

inicializar global FirstNumber como " "

inicializar global Operator como " "

cuando SixButton .Clic
ejecutar Llamar BuildNumber Digit " 6 "

cuando OneButton .Clic
ejecutar Llamar BuildNumber Digit " 1 "

cuando SevenButton .Clic
ejecutar Llamar BuildNumber Digit " 7 "

cuando TwoButton .Clic
ejecutar Llamar BuildNumber Digit " 2 "

cuando EightButton .Clic
ejecutar Llamar BuildNumber Digit " 8 "

cuando ThreeButton .Clic
ejecutar Llamar BuildNumber Digit " 3 "

```
cuando NineButton .Clic
ejecutar Llamar BuildNumber Digit " 9 "
```

```
cuando FourButton .Clic
ejecutar Llamar BuildNumber Digit " 4 "
```

```
cuando FourButton .Clic
ejecutar Llamar BuildNumber Digit " 4 "
```

```
cuando FiveButton .Clic
ejecutar Llamar BuildNumber Digit " 5 "
```

```
como BuildNumber Digit
ejecutar poner DisplayLabel . Texto como unir DisplayLabel . Texto
tomar Digit
```

```
cuando DecimalPointButton .Clic
ejecutar si no contiene texto DisplayLabel . Texto
cadena "."
entonces si ¿es un número? DisplayLabel . Texto
entonces poner DisplayLabel . Texto como unir DisplayLabel . Texto
" ."
sino poner DisplayLabel . Texto como " 0. "
```

```
cuando ClearErrorButton .Clic
ejecutar poner DisplayLabel . Texto como " "
```

```
cuando ChangeSignButton .Clic
ejecutar
  si
    no está vacío DisplayLabel . Texto
    entonces poner DisplayLabel . Texto como neg DisplayLabel . Texto
```

```
cuando ClearButton .Clic
ejecutar
  poner DisplayLabel . Texto como " "
  poner global FirstNumber a " "
  poner global Operator a " "
```

```
cuando BackArrowButton .Clic
ejecutar
  si
    no está vacío DisplayLabel . Texto
    entonces poner DisplayLabel . Texto como segmento de texto DisplayLabel . Texto
      inicio 1
      longitud longitud DisplayLabel . Texto - 1
```

```
cuando BackArrowButton .Clic
ejecutar
  si
    no está vacío DisplayLabel . Texto
    entonces poner DisplayLabel . Texto como segmento de texto DisplayLabel . Texto
      inicio 1
      longitud longitud DisplayLabel . Texto - 1
```

```
cuando AddButton .Clic
ejecutar
  Llamar StoreFirstNumberAndOperator Operator "+"
```

```
como StoreFirstNumberAndOperator Operator
ejecutar
  si
    ¿es un número? DisplayLabel . Texto
    entonces
      poner global FirstNumber a DisplayLabel . Texto
      poner global Operator a tomar Operator
      poner DisplayLabel . Texto como " "
    sino
      poner DisplayLabel . Texto como "number expected"
```

```
cuando SubtractButton .Clic
ejecutar
  Llamar StoreFirstNumberAndOperator Operator "-"
```

```
cuando MultiplyButton .Clic
ejecutar Lllamar StoreFirstNumberAndOperator Operator "x"
```

```
cuando EqualsButton .Clic
ejecutar
  si
    no está vacío tomar global Operator
  entonces
    poner DisplayLabel . Texto como Lllamar Calculate
      FirstNumber tomar global FirstNumber
      Operator tomar global Operator
      SecondNumber DisplayLabel . Texto
    poner global FirstNumber a " "
    poner global Operator a " "
  sino
    poner DisplayLabel . Texto como "operator expected"
```

```
cuando DivideButton .Clic
ejecutar Lllamar StoreFirstNumberAndOperator Operator "÷"
```

```
como Calculate FirstNumber Operator SecondNumber
resultado
  si
    comparar textos tomar Operator = "+"
  entonces
    tomar FirstNumber + tomar SecondNumber
  si no
    si
      comparar textos tomar Operator = "-"
    entonces
      tomar FirstNumber - tomar SecondNumber
    si no
      si
        comparar textos tomar Operator = "x"
      entonces
        tomar FirstNumber × tomar SecondNumber
      si no
        si
          comparar textos tomar Operator = "÷"
        entonces
          tomar FirstNumber / tomar SecondNumber
        si no
          "operator expected"
```