#io16

Autor: Gema Arnaiz Delgado

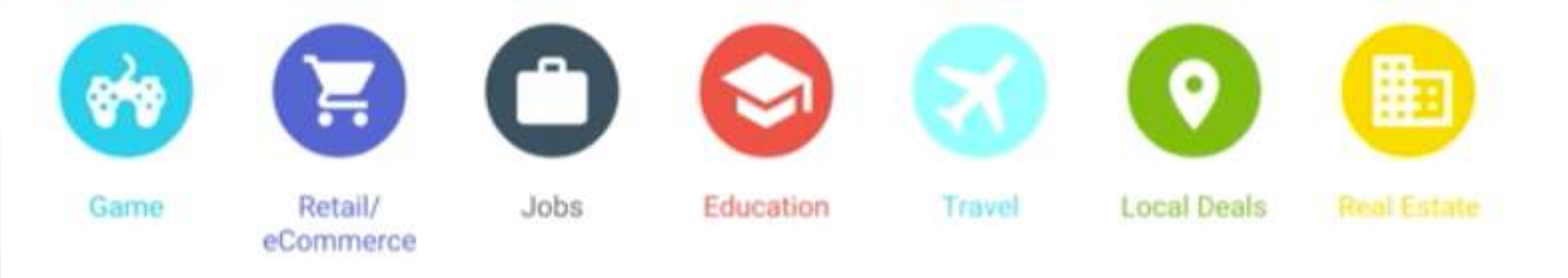
¿Qué es Firebase?

A partir de un servicio web nos ofrecen la posibilidad de programar aplicaciones con datos que se sincronizan en tiempo real a través de múltiples dispositivos, evitando muchas de las tareas engorrosas en las que tendríamos que dedicar tiempo al programar.

Plataformas

Firebase cuenta con una serie de librerías mediante las cuales podemos conectarnos y mantenernos suscritos a los cambios de los datos, compatibles con los sistemas más comunes como son iOS, Android y Web, pero también a varios lenguajes de programación del lado del servidor como podrían ser Python o PHP.

Tipos de aplicaciones



Características

Características

- **Analíticas**: Provee una solución gratuita para tener todo tipo de medidas (hasta 500 tipos de eventos), para gestionarlo todo desde un único panel.
- **Desarrollo**: Permite construir mejores apps, permitiendo delegar determinadas operaciones en Firebase, para poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad. Entre sus características destacan el almacenamiento, testeo, configuración remota, mensajería en la nube o autenticación, entre otras.
- **Crecimiento:** Permite gestionar los usuarios de las aplicaciones, pudiendo además captar nuevos. Para ello dispondremos de funcionalidades como las de invitaciones, indexación o notificaciones.
- **Monetización**: Permite ganar dinero gracias a AdMob.

Autenticación Firebase (DEVELOP)

#io16

Base de datos en tiempo real

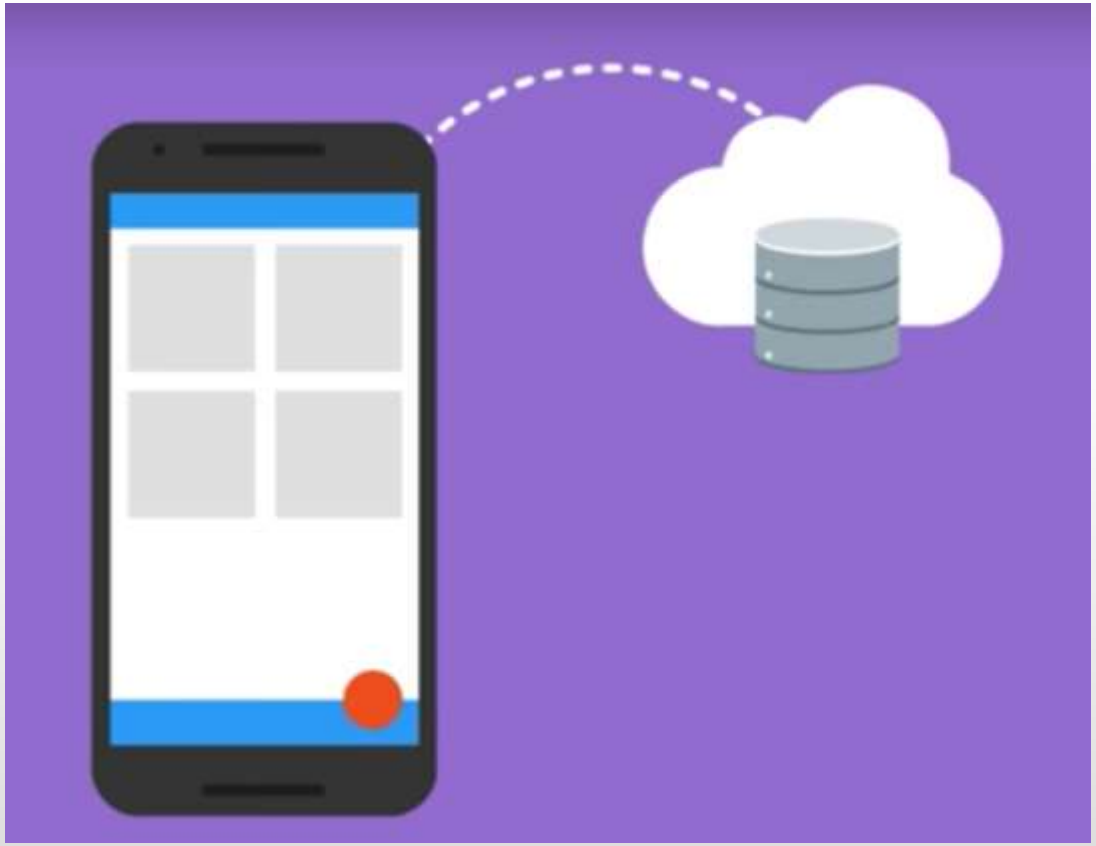
- Almacena y sincroniza información con no SQL en la nube.
- La información se sincroniza con todos los clientes en tiempo real, e incluso en modo offline.

Implementación

1	Integrate the Firebase Realtime Database SDKs.	Quickly include clients via Gradle, CococaPods, or a script include.
2	Create Realtime Database References.	Reference your JSON data, such as "users/user:1234/phone_number" to set data or subscribe to data changes.
3	Set Data and Listen for Changes.	Use this reference to write data or subscribe to changes.
4	Enable Offline Persistence.	Allow data to be written to the device's local disk so it can be available while offline.
5	Secure your data.	Use Firebase Realtime Database Security Rules to secure your data.

Mensajería en la nube

Firebase Almacenamiento

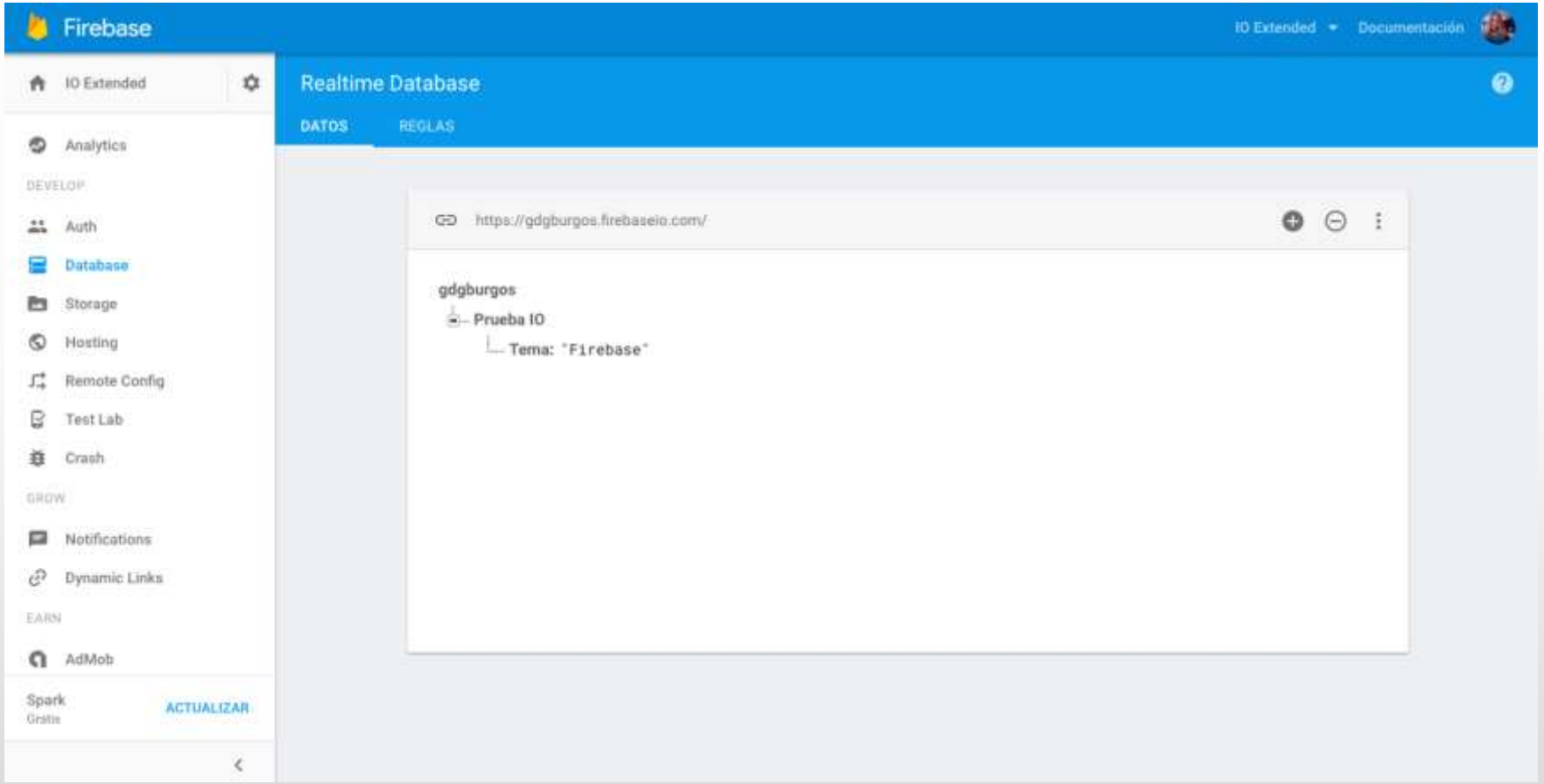


#io16

Estructura información

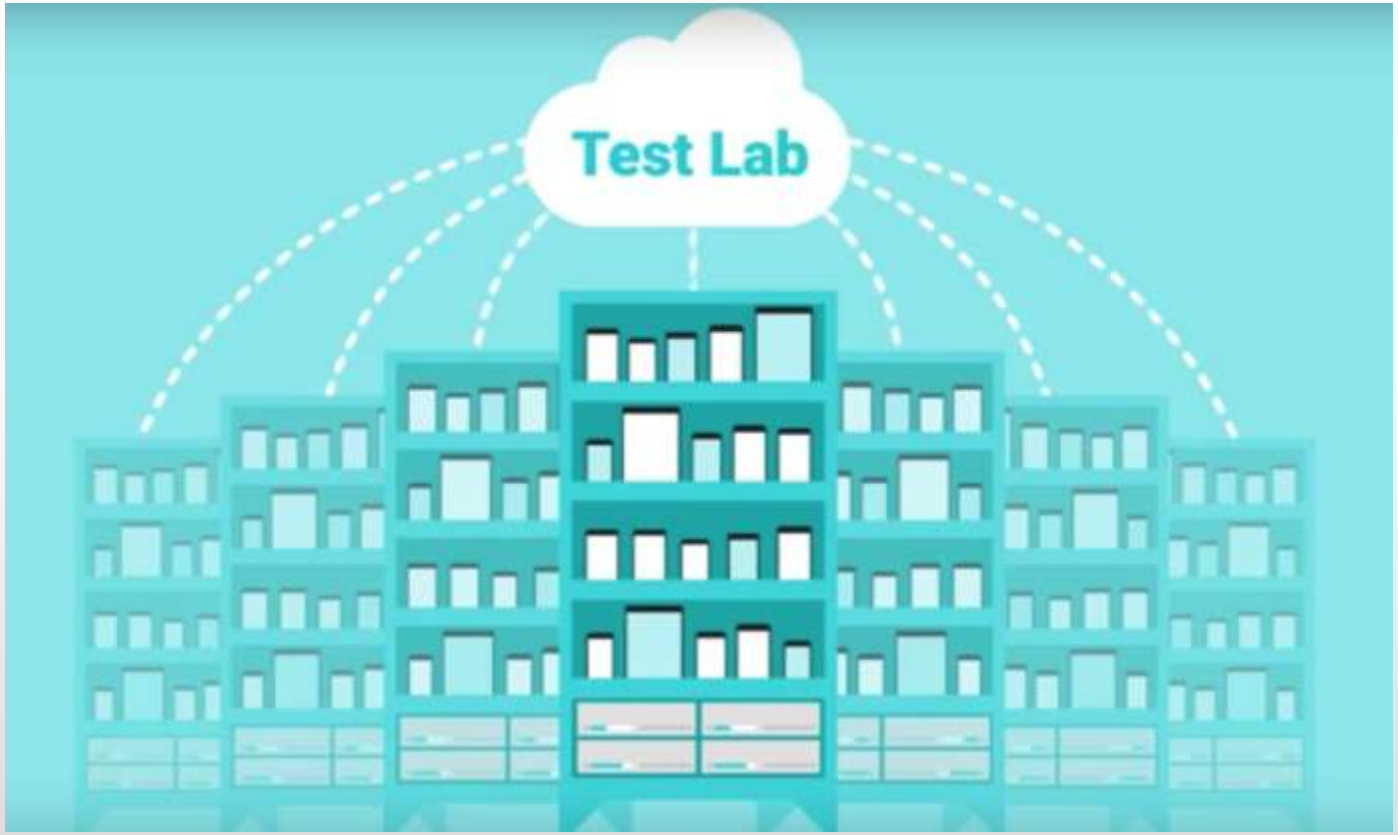
```
"messages": {
  "one": {
    "m1": {
      "name": "eclarke",
      "message": "The relay seems to be malfunctioning.",
      "timestamp": 14593618753337
    },
    "m2": { ... },
    "m3": { ... }
  },
  "two": { ... },
  "three": { ... }
}
```

Hola mundo Firebase

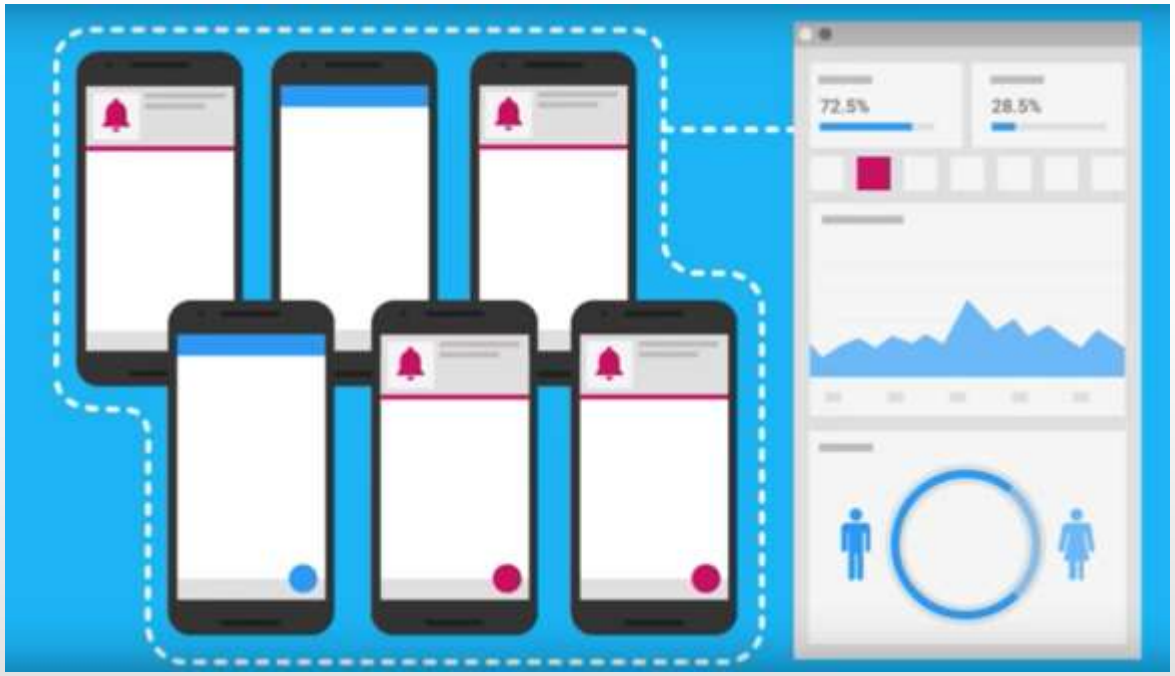


#io16

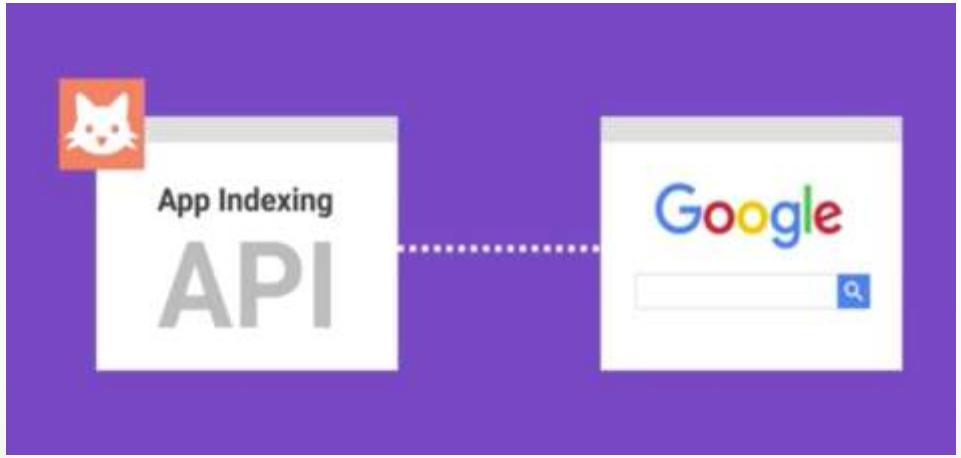
Test Lab

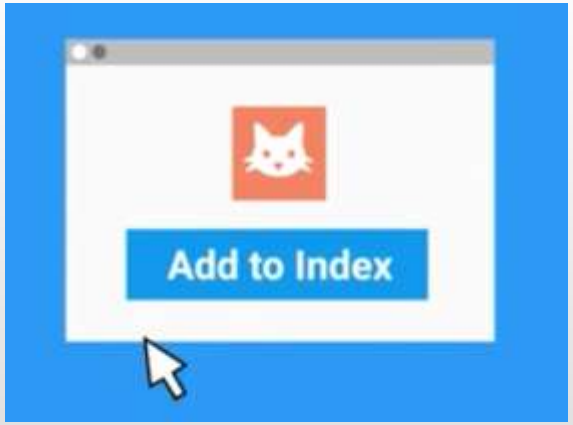


Notificaciones (GROW)



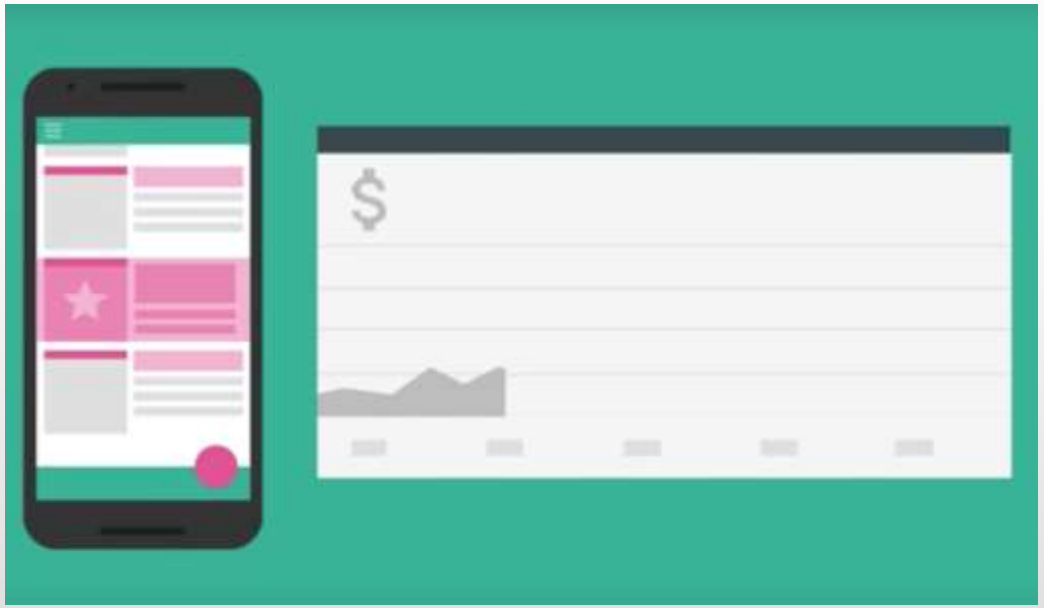
App Indexing





Firebase Invites

Google AdMob (EARN)



Implementación (Write)

- @import Firebase; //añadimos Firebase al proyecto
- [FIRApp configure]; //inicialiazamos
- self.ref = [[FIRDatabase database] reference]; //inicializamos la referencia
- [[[_ref child:@"users"] child:user.uid]
  setValue:@{@"username": username}]; /añadimos un nuevo usuario
- [[[[_ref child:@"users"] child:user.uid] child:@"username"] setValue:username];
//actualizar el nombre del usuario.

Listeners Firebase

```
[_commentsRef
        observeEventType:FIRDataEventTypeChildAdded
        withBlock:^(FIRDataSnapshot *snapshot) {
          [self.comments addObject:snapshot];
          [self.tableView insertRowsAtIndexPaths:@[
            [NSIndexPath indexPathForRow:[self.comments count] - 1 inSection:1]
          ]
                              withRowAnimation:UITableViewRowAnimationAutomatic];
        }];

[_commentsRef
 observeEventType:FIRDataEventTypeChildRemoved
 withBlock:^(FIRDataSnapshot *snapshot) {
   int index = [self indexOfMessage:snapshot];
   [self.comments removeObjectAtIndex:index];
   [self.tableView deleteRowsAtIndexPaths:@[[NSIndexPath indexPathForRow:index inSection:1]]
                         withRowAnimation:UITableViewRowAnimationAutomatic];
 }];
```

Información persistente

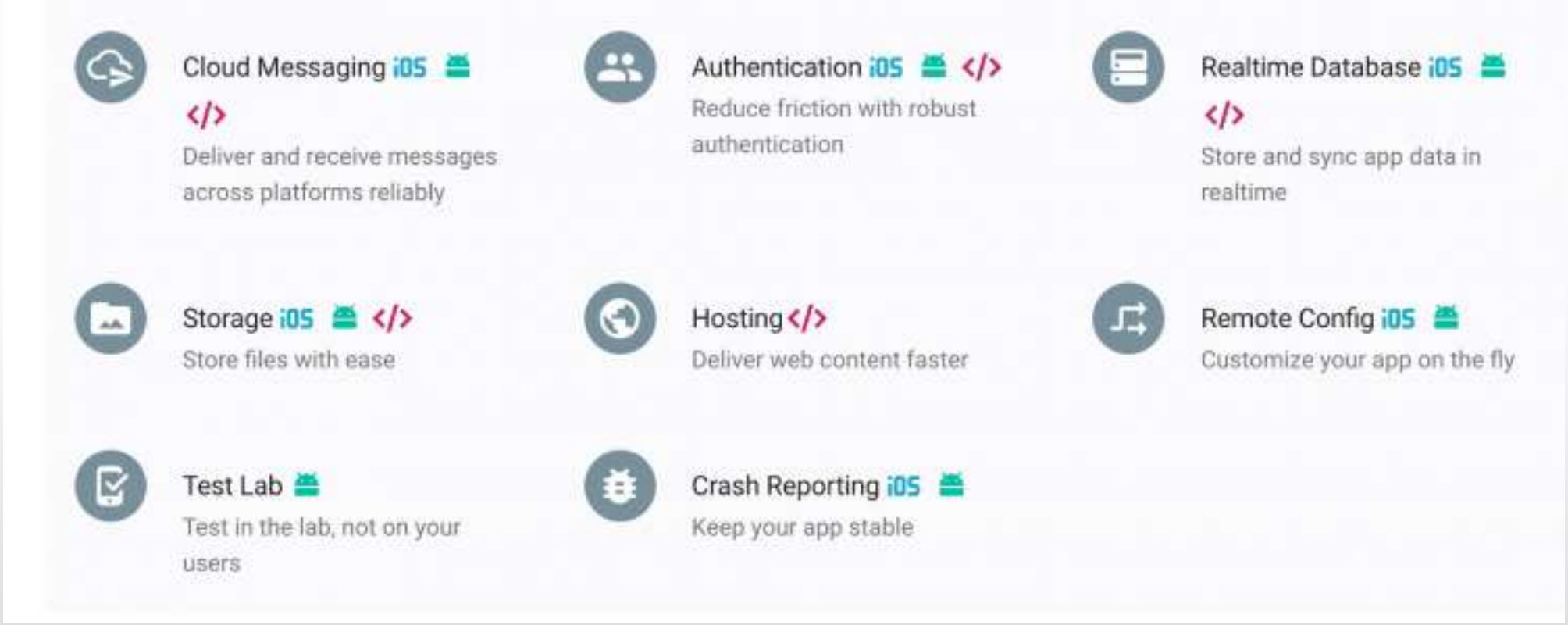
- [FIRDatabase database].persistenceEnabled = YES;

- FIRDatabaseReference *scoresRef = [[FIRDatabase database] referenceWithPath:@"scores"];
[scoresRef keepSynced:YES]; //escuchamos una localización específica

- [presenceRef onDisconnectRemoveValueWithCompletionBlock:^(NSError* error, FAUser* user) {
    if (error != nil) {
        NSLog(@"Could not establish onDisconnect event: %@", error);
    }
}];

#io16

Documentación



#io16