

## Exercício Final – Caixa do Supermercado Devs2Blu

Crie uma tela semelhante a imagem abaixo para simularmos as operações padrões de um supermercado e para praticarmos alguns conceitos visto até o momento:



**Regra 1** – Após desenhar a tela vamos criar uma classe para fazer a abstração de um Caixa para nossa aplicação.

Essa classe poderá ter o nome de *TCaixa*. E vamos criar em uma unit separada do formulário. Essa classe terá os atributos *SaldoInicial* e *SaldoAtual* ambos do tipo *Double* e terá o atributo *CaixaAberto* do tipo *Boolean*.

Como métodos podemos deixar disponíveis para outras classes as operações *AbrirCaixa*, *RetirarValor*, *AdicionarValor* e *FecharCaixa*.

**Obs.:** Lembre-se de utilizar as boas práticas e convenções do Delphi. Assim como os conceitos da Programação Orientada a Objetos.

**Regra 2** – Após criar a classe é hora de usarmos no formulário principal. Dessa forma podemos criar um atributo no formulário para o tipo da classe criada acima. Usar um case com um tipo enumerado para desviarmos o fluxo da execução das opções para seus respectivas métodos, ou seja, para cada operação do *RadioGroup* vamos criar um método separado para executar a ação da classe *TCaixa* e para fazermos algumas validações mencionado abaixo:

### Abrir Caixa

- Caso o usuário informar algum valor inválido devemos exibir uma mensagem de erro.
- Caso o objeto *TCaixa* não estiver criado, devemos criá-lo.
- Caso o caixa já esteja aberto devemos alertar ao usuário.
- Exibir no label *Resultado* alguma mensagem de que o caixa foi aberto.

### **Adicionar Dinheiro**

- Caso o usuário informar algum valor inválido devemos exibir uma mensagem de erro
- Caso o caixa não estiver aberto devemos alertar ao usuário.
- Exibir no label Resultado alguma mensagem do valor que foi adicionado ao caixa.

### **Retirar Dinheiro**

- Caso o usuário informar algum valor inválido devemos exibir uma mensagem de erro.
- Caso o caixa não estiver aberto devemos alertar o usuário.
- Exibir no label Resultado alguma mensagem do valor que foi retirado do caixa.

### **Exibir Saldo Atual**

- Bloquear o edit e o label de Input de Valor
- Caso o caixa não estiver aberto devemos alertar o usuário.
- Exibir o Saldo no label Resultado.

### **Fechar Caixa**

- Bloquear o edit e o label de Input de Valor.
- Caso o caixa não estiver aberto devemos alertar o usuário.
- Exibir uma mensagem no label Resultado com o valor final do caixa.

**Obs.:** Lembre-se de liberar o atributo *FCaixa* da memória no momento certo para não deixar “sujeira” e fazer as devidas verificações para não causar *AV (Access Violation)* no sistema.

**Regra 3 (plus)** – Adicionar um memo para exibir as operações realizadas no caixa.

Aqui podemos usar a classe *TStringList* para armazenar todas as operações. E criarmos um método na classe caixa que “varre” essa *TStringList* e exiba todas as operações no memo.