

# Introducción a Firebase AUTHENTICATION



# DOCUMENTACIÓN

<https://firebase.google.com/docs/auth/>

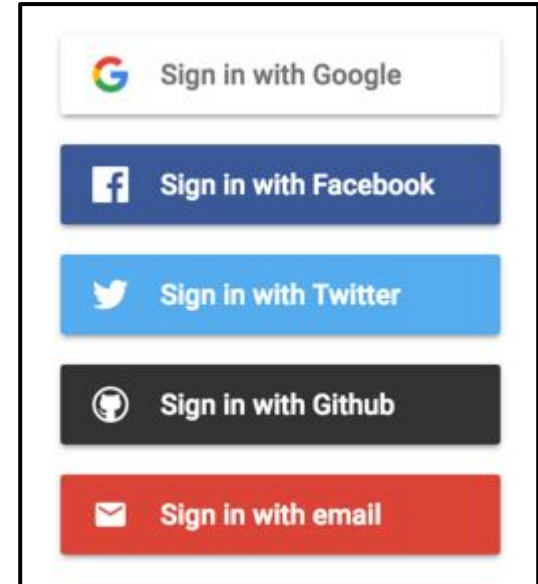
# Authentication

- La mayoría de las apps deben reconocer la identidad de un usuario. •
- Permite a una app guardar datos de este de manera segura en la nube. •
- Brinda la misma experiencia personalizada en todos sus dispositivos. •
- Soporta autenticación con varios proveedores de servicios. •



# Usuarios

- Los usuarios Firebase tienen las siguientes properties:
  - Id (Único)
  - eMail Address
  - Name
  - Photo
- Según el método de registración, (google, Facebook, autenticación custom, etc...) se llenarán más o menos campos de forma automática.



# Authentication

Entonces, cómo autenticar usuarios con Firebase desde nuestras aplicaciones iOS ?

# Agregar Firebase a nuestro proyecto

- **Trabajaremos simultáneamente:**
  - En la Consola Web de Firebase
  - En el proyecto XCode



# Agregar Firebase a nuestro proyecto

- Con CocoaPods, agregar en el Pod file:

```
pod 'Firebase'
```

- Además, se pueden agregar pods por cada funcionalidad requerida...

Pod	Servicio
pod 'Firebase/Core'	Solo incluye Analytics
pod 'Firebase/AdMob'	AdMob
pod 'Firebase/Messaging'	Cloud Messaging / Notifications
pod 'Firebase/Database'	Realtime Database
pod 'Firebase/Invites'	Invites
pod 'Firebase/DynamicLinks'	Dynamic Links
pod 'Firebase/Crash'	Crash Reporting
pod 'Firebase/RemoteConfig'	Remote Config
pod 'Firebase/Auth'	Authentication
pod 'Firebase/AppIndexing'	App Indexing
pod 'Firebase/Storage'	Storage



# En nuestro proyecto agregamos:

- Con CocoaPods, agregar en el Pod file:

```
pod 'Firebase'  
pod 'FirebaseUI'  
pod 'Firebase/Storage'  
pod 'Firebase/Database'
```

- Luego, `pod install` como siempre

# En la Consola Web:

- Es necesario contar con una cuenta de Firebase  
*Cualquier cuenta en Google es una cuenta en Firebase*
- URL de la Consola: <https://console.firebase.google.com/>
- Una vez en la consola:



# En la Consola Web:



ID del paquete de iOS ?

▼ Identity

Display Name	<input type="text" value="LoginFirebase"/>
Bundle Identifier	<input type="text" value="ar.com.learnit.LoginFirebase"/>
Version	<input type="text" value="1.0"/>
Build	<input type="text" value="1"/>

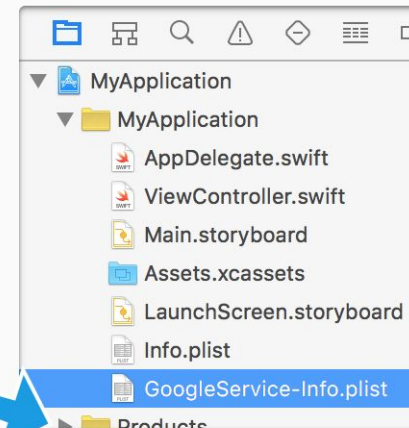


# En la Consola Web:

## Instrucciones de Xcode

Alternativas: [Unity](#) [C++](#)

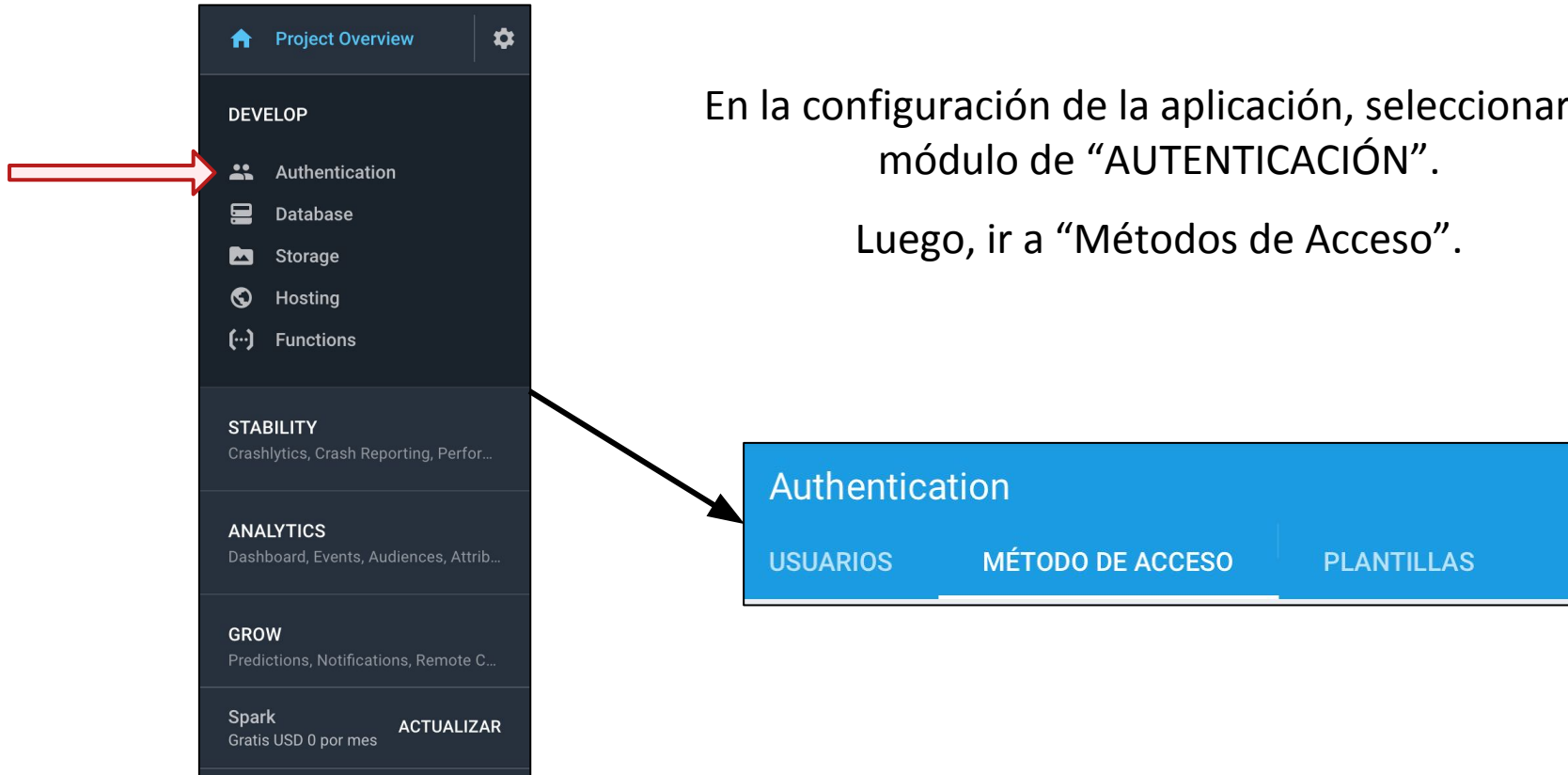
1.  Descargar GoogleService-Info.plist
2. Coloca el archivo **GoogleService-Info.plist** que acabas de descargar en el directorio raíz de tu proyecto de Xcode y agrégalo a todos los destinos.





# En la Consola Web:

En la configuración de la aplicación, seleccionar el módulo de “AUTENTICACIÓN”.

Luego, ir a “Métodos de Acceso”.



# En la Consola Web:

Proveedores de acceso	
Proveedor	Estado
 Correo electrónico/contraseña	Habilitado
 Teléfono	Inhabilitado
 Google	Habilitado
 Facebook	Inhabilitado
 Twitter	Inhabilitado
 GitHub	Inhabilitado
 Anónimo	Habilitado

Luego, habilitar y configurar los proveedores de Login que queremos en nuestra app.

## IMPORTANTE

Para utilizar cualquiera de los proveedores, hay que registrar nuestra app en el sitio de developers de dichos proveedores

# En XCode:

- Agregar en el AppDelegate el siguiente código para inicializar Firebase •

```
import Firebase

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(_ application: UIApplication,
                    didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?)
        -> Bool {

        FirebaseApp.configure()
        return true
    }
}
```

# En XCode:

- **Autenticación con email y contraseña con FirebaseAuthUI** ●

Importar, según sea necesario, el AuthUI del provider a usar.

```
import Firebase
import FirebaseAuthUI
import FirebaseAuthUI
import FirebaseAuthUI
import FirebaseAuthUI
```

Luego, configurar el defaultAuthUI con el/los provider/s seleccionado/s.

```
var googleProvider = FUIGoogleAuth()
var facebookProvider = FUIFacebookAuth()
var twitterProvider = FUITwitterAuth()
let providers:[FUIAuthProvider] = [googleProvider, facebookProvider, twitterProvider]
FUIAuth.defaultAuthUI()?.providers = providers
```



# En XCode:

- **Responder a los cambios de estado** ●

Agregar un listener que se entera de los cambios de Authentication  
(por ejemplo, si hubo un 'log in' o un 'log out')

```
Auth.auth().addStateChangeListener { (auth: Auth, user: User?)  
in  
    // se procesa el evento  
}
```

Dentro del closure, se puede manejar el evento de acuerdo a las acciones necesarias como resultado de lo que sucedió