

Estilos IONIC

José Santorcuato T

En este proyecto vamos a trabajar con los estilos, comenzando por el icono, splashscreen y la propia aplicación(UI). Vamos a crear además el intro de nuestra aplicación, la pantalla que aparece la primera vez que ejecutamos nuestra app.

Comenzaremos por el esqueleto, agregando en primer lugar un slide intro

```
ionic start estilos blank
```

```
ionic g tabs misTabs
```

Intro **Inicio** Resultados Perfil

```
ionic cordova plugin add cordova-sqlite-storage
```

```
npm install --save @ionic-native/sqlite
```

Comenzaremos creando la típica intro que se le muestra a un usuario al instalar por primera vez la aplicación.

Ejemplo completo en 2.estilos introStorage

En app.components.ts

```
import { Component } from '@angular/core';
import { Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { IntroSliderPage } from '../pages/intro-slider/intro-slider';
import { LoadingPage } from '../pages/loading/loading';
import { IonicPage, NavController, NavParams, ViewController } from 'ionic-angular';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  public viewCtrl: ViewController;
  rootPage = LoadingPage;
  constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
    platform.ready().then(() => {
      // Okay, so the platform is ready and our plugins are available.
      // Here you can do any higher level native things you might need.
      statusBar.styleDefault();
      splashScreen.hide();

      platform.registerBackButtonAction(() => {
        platform.exitApp();
      });
    });
  }
}
```

App.modules.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { SplashScreen } from '@ionic-native/splash-screen';
import { StatusBar } from '@ionic-native/status-bar';
```

```
import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { IntroSliderPage } from '../pages/intro-slider/intro-slider';
import { MisTabsPage } from '../pages/mis-tabs/mis-tabs';
import { Storage } from '@ionic/storage';
import { IonicStorageModule } from '@ionic/storage';
import { LoadingPage } from '../pages/loading/loading';
```

```
@NgModule({
  declarations: [
    MyApp,
    HomePage,
    IntroSliderPage,
    MisTabsPage,
    LoadingPage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
    IonicStorageModule.forRoot()
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    IntroSliderPage,
    LoadingPage,
    MisTabsPage
  ],
  providers: [
    StatusBar,
    SplashScreen,
    {provide: ErrorHandler, useClass: IonicErrorHandler}
  ]
})
export class AppModule {}
```

Intro-slider.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { Storage } from '@ionic/storage';
import { MisTabsPage } from '../mis-tabs/mis-tabs';

/**
 * Generated class for the IntroSliderPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for more info on
 * Ionic pages and navigation.
 */

@IonicPage()
@Component({
  selector: 'page-intro-slider',
  templateUrl: 'intro-slider.html',
})
export class IntroSliderPage {

  public options;

  constructor(private nav: NavController,
               private storage: Storage) {
    this.options = {
      show: false
    };
  }

  complete() {
    if (!this.options.show) {
      //Save flag if don't need to show intro slider on next app startup
      this.storage.set('showIntroSlider', 0).then(() => {
        // we don't actually need this callbacks in such simple example,
        // but in real apps it will be useful
      });
    }
    this.nav.setRoot(MisTabsPage);
  }
}
```

loading.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { IntroSliderPage } from '../intro-slider/intro-slider';
import { MisTabsPage } from '../mis-tabs/mis-tabs';
import { Storage } from '@ionic/storage';

/**
 * Generated class for the LoadingPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for more info on
 * Ionic pages and navigation.
 */

@IonicPage()
@Component({
  selector: 'page-loading',
  templateUrl: 'loading.html',
})
export class LoadingPage {

  constructor(private nav: NavController,
              private storage: Storage) {

  }

  ionViewDidEnter() {
    this.storage.get('showIntroSlider').then((v) => {
      if (v == 0) {
        this.nav.push(MisTabsPage);
      }
      else {
        this.nav.push(IntroSliderPage);
      }
    }, () => {
      this.nav.push(MisTabsPage);
    });
  }
}
```

Estilo

Es importante señalar que IONIC apunta a una lógica y estética **Material Design**.

Podemos utilizar Sass, para agregar nuevos **colores, tipografías, objetos UI, etc.**

Puedes obtener nuevos temas en <https://bootswatch.com/>

Ejemplo:

<https://bootswatch.com/spacelab/>

Variables

<https://bootswatch.com/4/spacelab/variables.scss>

<https://bootswatch.com/4/spacelab/bootswatch.scss>

Los archivos a editar:

src/theme/variables.scss

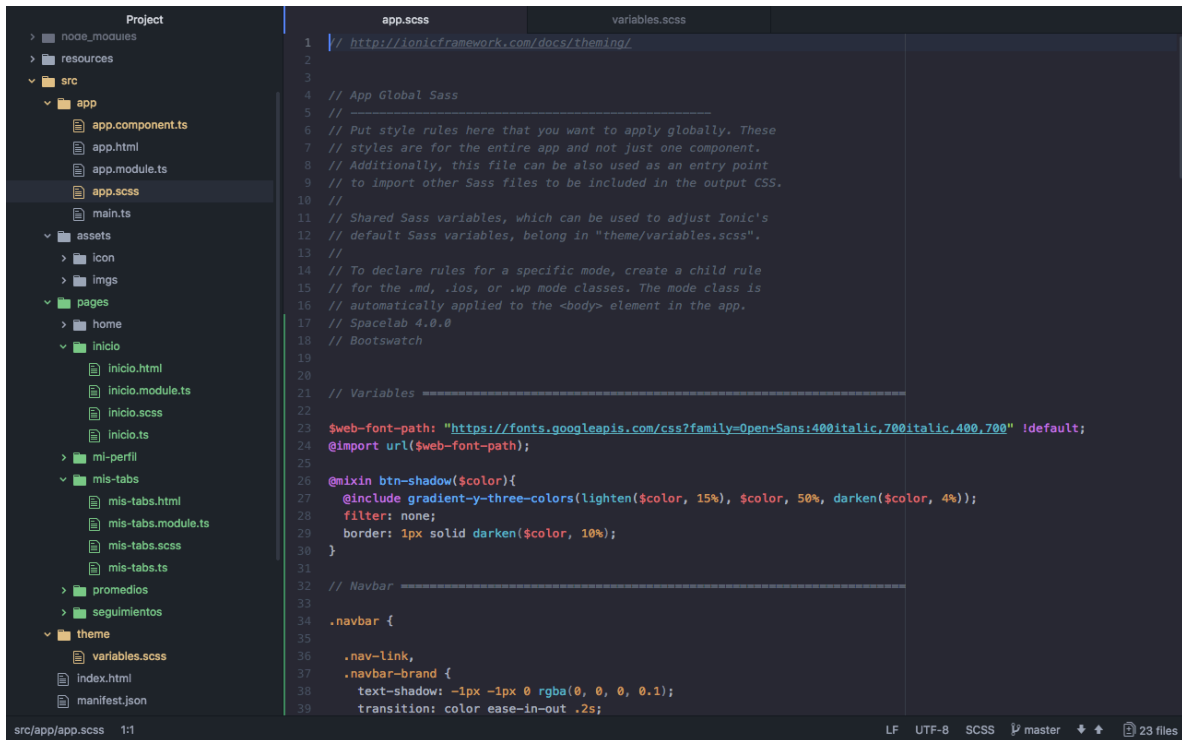
src/app/app.scss

***Cada componente posee su propio html y css**

Ubicación archivos

src/theme/variables.scss

src/app/app.scss



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure, with the following files and folders:

- node_modules
- resources
- src
 - app
 - app.component.ts
 - app.html
 - app.module.ts
 - app.scss
 - main.ts
 - assets
 - icon
 - imgs
 - pages
 - home
 - inicio
 - inicio.html
 - inicio.module.ts
 - inicio.scss
 - inicio.ts
 - mi-perfil
 - mis-tabs
 - mis-tabs.html
 - mis-tabs.module.ts
 - mis-tabs.scss
 - mis-tabs.ts
 - promedios
 - seguimientos
 - theme
 - variables.scss
- index.html
- manifest.json

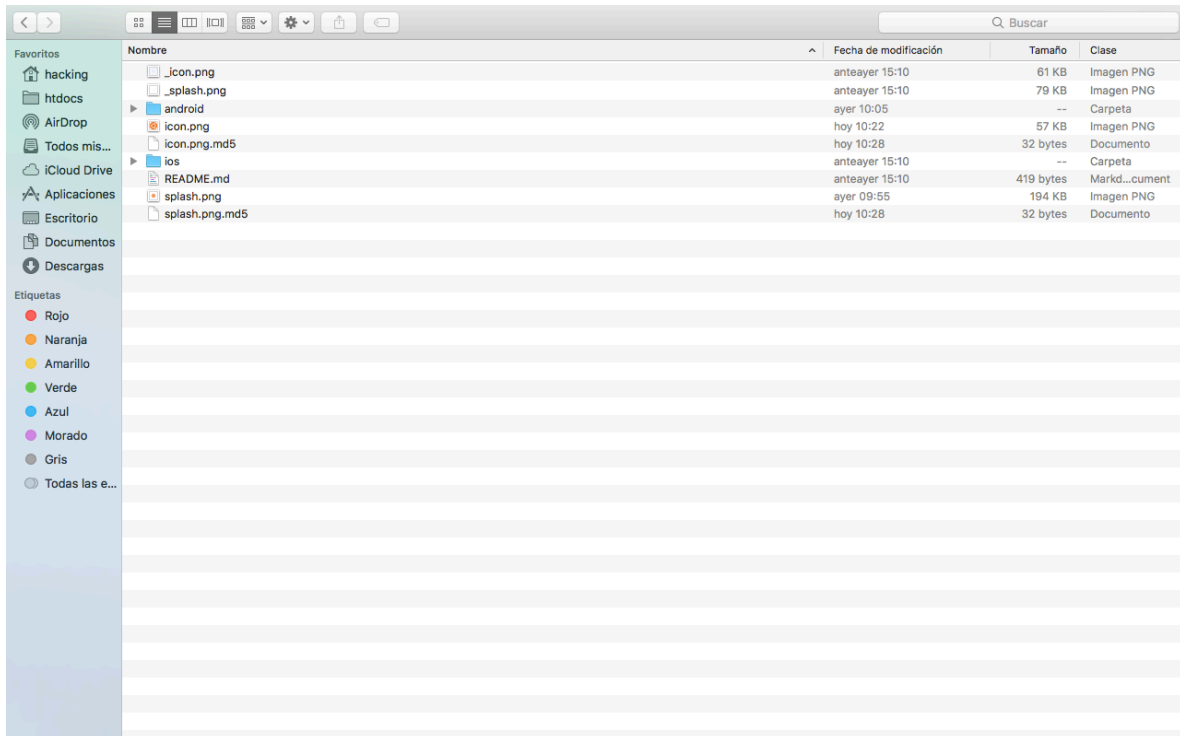
The code editor shows the content of the `app.scss` file, which is a Sass file for the application. It includes comments and CSS rules for the application's styling.

```
1 // http://ionicframework.com/docs/theming/
2
3
4 // App Global Sass
5 // -----
6 // Put style rules here that you want to apply globally. These
7 // styles are for the entire app and not just one component.
8 // Additionally, this file can be also used as an entry point
9 // to import other Sass files to be included in the output CSS.
10 //
11 // Shared Sass variables, which can be used to adjust Ionic's
12 // default Sass variables, belong in "theme/variables.scss".
13 //
14 // To declare rules for a specific mode, create a child rule
15 // for the .md, .ios, or .wp mode classes. The mode class is
16 // automatically applied to the <body> element in the app.
17 // Spacelab 4.0.0
18 // Bootswatch
19
20
21 // Variables -----
22
23 $web-font-path: "https://fonts.googleapis.com/css?family=Open+Sans:400italic,700italic,400,700" !default;
24 @import url($web-font-path);
25
26 @mixin btn-shadow($color){
27   @include gradient-y-three-colors(lighten($color, 15%), $color, 50%, darken($color, 4%));
28   filter: none;
29   border: 1px solid darken($color, 10%);
30 }
31
32 // Navbar -----
33
34 .navbar {
35
36   .nav-link,
37   .navbar-brand {
38     text-shadow: -1px -1px 0 rgba(0, 0, 0, 0.1);
39     transition: color ease-in-out .2s;
```

Lo primero que haremos será cambiar el splashscreen y el icono

Las imágenes de splashscreen tienen una dimensión de 2732x2732píxeles y el icono de 1024x1024 ambas en PNG.

`/resources`



`/resources`

Reemplazamos la imagen splashscreen.png e icon.png por la nuestra y ejecutamos

```
ionic cordova resources android
```

Cada cambio de contenido debemos ejecutar este comando, se redimensionan imágenes, iconos, etc.

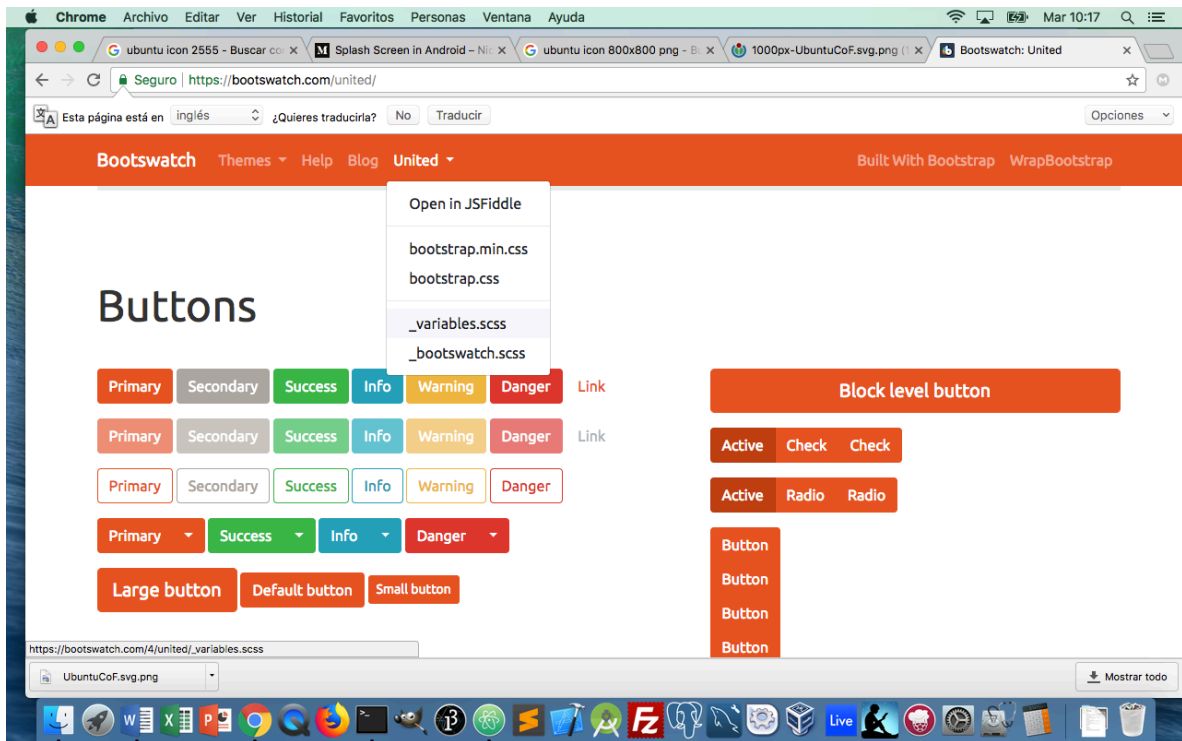
Bootswatch

Agregaremos el estilo de <https://bootswatch.com/united/> , es el tema de Ubuntu

Ahí se encuentran los archivos que contienen las variables scss que luego agregaremos en

```
src/app/theme variable.scss
```

```
src/app/app.css bootswatch.scss
```



Tipografía

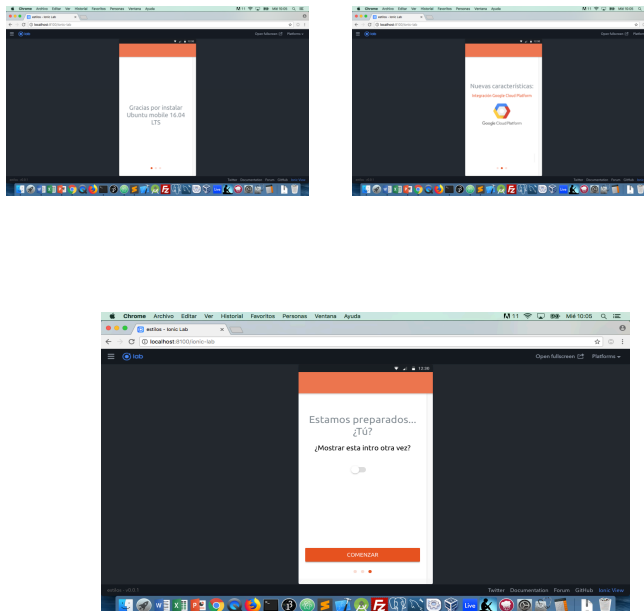
Editamos el archivo **intro.scss**

```
page-intro {  
  
  h1{  
  
    font-family: $font-family-sans-serif;  
  }  
  
}
```

Esa variable **\$font-family-sans-serif** está definida en variables así:

```
$font-family-sans-serif: "Ubuntu", -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,  
"Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol"  
!default;
```

Revisa la carpeta estilos2, he modificado el ejemplo de arriba para que luzca así:



Material complementario

Ionic tiene en sus html mucha similitud con Bootstrap, maneja el mismo concepto de grilla, con prácticamente los mismo nombres.

Acá está la especificación de la grilla y los componentes, botones, selectores, checkbox, inputs, fabs, modales, etc.

<https://ionicframework.com/docs/api/components/grid/Grid/>