

Laboratório nº: 4

Data: domingo, 10 de maio de 2020

Turma: 61D

Grupo: **06**

Número e nome dos alunos presentes:

Número	Nome
43861	Francisco Chicharro
43874	João Florentino
Click or tap here to enter text.	Click or tap here to enter text.

1. Objetivo da atividade (descrição por palavras simples do que entendeu como objetivo da atividade);
➔ Utilizar a API Java para acesso ao serviço Google Firestore.

2. Indicação das tecnologias e as ferramentas (*tools*) utilizadas;

- ➔ Serviço Firestore da Google Cloud Platform;
- ➔ IntelliJ IDEA;

3. Descrição da arquitetura das partes (componentes) envolvidas, com eventuais diagramas:

A partir de um ficheiro CSV que continha dados sobre uma ocupação temporária de espaços na realização de eventos na cidade de Lisboa, implementou-se uma aplicação Java que, a partir da utilização do código fornecido pelos docentes, fosse possível criar uma coleção de documentos na base de dados do Firestore usando a API Java Firestore. Este ficheiro continha um conjunto de parâmetros sendo que o tratamento de dados foi previamente implementada a partir de classes.

Assim, alterou-se o ficheiro pom.xml para que este tivesse a dependência do Firestore, sendo possível utilizar as bibliotecas que esta dependência disponibilizava. De seguida, o grupo obteve por criar duas classes: OcupacaoTemporariaStorage e FirestoreOperations.

A classe OcupacaoTemporaria representa o nosso programa principal, onde é semelhante do último relatório, se faria a ligação com a GCP a partir da variável ambiente GoogleCredentials. Fez-se então um menu que disponibiliza várias funcionalidades a serem implementadas no projeto

A classe FirestoreOperations, também é semelhante do último relatório, é a classe onde estão implementadas todas as funcionalidades necessárias para a criação da base de dados. Nesta classe colocou-se também as funções fornecidas no enunciado que permitiam processar o ficheiro CSV.

Começou-se então por criar um documento tendo este uma coleção ao qual se deu o nome de OcupacaoTemporaria:

```
##### MENU #####
Options for Google Firestore Operations:
0: Inserir documentos
1: Mostrar conteúdo por ID
2: Apagar campo do documento
3: Interrogação simples
4: Interrogação composta
.....
99: Exit
Enter an Option:0
Qual o path do ficheiro?
D:\Documentos\Universidade\6 Semestre\Computacao na Nuvem\Trabalhos_CN\Lab4\OcupacaoEspacosPublicos.csv
Qual o nome da collection?
OcupacaoTemporaria
Documento inserido com sucesso!!
```

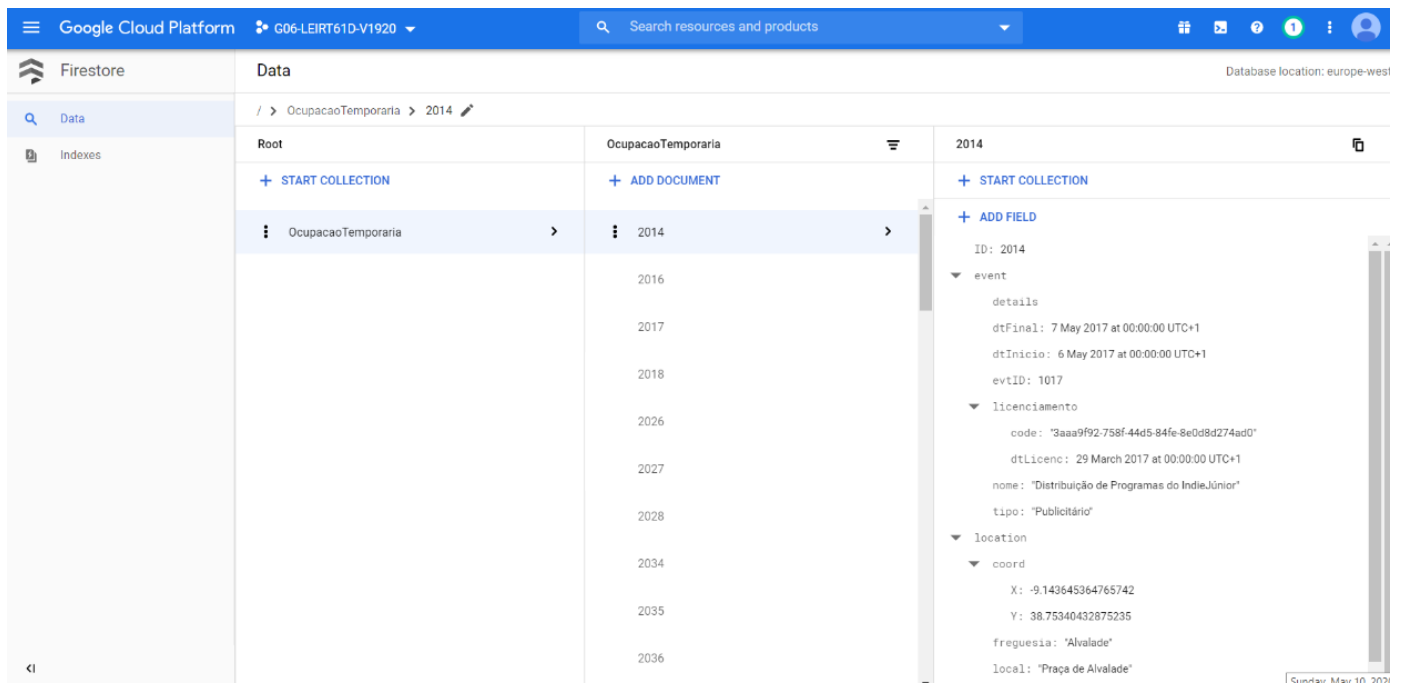
Assim a primeira funcionalidade solicitada foi : Apresentar o conteúdo de um documento a partir de um ID:

É criado um Document Reference que contém a informação do nome da coleção e do ID do documento onde a partir do método get() é possível mostrar um documento com as informações que são passadas pelo utilizador, apresentando assim o tipo de documento procurado:

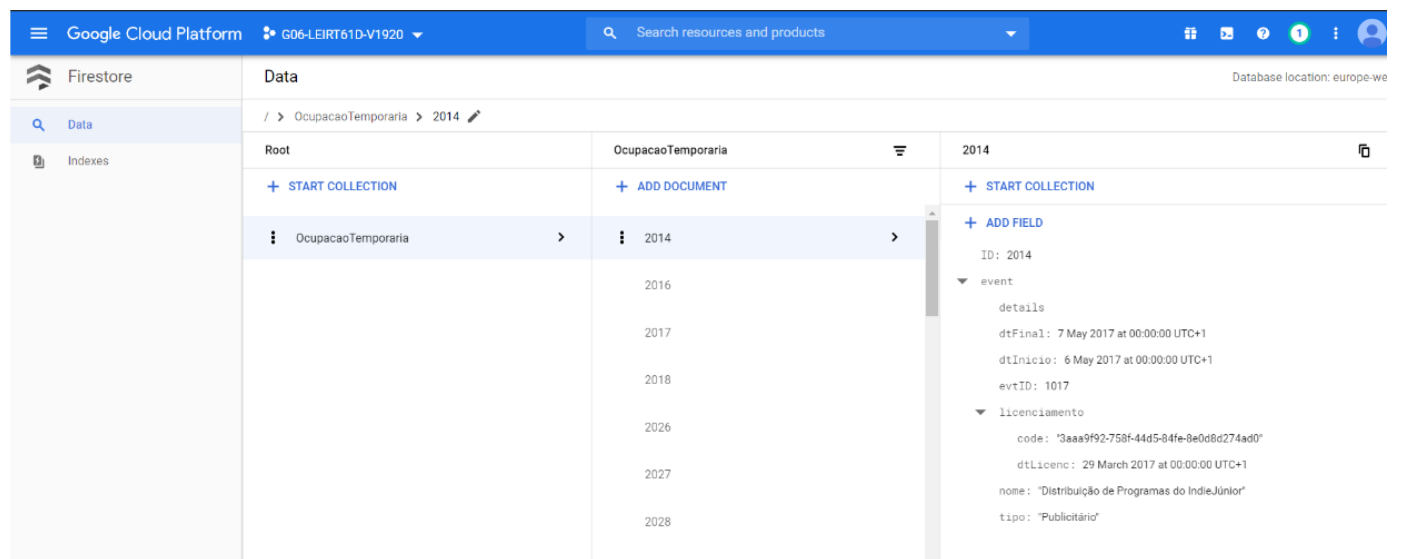
```
Run: OcupacaoTemporariaStorage
"D:\Program Files\Java\jdk1.8.0_241\bin\java.exe" ...
Current Project ID:g05-leirt61d
##### MENU #####
Options for Google Firestore Operations:
0: Inserir documentos
1: Mostrar conteúdo por ID
2: Apagar campo do documento
3: Interrogação simples
4: Interrogação composta
.....
99: Exit
Enter an Option:1
Qual o nome da Collection?
OcupacaoTemporaria
Qual o ID do documento?
2014
Document Data: {location={coord={X=-9.143645364765742, Y=38.75340432875235}, local=Praça de Alvalade, point=GeoPoint { latitude=-9.143645364765742, long
```

A seguinte funcionalidade pedida pretendia apagar um campo de um documento, dado um ID e o nome do campo. Assim, seguindo a mesma estratégia do Document Reference anterior, criou-se um Hash Map de nome updates que, através do método put() e dado o nome do campo a ser apagado , fosse possível fazer delete do mesmo. Assim, docRef atualizaria de seguida o seu documento usando o update():

Firestore antes de eliminar o campo location:



Firestore depois de eliminado o campo location:



De seguida, foi solicitada a realização de interrogação simples para obter todos os documentos de uma determinada freguesia.

Usando a mesma estratégia do documento, criou-se um FieldPath que conseguisse aceder ao parâmetro location e consequentemente á freguesia. Com este FieldPath, criou-se então uma Query onde, através do método whereEqualTo(), foi possível procurar todos os documentos que tivessem a freguesia passada pelo utilizador:

```

##### MENU #####
Options for Google Firestore Operations:
0: Inserir documentos
1: Mostrar conteúdo por ID
2: Apagar campo do documento
3: Interrogação simples
4: Interrogação composta
.....
99: Exit
Enter an Option:3
Qual o nome da Collection?
OcupacaoTemporaria
Qual o nome da freguesia?
Alvalade
2014
2113

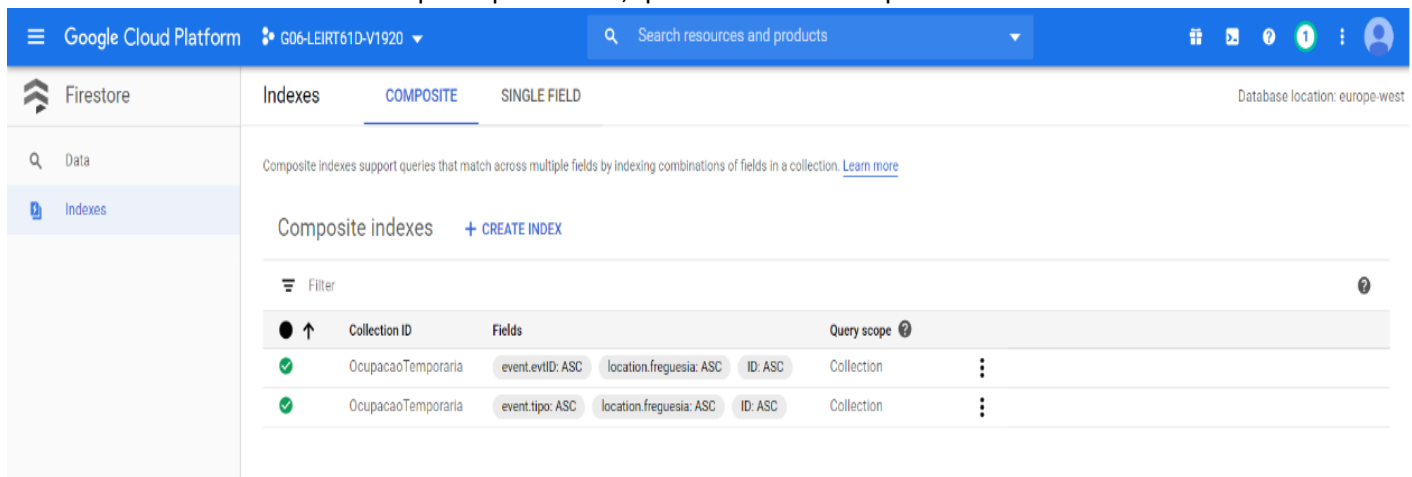
```

Por último, realizou-se uma interrogação composta onde os critérios seriam os seguintes:

- i. Maior que determinado ID
- ii. Igual a uma determinada freguesia
- iii. Igual a um determinado tipo de evento

À semelhança da funcionalidade anterior, esta última permite agora fazer mais do que uma interrogação, tomando assim o nome de interrogação composta. Assim, foram criados dois FieldPaths, um que acesse ao campo location e outro ao campo event, obtendo a freguesia e o tipo de evento respetivamente. Assim, criou-se uma query que procurava os docRef a partir de um determinado ID pelo método whereGreaterThan(), seguido por dois métodos whereEqualTo() um para a freguesia e outro para o tipo de Evento.

Foi necessário criar um índice composto para a GCP, que satisfizesse este pedido:



Obtendo assim todos os documentos da coleção que satisfizessem a query:

```
##### MENU #####
Options for Google Firestore Operations:
0: Inserir documentos
1: Mostrar conteúdo por ID
2: Apagar campo do documento
3: Interrogação simples
4: Interrogação composta
.....
99: Exit
Enter an Option:4
Qual o nome da Collection?
OcupacaoTemporaria
Qual o nome do documento?
2018
Qual o nome da freguesia?
Belém
Tipo do evento?
Desportivo
2028:Doc:{location={coord={X=-9.206537712279301, Y=38.69711029981643}, local=Praça do Império, point=GeoPoint { latitude=-9.206537712279301, longitude=38.69711029981643 }, fregue
```

4. Resumo dos problemas encontrados e as soluções aplicadas:

- ➔ O teste da interrogação composta não teve sucesso imediato, uma vez que foi necessário criar índices compostos na GCP, sendo esta situação rapidamente resolvida utilizando a secção Indexes do Firestore.

5. Indicação se a solução final é executável e demonstrável

A solução final é executável para todas as funcionalidades implementadas assim como demonstrável.

6. Conclusões e lições aprendidas

Com este trabalho aprendeu-se a implementar funcionalidades que permitiam lidar com o Firestore da GCP, sendo possível manipular um documento fazendo várias solicitações á base de dados do serviço.

7. Auto-avaliação qualitativa por parte dos alunos

Bom

