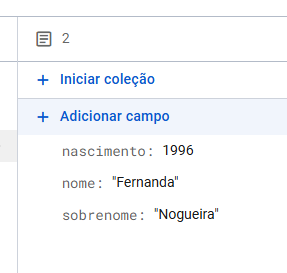
Novamente vamos criar um método para isolar a ação que vamos fazer, e vamos chamar esse método dentro de onClick

# Método para atualizar os dados

Nesse método estamos atualizando somente o sobrenome

private fun atualizarRemoverDados() {  
 */\*\*  
 \* Para atualizar ou remover precisamos selecionar o item que queremos.  
 \* Temos o usuario 1 e usuario 2  
 \* Vamos atualizar o usuario 2 "FERNANDA"  
 \*/* /\* Dados que queremos atualizar, nesse caso colocamos tudo igual, mudamos somente o sobrenome \*/  
  
 val dados = *mapOf*(  
 "nome" *to* "Fernanda",  
 "sobrenome" *to* "Nogueira",  
 "nascimento" *to* 1996  
 )  
  
 */\*\* Precisamos de uma referencia, para o dado que queremos alterar  
 \* Nesse exemplo aqui de baixo estamos pegando a referencia de onde esta os dados da Fernanda  
 \*/* val referencia = bancoDeDados  
 .collection("usuarios")  
 .document("2")  
  
  
 */\*\*  
 \* Apos pegar a referencia da Fernanda, vamos ter um objeto DocumentReference  
 \* Ai com esse objeto usamos o objeto set e passamos os novos dados.  
 \*/* referencia.set(dados)  
  
  
}



Agora vamos imaginar que eu queira remover o nascimento

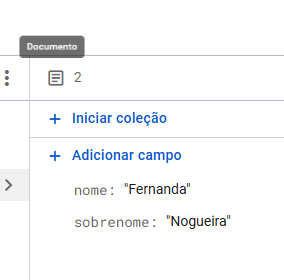
# Removendo dados / Substituindo tudo

Preciso so remover a linha que eu quero remover, e não vai ter mais nascimento

Nesse exemplo ele esta substituindo todos os dados por completo, sendo assim se você remover os outros dados ele vai remover também da coluna.

private fun atualizarRemoverDados() {  
 */\*\*  
 \* Para atualizar ou remover precisamos selecionar o item que queremos.  
 \* Temos o usuario 1 e usuario 2  
 \* Vamos atualizar o usuario 2 "FERNANDA"  
 \*/* /\* Dados que queremos atualizar, nesse caso colocamos tudo igual, mudamos somente o sobrenome \*/  
  
 val dados = *mapOf*(  
 "nome" *to* "Fernanda",  
 "sobrenome" *to* "Nogueira",  
 //"nascimento" *to* 1996 // Preciso apenas remover essa linha

E não vai ter mais nascimento  
 )  
  
 */\*\* Precisamos de uma referencia, para o dado que queremos alterar  
 \* Nesse exemplo aqui de baixo estamos pegando a referencia de onde esta os dados da Fernanda  
 \*/* val referencia = bancoDeDados  
 .collection("usuarios")  
 .document("2")  
  
  
 */\*\*  
 \* Apos pegar a referencia da Fernanda, vamos ter um objeto DocumentReference  
 \* Ai com esse objeto usamos o objeto set e passamos os novos dados.  
 \*/* referencia.set(dados)  
  
  
}



# Atualizando dados

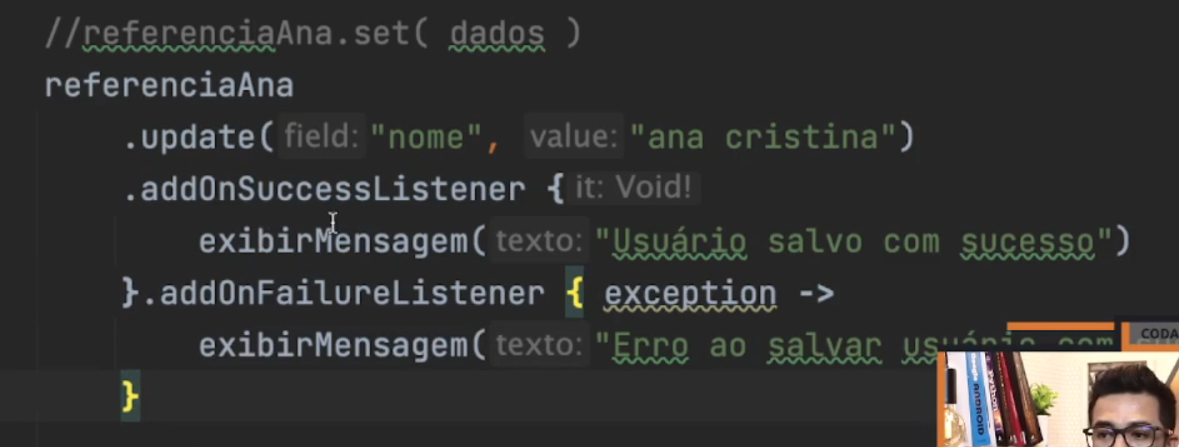
Nesse modelo aqui podemos dizer qual dado queremos atualizar, em vez de substituir todos os dados de uma vez só.

private fun atualizarRemoverDados() {  
 */\*\*  
 \* Para atualizar ou remover precisamos selecionar o item que queremos.  
 \* Temos o usuario 1 e usuario 2  
 \* Vamos atualizar o usuario 2 "FERNANDA"  
 \*/* /\* Dados que queremos atualizar, nesse caso colocamos tudo igual, mudamos somente o sobrenome \*/  
  
 val dados = *mapOf*(  
 "nome" *to* "Fernanda",  
 "sobrenome" *to* "Nogueira"  
 //"nascimento" to 1996 Removendo a linha não vai ter mas nascimento no documento da fernanda  
 )  
  
 */\*\* Precisamos de uma referencia, para o dado que queremos alterar  
 \* Nesse exemplo aqui de baixo estamos pegando a referencia de onde esta os dados da Fernanda  
 \*/* val referencia = bancoDeDados  
 .collection("usuarios")  
 .document("2")  
  
  
 */\*\*  
 \* Apos pegar a referencia da Fernanda, vamos ter um objeto DocumentReference  
 \* Ai com esse objeto usamos o objeto set e passamos os novos dados.  
 \* Dessa forma vamos substituir todos os valores, e se faltar algum valor la em cima na coluna ele vai deletar a coluna  
 \*/* //referencia.set(dados)  
  
 */\*\*  
 \* Dessa forma vamos atualizar somente o valor que queremos  
 \* Nesse exemplo vamos utilizar chave : valor.  
 \*/* referencia.update("nome", "Fernanda Ferreira")  
  
  
}

Também podemos usar os métodos para verificar se deu tudo certo em atualizar

Fazendo dessa forma

referencia.update("nome", "Fernanda Ferreira")  
 .addOnSuccessListener **{** exibirMensagem("Documento atualizado com sucesso!")  
 **}** .addOnFailureListener **{** exibirMensagem("Falha ao atualizar documento!")  
 **}**



# Removendo uma referencia por completo

Nesse exemplo estamos removendo a referencia por completo

private fun atualizarRemoverDados() {  
 */\*\*  
 \* Para atualizar ou remover precisamos selecionar o item que queremos.  
 \* Temos o usuario 1 e usuario 2  
 \* Vamos atualizar o usuario 2 "FERNANDA"  
 \*/* /\* Dados que queremos atualizar, nesse caso colocamos tudo igual, mudamos somente o sobrenome \*/  
  
 val dados = *mapOf*(  
 "nome" *to* "Fernanda",  
 "sobrenome" *to* "Nogueira"  
 //"nascimento" to 1996 Removendo a linha não vai ter mas nascimento no documento da fernanda  
 )  
  
 */\*\* Precisamos de uma referencia, para o dado que queremos alterar  
 \* Nesse exemplo aqui de baixo estamos pegando a referencia de onde esta os dados da Fernanda  
 \*/* val referencia = bancoDeDados  
 .collection("usuarios")  
 .document("2")  
  
  
 */\*\*  
 \* Apos pegar a referencia da Fernanda, vamos ter um objeto DocumentReference  
 \* Ai com esse objeto usamos o objeto set e passamos os novos dados.  
 \* Dessa forma vamos substituir todos os valores, e se faltar algum valor la em cima na coluna ele vai deletar a coluna  
 \*/* //referencia.set(dados)  
  
 */\*\*  
 \* Dessa forma vamos atualizar somente o valor que queremos  
 \* Nesse exemplo vamos utilizar chave : valor.  
 \*/*// referencia.update("nome", "Fernanda Ferreira")  
// .addOnSuccessListener {  
// exibirMensagem("Documento atualizado com sucesso!")  
// }  
// .addOnFailureListener {  
// exibirMensagem("Falha ao atualizar documento!")  
// }  
  
  
 */\*\*  
 \* Dessa forma também podemos remover a referencia no caso remover todo o documento  
 \*/* referencia.delete()  
  
  
 }