

# CALCULADORA

---

Simple, horizontal

---

## Concepto de práctica.

En esta práctica se pide la realización de una calculadora simple y en orientación horizontal.

El punto de partida de la práctica fue el de la realización de la interfaz. En un principio se desarrollaron dos interfaces con diseño diferente para la misma. El primero minimalista abstracto y el segundo minimalista con toques de skeumorfismo (el parecido al la de iOS).

Se optó seguir el desarrollo con el diseño clásico (o parecido al de iOS).

Una vez solucionado el tema de la interfaz pasamos a la realización de la funcionalidad.

Lo primero fue crear las @property para los botones y función de memoria para después crear los métodos necesarios para que la calculadora funcione.

Lo siguiente que hicimos fue la conexión de los elementos a sus respectivos métodos y la programación de los mismos.

## Código.

### viewController.h

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@property (weak, nonatomic) IBOutlet UILabel *label;
@property (weak, nonatomic) IBOutlet UIButton *button_ac;
@property (weak, nonatomic) IBOutlet UIButton *button_m;
@property (weak, nonatomic) IBOutlet UIButton *button_mr;

@property (weak, nonatomic) IBOutlet UIButton *button_coma;

//Property que nos sirve para la memoria
@property NSString *memoria;

- (IBAction)pulsa_num:(id) sender;
- (IBAction)operation:(id) sender;
- (IBAction)equal:(id) sender;
- (IBAction)pulsa_ac:(id) sender;

- (IBAction)pulsa_m:(id) sender;
- (IBAction)pulsa_mr:(id) sender;

@end
```

### viewController.m

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

//variable para ir almacenando los valores que se van introduciendo en
pantalla
NSString *primer_valor, *segundo_valor, *operacion;
NSString *concat1=@"", *concat2=@"";
```

```

double resultado;
bool clickOperacion = false, clickEqual = false;
float num1 = 0.0, num2 = 0.0;

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a
nib.

    //Iniciación de la memoria
    _memoria = @"";
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)pulsa_num:(id)sender
{
    NSString *texto = [sender currentTitle];
    [_label setText:texto];

    //Rellenamos el primer parametro
    if(!(clickOperacion) && !(clickEqual))
    {

        //recogemos el valor introducido
        primer_valor = texto;

        //concatenamos en un String los numeros introducidos
        concat1 = [NSString stringWithFormat:@"%@"
%@",concat1,primer_valor];

        //el String lo convertimos en float
        num1 = [concat1 floatValue];

        //si introducimos una coma se deshabilita este botón
        if ([primer_valor isEqualToString:@"."] )
        {
            [_button_coma setEnabled:NO];
        }

        //Imprimimos el valor en modo String en la calculadora
        [_label setText:concat1];
    }

    //Rellenamos el segundo valor (si clickOperacion es true)
    if(clickOperacion)
    {
        //recogemos el valor introducido
        segundo_valor = texto;

        //concatenamos en un String los numeros introducidos
        concat2 = [NSString stringWithFormat:@"%@"
%@",concat2,segundo_valor];

        //el String lo convertimos en float
        num2 = [concat2 floatValue];

        //si introducimos una coma se deshabilita este botón
    }
}

```

```

        if ([segundo_valor isEqualToString:@"."])
        {
            [_button_coma setEnabled:NO];
        }

        //Imprimimos el valor en modo String en la calculadora
        [_label setText:concat2];
    }

}

//Controlamos el tipo de operacion (+, -, /, *)
- (IBAction)operation:(id)sender
{
    NSString *texto = [sender currentTitle];

    //habilitamos el botón coma para poder introducirlo en el siguiente
    número
    [_button_coma setEnabled:YES];

    //controlamos la siguiente operación
    clickOperacion = true;

    if([texto isEqualToString:@"+"])
    {
        operacion = @"+";
    }else if([texto isEqualToString:@"-"])
    {
        operacion = @"-";
    }else if([texto isEqualToString:@"*"])
    {
        operacion = @"*";
    }else if([texto isEqualToString:@"/"])
    {
        operacion = @"/";
    }

}

//Si clickamos en el símbolo igual
- (IBAction)equal:(id)sender
{
    //Si solo se ha introducido la primera operación y no se ha
    introducido ningún símbolo
    if(operacion == nil)
    {
        [_label setText:concat1];
        primer_valor = concat1;
        concat1 = concat2 = segundo_valor = @"";
        clickOperacion = false;
    }

    //Según la operación elegida hacemos una suma, resta, multiplicación
    o división
    if([operacion isEqualToString:@"+"])
    {
        resultado = num1 + num2;
    }else if([operacion isEqualToString:@"-"]){
        resultado = num1 - num2;
    }else if([operacion isEqualToString:@"*"]){

```

```
        resultado = num1 * num2;
    }else if([operacion isEqualToString:@" / "]){
        resultado = num1 / num2;
    }

    if(operacion != nil)
    {
        //devolvemos el resultado
        [_label setText:[NSString stringWithFormat:@"%%.2f",resultado]];
        clickEqual = clickOperacion = false;
        num1 =resultado;
        concat1 = concat2 = segundo_valor = primer_valor =@" ";
        operacion = nil;
    }

    //Habilitamos la coma
    [_button_coma setEnabled:YES];
}

//Dejamos todo a 0
- (IBAction)pulsa_ac:(id)sender
{
    primer_valor = @" ";
    segundo_valor = @" ";
    concat1 = @" ";
    concat2 = @" ";
    [_label setText:@"0.0"];
    clickOperacion = false;
}

- (IBAction)pulsa_m:(id)sender //Implementa funcionalidad de M
{
    _memoria = [_label text];
    NSLog(@"%@", _memoria);
}

- (IBAction)pulsa_mr:(id)sender //Implementa funcionalidad MR
{
    [_label setText:_memoria];
}

@end
```