

Incorporando fontes personalizadas



Kaê Angeli Coutinho

Por que usa-las?

- Identidade
- Autenticidade
- Inovação
- Estética

Onde adquiri-las?

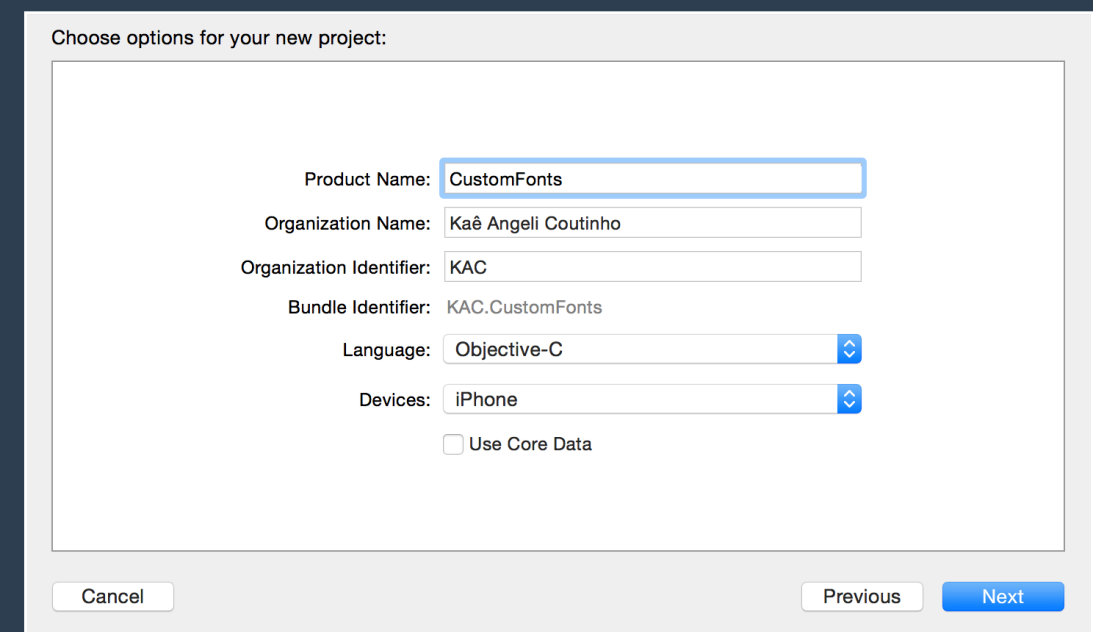
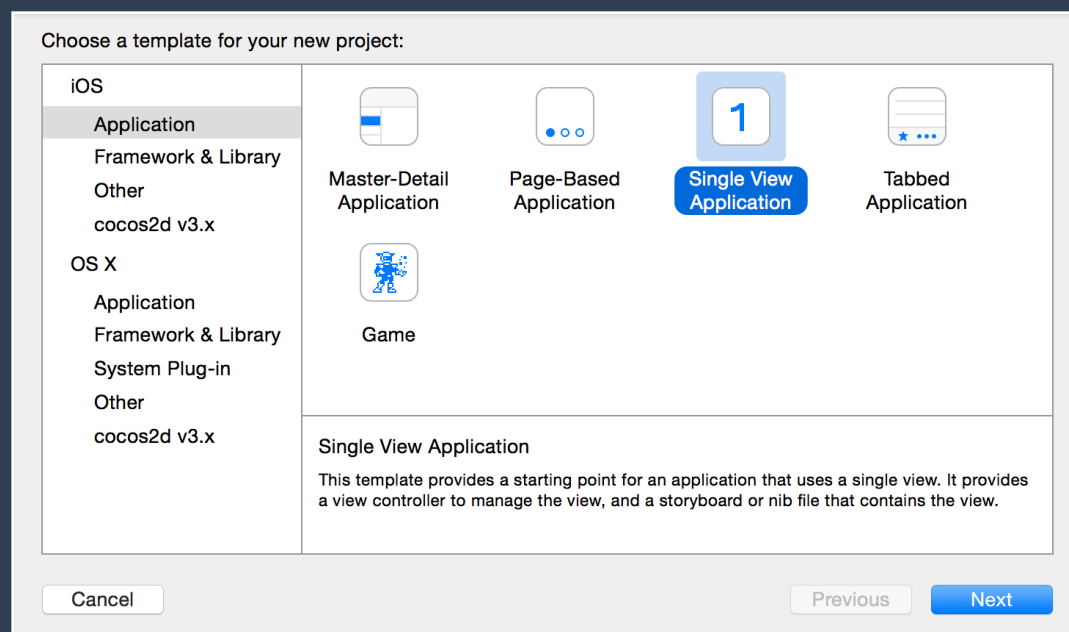
- Qualquer formato é suportado pela IDE Xcode
- Extensões mais comuns são OTF, TTF e FNT
- Diversos repositórios na web
 - www.dafont.com
 - www.1001freefonts.com
 - www.fontsquirrel.com
 - www.fontfabric.com
 - www.urbanfonts.com
- Atenção para o tipo de licença da fonte escolhida

Como incorpora-las?

- É necessário adiciona-las ao projeto assim como qualquer outra dependência
- Adiciona-las ao aplicativo alvo
- Incluir os arquivos no invólucro de compilação
- Modificar o property list para reconhecer as novas fontes
- Encontrar o nome da fonte que fora adicionada

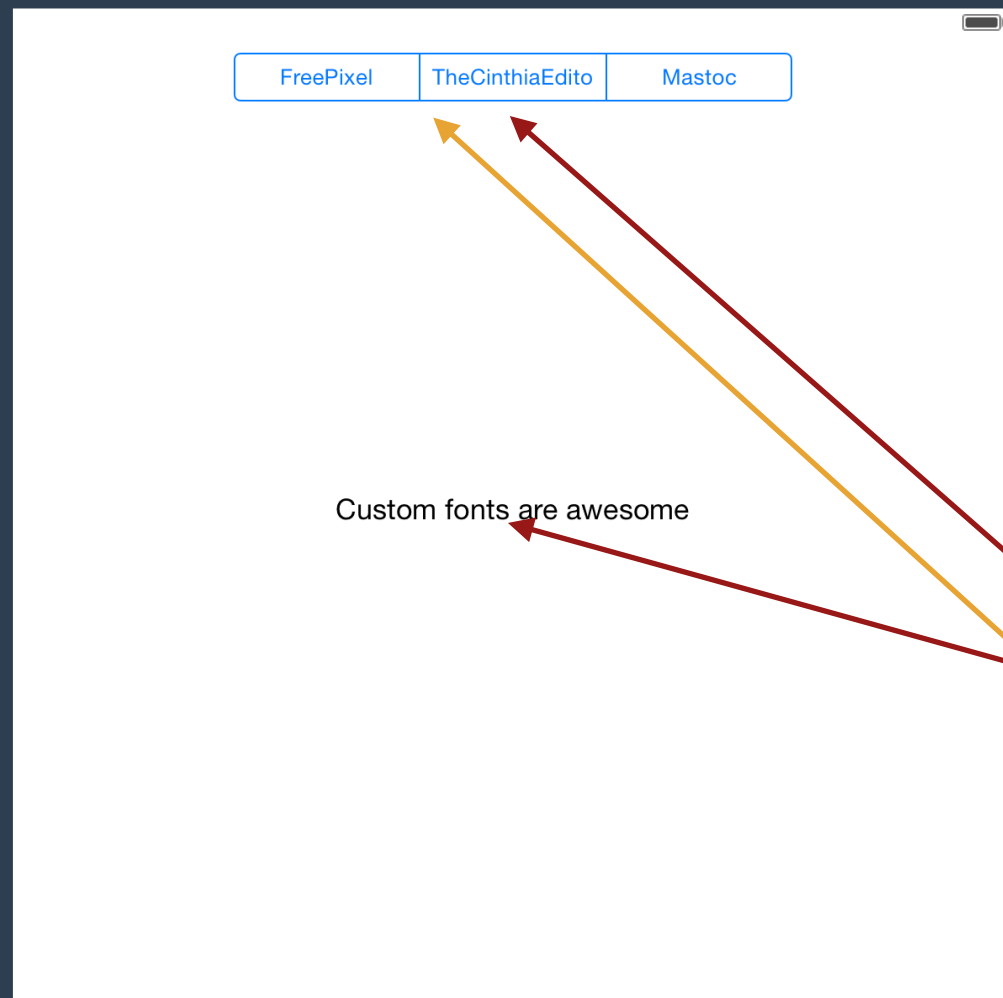
Mãos à obra

- Vamos começar um novo projeto no Xcode



Kaê Angeli Coutinho

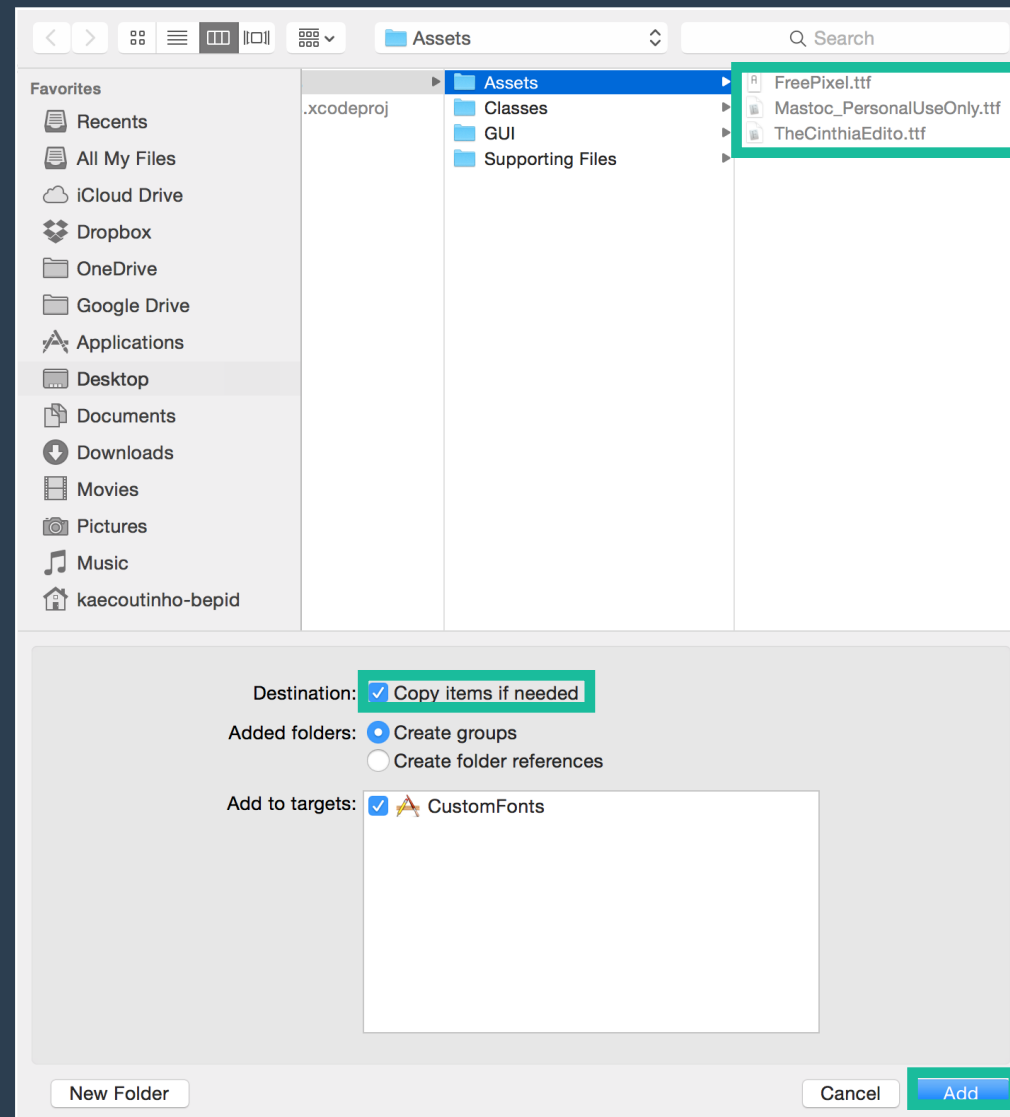
- Começaremos o desenvolvimento atrelando a interface gráfica com ViewController.h



```
1 //
2 // ViewController.h
3 // CustomFonts
4 //
5 // Created by Kaê Coutinho on 11/17/14.
6 // Copyright (c) 2014 Kaê Angeli Coutinho. All rights reserved.
7 //
8
9 #import <UIKit/UIKit.h>
10 #define FONT_FREEPIXEL @"FreePixel-Regular"
11 #define FONT_THECINTHIAEDITO @"TheCinthiaEdito"
12 #define FONT_MASTOC @"MastocPersonalUseOnly"
13 #define FONT_SIZE 28
14
15 @interface ViewController : UIViewController
16
17 @property (weak, nonatomic) IBOutlet UISegmentedControl * fontsSegmentedControl;
18 @property (weak, nonatomic) IBOutlet UILabel * fontLabel;
19
20 -(IBAction)changeFont:(UISegmentedControl *)sender;
21
22 @end
```

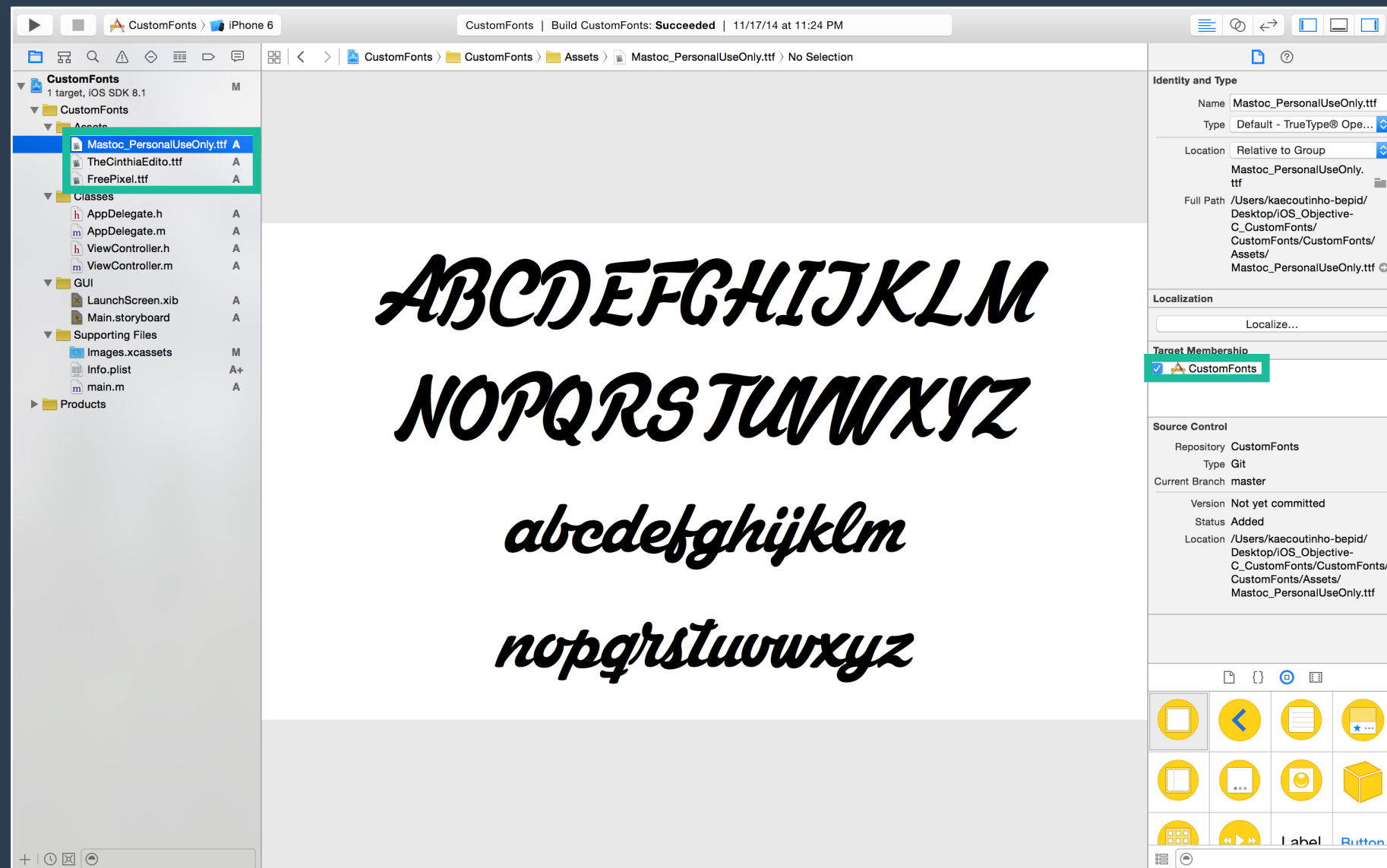
● IBOutlet
● IBAction

- Vamos agora para o processo de adicionar as fontes personalizadas no projeto

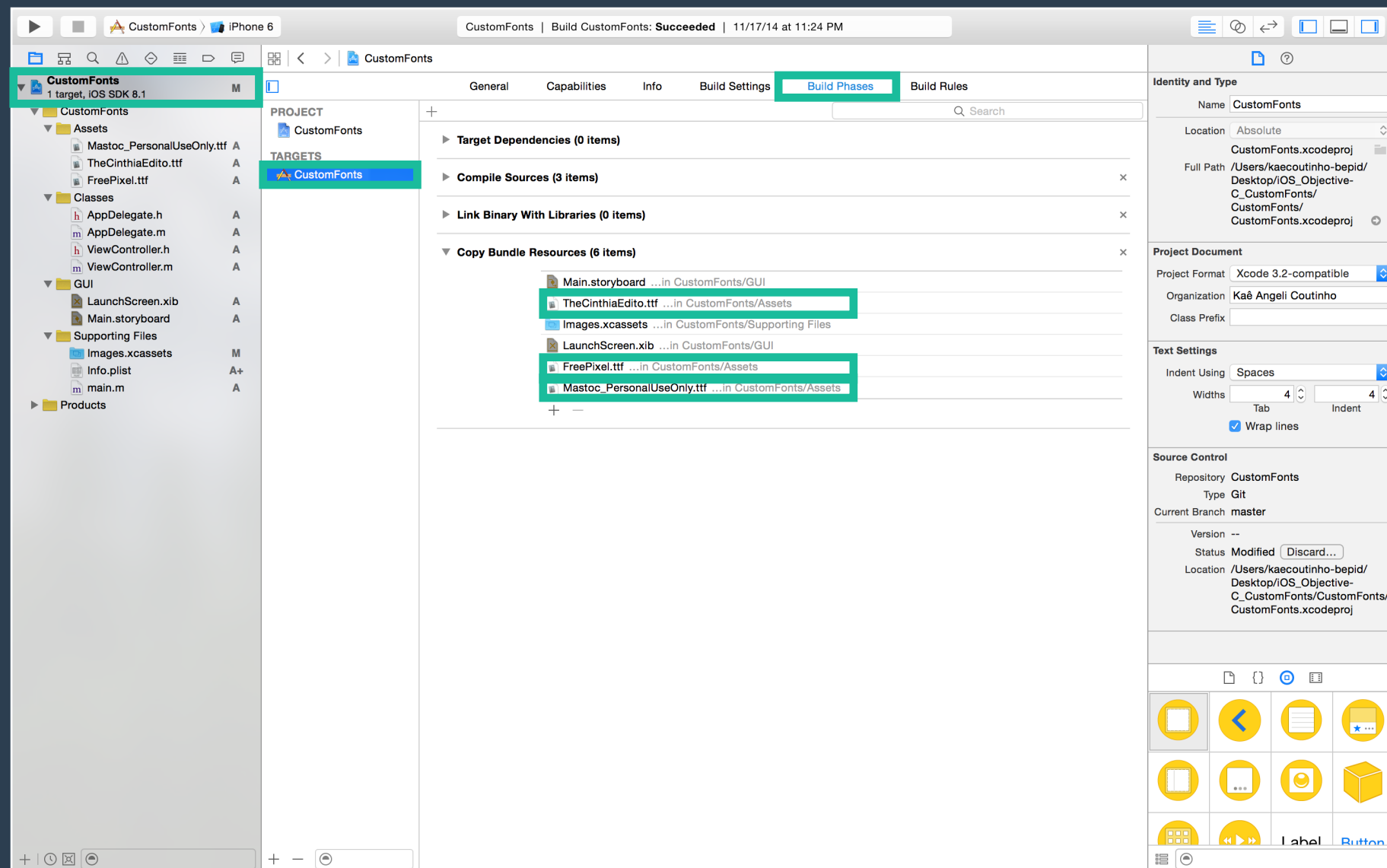


Kaê Angeli Coutinho

- Precisamos checar se as fontes estão no invólucro de compilação e fazem parte do aplicativo alvo

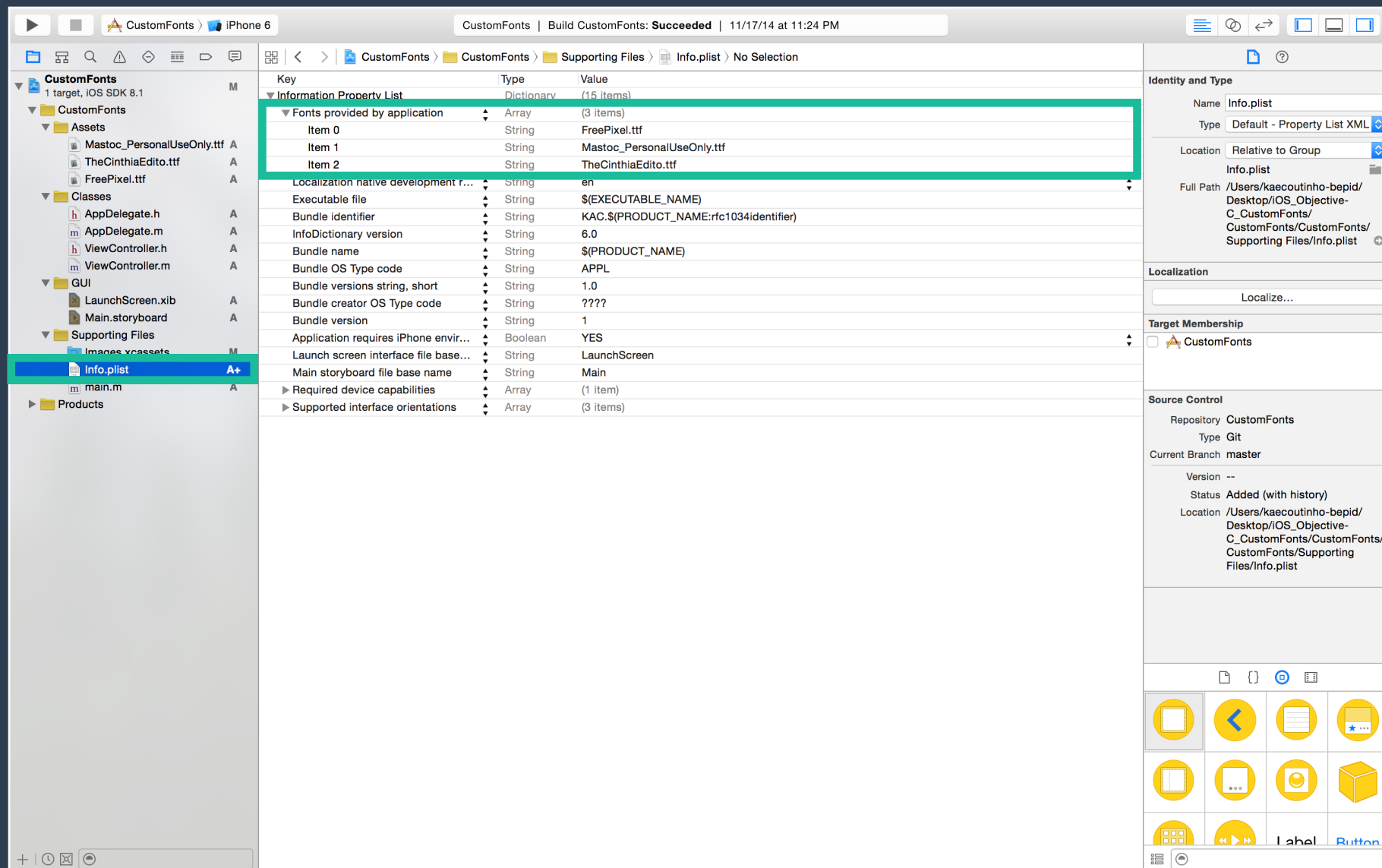


Kaê Angeli Coutinho



Kaê Angeli Coutinho

- Finalmente, precisamos modificar a property list para reconhecer as novas fontes



- Já com as fontes adicionadas, é necessário descobrir o nome real das fontes

```
-(void)viewDidLoad
{
    [super viewDidLoad];
    for(NSSString * family in [UIFont familyNames])
    {
        // Filtrando fontes para não exibir fontes de sistema
        if([family containsString:@"Free"] || [family containsString:@"Cinthia"] || [family containsString:@"Mastoc"])
        {
            for(NSSString * fontName in [UIFont fontNamesForFamilyName:family])
            {
                NSLog(@"\n Family: %@\n Font name: %@",family,fontName);
            }
        }
    }
}
```

```
2014-11-19 12:04:09.483 CustomFonts[3613:138884]
Family: Free Pixel
Font name: FreePixel-Regular
2014-11-19 12:04:09.483 CustomFonts[3613:138884]
Family: The Cinthia Edito
Font name: TheCinthiaEdito
2014-11-19 12:04:09.483 CustomFonts[3613:138884]
Family: Mastoc Personal Use Only
Font name: MastocPersonalUseOnly
```

- Pronto, com os nomes das fontes já adquiridos, vamos definir macros para eles e um tamanho de fonte padrão

```
//  
// ViewController.h  
// CustomFonts  
//  
// Created by Kaê Coutinho on 11/17/14.  
// Copyright (c) 2014 Kaê Angeli Coutinho. All rights reserved.  
//  
  
#import <UIKit/UIKit.h>  
#define FONT_FREEPIXEL @"FreePixel-Regular"  
#define FONT_THECINTHIAEDITO @"TheCinthiaEdito"  
#define FONT_MASTOC @"MastocPersonalUseOnly"  
#define FONT_SIZE 28  
  
@interface ViewController : UIViewController  
  
@property (weak, nonatomic) IBOutlet UISegmentedControl * fontsSegmentedControl;  
@property (weak, nonatomic) IBOutlet UILabel * fontLabel;  
  
-(IBAction)changeFont:(UISegmentedControl *)sender;  
  
@end
```

- Todos os requisitos para utilizarmos fontes personalizadas foram atendidos, é hora de implementar os IBActions

```
-(IBAction)changeFont:(UISegmentedControl *)sender
{
    switch (sender.selectedSegmentIndex)
    {
        case 0:
            [fontLabel setFont:[UIFont fontWithName:FONT_FREEPIXEL size:FONT_SIZE]];
            break;
        case 1:
            [fontLabel setFont:[UIFont fontWithName:FONT_THECINTHIAEDITO size:FONT_SIZE]];
            break;
        case 2:
            [fontLabel setFont:[UIFont fontWithName:FONT_MASTOC size:FONT_SIZE]];
            break;
        default:
            break;
    }
}
```

- Por último, mas não menos importante, vamos implementar outros dois métodos

```
-(void)viewDidLoad
{
    [super viewDidLoad];
    [self initialize];
}
```

```
-(void)initialize
{
    [[self fontsSegmentedControl] setSelectedSegmentIndex:0];
    [self changeFont:fontsSegmentedControl];
}
```

Concluindo

- Neste tutorial, você aprendeu alguns conceitos importantes, como:
 - Incorporar fontes personalizadas em um aplicativo
 - Manipulações de property lists
 - Utilização do componente de interface gráfica UISegmentedControl

