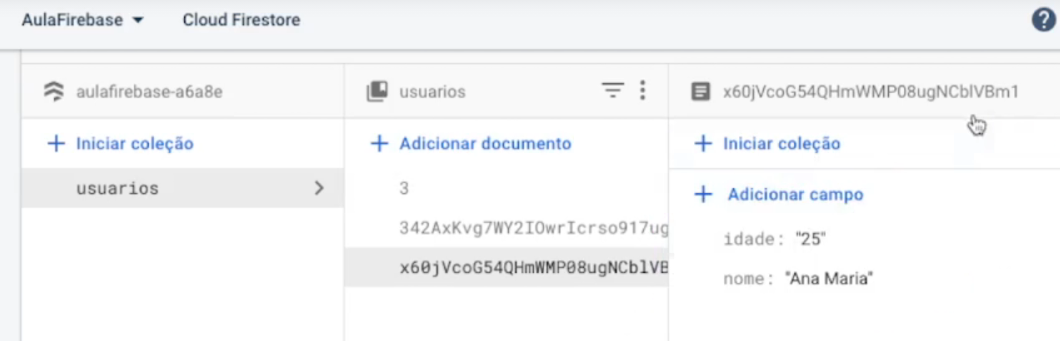
Atenção



# Com o método GET

Com o método get vamos recuperar os dados de forma instantânea, mas se esses dados forem alterados, vamos precisar recuperar eles novamente.

private fun recuperarDadosComGET(){  
  
 //salvarDadosUsuario("AllephNogueira", "1994")  
  
 /\* Aqui vamos recuperar os dados do usuario  
 vamos recuperar os dados do usuario que esta logado \*/  
  
 val idUsuarioLogado = autenticacao.*currentUser*?.*uid* if (idUsuarioLogado != null) {  
 val referenciaUsuario = bancoDeDados  
 .collection("usuarios")  
 .document(idUsuarioLogado)  
  
 /\* Get = pegar  
 Vamos pegar os dados do usuario que vai retornar da nossa consulta a cima  
 Atençao o GET pegamos esse dado de uma unica vez, como é a data de nascimento e isso não vai ficar sendo alterado  
 Podemos recuperar com o GET  
 \*/  
 referenciaUsuario  
 .get()  
 .addOnSuccessListener **{** documentSnpashot **->** /\* Aqui dentro se conseguir recuperar os dados, vamos ter um documentSnapshot  
 Snapshot = instantanea ( algo pego na hora )  
 Ou seja ele vai capturar o documento como esta naquele momento.  
 \*/  
  
 /\* Aqui estamos recuperando o documento na hora  
 o data = os dados que estao la dentro.  
 \*/  
 val dados = documentSnpashot.*data* /\* Atenção os dados é do tipo map, então ele vai ter chave e valores.  
 Atençao os dados que vem do servidor podem ser nulos, então voce deve verificar também  
 \*/  
  
 if (dados != null) {  
 /\* Se ele nao retornar null, vamos poder pegar os dados  
 Atençao as colunas devem esta da mesma forma que voce esta inserindo aqui  
 Nesse modelo a baixo voce esta inserindo a chave da e ele vai retornar o valor daquela chave.  
 \*/  
  
 val nome = dados["nome"]  
 val anoNascimento = dados["anoNascimento"]  
  
 /\* Agora vamos inserir esses dados na tela  
  
 \*/  
  
 binding.textResultado.*text* = "Usuario: $nome - nascido em $anoNascimento"  
   
   
 }  
  
  
 **}** .addOnFailureListener **{** exibirMensagem("Falha em recuperar os dados")  
 **}** }  
  
  
}

# Capturando os dados sempre atualizados (Melhor maneira)

Dessa forma vamos recuperar os dados sempre atualizados, sem precisar ficar fazendo get a cada vez que precisar atualizar.

Atenção ao Listener = ele fica ouvindo, procurando por atualizações.

Ou seja o listener fica sempre ouvindo/olhando o banco de dados, quando ele achar que tem algo diferente ele vai atualizar.

private fun recuperarDadosUsuariosAtualizados() {  
 // dessa forma sempre que tiver alguma atualização os dados vai ser atualizado também.  
 val idUsuarioLogado = autenticacao.*currentUser*?.*uid* // Pegar id do usuario LOGADO  
  
 if (idUsuarioLogado != null) { // Salvar a referencia do usuario  
 val referenciaUsuario = bancoDeDados  
 .collection("usuarios")  
 .document(idUsuarioLogado)  
  
  
 // Agora vamos pegar os dados e sempre que esses dados mudarem vamos atualizar  
 // O Firebase vai sempre notificar a gente de mudanças.  
 referenciaUsuario.**addSnapshotListener{**documentSnapshot, erro **->** val dados = documentSnapshot?.*data* // Precisamos verificar porque os valores/colunas podem nao existir.  
 if (dados != null) {  
 val nome = dados["nome"]  
 val anoNascimento = dados["anoNascimento"]  
  
 val texto = "Seja bem vindo: $nome - $anoNascimento"  
 binding.textResultado.*text* = texto  
 }  
 **}** }  
  
}

# Agora imagina que queremos trazer a lista com todos os usuários

private fun listaComTodosOsUsuarios() {  
 /\* Dessa maneira vamos recuperar uma lista de documentos,   
 Depois vamos entrar dentro de cada um deles e vamos recuperar seus dados.  
 \*/  
   
 val referenciaLista = bancoDeDados  
 .collection("usuarios")  
  
 referenciaLista.addSnapshotListener **{** querySnapshot, error **->** // Reparar que agora estamos trazendo um query e nao um documento  
 // Porque estamos trazendo varios documentos  
  
 // Dessa forma vamos trazer uma lista de documentos  
 val listaDocuments = querySnapshot?.*documents* // Vamos guardar os dados que percorremos dentro da variavel  
 var listaResultado = ""  
 listaDocuments?.*forEach* **{** documentSnapshot **->** // percorrer essa lista de documentos  
 // Quando entramos dentro de documents, vamos ter as colunas/ DocumentSnapshot  
 /\* Atenção como estamos percorrendo um documents, agora vamos ter as colunas  
 \* E ai fazemos a mesma coisa que fizemos nos outros.  
 \*  
 \* Resumindo vamos percorrer cada um dos documentos e pegar os dados deles.\*/  
 val dados = documentSnapshot.*data* if (dados != null) {  
 val nome = dados["nome"]  
 val anoNascimento = dados["anoNascimento"]  
  
 listaResultado += "nome: $nome - nascimento: $anoNascimento \n"  
 }  
  
 **}** binding.textResultado.*text* = listaResultado  
  
 **}**}