

Utilização de shell script e interfaces gráficas para construção de software educacional livre

Roberto Franciscatto - PGIE/UFRGS - roberto.franciscatto@gmail.com
Liliana Maria Passerino - PGIE/UFRGS - liliana@cinted.ufrgs.br
Andressa Vergütz - UFSM/CAFW - andressavergutz@gmail.com
Kamile A. Wahlbrinck - UFSM/CAFW - kamilewahlbrinck@gmail.com
Vitor H. Lopes - UFSM/CAFW - vhugo_1@hotmail.com

Resumo. *O artigo apresenta uma visão geral sobre as principais características e funcionalidades da linguagem para criação de scripts automatizados no sistema operacional Linux, denominada shell script, bem como, as interfaces gráficas Zenity e Yade que permitem a criação de softwares educacionais com maior interação pelos usuários utilizando software livre. No decorrer do artigo são demonstrados exemplos de como se pode utilizar a integração de shell script e interfaces gráficas, através da implementação de softwares educacionais desenvolvidos e testados, bem como os resultados obtidos com a aplicação dos mesmos no ambiente escolar, servindo de suporte no processo de ensino-aprendizagem e os benefícios que estes softwares podem trazer aos alunos que os utilizam.*

Palavras-chaves: *software educacional, shell script, interfaces gráficas, linux.*

Abstract. *The article presents an overview of the main features and functionality of the language to create automated scripts on the Linux operating system, called shell script as well, and the Yade Zenity graphical interfaces that enable the creation of educational software with greater interaction by users using free software. Throughout the article examples of how to use the script shell integration and graphical interfaces, through the implementation of educational software developed and tested, and the results obtained from the application of the same in school environments are presented, serving to support the process teaching-learning and the benefits that this software can bring to students who use them.*

Keywords: *educational software, shell scripting, graphical interfaces, linux.*

1. Introdução

Há algum tempo o sistema operacional Linux tem feito parte do contexto educacional através de diferentes distribuições e aplicativos que contemplam não só propagar o uso de tais ferramentas, como oferecer uma gama de recursos que pode ser utilizada de forma educacional tanto para professores, quanto alunos, das mais diversas faixas etárias e níveis de ensino. Neste contexto abordamos neste trabalho o uso de shell script e seus diversos recursos que podem ser aproveitados como suporte para a construção de softwares educacionais livres.

Para que se possa entender como esta construção é possível, precisamos inicialmente entender o conceito de shell e script separadamente. Shell é o responsável

pela interação entre usuário e o kernel em um sistema operacional (entende-se kernel como o núcleo de um sistema operacional Linux). Ele recebe os comandos digitados pelos usuários e os executa. O shell é o “terminal” no Linux ou o “prompt de comandos” no Windows. Já o script é um arquivo que armazena uma sequência de comandos que são executados linha por linha pelo shell (AMORIM, 2006).

A combinação da escrita de scripts e sua execução pelo shell é o princípio para o desenvolvimento de inúmeras aplicações que podem ser construídas e executadas no sistema operacional Linux, abrindo desta forma, várias possibilidades para a criação de software educacional que pode ser aplicado a diferentes contextos.

Este artigo tem por objetivo descrever o uso de shell script na construção de aplicativos juntamente com as interfaces gráficas Zenity e Yad para a implementação de software educacional. Este trabalho tem por diferencial explorar as possibilidades e recursos que o shell script e as interfaces gráficas de desenvolvimento para Linux propiciam para o desenvolvimento dos mais variados tipos de softwares educacionais, uma vez que a grande maioria dos trabalhos encontrados na literatura exploram o uso de aplicativos livres prontos ou que já fazem parte do pacote dos sistemas operacionais Linux.

O presente trabalho está organizado da seguinte forma: na seção 2 é descrito o que é shell script e como ele funciona, bem como, as ferramentas gráficas para interação em shell script, Zenity e Yad. Na seção 3 são apresentados os artigos relacionados ao tema proposto. Na seção 4 é descrito como foi realizado o processo de planejamento das atividades e desenvolvimento das soluções educacionais livres, com ênfase para o estudo de caso do software “Quiz Educacional”. Na seção 5 são demonstrados os resultados obtidos pelo uso da aplicação e por fim, na seção 6 as conclusões finais.

2. Entendendo o Shell Script

Quando pensamos na automação de tarefas em ambientes Linux logo remetemo-nos ao shell script. Este foi desenvolvido e disponibilizado como recurso para construção de instruções, com uma grande quantidade de possibilidades. Chamamos de shell script um arquivo contendo comandos do terminal sob a forma de um programa executável. Para execução desses scripts é necessário um interpretador de comandos, que são programas feitos para intermediar o usuário e seu sistema. Através destes interpretadores, o usuário digita um comando e o interpretador o executa no sistema. O *bash* é um exemplo dos interpretadores existentes e geralmente o mais usado por ser o shell padrão na maioria das distribuições Linux (GOMES, 2013).

Considerada uma linguagem de fácil aprendizado, o shell script possui inúmeros recursos para automatizar uma série de tarefas do sistema operacional, permitindo integrar tarefas complexas com o *cron* (recurso dos sistemas Linux que permite o agendamento da execução de determinados comandos.), operações com múltiplos arquivos e diretórios, bem como a construção de aplicações para diferentes fins (redes de computadores, backup, segurança, software educacional, entre outros) (HAX, 2010).

Uma implementação de um programa em shell script, se dá, pela escrita de códigos entendíveis pelo interpretador de comandos, que nada mais é do que uma sequência lógica e estruturada para a resolução de um determinado problema. O código (sequência de comandos) deve ser escrito em um editor de texto e salvo. Para executá-lo é necessário ter permissão de execução. Normalmente uma aplicação criada em shell script é bastante utilizada para instalações/configurações, geração de relatórios, entre

outros fins, conforme descrito no parágrafo anterior (CISNEIROS, 2004), (VIRGILIO, 2007).

Um programa escrito em shell script, em sua forma primitiva contempla somente a interação modo texto. Como forma de criar mais opções na qual o usuário possa manipular e interagir com o sistema, foram criadas interfaces gráficas, dando ainda mais opções à criação de aplicativos profissionais com a utilização de shell scripts. Entre os mais usuais estão o Zenity e o YAD. O Zenity utilizado para criar a parte gráfica de aplicações feitas em scripts possui recursos conhecidos dos usuários de sistemas multi tarefas como, por exemplo, o uso de botões, menus, caixas de diálogos, etc. Já o YAD é ainda mais robusto que o Zenity e conta com recursos que o Zenity não contempla como a manipulação de imagens, por exemplo (HAX, 2010).

2.1 Zenity e YAD como solução gráfica ao Shell Script

O Zenity, conforme descrito anteriormente é a ferramenta para construção de interfaces gráficas junto ao shell script. O Zenity foi criado devido aos usuários não estarem acostumados com a interface modo texto exibida pelo terminal. Os usuários em geral necessitam das caixas de diálogo com os botões instruindo o que fazer. O Zenity exibe as janelas usando GTK+ (GTK: é uma biblioteca do Gnome que armazena os componentes da interface gráfica) a partir da linha de comando em shell script (CARTWRIGHT, 2009).

O Zenity oferece uma série de componentes gráficos como, barras de progresso, janelas de erros e mensagens, porém, não oferece inclusão de imagens e é limitado quanto a tarefas mais avançadas necessárias a ambientes gráficos. Como solução a tal problema tem-se o YAD. O YAD é um conjunto de recursos de software que traz diversas melhorias, além da possibilidade de inserção de imagens, também possibilita o uso de formulários ajudando na interação com o usuário. O YAD, assim como o Zenity, são ferramentas para aplicar interface gráfica ao shell script de maneira mais aperfeiçoada, permitindo diálogos adicionais, botões customizados, entre outros e utiliza os recursos do GTK 2 em sua implementação (PORTELA, 2011).

3. Trabalhos relacionados

Por tratar-se de ferramentas com inúmeras opções, como é o caso do shell script e as interfaces gráficas associadas, diferentes soluções são desenvolvidas para esta plataforma, conforme alguns trabalhos relacionados abaixo.

No trabalho de (SOUZA, 2003), a ideia foi o desenvolvimento de um servidor de backup inteligente utilizando a linguagem shell script, onde através do compartilhamento dos computadores da rede, uma aplicação realizava os backups de todas as máquinas na rede (diretórios compartilhados) de forma centralizada e com controle de versionamento, permitindo desta forma um controle e gerenciamento de backups em discos rígidos internos e externos, como forma de prevenção e plano de continuidade dentro da organização.

Outros trabalhos, com cunho educacional, também são propostos através da implementação utilizando shell script, como é o caso do trabalho de (PEREIRA, 2013) que descreve o desenvolvimento de uma ferramenta de apoio didático para a disciplina de administração em redes de computadores. Neste trabalho foi desenvolvida uma ferramenta que visava facilitar a tarefa dos docentes na área da tecnologia da

informação e o aprendizado dos alunos. A ferramenta prevê o gerenciamento de redes de computadores para verificar o estado da mesma, como o *status* de um link, verificação de rotas, *ping*, através de uma interface simples, porém de fácil uso e entendimento.

O trabalho de (BRUCKSCHEN, 2007) relata o desenvolvimento de software educacional proporcionando acessibilidade aos alunos com necessidades educacionais especiais. Este trabalho apresenta um relato de desenvolvimento de software educacional acessível e livre, voltado para inclusão de pessoas com deficiências visuais. Para tal desenvolvimento, houve a interação de uma equipe multidisciplinar envolvendo inclusive os alunos e seus professores, além de profissionais da Pedagogia e Informática.

4. Planejamento das atividades e desenvolvimento das soluções educacionais livres

A ideia deste trabalho foi despertar nos alunos as potencialidades de desenvolvimento de ferramentas educacionais utilizando o poder da linguagem de criação de scripts (shell script) bem como, capacitá-los a criar suas próprias soluções, uma vez que uma linguagem de programação permite uma infinidade de recursos de desenvolvimento de aplicações.

Durante a disciplina denominada “Administração de Sistemas Linux” ministrada junto ao Curso Superior de Tecnologia em Sistemas para Internet, os alunos tiveram 15 aulas sobre os princípios e práticas para o desenvolvimento e criação de softwares que utilizam shell script com o uso de interfaces gráficas (Zenity e YAD) permitindo a criação de softwares diversos, com ênfase para softwares educacionais que pudessem ser produzidos para utilizações futuras por alunos de diferentes níveis de ensino.

Dentre os projetos finais que foram estipulados para desenvolvimento pelos alunos, os protótipos em questão deveriam ser desenvolvidos com o objetivo de sua utilização como uma ferramenta de auxílio à aprendizagem em sala de aula de forma simples, dinâmica e divertida despertando no aluno vontade de utilizá-lo para testar seu nível de conhecimento sobre determinado assunto.

Os projetos desenvolvidos foram simulados, *quiz*, jogos de associações, ferramentas de automação de tarefas, entre outros. Como exemplo, para este artigo, faremos uma breve abordagem de um dos materiais desenvolvidos para o fim educacional, que consistia de um “Quiz Educacional” sobre o sistema operacional Linux. O projeto “Quiz Educacional” foi desenvolvido para que pudesse ser utilizado de forma simples e prática, podendo alterar suas perguntas e personalizá-lo de acordo com o conteúdo estudado.

4.1 Um exemplo de software produzido – Quiz Educacional

Abordaremos nesta seção o software “Quiz Educacional”, como exemplo de material produzido, sua implementação e forma de funcionamento. A ideia neste contexto é mostrar o processo de criação de softwares educacionais livres, através dos recursos de programação existentes no Linux (shell script, Zenity e YAD) e os resultados finais obtidos.

O aplicativo “Quiz Educacional”, foi desenvolvido utilizando a linguagem shell script juntamente com as ferramentas Zenity e Yad. A maior parte do código foi escrita

em shell script e as ferramentas Zenity e Yad foram utilizadas apenas para a parte gráfica, ou seja, para criar as telas do jogo, o que facilita a interação do usuário com o protótipo. Uma das maiores dificuldades relatadas no desenvolvimento do aplicativo foi fazer a concatenação dos dados das perguntas com o *quiz* para comparar se a resposta do usuário era correta, já que as mesmas estavam em arquivos separados para melhor praticidade do software. Um exemplo de código utilizado neste protótipo é mostrado na figura 1.

```
quiz.sh
1 #!/bin/bash
2 #abertura do jogo
3 yad --image='quiz.jpg' --title='Quiz-Linux!' --button='Jogar':"0" --button='Sair':"1" --window-icon='vux.png'
4 #se ele apertou o botão jogar entao faca
5 if [ $? == 0 ]
6 then
7 #guarda o nome da pessoa que for jogar
8 apelido=$(
9 yad --entry --text='Bem vindo ao Quiz-Linux! Para começar, entre com um apelido:' --entry-label='Apelido' --entry-text='Ex: Tux' --completion --edit
10 #chama as perguntas de 1 a 30
11 for ((i = 1; i <=2; i++))
12 {
13 #diz qual e o arquivo que ele deve acessar
14 ARQUIVO=$i.pergunta
15 #guarda a resposta correta
16 RESP=$(egrep '^R.*' "$ARQUIVO" | cut -d'|' -f2)
17 #pega a pergunta
18 PERGUNTA=$(egrep '^P.*' "$ARQUIVO" | cut -d'|' -f2)
19 OP1=$(egrep '^1.*' "$ARQUIVO")
20 OP2=$(egrep '^2.*' "$ARQUIVO")
21 OP3=$(egrep '^3.*' "$ARQUIVO")
22 OP4=$(egrep '^4.*' "$ARQUIVO")
23 OP5=$(egrep '^5.*' "$ARQUIVO")
24 #guarda a resposta do usuario
25 q1=${
26 zenity --list --text="$apelido esta é a pergunta: $PERGUNTA" --radiolist --column="" --column="" FALSE "$OP1" FALSE "$OP2" FALSE "$OP3" FALSE "$OP4" FALSE
27 if [ $? == 0 ]
28 then
29 case $q1 in
30 $RESP)
31 yad --text='Você acertou! Parabéns!' --title='Parabéns!' --button='OK':"0" --image='tux-ninja.png' --image-on-top
32 cont=$(( $cont + 1 ))
```

Figura 1 – Exemplo de código shell script utilizado no software “Quiz Educacional”

Através da figura 1, é possível observar o código principal do jogo e perceber que a linguagem shell script e as linguagens próprias para criação das interfaces em Zenity e YAD são combinadas para que uma possa dar suporte à outra, ou seja, uma produz (renderiza) a interface e a outra é responsável pela parte lógica do jogo que é executada cada vez que o usuário clica ou aciona uma ação neste aplicativo. Na figura 2, é mostrada a tela inicial do “Quiz Educacional” em execução.



Figura 2 – Tela inicial do Jogo e Apelido do usuário

Para dar início ao “Quiz Educacional” é preciso optar pelo botão “Jogar” que irá executar a tela do “login”, onde o usuário deve escolher um apelido para usar durante o jogo. Após a escolha do apelido iniciará o jogo com uma pergunta e as opções para resposta (Figura 3). O usuário deve selecionar uma delas e prosseguir o jogo pressionando o botão “OK”. A próxima tela irá informar ao usuário se a alternativa escolhida foi à correta ou não (Figura 3), caso esteja errada irá indicar a alternativa certa. Caso o usuário deseje finalizar o jogo sem ter terminado o mesmo, ele pode optar pela opção cancelar, assim o jogo será finalizado e uma tela de fim de jogo será mostrada. Quando o usuário terminar de responder todas as questões aparecerá uma tela

indicando que o jogo chegou ao fim e disponibilizando o placar do mesmo (Figura 3). Ao clicar em “Placar” aparece outra tela indicando quantas questões foram respondidas corretamente e quantos erros o usuário teve, disponibilizando um material de apoio sobre “Comandos Linux” e oferecendo a opção de jogar novamente.



Figura 3 – Perguntas do Quiz, Sinalização de Acerto pelo Software e Totalização de Pontos

Depois de finalizar o jogo o usuário poderá jogar quantas vezes quiser novamente. A vantagem deste software é o aprendizado que se ganha respondendo as questões do “Quiz Educacional”, que podem ser editadas, quanto à quantidade de questões a serem inseridas no programa, bem como, o tema a ser ministrado. Isto garante que o jogo pode ser aplicado a qualquer tema que se deseja estudar ou avaliar, permitindo ao aluno revisar conteúdos já ministrados, testar suas habilidades e estudar de uma forma que esteja jogando (ou sendo desafiado) ao mesmo tempo. Ao professor tem-se a possibilidade de criar diferentes tipos de materiais, para conteúdos específicos e propor aos alunos atividades não presenciais que possam permitir a eles aprender enquanto estão em horários ociosos e acompanhar a sua evolução através da pontuação obtida, relatórios ou ainda desafiando-os a criarem seus próprios jogos no futuro.

5. Análise e Resultados Encontrados

Quanto à análise e validação dos trabalhos desenvolvidos, escolheu-se o “Quiz Educacional”, como forma de testar sua utilidade e demais critérios estipulados (conforme Tabela 1), junto a uma turma do Curso Técnico em Informática, da instituição ABC (nome fictício), na disciplina de Sistemas Operacionais. Um total de aproximadamente 40 alunos, distribuídos no laboratório de informática, utilizaram a ferramenta “Quiz Educacional”, com 20 questões sobre o tema da aula em questão. Após utilizarem a ferramenta em sua totalidade os mesmos responderam a um questionário on-line, contendo as seguintes perguntas, conforme Tabela 1:

Tabela 1. Perguntas de avaliação sobre a ferramenta

1 - Quanto à interface da aplicação, o que você achou?
2 - Quanto à facilidade em utilizar a aplicação, o que você achou?
3 - Com relação ao tempo de resposta da aplicação, o que você achou?
4- Quanto às cores, menu e distribuição da aplicação na tela, o que você achou?
5 - Você acha que aplicações como esta podem lhe ajudar a entender melhor um conteúdo ministrado em sala de aula?

6 – Você utiliza o Sistema Operacional Linux em seu dia-a-dia?

Para todas as perguntas da tabela 1, havia uma escala de 1 a 4 (sendo “1” para ruim, “2” para regular, “3” para bom e “4” para ótimo). Esta escala tinha por objetivo medir a satisfação do usuário relativo a itens específicos da aplicação. Os percentuais obtidos nas perguntas acima são apresentados na Tabela 2.

Tabela 2. Respostas obtidas sobre a aplicação em Shell Script

1 - Quanto à interface da aplicação, o que você achou?	1 = 10 %	2 = 20%	3 = 50%	4 = 20%
2 - Quanto à facilidade em utilizar a aplicação, o que você achou?	1 = 0 %	2 = 25%	3 = 45%	4 = 30%
3 - Com relação ao tempo de resposta da aplicação, o que você achou?	1 = 10 %	2 = 30%	3 = 35%	4 = 25%
4- Quanto às cores, menu e distribuição da aplicação na tela, o que você achou?	1 = 0 %	2 = 25%	3 = 45%	4 = 30%
5 - Você acha que aplicações como esta podem lhe ajudar a entender melhor um conteúdo ministrado em sala de aula?	Sim: 95%		Não: 5%	
6 – Você utiliza o Sistema Operacional Linux em seu dia-a-dia?	Não, nunca utilizei: 15%	Já utilizei algumas vezes: 80 %	Utilizo o Windows e o Linux em meu computador: 5%	Utilizo somente o Linux em meu computador: 0%

Ao analisarmos as respostas obtidas pelos alunos, relativo à pergunta 01, cerca de 70% aprovaram a interface do software o que demonstra que os recursos gráficos utilizados na aplicação foram satisfatórias para seus utilizadores. Quanto à facilidade de uso da aplicação cerca de 75% acharam de fácil utilização uma vez que o software desenvolvido tinha um fim específico. No fator tempo de resposta e usabilidade do mesmo, mais de 60% dos usuários aprovaram tais critérios. Um dado importante obtido com o questionário on-line refere-se a utilização de softwares educacionais no dia-a-dia dos alunos, cerca de 95% concordam que a utilização de tais softwares podem ajudar a entender melhor um conteúdo ministrado em sala de aula. Por fim, quanto a utilização de softwares livres no contexto diário dos alunos, mesmo se tratando de um curso técnico em informática, ainda são poucos os alunos que utilizam somente Linux ou Linux e Windows juntos em seu computador e/ou notebook pessoal, perfazendo um percentual de aproximadamente 5% dos alunos que participaram desta pesquisa, o que demonstra o grande potencial e importância de trabalhos voltados a tal fim.

6. Conclusões e Trabalhos Futuros

Através deste trabalho, foi possível observar a importância da utilização de tecnologias livres que complementem o conteúdo ministrado em sala de aula, seja através de jogos, simulados, *quiz*, entre outros softwares educacionais que venham a ser desenvolvidos para este propósito. Ainda, a correta utilização de ferramentas livres como o Linux, propiciam um ambiente repleto de recursos que podem ser utilizados para criar ferramentas dos mais diversos fins, com o intuito de tornar a aula mais dinâmica e atrativa para os alunos.

Entretanto, percebe-se que o ensino da programação e da utilização de softwares livres são fundamentais cada vez mais cedo na vida dos alunos para que possamos em um futuro próximo ter condições de desenvolver softwares educacionais de qualidade, que tenham sua eficácia comprovada e que possam ajudar cada vez mais no processo de ensino-aprendizagem.

Como trabalhos futuros, pretende-se desenvolver um repositório web, com um conjunto de soluções educacionais livres, que possa ser divulgada e utilizada por educadores, alunos e comunidade em geral, com o intuito destes proporem melhorias e contribuições.

Referências

- AMORIM, L. A. **Introdução ao uso de shell script**. Disponível em: <<http://www.vivaolinux.com.br/artigo/Introducao-ao-uso-de-Shell-Script>>. Acesso em: 18 de Setembro de 2013.
- BRUCKSCHEN, M., RIGO, S. J., FAGUNDES, É. **Desenvolvimento de software educacional livre e inclusão de alunos com deficiência visual**. Ciclo de Palestras sobre Novas Tecnologias na Educação. CINTED/UFRGS. V. 5 N° 2, Dezembro, 2007.
- CARTWRIGHT, R. **Zenity: Interface gráfica para scripts, parte 1**. Disponível em: <<http://www.hardware.com.br/artigos/zenity/>>. Acesso em: 23 de setembro de 2013.
- CISNEIROS, H. **Programando em Shell Script**. Disponível em: <http://www.devin.com.br/shell_script/>. Acesso em: 09 de Outubro de 2013.
- GOMES, R. F. **Introdução ao Shell Script no Linux**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-shell-script-no-linux/25778>>. Acesso em: 05 de Outubro de 2013.
- HAX, V. **Introdução à Shell Script**. Disponível em: <<http://www.slideshare.net/viniushax/introduco-shellscript>>. Acesso em: 30 de Setembro de 2013.
- PEREIRA, F.A., SILVA, J. M.; COSTA, A. P. **Desenvolvimento de uma Ferramenta de Apoio Didático para Disciplina de Administração em Redes de Computadores**. Disponível em: <<http://www.fatecbauru.edu.br/ojs/index.php/CET/article/download/58/53>>. Acesso em: 19 de Outubro de 2013.

PORTELA, R. A. **Apresentando o YAD.** Disponível em: <<http://www.vivaolinux.com.br/artigo/Apresentando-o-Yad-zenity-melhorado/>>. Acesso em: 19 de Agosto de 2013.

SOUZA, F. R. N. O.; FARIA, M. N.; DIAS, S. T.; VIANNA, M. B. M. **Desenvolvimento de um Servidor de Backup Inteligente Utilizando a Linguagem Shell Script em Linux.** Disponível em: <<http://www3.iesampa.edu.br/ojs/index.php/computacao/article/view/238/229>>. Acesso em 27 de novembro de 2013.

VIRGILIO, J. **O que é Shell Script?.** Disponível em: <<http://www.vivaolinux.com.br/artigo/O-que-e-Shell-Script>>. Acesso em 03 de Agosto de 2013.