

GABARITO

Projete uma aplicação em Lazarus para um operador de uma Lan House controlar o tempo e o crédito de cada usuário. Nesta Lan House, cada minuto de uso de um computador custa R\$ 0,10. Esta possui 3 computadores: C1 , C2 e C3 . Quando o usuário chega, ele faz um depósito e ganha um crédito em minutos no valor do depósito. Por exemplo, se ele faz um depósito de R\$ 3,00 ele ganha um crédito de 30 minutos para usar um computador.

- Formulário Principal

Neste formulário, o operador deve entrar com o nome do usuário, a máquina que este vai usar, o valor em Reais que o usuário vai colocar de depósito e a hora (hh:mm).

Além deste campos, o formulário deverá ter mais 2 campos (que não serão preenchidos pelo operador) onde será calculado o valor do crédito, em minutos, que o usuário possui e um outro campo com a hora (hh:mm) que ele saiu do computador.

Este formulário tem 3 botões: um para limpar os dados dos campos, outro para fechar a aplicação e outro para abrir o novo formulário para controle do usuário que está começando.

- Formulário Secundário

Cada vez que um usuário é alocado a um computador, é aberto um novo formulário. Neste formulário, o nome do usuário, o valor depósito, o computador que ele está usando, o valor calculado do crédito, em minutos, e a hora de entrada (hh:mm) devem ser trazidos do formulário principal.

Este formulário deve ter um campo para o operador entrar com a hora (hh:mm) que o usuário terminou seu uso. Caso o operador entre com uma hora inferior a hora de entrada uma mensagem de erro deve ser dada.

Este formulário deve ter um único botão para encerrar a seção do usuário e fechar o formulário. Neste fechamento, valores atualizados de depósito/dívida, crédito/debito em minutos e hora de encerramento devem ser trazidos para o formulário principal, além do nome e computador do usuário.

1. Faça o projeto desta aplicação e implemente em Lazarus.
2. O código fonte deve executar no Lazarus para ambiente Linux.

Solução

Para criar uma aplicação com as especificações solicitadas o Formulário Principal, Figura 1, deve conter os seguintes campos editáveis (com o uso do componente TEdit):

- Nome do usuário: Para identificar o cliente que está utilizando o computador.
- Valor em reais: valor que será convertido em créditos. Essa conversão pode ocorrer, por exemplo, ao pressionar enter nesse campo, configurando o evento "OnKeyPress".
- Hora: Hora inicial de utilização.

Como só existem 3 máquinas que podem ser utilizadas e suas identificações são conhecidas, pode-se utilizar o componente TComboBox.

E dois campos não editáveis (propriedade "Enabled" igual a False):

- Hora de saída: esse valor é preenchido no formulário secundário.
- Créditos: créditos em minutos do usuário. Esse valor é alterado quando algum valor é adicionado no campo "Valor em reais" ou após fechar o formulário secundário.

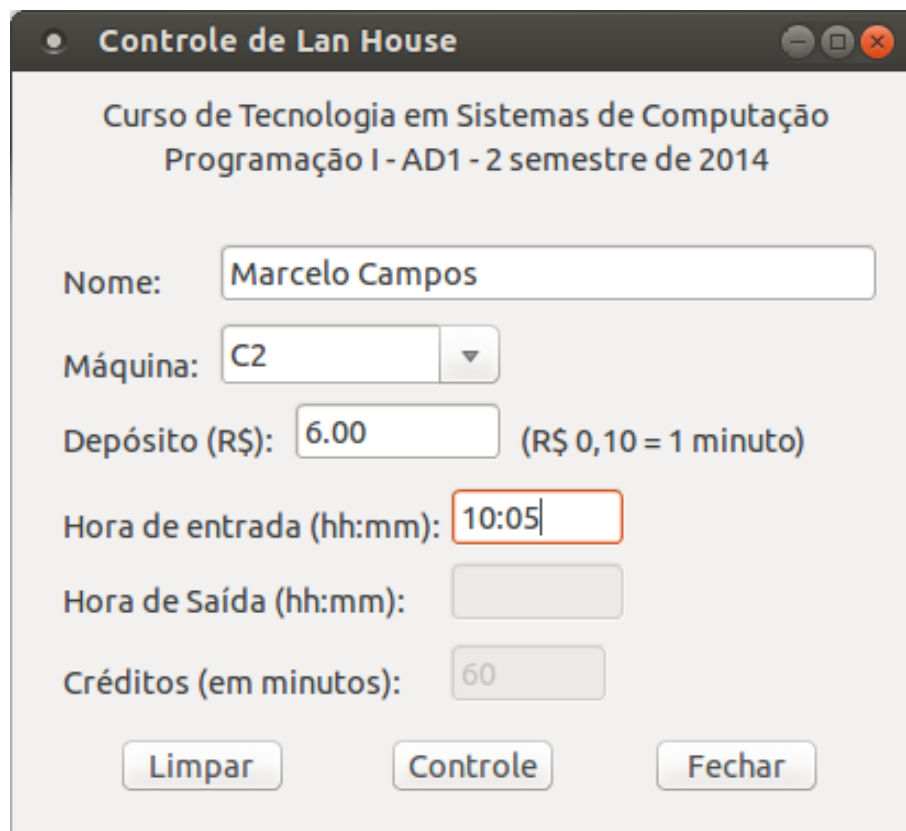


Figure 1: Formulário Principal.

Para abrir o formulário secundário é necessário acrescentar no Formulário Principal o seguinte código:

- Logo após o comando "implementation", acrescenta-se:

Uses

UnitSecundario ;

- e no botão "Controle", no evento "OnClick", o seguinte código:

FormSecundario . show ;

Além disso no botão "Controle" deve-se colocar o código para enviar as informações que serão utilizadas no formulário secundário, e todos os campos do formulário principal devem ser apagados para permitir que outros usuários possam ser alocados em outros computadores.

Figure 2: Formulário Secundário.

O Formulário Secundário, Figura 2, é formado por 5 campos não editáveis, propriedade Enable igual a False (TEdits com informações que vieram do formulário principal).

- Nome, Máquina, Depósito, Hora de entrada, Créditos.

E um campo editável para informar a hora de saída.

Após digitar a hora de saída e pressionar o botão Fechar, os valores de depósito/dívida e crédito/débito devem ser atualizados no formulário principal. Além disso, os outros campos devem ser copiados para seus respectivos locais.

Para que o Formulário Secundário possa enviar as informações no Formulário Principal, deve-se acrescentar, logo após o comando "implementation" o código abaixo.

Uses

```
UnitPrincipal;
```

Caso o operador entre com uma hora de saída inferior a hora de entrada, uma mensagem de erro deve ser dada. Para isso pode-se utilizar o seguinte comando:

```
MessageDlg( 'Hora_de_saída_inferior_a_hora_de_entrada.', mtCustom, [mbYes], 0);
```