requisitos

- Conhecimentos em Python
- Conhecimentos em Linux
- VMWare
- Imagem Maemo SDK obtida em: http://maemovmware.garage.maemo.org/

Desejável

- Conhecimentos de PyGTK
- Um *Internet Tablet* (N770, N800, N810)

Internet Tablets

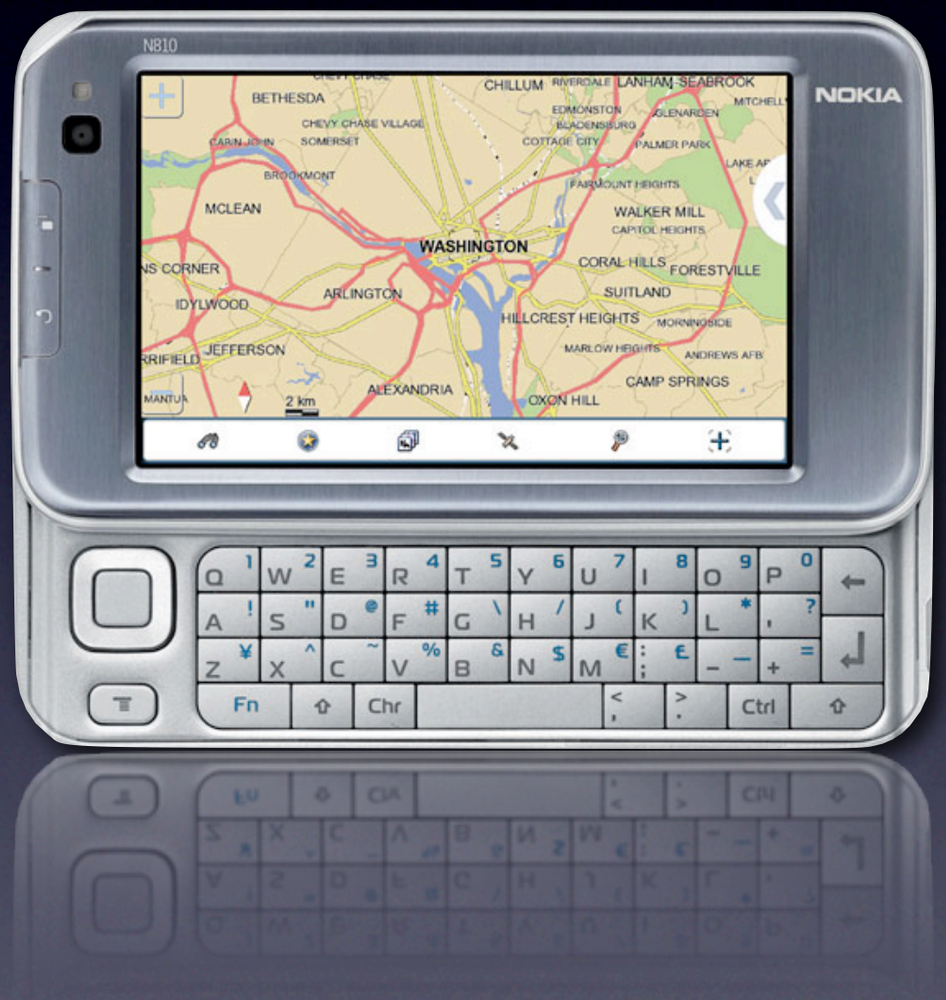


Dispositivos portáteis voltados para uso da Internet.

Nokia N800

- OMAP 2420
- ARM11 330MHz
- 800x480x16
- WLAN, Bluetooth, USB
- Câmera VGA
- 2x cartões SD

Nokia N810



- OMAP 2420
- ARM11 400MHz
- 800x480x16
- WLAN, Bluetooth, USB
- Câmera VGA
- 1x cartão miniSD
- GPS
- Teclado

- Plataforma desenvolvida pela Nokia para equipar a sua linha de *Internet Tablets*
- Baseada no Debian Linux
- Utiliza uma série de componentes conhecidos no universo do Software Livre

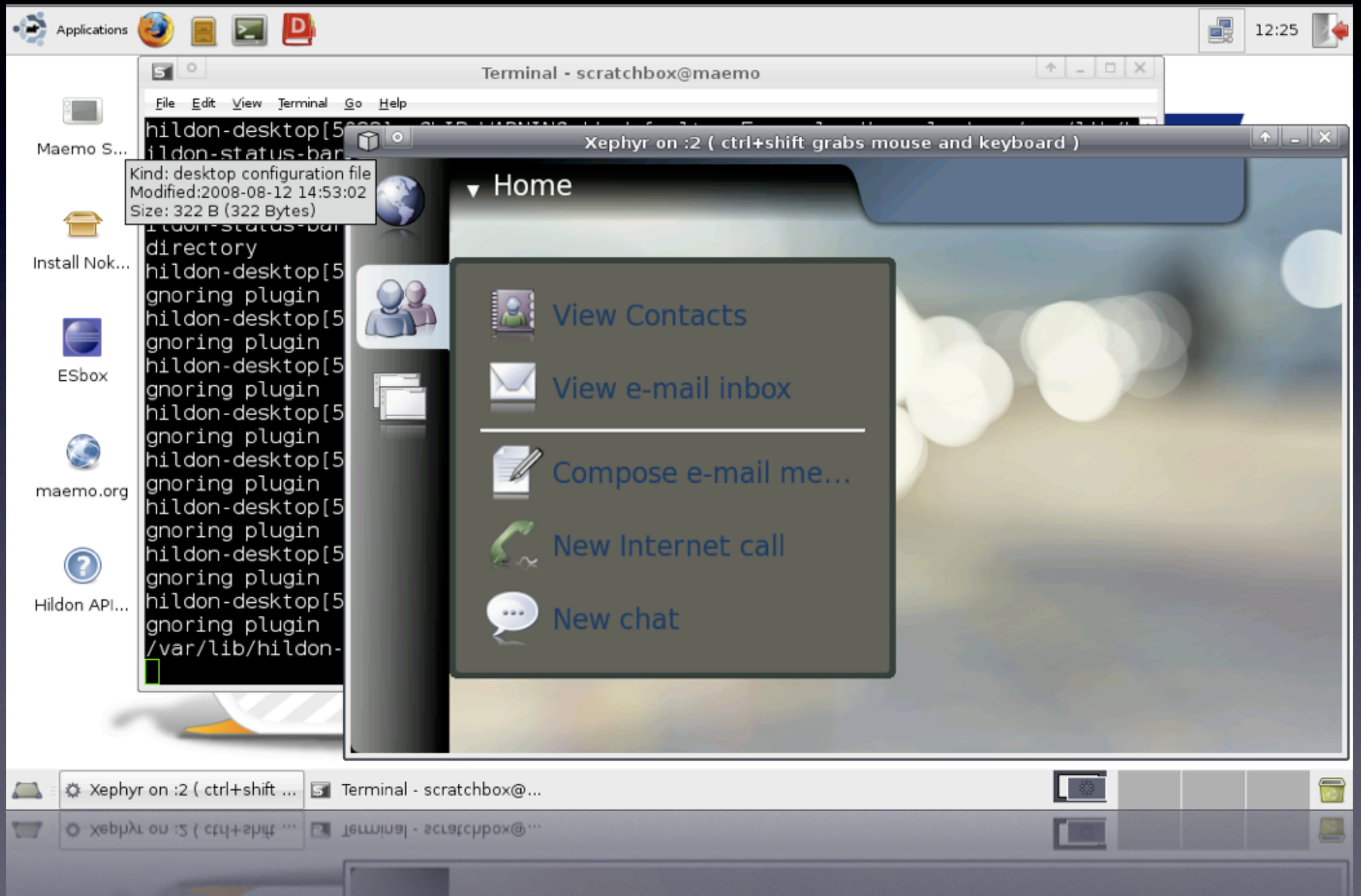
Ambiente de Desenvolvimento

- Scratchbox
  - Ambiente para cross-compiling
    - SDK_X86 - arquitetura ‘host’
    - SDK_ARMEL - arquitetura do dispositivo
- Xephyr
  - Servidor X para emulação da interface gráfica

Iniciando o ambiente

- Entrando no Scratchbox:
  - /scratchbox/login ou
- Iniciando os serviços:
  - af-sb-init.sh start
- Executando aplicações gráficas:
  - run-standalone.sh programa

A primeira impressão...

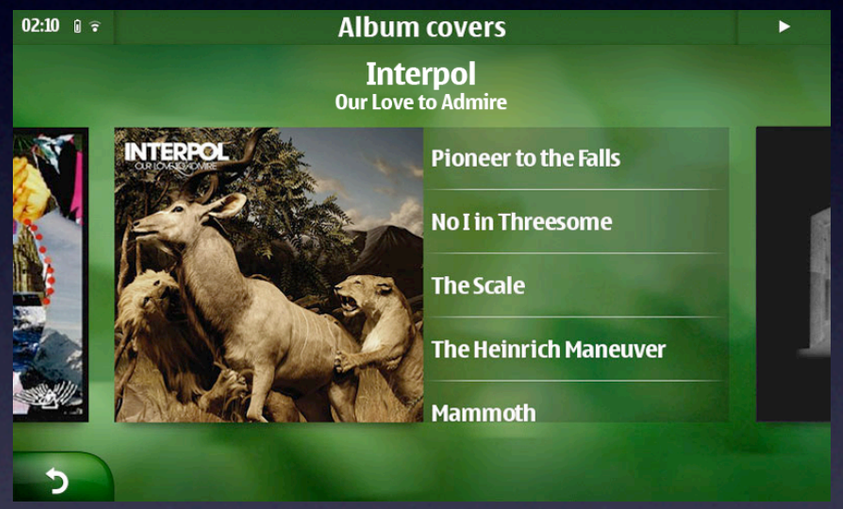


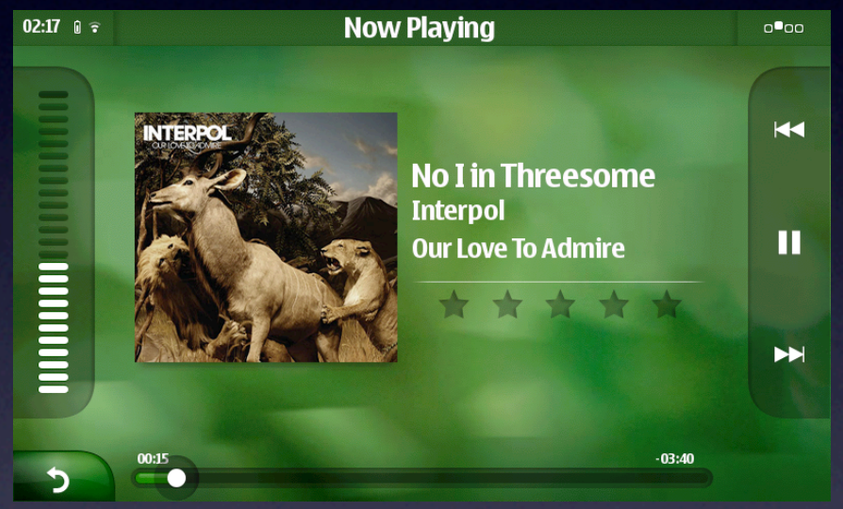
Python para Maemo

- Principal alternativa à linguagem C para desenvolvimento na plataforma
- Bastante difundida no mundo do Software Livre
- Mantido pelo INdT de Recife
- http://pymaemo.garage.maemo.org

Projetos em Python

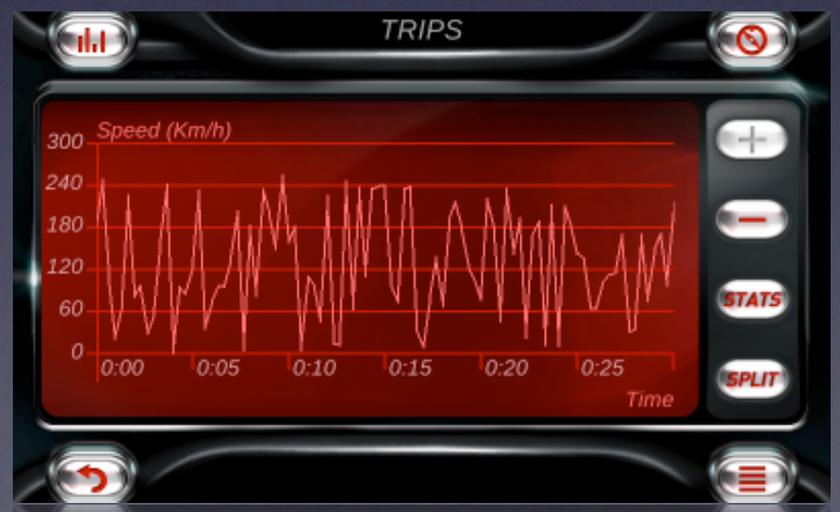
Canola





Projetos em Python

Carman



Bibliotecas Python

- Gerais
  - PyGTK/Gobject
  - D-BUS
  - GStreamer
  - Bluez
  - Pyrex
- PIL
- Gnome VFS/GConf
- Específicas Maemo
  - Hildon
  - LibOSSO
  - OSSO-Addressbook

Easy

- Facilita ainda mais o desenvolvimento para Maemo
- Desenvolvido pela UFCG com apoio do INdT
- Incorpora a biblioteca Eagle para desenho de Interfaces gráficas
- http://easy.garage.maemo.org/

Vamos começar...

Aplicação de exemplo

- Lista de tarefas
- Adicionar Tarefas
- Remover Tarefas
- Editar Tarefas
- Marcá-las como ‘prontas’
- Gravar as informações automaticamente

Lembretes importantes

- Use sempre:
  - python2.5
- O easy não vem instalado por padrão
  - Adicione...

    deb http://repository.maemo.org/extras-devel diablo free non-free
  - ... no /etc/apt/sources.list
  - apt-get install easy

Módulos & Armazenamento

```
#!/usr/bin/env python2.5

import shelve
from easy import ui

storage = shelve.open("tasks.dat", 'c')
```

Definindo a Tabela

```
table = ui.Table(
        id="todo_table",
        headers=("Done", "Description"),
        types=(bool, str),
        editable=True,
        data_changed_callback=change_task,
)
```

Manipulando a tabela

```
def change_task(app, table, data):
    global storage

    key, task = data

    if task is None:
        del storage[str(key)]
    else:
        storage[str(key)] = tuple(task)
    storage.sync()
```

Tela principal

```
ui.App(
   id="todo_app",
   title="To Do List",
center=(table,),
bottom=(
   ui.Button(
      id="quit_button",
      label="_Quit",
      expand_policy=ui.ExpandPolicy.All(),
      callback=exit)))
```

Fechar a aplicação

```
def exit(app, widget):
    storage.close()
    app.close()
```

Populando a tabela

```
def populate_table():
  global storage
  global table

  ks = int(k) for k in storage.keys()
  for k in sorted(ks):
    table.append( storage[str(k)] )
```

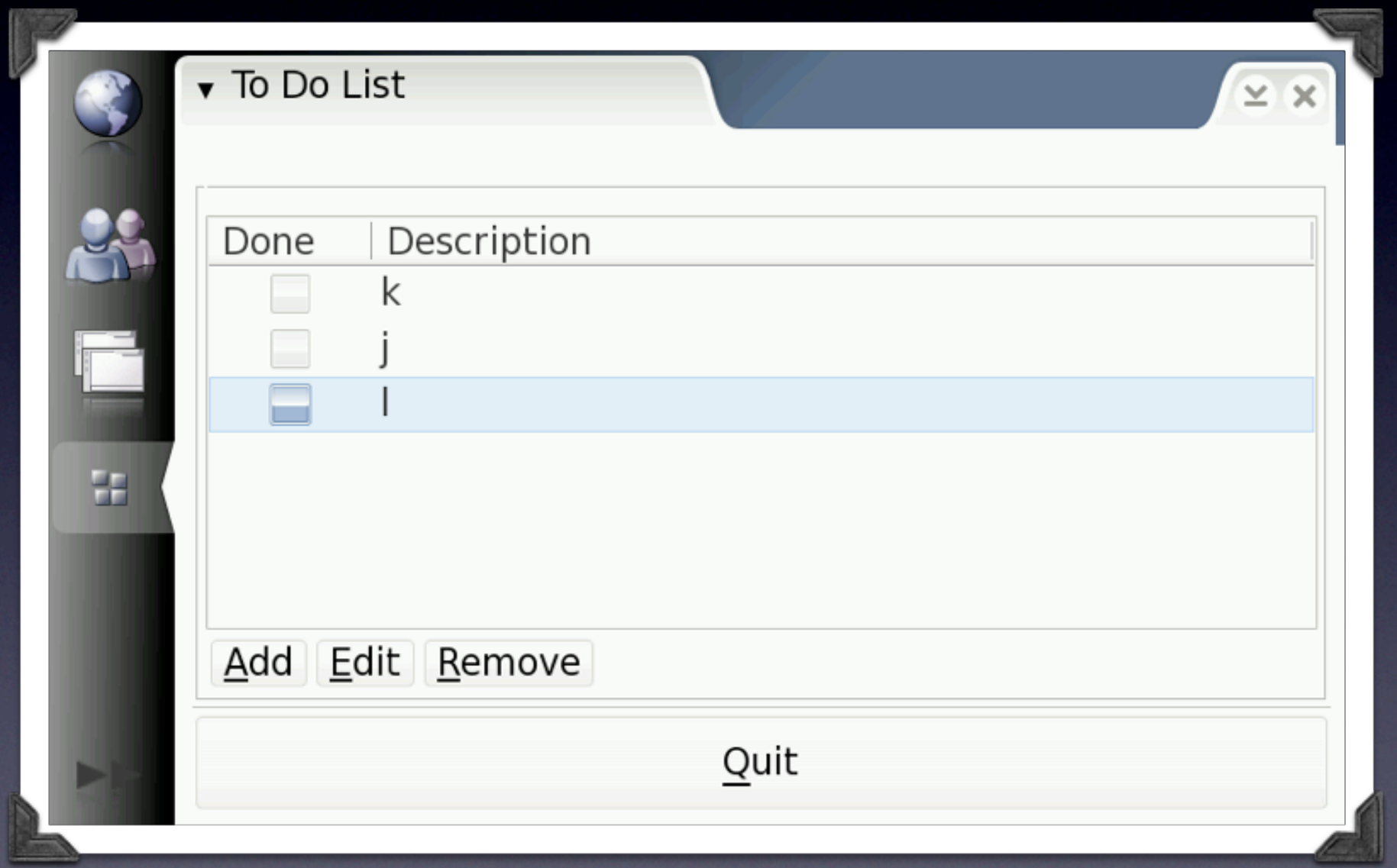
Rodando

```
if __name__ == "__main__":
    populate_table()
    ui.run()
```

Hora de executar...

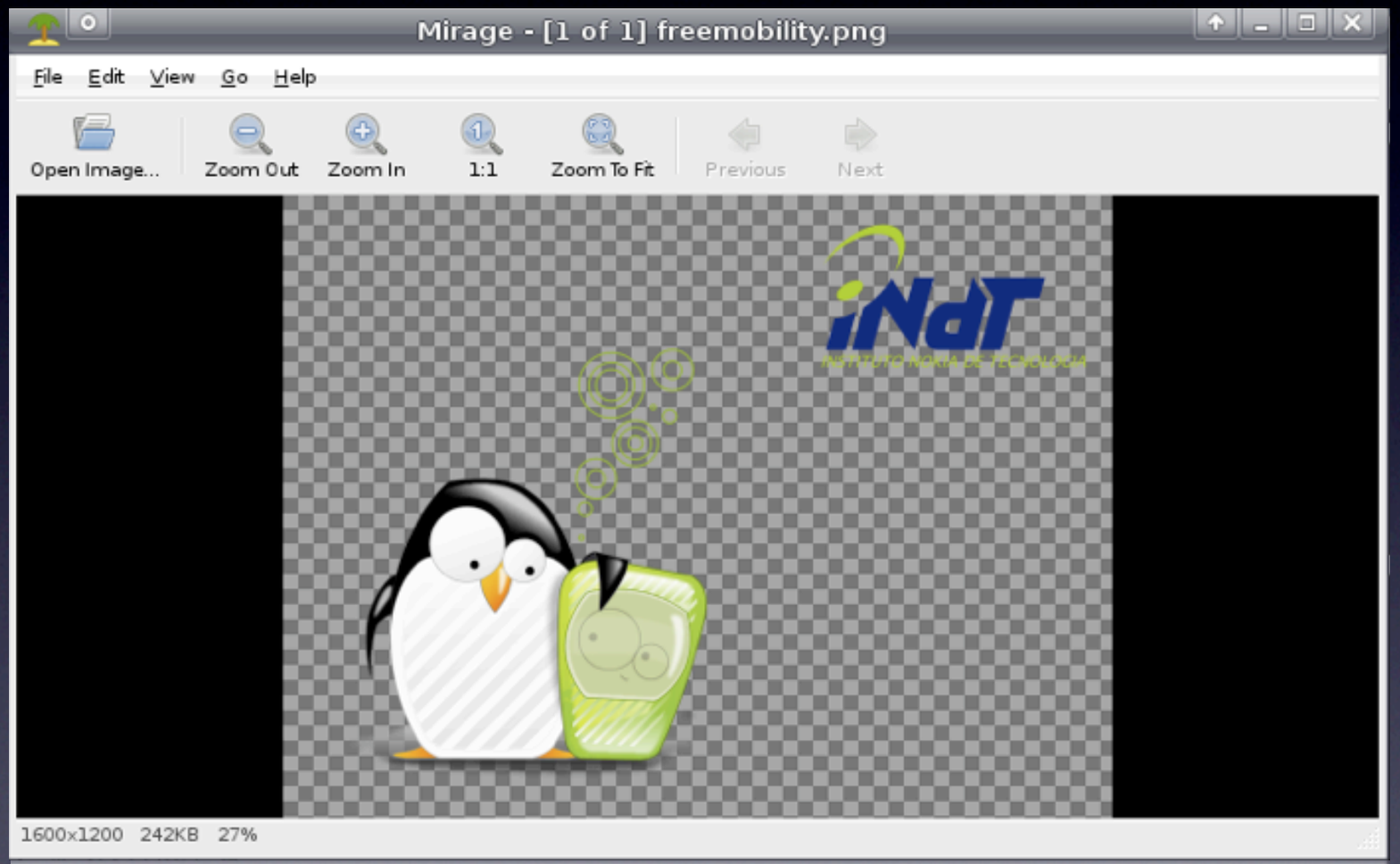
```
SDK_X86:~> run-standalone.sh python2.5 tasks.py
```

... e pronto.



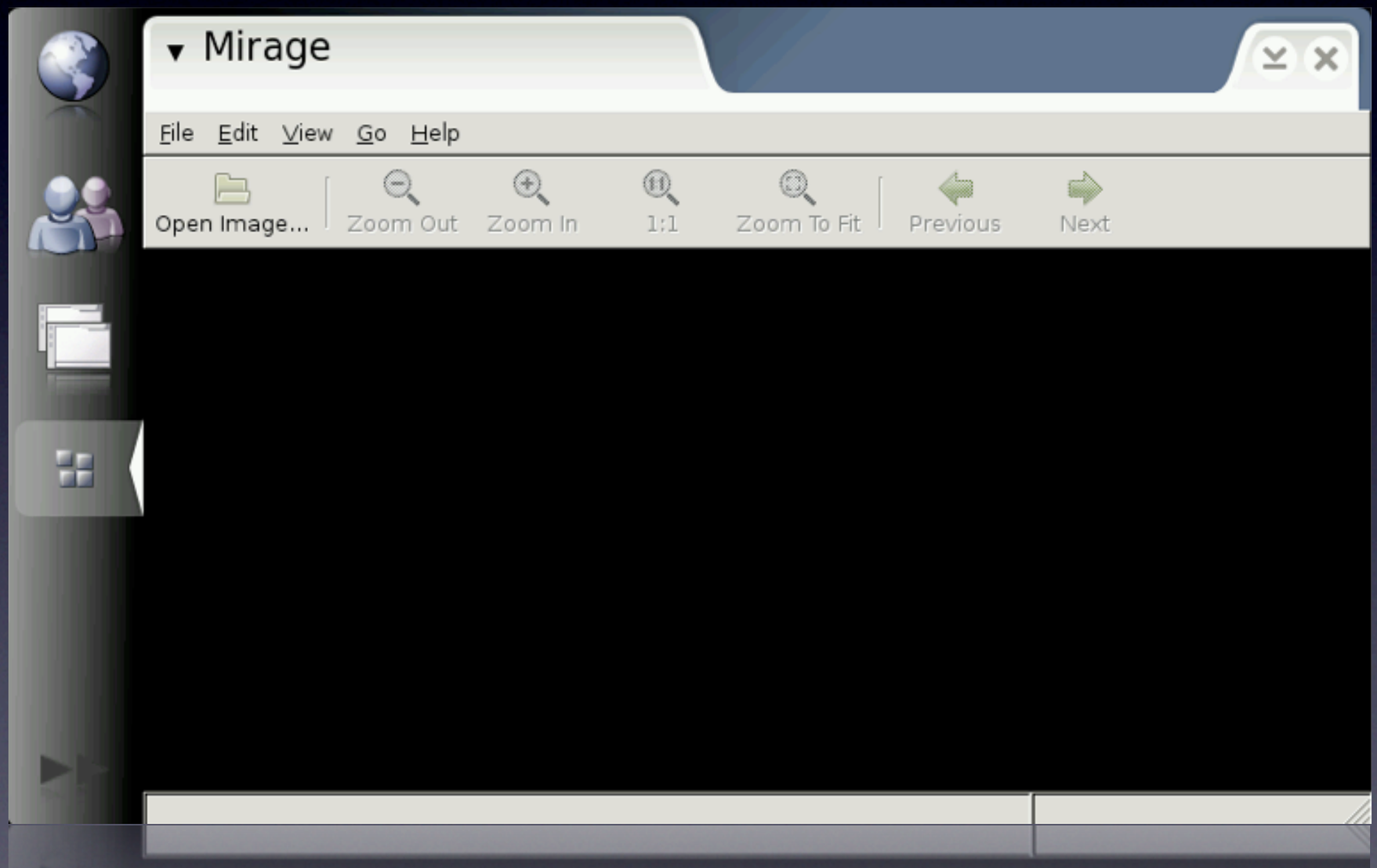
Portando uma aplicação

A Aplicação



Desktop

A Aplicação



Maemo sem adaptação...

Passo a passo do porte

- Substituir `gtk.Window(gtk.WINDOW_TOPLEVEL)` por `hildon.Window()`
- Fazer ‘reparent’ dos menus e toolbars
- Substituir diálogos GTK+ pelos do Hildon
- Adaptações para espaço reduzido na tela

Substituindo a Window

```
@@ -26,6 +26,7 @@
 pygtk.require('2.0')
 import gtk
 import gtk.gdk
+import hildon
 import os
 import sys, getopt
 import ConfigParser
@@ -478,7 +479,7 @@
         """

     # Create interface
-    self.window = gtk.Window(gtk.WINDOW_TOPLEVEL)
+    self.window = hildon.Window()
     self.update_title()
     icon_path = self.find_path('mirage.png')
     try:
```

Novos pais para o Menu

```
@@ -498,7 +498,10 @@
     self.refresh_recent_files_menu()

   self.window.add_accel_group(self.UIManager.get_accel_group())
     self.menubar = self.UIManager.get_widget('/MainMenu')
-    vbox.pack_start(self.menubar, False, False, 0)
+    self.menu = gtk.Menu()
+    for item in self.menubar:
+       item.reparent(self.menu)
+    self.window.set_menu(self.menu)
     self.set_slideshow_sensitivities()
     self.toolbar = self.UIManager.get_widget('/MainToolbar')
     vbox.pack_start(self.toolbar, False, False, 0)
```

A barra de ferramentas

```
@@ -504,7 +504,7 @@
     self.window.set_menu(self.menu)
     self.set_slideshow_sensitivities()
     self.toolbar = self.UIManager.get_widget('/MainToolbar')
-    vbox.pack_start(self.toolbar, False, False, 0)
+    self.window.add_toolbar(self.toolbar)
     self.layout = gtk.Layout()
     self.vscroll = gtk.VScrollbar(None)
     self.vscroll.set_adjustment(self.layout.get_vadjustment())
```

O diálogo correto

```
@@ -1353,7 +1353,7 @@
        self.save_image_now(self.currimg_name, gtk.gdk.pixuf_...

   def save_image_as(self, action):
-    dialog = gtk.FileChooserDialog(title=_("Save
As"),action=gtk.FILE_CHOOSER_ACTION_SAVE,buttons=(gtk.STOCK_CANC
EL,gtk.RESPONSE_CANCEL,gtk.STOCK_SAVE,gtk.RESPONSE_OK))
+    dialog =
hildon.FileChooserDialog(self.window,gtk.FILE_CHOOSER_ACTION_SAV
E)
     dialog.set_default_response(gtk.RESPONSE_OK)
     filename = os.path.basename(self.currimg_name)
     filetype = None
```

Versão final

Distribuição

Pacotes Debian

distutils

Progama &
Biblioteca Python

Ajustando os arquivos

```
setup.py
@@ -23,7 +23,7 @@
      ext_modules = [Extension('imgfuncs', ['imgfuncs.c'])],
      scripts = ['mirage'],
      data_files=[('share/mirage', ['README', 'COPYING',...
-        ('share/applications', ['mirage.desktop']),
+        ('share/applications/hildon', ['mirage.desktop']),
         ('share/pixmaps', ['mirage.png']),
         ('share/locale/ru/LC_MESSAGES', ['locale/ru/LC_M...
         ('share/locale/pl/LC_MESSAGES', ['locale/pl/LC_M...
```

Arquivo .desktop

```
mirage.desktop
@@ -1,10 +1,10 @@
 [Desktop Entry]
 Name=Mirage
 Comment=A fast GTK+ Image Viewer
-Exec=mirage %U
+Exec=mirage
 Terminal=false
 Type=Application
-Icon=mirage.png
+Icon=mirage
 Categories=GTK;Application;Graphics;
 Version=0.8.3
 Encoding=UTF-8
```

Mais Informações

- http://www.maemo.org
- http://pymaemo.garage.maemo.org
- http://easy.garage.maemo.org
- http://www.pygtk.org
- http://code.google.com/p/eagle-py
- http://openbossa.indt.org/canola
- http://openbossa.indt.org/carman

Contatos

e-mail: osantana@triveos.com
google talk: osantana@gmail.com