

## APRESENTANDO O YAD - "ZENITY MELHORADO"

Autor: Raimundo Alves Portela <rai3mb at gmail.com>

Data: 04/06/2011

### APRESENTANDO O YAD

Nesse artigo vou mostrar uma maravilhosa ferramenta para GUI (Interface Gráfica com o Usuário), que acabou de sair do forno, o *Yad*, um fork do *Zenity* que vem com diversas melhorias, dentre elas a possibilidade de fazer formulários, melhorando a interação do usuário com nossos scripts em shell.

O bichinho é tão recente que nem o manual cobre todas as opções disponíveis, cito um exemplo mais adiante no uso de diálogo para selecionar fontes.

O autor do Yad é o ucraniano Victor Ananjevsky, e a página do projeto pode ser encontrada no:

- ▶ <http://code.google.com/p/yad/> (<http://code.google.com/p/yad/>)

Mesmo nessa página ainda há pouca informação sobre o uso do Yad, o melhor lugar que achei no momento é o manual dele (man yad) :-).

Caso forem buscar informações, pouca coisa ainda está disponível, o manual está datado em "Marth 24, 2011", e ainda tem o fato de Yad ser o nome de uma ferramenta "...usada em para apontar o texto durante a leitura do Torá..." (Fonte: <http://en.wikipedia.org/wiki/Yad> (<http://en.wikipedia.org/wiki/Yad>)). Então ao buscar informações, busque por 'yad fork zenity' vai ser mais fácil.

Ok, vamos em frente pois no artigo pretendo mostrar vários recursos do Yad, com um pequeno exemplo de cada um.

### INSTALAÇÃO

Baixe a versão mais recente do Yad no link abaixo:

- ▶ <http://code.google.com/p/yad/downloads> (<http://code.google.com/p/yad/downloads/>)

Descompacte:

```
$ tar -Jxf arquivo.tar.xz
```

Instale:

```
$ ./configure
```

```
$ sudo make
```

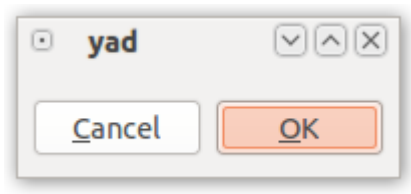
```
$ sudo make install
```

Pode acontecer dele reclamar de alguma dependência, aqui no meu *Debian*, reclamou do *libgtk2*, basta instalar com:

```
$ sudo apt-get install libgtk2.0-dev
```

Após instalar, podemos testar com:

```
$ yad
```



Prontinho, vamos à parte divertida e viciante, programar :-)

## POSSIBILIDADES E EXEMPLOS DE USO

A sintaxe básica é :

```
yad [--tipo-dialogo] [--options]
```

Abaixo mostro exemplo com os tipos de diálogo:

- calendar (calendário)
- color (paleta de cores)
- entry (entrada de dados)
- icons (mostra uma caixa com ícones de atalho para aplicações)
- file (diálogo para selecionar arquivos)
- font (diálogo para seleção de fontes)
- form (aeeeeeeee formulários)
- list (diálogo com ítems em lista)
- notification (mostra um ícone da barra de notificação do sistema)
- progress (diálogo de progresso)
- text-info (mostra o conteúdo de um arquivo texto)
- scale (diálogo para seleção de valor, usando uma escala)

Vamos lá, para cada um dos exemplos coloco o script e depois algumas imagens da sua execução.

--calendar:

Mostra um calendário permitindo selecionar a data e envia o valor para a saída padrão

Exemplo:

```
#!/bin/bash
# uso do yad com --calendar

# mostra um calendário iniciando com o dia 20/05/2011
# guarda o valor selecionar na variável $DATA

DATA=$(
    yad --calendar \
        --day=20 \
        --month=5 \
        --year=2011 \
        --date-format=%dV%mV%Y \
        --title=Calendario \
        --center \ # disposição do diálogo na tela
)

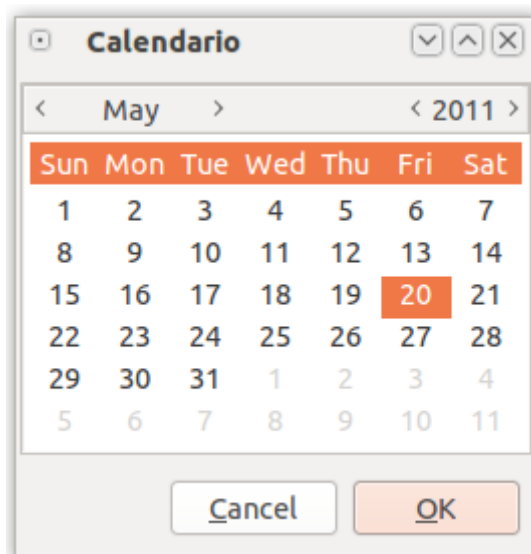
# mostra um diálogo informando a $DATA selecionada
yad --title="AVISO" \    --text="Você selecionou a data $DATA"
#.EOF
```

Salve com o nome de "calendar.sh", dê permissão de execução:

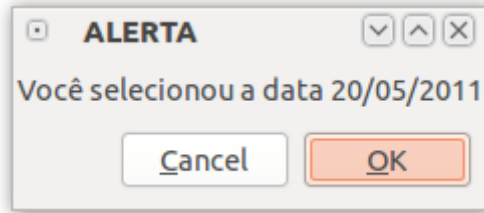
```
$ chmod +x calendar.sh
```

Execute:

```
$ ./calendar.sh
```



Após selecionar uma data:



## EXEMPLO COM YAD --COLOR

yad --color:

Diálogo de seleção de cores, permite selecionar uma determinada cor usando a paleta de cores, editando diretamente por código, etc. Bom para saber o código de uma cor específica.

Exemplo:

```
#!/bin/bash
# uso do yad - com color
# permite selecionar determinada cor numa paleta de cores
# e envia o valor para a saída padrão, no caso armazenei na variável $COR

COR=$(
yad --color \
    --init-color="#FFFFFF" \ #cor que inicialmente fica selecionada na paleta de cores.
    --palette \
)

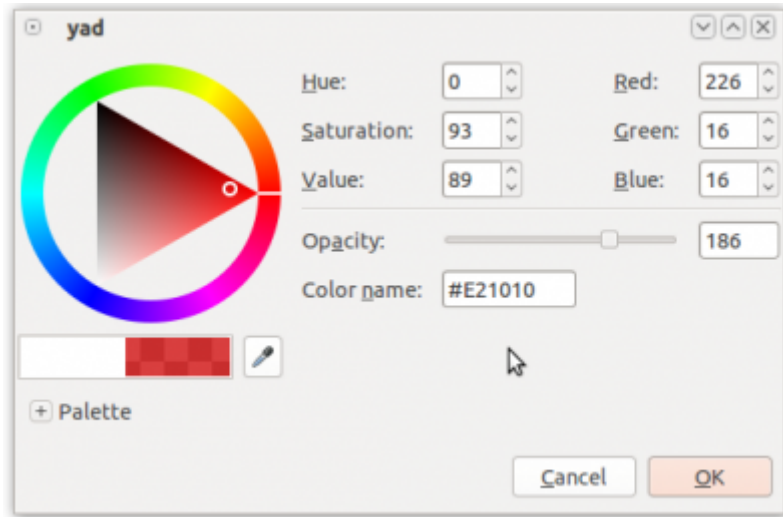
yad --title="YAD COM COLOR" \
    --text="Você selecionou a cor $COR"
#.EOF
```

Salve com o nome de "color.sh", dê permissão de execução:

```
$ chmod +x color.sh
```

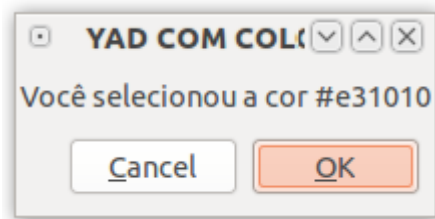
Execute:

```
$ ./color.sh
```



(//img.vivaolinux.com.br/imagens/artigos/comunidade/color01.png)

Após selecionar uma cor:



## EXEMPLOS 1

### EXEMPLO COM YAD --ENTRY

yad --entry:

Diálogo de entrada de texto, onde podemos fazer entrada de dados, modificando a forma como essa entrada será feita, podendo ser do tipo senha(password), texto e número.

```
#!/bin/bash
# yad com --entry
# recebe valores e armazena em variáveis
NOME=$(
  yad --entry \
    --entry-label="Nome" \
    --entry-text="seu nome aqui" \
    --completion \
    --editable \
    --title="USO DO YAD COM --entry" \
    --width="300" \
```

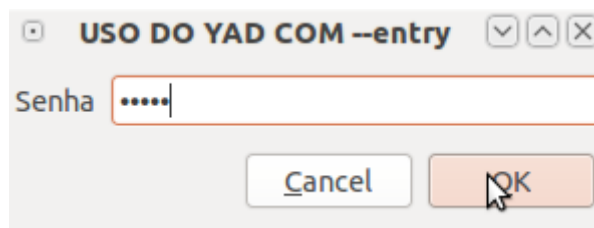
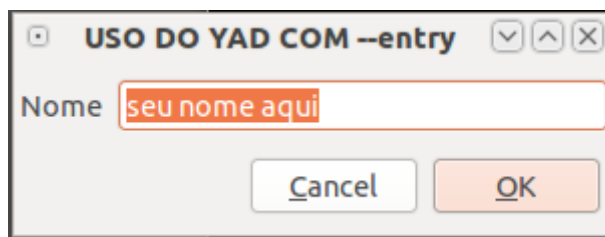
```
)  
  
SENHA=$(  
  yad --entry \  
  --entry-label="Senha" \  
  --entry-text="123456" \  
  --completion \  
  --editable \  
  --hide-text \#senha – esconde o texto digitado apresentando apenas um "." ponto  
  --title="USO DO YAD COM --entry" \  
  --width="300" \  
)  
  
yad --title="YAD COM --entry" \  
  --text="Nome: $NOME, Senha: $SENHA" \  
  --width="300" \  
  --height="100"  
  
#.EOF
```

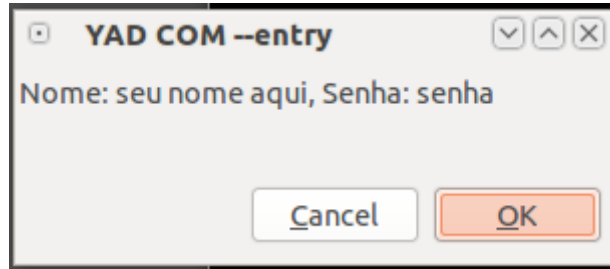
Salve com o nome de entry.sh, dê permissão de execução:

```
$ chmod +x entry.sh
```

Execute:

```
$ ./entry.sh
```





## EXEMPLO COM YAD --ICONS

yad --icons:

Diálogo que mostra uma caixa com os ícones de atalho (formato .desktop).

Exemplo:

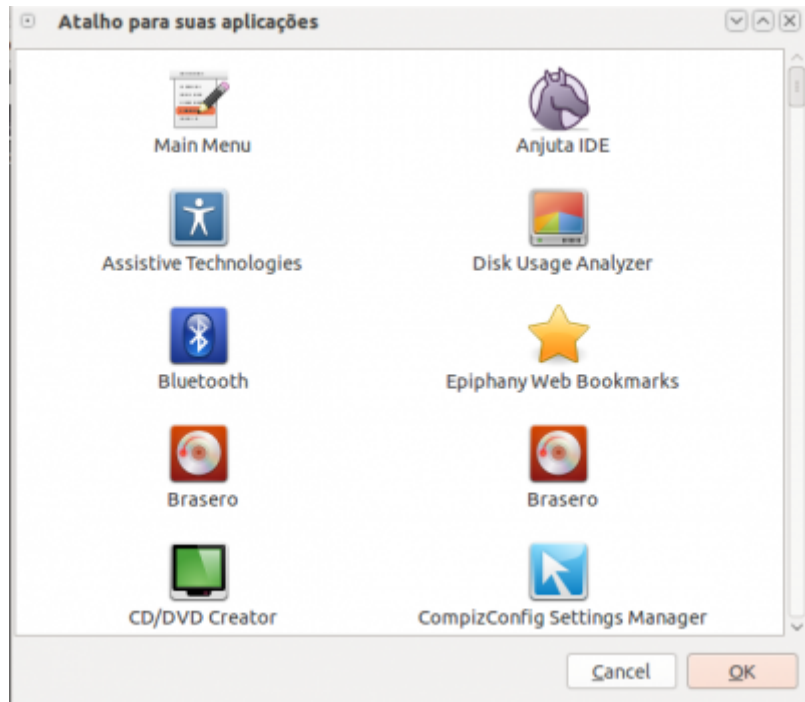
```
#!/bin/bash
# yad com icons
# mostra os ícones das aplicações instaladas no seu sistema
# você pode abrir uma aplicação clicando no ícone correspondente
yad --icons \
    --read-dir="/usr/share/applications" \
        --width="600" \
        --height="500"
#.EOF
```

Salve com o nome de icons.sh, dê permissão de execução:

```
$ chmod +x icons.sh
```

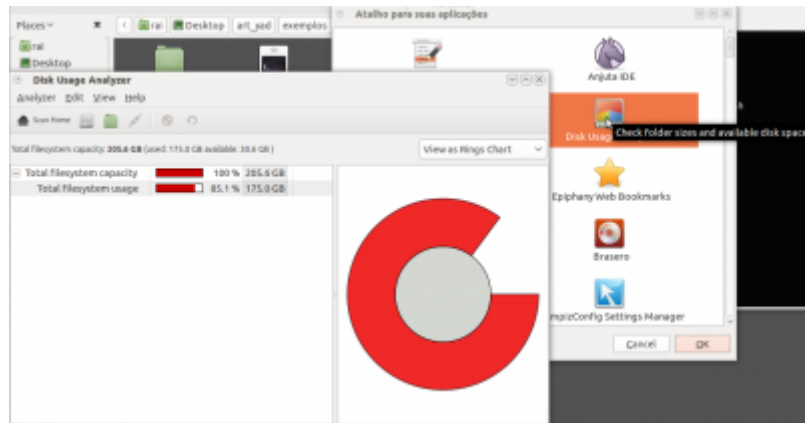
Execute:

```
$ ./icons.sh
```



(//img.vivaolinux.com.br/imagens/artigos/comunidade/icons01.png)

Após clicar duas vezes em um ícone ele abre aplicação correspondente:



(//img.vivaolinux.com.br/imagens/artigos/comunidade/icons02.png)

## EXEMPLO COM YAD --FILE

yad --file:

Dialogo que permite selecionar um arquivo/diretório, retornando o caminho do mesmo.

Exemplo:

```
#!/bin/bash
# yas com file
# permite selecionar um arquivo/diretório através de caixa de diálogo
# retorna o caminho do arquivo/diretório, deixando a nós o critério de fazer
# o que quiser com ele, nesse caso só vou mostrar o tamanho
ARQUIVO=${}
```



```

yad --file \
    --directory \
    --title="Selecione um diretório para ver seu tamanho" \
    --width="500" \
    --height="300" \
)

TAMANHO=$(du -sh $ARQUIVO | cut -d"/" -f 1)

yad --title="Tamanho do diretório" \
    --text="Tamanho do diretório $ARQUIVO = $TAMANHO" \
    --width="400" \
    --height="100" \
    --button=gtk-ok:0

# a opção button=gtk-ok:0 - mostra apenas o botão de ok no dialog
# e seu código de retorno é 0 (zero)
#.EOF

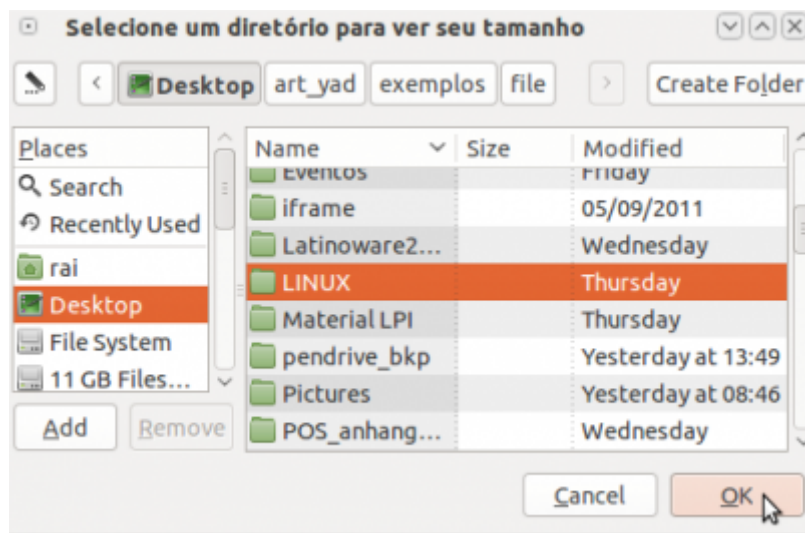
```

Salve com o nome de "file.sh", dê permissão de execução:

```
$ chmod +x file.sh
```

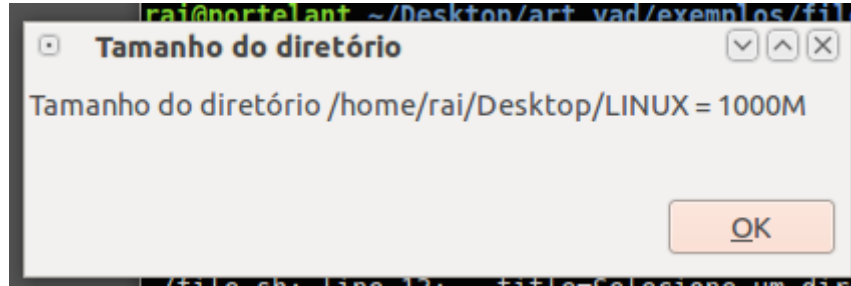
Execute:

```
$ ./file.sh
```



(//img.vivaolinux.com.br/imagens/artigos/comunidade/file01.png)

Após selecionar o diretório ele mostra o tamanho do mesmo:



## EXEMPLO COM YAD --FONT

yad --font:

Mostra um diálogo para seleção de fontes.

Exemplo: (detalhe o manual não mostra que podemos colocar o tamanho da fonte depois do nome, descobrir testando mesmo).

```
#!/bin/bash
# yad com font
# mostra um diálogo de seleção de fonte
# retorna o nome da fonte selecionada e tamanho selecionado, gravando num html
RESULTADO=$(
  yad --font \
    --fontname="Ubuntu",14 \
    --preview="Texto para exemplo do uso de fonte yad é 10 :-)" \
    --title="Selecione uma fonte" \
)
# usa expressão regular para pegar apenas o texto
FONTE=$(echo $RESULTADO | grep -o '[:alpha:]*')
# usa expressão regular para pegar apenas o tamanho da fonte, 2 dígitos
TAMANHO=$(echo $RESULTADO | egrep -o '[0-9]{2}')

# grava o resultado num arquivo html para fazer um teste um pouco útil :-)
echo "texto com a fonte $FONTE e TAMANHO $TAMANHO " > arq_teste.htm

# abre o firefox para fazer um teste
firefox file://$PWD/arq_teste.htm

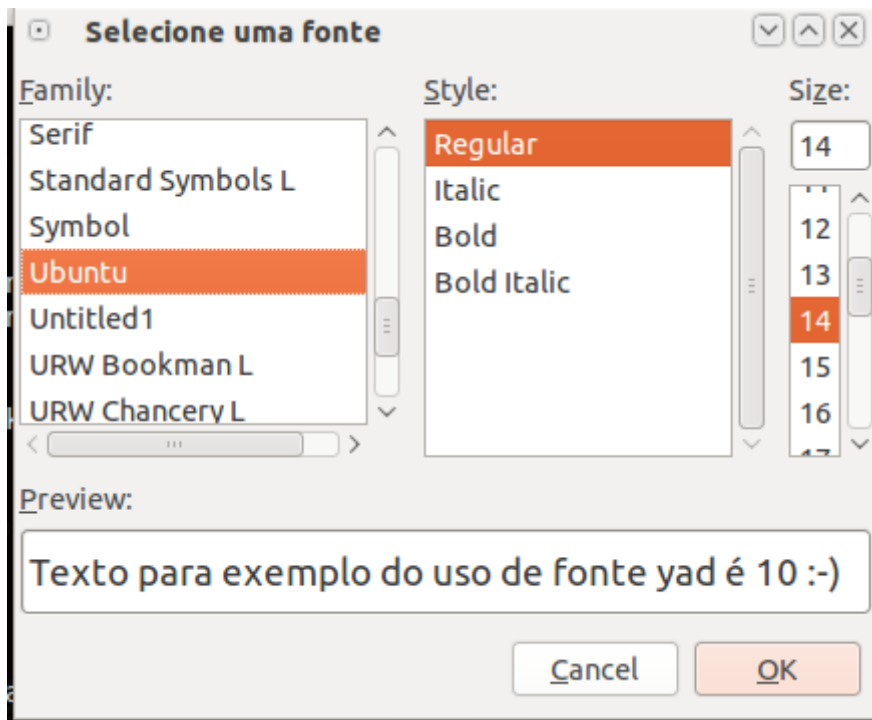
#.EOF
```

Salve com o nome de "font.sh", dê permissão de execução:

```
$ chmod +x font.sh
```

Execute:

\$ ./font.sh



Após selecionar a fonte e tamanho, abre um página do *Firefox* com o resultado:



(//img.vivaolinux.com.br/imagens/artigos/comunidade/font02.png)

## EXEMPLO COM YAD --FORM

```
yad --form;
```

Uma das novidades que mais gostei: FORMULÁRIOS, não dava pra fazer isso no *Zenity*, e limitava as entradas do usuário tendo que aprensetar várias telas para coletar dados, o que não é muito prático.

Exemplo:

```
#!/bin/bash
# yad com FORMULARIOS, que bacana, tem algumas novidades aprecie com calma
HOJE=$(date +%d/%m/%Y)
touch $HOME/cad_usuario.txt

VAR_FORM=$( \
  yad --form \
```

```
--title="CADASTRO DE USUÁRIOS" \
--width=400 \
--height=400 \
--image="accessories-text-editor" \
--field="Cadastrando em":RO "$HOJE" \
--field="Nome:" "" \
--field="Sexo":CB Masculino!Feminino \
--field="Login:" "$USER" \
--field="Senha:":H "123456" \
--field="Ler livro sobre:":RO "" \
--field="Programação":CHK FALSE \
--field="Redes":CHK FALSE \
--field="Linux":CHK TRUE \
--field="Quantos Filhos:":NUM 0!0..10!1 \
--field="Foto:":FL "$HOME/Pictures" \
--field="Salvar dados em:":DIR "$HOME" \
--field="Cor preferida:":CLR "#00DDDD" \
)
```

```
DATA=$(echo "$VAR_FORM" | cut -d'|' -f 1)
```

```
NOME=$(echo "$VAR_FORM" | cut -d'|' -f 2)
```

```
SEXO=$(echo "$VAR_FORM" | cut -d'|' -f 3)
```

```
LOGIN=$(echo "$VAR_FORM" | cut -d'|' -f 4)
```

```
SENHA=$(echo "$VAR_FORM" | cut -d'|' -f 5)
```

#notem que pulei o campo 6, pois o mesmo nao retorna valor útil para nós

```
LIVRO_PROGRAMACAO=$(echo "$VAR_FORM" | cut -d'|' -f 7)
```

```
LIVRO_REDES=$(echo "$VAR_FORM" | cut -d'|' -f 8)
```

```
LIVRO_LINUX=$(echo "$VAR_FORM" | cut -d'|' -f 9)
```

```
QTD_FILHOS=$(echo "$VAR_FORM" | cut -d'|' -f 10)
```

```
FOTO=$(echo "$VAR_FORM" | cut -d'|' -f 11)
```

```
SALVAR_EM=$(echo "$VAR_FORM" | cut -d'|' -f 12)
```

```
COR=$(echo "$VAR_FORM" | cut -d'|' -f 13)
```

# repare que o campo \$QTD\_FILHOS retorna um valor no formato 0.000000

# vc pode usar uma expressão regular, ou o cut, para capturar apenas os digitos antes do .

```
QTD_FILHOS=$(echo $QTD_FILHOS | cut -d"." -f 1)
```

#apenas vou mostrar os campos e valores de retorno em um html

```
DADOS_USUARIO="<table border=1><br/>
```

```
<tr><th>CAMPO</th><th>VALOR</th></tr>
```

```
<tr><td>DATA</td><td>$DATA</td></tr>
```

```
<tr><td>NOME</td><td>$NOME</td></tr>
```

```
<tr><td>SEXO</td><td>$SEXO</td></tr>
```

```
<tr><td>LOGIN</td><td>$LOGIN</td></tr>
```

```
<tr><td colspan=2 align=center>LIVROS</td></tr>
```

```
<tr><td>PROGRAMACAO</td><td>$LIVRO_PROGRAMACAO</td></tr>
```

```
<tr><td>REDES</td><td>$LIVRO_REDES</td></tr>
```

```
<tr><td>LINUX</td><td>$LIVRO_LINUX</td></tr>
```

```
<tr><td>QTD_FILHOS</td><td>$QTD_FILHOS</td></tr>
<tr><td>FOTO</td><td><img src='$FOTO' /></td></tr>
<tr><td>DIRETORIO</td><td>$SALVAR_EM</td></tr>
<tr><td>COR</td><td bgcolor='$COR'></td></tr>
</table>"
```

```
echo $DADOS_USUARIO > cad_usuario.htm
```

```
firefox cad_usuario.htm
```

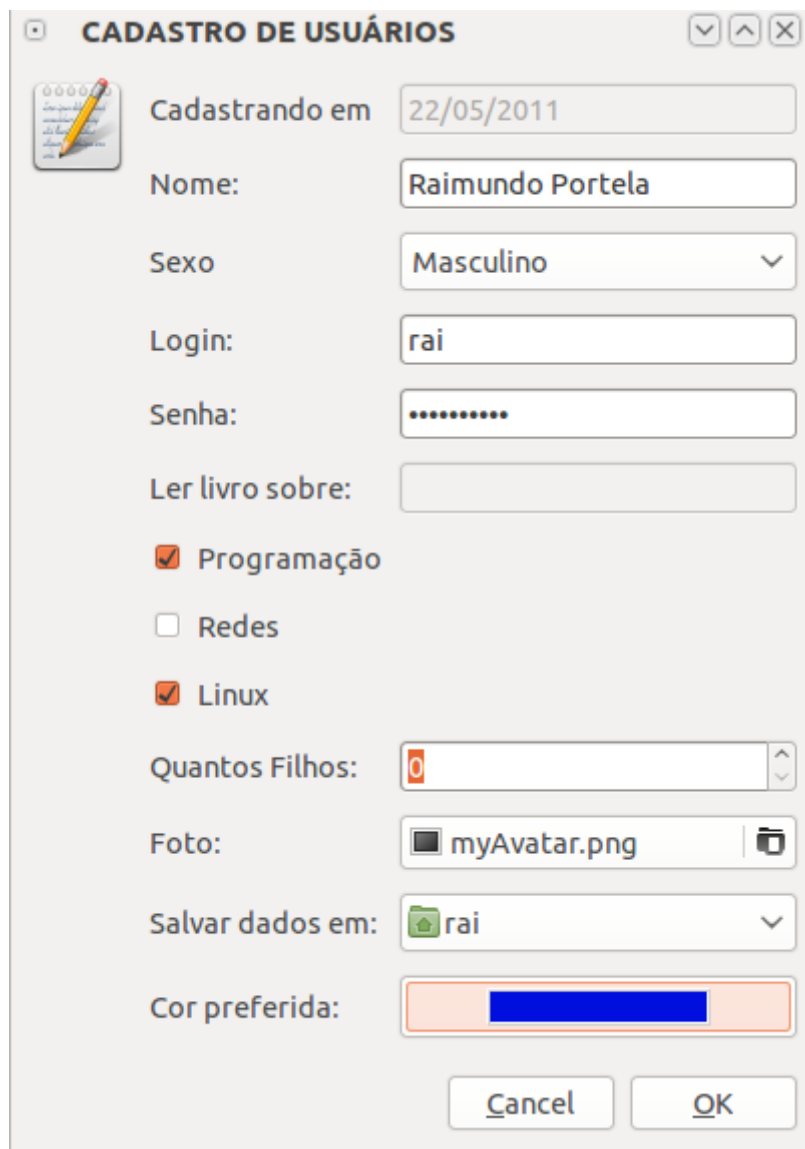
```
#.EOF
```

Salve com o nome de form.sh, der permissão de execução:

```
$ chmod +x form.sh
```

Execute:

```
$ ./form.sh
```



**CADASTRO DE USUÁRIOS**

Cadastrando em: 22/05/2011

Nome: Raimundo Portela

Sexo: Masculino

Login: rai

Senha: .....

Ler livro sobre:

- Programação
- Redes
- Linux

Quantos Filhos: 0


Foto: myAvatar.png

Salvar dados em: rai

Cor preferida: [Blue]

Cancel OK

Após preencher o formulário:

CAMPO	VALOR
DATA	22/05/2011
NOME	Raimundo Portela
SEXO	Masculino
LOGIN	rai
LIVROS	
PROGRAMACAO	TRUE
REDES	FALSE
LINUX	TRUE
QTD_FILHOS	0
FOTO	
DIRETORIO	/home/rai
COR	

## EXEMPLOS 2

### EXEMPLO COM YAD --LIST

yad --list:

Mostra um diálogo do tipo lista de seleção.

Exemplo:

Fiz um pequeno exemplo que usa uma lista com uma coluna do tipo texto e outra do tipo imagem

```
#!/bin/bash
# yas com list
# Permite ao escolher uma personalista na lista e ver detalhes sobre ela na internet :-)
# Tem uma coluna com o nome e outra com a imagem da personalidade, se alguém
```

```
# copiar esse script, vai aparecer os nomes, mas não as imagens :-), pois elas no meu
# disco, é apenas um exemplo
```

```
show_list() {
```

```
ESCOLHIDO=$( \
```

```
  yad --list \
```

```
  --title="Detalhes sobre Personalidades" \
```

```
  --column="ID":NUM \
```

```
  --column="PERSONALIDADE":TEXT \
```

```
  --column="FOTO":IMG \
```

```
  --print-column=1 \
```

```
  --hide-column=1 \
```

```
  --width="300" \
```

```
  --height="400" \
```

```
  --button="Sair":1 \
```

```
  --button="Ver detalhes":0 \
```

```
  1 "rai3mb :-)" "myAvatar.png" \
```

```
  2 "Linus Torvalds" "linus.png" \
```

```
  3 "Richard Stallman" "stallman.png" \
```

```
  4 "Mad Dog" "maddog.png" \
```

```
)
```

```
# ver se o usuário clicou em 'sair' or no 'x' da janela
```

```
ACAO="$?"
```

```
test "$ACAO" -eq "1" || test "$ACAO" -eq "252"
```

```
if [ "$?" -eq "0" ]; then
```

```
  exit
```

```
fi
```

```
#captura apenas o numero da opcao selecionada
```

```
ESCOLHIDO=$(echo $ESCOLHIDO | egrep -o '[0-9]')
```

```
# de acordo com a opcao selecionada, abre o firefox com detalhes da personalidade
```

```
case "$ESCOLHIDO" in
```

```
  1)
```

```
    firefox http://www.vivaolinux.com.br/perfil/verPerfil.php?login=rai3mb
```

```
    show_list
```

```
  ;;
```

```
  2)
```

```
    firefox http://pt.wikipedia.org/wiki/Linus_Torvalds
```

```
    show_list
```

```
  ;;
```

```
  3)
```

```
    firefox http://pt.wikipedia.org/wiki/Stallman
```

```
    show_list
```

```
  ;;
```



```
4)
firefox http://pt.wikipedia.org/wiki/Jon_Hall
show_list
;;
*)
yad --image="face-angry" \
    --title="Alerta" \
    --text "Você não selecionou uma personalidade!" \
    --button="Voltar e Selecionar"
show_list
;;
esac
}
show_list

#.EOF
```

Salve com o nome de list.sh, dê permissão de execução:

```
$ chmod +x list.sh
```

Execute:

```
$ ./list.sh
```



Após selecionar uma personalidade:



(//img.vivaolinux.com.br/imagens/artigos/comunidade/list02.png)

## EXEMPLO COM YAD --NOTIFICATION

yad --notification:

Mostra um ícone na área de notificação do sistema, podemos colocar um texto a ser mostrado quando o mouse estiver em cima dele, e uma ação para executar ao clicar.

Exemplo:

```
#!/bin/bash
# yad com --notification
# mostra uma notificação no painel, na área de notificação
# neste exemplo verifico a hora e uma tarefa associada a mesma
HORA=$(date +%H)
if [ "$HORA" -lt 06 ]; then
    FAIXA="1"
else
    if [ "$HORA" -lt 12 ]; then
        FAIXA="2"
    else
        if [ "$HORA" -lt 18 ]; then
            FAIXA="3"
        else
            if [ "$HORA" -le 24 ]; then
                FAIXA="4"
            fi
        fi
    fi
fi
case "$FAIXA" in
    1)
        MSG="Bom sonhos...."
        ;;
    2)
        MSG="Bom dia \o/"
```

```

;;
3)
MSG="Boa tarde"
;;
4)
MSG="Aproveite o fim do dia"
;;
esac
yad --notification \
--image="appointment-soon" \
--text "$MSG - clique para atualizar o sistema" \
--command "gnome-terminal --command='sudo apt-get upgrade'"

#.EOF

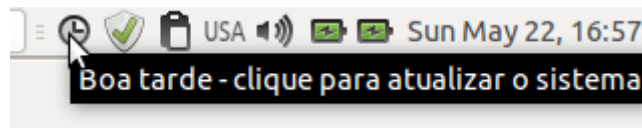
```

Salve com o nome de "notification.sh", dê permissão de execução:

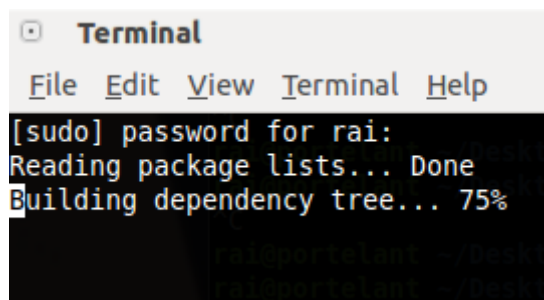
```
$ chmod +x notification.sh
```

Execute:

```
$ ./notification.sh
```



Após clicar, abre um *Gnome-terminal* e pede a senha pra depois atualizar:



## EXEMPLO COM YAD --PROGRESS

```
yad --progress:
```

Mostra um diálogo de progresso.

```
#!/bin/bash
# yad com --progress
# mostra uma barra de progresso para determinada ação, sendo que ler linha a linha
# dar entrada padrão
# neste exemplo busco arquivos .png na pasta /home do usuário

find $HOME -name '*.png' | \
yad --progress \
    --title "Progresso da busca" \
    --width="300" \
    --progress-text="Buscando imagens PNG" \
    --pulsate \
    --auto-close \
    --auto-kill # mata o processo anterior caso usuario clique em cancelar

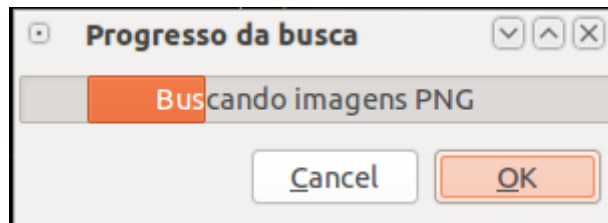
#.EOF
```

Salve com o nome de "progress.sh", dê permissão de execução:

```
$ chmod +x progress.sh
```

Execute:

```
$ ./progress.sh
```



## CONCLUSÃO

### EXEMPLO COM YAD --TEXT-INFO

yad --text-info:

Mostra diálogo com informações sobre texto.

Exemplo:

```
#!/bin/bash
# yad com --text-info
# exemplo para selecionar um arquivo e ver seu conteúdo (editá-lo se desejar)
ARQUIVO=$(
  yad --file \
    --filename="$HOME/" \
    --title="Selecione um arquivo" \
    --width="500" \
    --height="400" \
)
# testa se o usuário quer cancelou
ACAO="$?"
test "$ACAO" -eq "1" || test "$ACAO" -eq "252"
if [ "$?" -eq 0 ]; then
  exit
fi

# filename - nome do arquivo, editable - permite edição, fore - cor do texto
# back - cor do fundo, fontname - fonte a ser usada, justify - alinhamento do texto
# margins - tamanho da margem, tail - barra de rolagem
MOD_ARQUIVO=$( \
  yad --text-info \
    --title="Visualizando o arquivo $ARQUIVO" \
    --filename="$ARQUIVO" \
    --editable \
    --fore="#00FF00" \
    --back="#000000" \
    --fontname="Ubuntu" \
    --justify="left" \
    --margins="2" \
    --tail \
    --width="500" \
    --height="400" \
    --button="Cancelar":1 \
    --button="Salvar":0 \
)
# testa se o usuário quer cancelou
ACAO="$?"
test "$ACAO" -eq "1" || test "$ACAO" -eq "252"
if [ "$?" -eq 0 ]; then
  exit
fi

# salva as modificações no arquivo
echo $MOD_ARQUIVO \n > $ARQUIVO
```

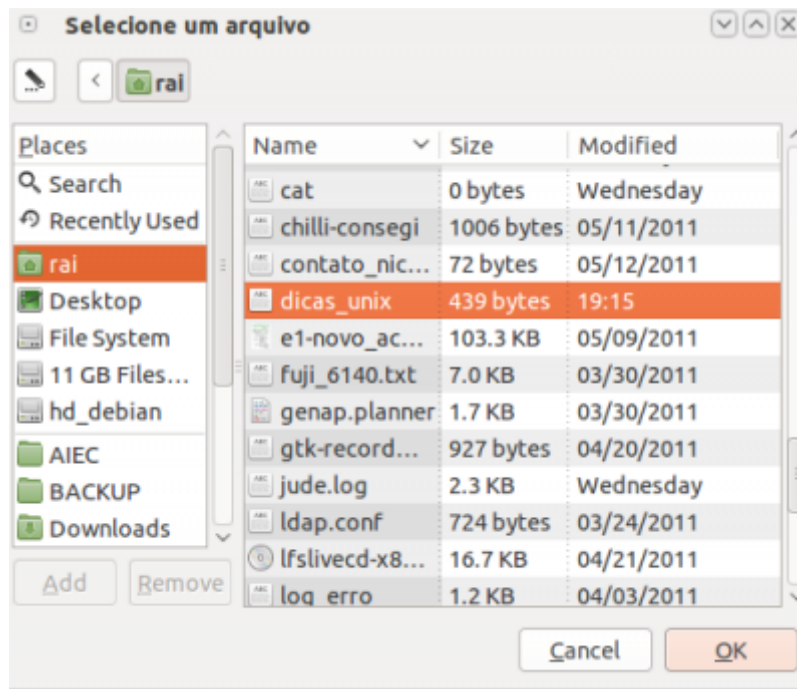
```
#.EOF
```

Salve com o nome de "text\_info.sh", dê permissão de execução:

```
$ chmod +x text_info.sh
```

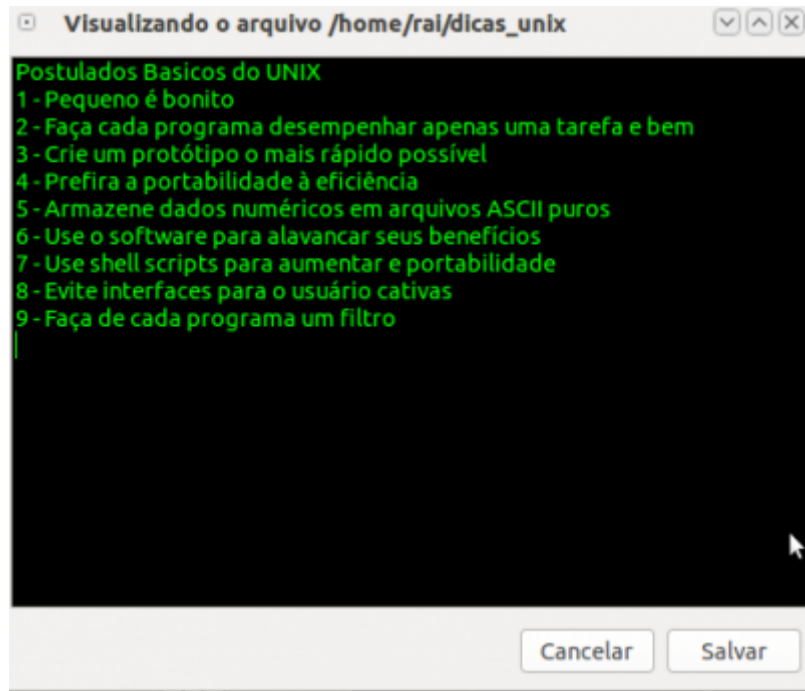
Execute:

```
$ ./text_info.sh
```



(//img.vivaolinux.com.br/imagens/artigos/comunidade/text\_info01.png)

Após selecionar o arquivo:




(//img.vivaolinux.com.br/imagens/artigos/comunidade/text\_info02.png)

## EXEMPLO COM YAD --SCALE

yad --scale:

Mostra um diálogo de escala.

```
#!/bin/bash
# yad com --scale
# neste exemplo uso o scale para permitir escolher o tamanho da imagem numa pagina html
#armazena o valor escolhido na escala
VALOR=$(
  yad --scale \
    --title="Escolha o tamanho da imagem" \
    --value="100" \
    --min-value="100" \
    --max-value="500" \
    --step="50" \
)

#grava o conteúdo a ser gravado em uma pagina html com o tamanho da imagem escolhido
CONTEUDO="Vux 
Todos os direitos reservados ao VOL :$

echo $CONTEUDO > scale.htm

#abre o arquivo no firefox
firefox scale.htm
```

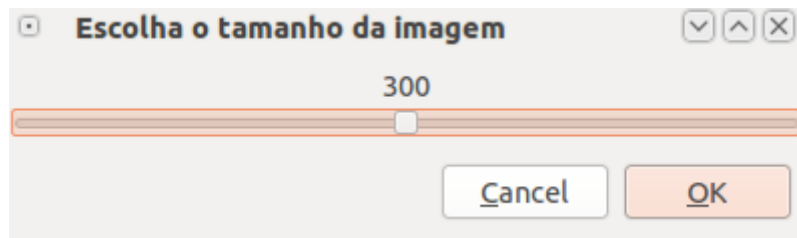
```
#.EOF
```

Salve com o nome de "scale.sh", dê permissão de execução:

```
$ chmod +x scale.sh
```

Execute:

```
$ ./scale.sh
```

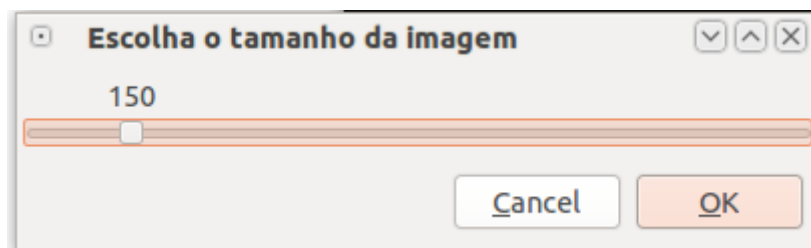


Após selecionar:



Vux  
Todos os direitos reservados ao VOL :-)

Mais um teste:



Após selecionar:





Vux  
Todos os direitos reservados ao VOL :-)

## CONCLUSÕES E AGRADECIMENTOS

Estou adorando conhecer o *Yad*, espero que a comunidade goste também, quanto mais pessoas apoiarem esse projeto, seja usando, seja divulgando, mais temos chances que ele se torne padrão futuramente nas distribuições mais usadas, como *Ubuntu*.

As possibilidades com o *Yad* são várias, depende apenas da criatividade, meus exemplos foram apenas demonstrativos. No site do projeto, temos alguns exemplos bem interessantes, inclusive um do brasileiro Julio Neves, que aliás foi quem tive o prazer de conversar rapidinho nos corredores do CONSEGI 2011, e durante a conversa o Julio me falou sobre o *Yad*, então fiquei curioso e essa curiosidade resultou nesse trabalho :-)

O Julio Neves apresentou uma palestra sobre o *Yad* no CONSEGI, não pude assistir pois estava ajudando no Install Fest pela manhã e à tarde ia trabalhar.

- ▶ *Yad* uma nova e inovadora interface gui - [consegi.gov.br \(http://www.consegi.gov.br/oficinas/yad-uma-nova-e-inovadora-interface-gui/oficina\\_view?iO=yad-uma-nova-e-inovadora-interface-gui\)](http://www.consegi.gov.br/oficinas/yad-uma-nova-e-inovadora-interface-gui/oficina_view?iO=yad-uma-nova-e-inovadora-interface-gui)

Obrigado Julio Neves e Parabéns Victor Ananjevsky, o *Yad* é muito maravilhoso!!!!



(//img.vivaolinux.com.br/imagens/artigos/comunidade/elinis\_julio\_rai.JPG)

[↩ Voltar \(verArtigo.php?codigo=12713\)](#)