cambia el comportamiento y el aspecto de la app sin que los usuarios tengan que descargar una actualización.

**Se usa cuando:**

* Rediseños , reimplementaciones , bug fix
* Lanzamientos progresivos (soft/hard launch )
* Habilitar/desabilitar una funcionalidad

**Como funciona:**

1. App solicita configuracion a los server del firebase
2. Firebase evalue condiciones y devuelve la adecuada
3. La app utiliza la configuracion

**Acceder a Firebase Remot config :**

val remoteConfig: FirebaseRemoteConfig = Firebase.remoteConfig

**Establecer intervalo de recuperacin minimo , para permitir actualziaciones frecuentes**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Asignar valores por defecto para las configuraciones**

**Por codigo :**

Imagen que contiene Texto

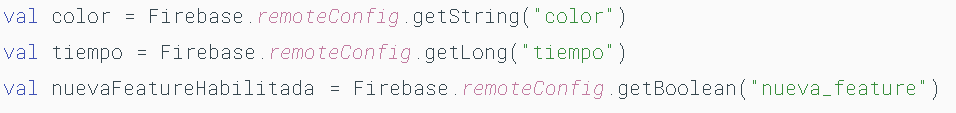
Descripción generada automáticamente

**Por XML :**

Texto

Descripción generada automáticamente con confianza media

**Buscar configuraciones remotas:**



**Agregar estos valores al objeto Remote Config**

**Se pueden establecer valores de parametoros en el backend de Remote config (consola firebase)**

**Authentication/ Identification**

**Identificacion :** Capacidad de comprobar a un usuario , es algo que debe diferenciarlo del resto

**Autenticaction :** Proceso que valida la identidad del usuario , nos dara el acceso al sistema

**Autorizacion :**  Define si el suuario tiene acceso al recurso que intenta utilziar

**Servicios de autenticación de terceros:**

**Oauth2** : Obtiene el permiso del usuario para cceder a un recurso en su nombre (usuario ya identificado y autenticado)

**OpenID:** Obtencion digital de un usuario , identifica y autentica al usuario

**OpenID Conecct:** (unifica los dos mundos)

* Conoce la identidad del usuario
* Podemos actuar sobre recursos autorizados por el usuario

**Token :**  Rpresenta la identidad de un usuario.

**Firebase Auth**

* Resuelve el proceso de identificacion y autenticacion de usuario
* Se encarga de recordar y mantener logueado al usuario
* Permite incluir servicios de auth de terceros (OPENID – OAUTH2)
* Permite administrar a los usuarios desde la consola

**Tipos de autenticacion**

**Directa :**

* Recomendada por Firebase
* Nos provee una UI y un flujo controlado y customziable

**Autenticacion con email/password:**

* Debemos implementar nuestro propio flujo de auth y nuestra UI.
* Experiencia customizable
* Mayor tiempo de implementacion

**Integracion con identidades :**

* Facebook , google , github etc.

**Utilizando numero de telefono:**

* Envia un sms

**Sistemas propios de aut e identificacion:**

**Usuarios Anonimos :**

* Temporales
* Usuario pueda probar neustra app
* Firebase tiene un flujo para convertir una cuenta anonima en uan identificada

**RelTime data base**

Base de datos alojada en la nube que guarda los datos en formato JSON

Mantiene sincronziada una bdd local conuna remota y recibe actualziacione sen tiempo real

**¿ Que es una database?**

Conjunto de datos perteneciente a un contexto y almacenados sistematicamente

**Existen Dos formas de almacenar**

* BDD remotas (servidor)
* BDD locales (telefono)

**Tipos de BDD:**

* Relacionales
  + Tablas , columnas y registros
  + Evitan redundancia de datos
* No relacionales
  + Info guardadad en documentos u objetos
  + Mas veloces
  + SQLite , MongoDB

**Caractersiticas :**

* Almacena datos en la consulta de Firebase
* No relaciona
* Cache automatico
* Los datos se sincronizan cuando hay conexión
* Se accede a lso datos con una ruta similar a una URL

**INTEGRACION:**

1. IMPLEMENTATIO DATABASE
2. Se accede a realtime con

Texto

Descripción generada automáticamente con confianza baja

De esta manera, puedes guardar diversos tipos de datos en la base de datos, incluso objetos Java.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**.child =**  Trae datos de la BD

**Sincronizacion Como mantener actualziada la BDD REMOTA :**

Polling : El cliente pregutna al servidor por nuev ainfo cada cierto tiempo

* Si no hay conexión al realizar la peticion , la siguient realizara l sin probmeas
* No provee info en tiempo real
* Crea hilo por cada peticion , mas procesamiento y mayor consumo de bateria

Websocket : Provee info en tiempo real , es mas compleja de implementar (manejo de erroes , no conexión etc)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

onChildAdded :Se usara cuando se agrege un nuevo child

onChildChanged: Cuando la localizacion del child cambie

onChildRemoved : Cuando se elimina un child

onChildMoved : cuandos e movo el child

onChildCancelled: Se activara cuando listener remita que fallo el servidor