Universidad Europea

EXAMEN FINAL BIG DATA I

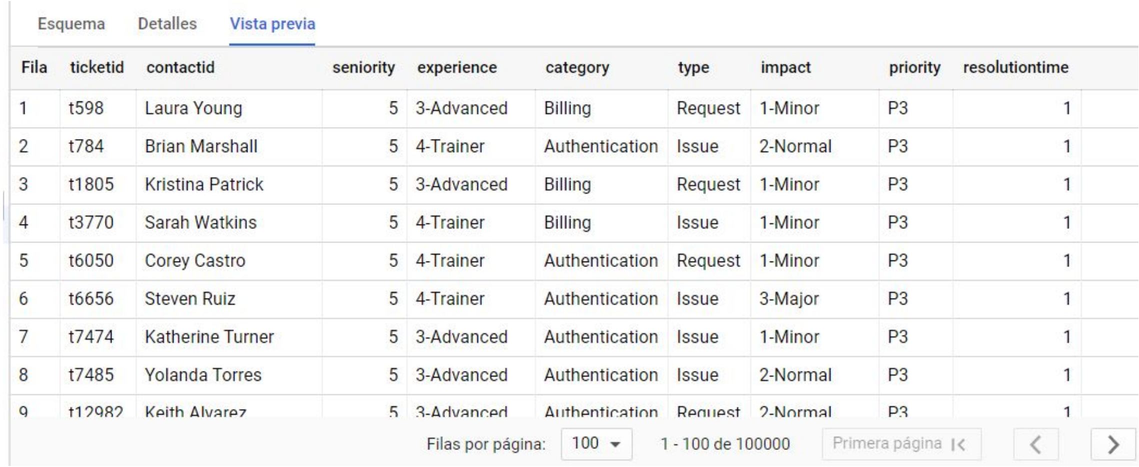
Sofía Corral Caballero

Número expediente: 21846911

EJERCICIO 3:

a) ¿Cuántas observaciones tiene el dataset?

Tiene 100.000 observaciones



Fila	ticketid	contactid	seniority	experience	category	type	impact	priority	resolutiontime
1	t598	Laura Young	5	3-Advanced	Billing	Request	1-Minor	P3	1
2	t784	Brian Marshall	5	4-Trainer	Authentication	Issue	2-Normal	P3	1
3	t1805	Kristina Patrick	5	3-Advanced	Billing	Request	1-Minor	P3	1
4	t3770	Sarah Watkins	5	4-Trainer	Billing	Issue	1-Minor	P3	1
5	t6050	Corey Castro	5	4-Trainer	Authentication	Request	1-Minor	P3	1
6	t6656	Steven Ruiz	5	4-Trainer	Authentication	Issue	3-Major	P3	1
7	t7474	Katherine Turner	5	3-Advanced	Authentication	Issue	1-Minor	P3	1
8	t7485	Yolanda Torres	5	3-Advanced	Authentication	Issue	2-Normal	P3	1
9	t12982	Keith Alvarez	5	3-Advanced	Authentication	Request	2-Normal	P3	1

Filas por página: 100 ▾ 1 - 100 de 100000

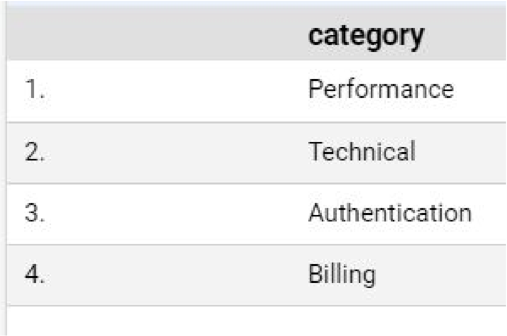
b) ¿Cuántas categorías tiene el atributo category?

Ejecutamos la siguiente query:

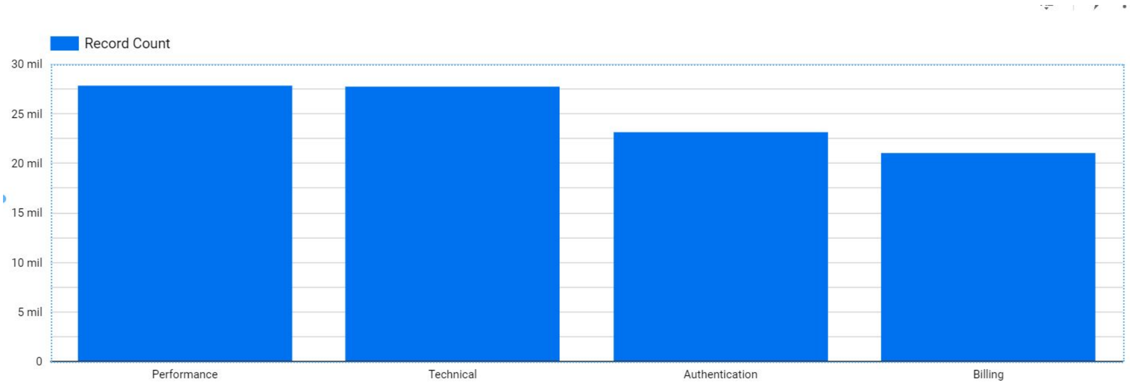
```
SELECT*FROM `examensofia.dataset.tabla`
```

Abrimos dataStudio:

Hay 4 categorías diferentes (Performance, Technical, Authentication y Billing)

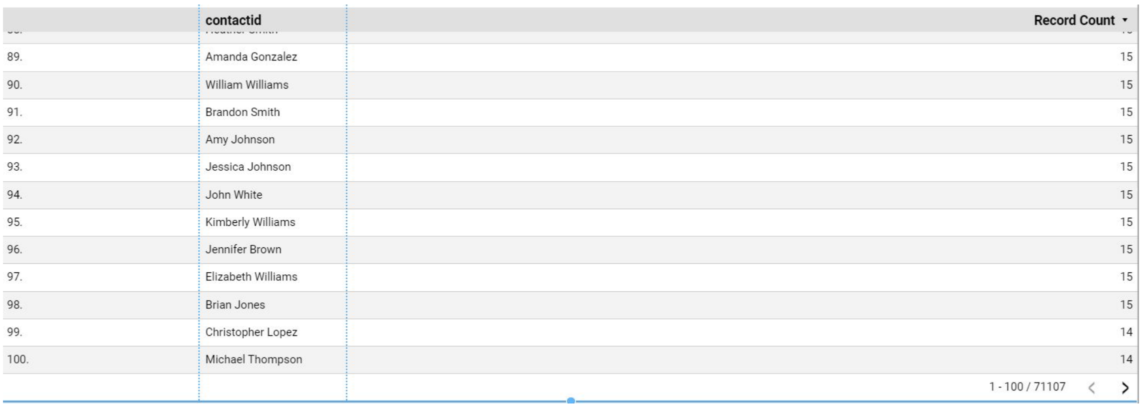


	category
1.	Performance
2.	Technical
3.	Authentication
4.	Billing



c) ¿Cuántos empleados diferentes hay en el dataset?

Hay 71.107 empleados diferentes



	contactid	Record Count ▾
89.	Amanda Gonzalez	15
90.	William Williams	15
91.	Brandon Smith	15
92.	Amy Johnson	15
93.	Jessica Johnson	15
94.	John White	15
95.	Kimberly Williams	15
96.	Jennifer Brown	15
97.	Elizabeth Williams	15
98.	Brian Jones	15
99.	Christopher Lopez	14
100.	Michael Thompson	14

1 - 100 / 71107 < >

EJERCICIO 4:

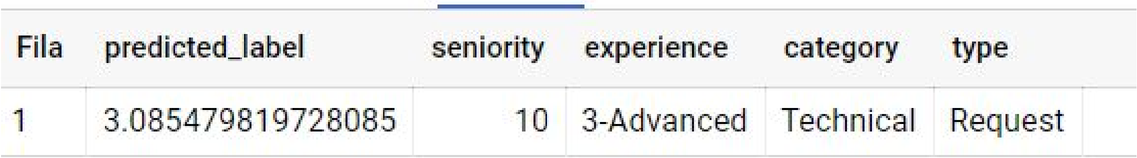
ANTES DE RESPONDER A LAS PREGUNTAS HAY QUE HACER:

```
CREATE OR REPLACE MODEL `examensofia.dataset.predicion` OPTIONS (model_type =
'linear_reg') AS

SELECT

seniority, experience, category,type,

resolutiontime as label FROM

`examensofia.dataset.tabla`



WITH eval_table AS (SELECT seniority, experience, category,type,resolutiontime
as label

FROM `examensofia.dataset.tabla` )

SELECT *

FROM ML.EVALUATE(MODEL `examensofia.dataset.predicion`, TABLE eval_table)
```

a) Para un ticket : nivel-> 10 seniority, '3-Advanced' en experience, 'Technical' en category, 'Request' como type. CUANTO DIAS ?

Query:

```
WITH pred_table AS (SELECT 10 as seniority, '3-Advanced' as experience,
'Technical' as category, 'Request' as type)

SELECT * FROM ML.PREDICT(MODEL `examensofia.dataset.predicion`, TABLE
pred_table)
```



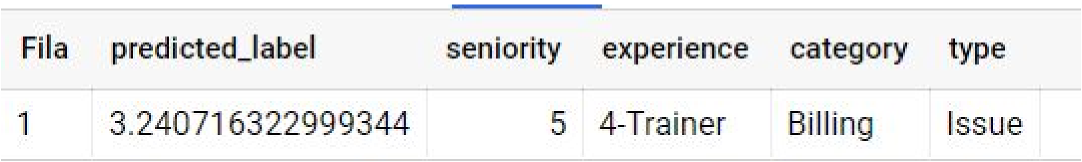
Fila	predicted_label	seniority	experience	category	type
1	3.085479819728085	10	3-Advanced	Technical	Request

3.0854..., Como es más de 3, sería 4

b) Para un ticket : nivel-> 5 seniority, ' 4-Trainer' en experience, 'Billing' en category, 'Issue' como type. CUANTO DIAS ?

Query:

```
WITH pred_table AS (SELECT 5 as seniority, '4-Trainer' as experience,
'Billing' as category, 'Issue' as type)

SELECT * FROM ML.PREDICT(MODEL `examensofia.dataset.predicion`, TABLE
pred_table)
```

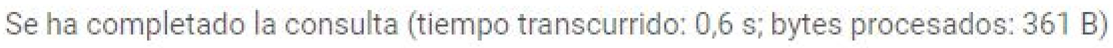


Fila	predicted_label	seniority	experience	category	type
1	3.240716322999344	5	4-Trainer	Billing	Issue

3.2407..., como es más de 3, sería 4

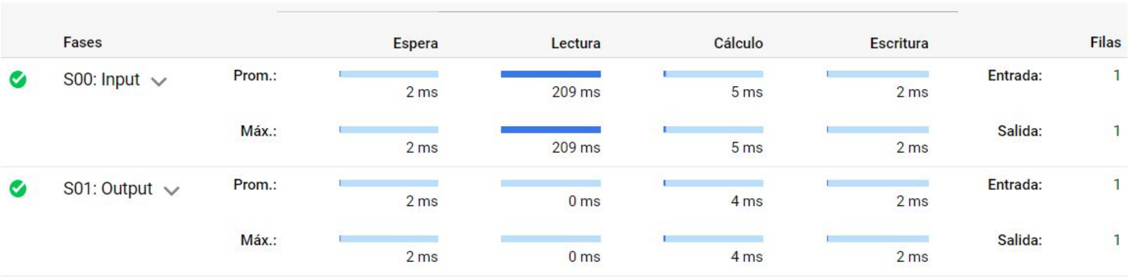
c) Cuánto tiempo tardo el cluster de bigquery en responder a la pregunta del punto 4a ?

En esta captura se ve el tiempo



Se ha completado la consulta (tiempo transcurrido: 0,6 s; bytes procesados: 361 B)

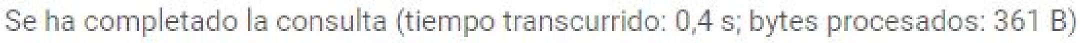
En esta otra se ve el proceso que realiza el cluster



	Fases		Espera	Lectura	Cálculo	Escritura		Filas
✓	S00: Input	Prom.:	2 ms	209 ms	5 ms	2 ms	Entrada:	1
		Máx.:	2 ms	209 ms	5 ms	2 ms	Salida:	1
✓	S01: Output	Prom.:	2 ms	0 ms	4 ms	2 ms	Entrada:	1
		Máx.:	2 ms	0 ms	4 ms	2 ms	Salida:	1

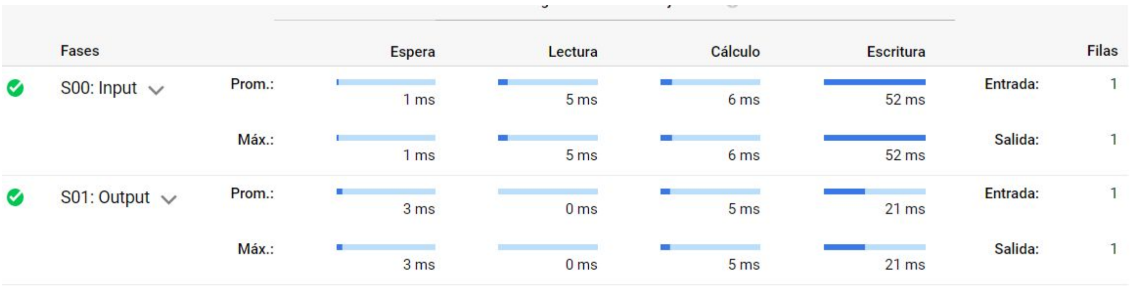
d) Cuántos megabytes uso el cluster de bigquery en responder a la pregunta del punto 4b?

Se ve el tiempo y el espacio que ocupa



Se ha completado la consulta (tiempo transcurrido: 0,4 s; bytes procesados: 361 B)

Aquí se ve lo que hace el cluster:

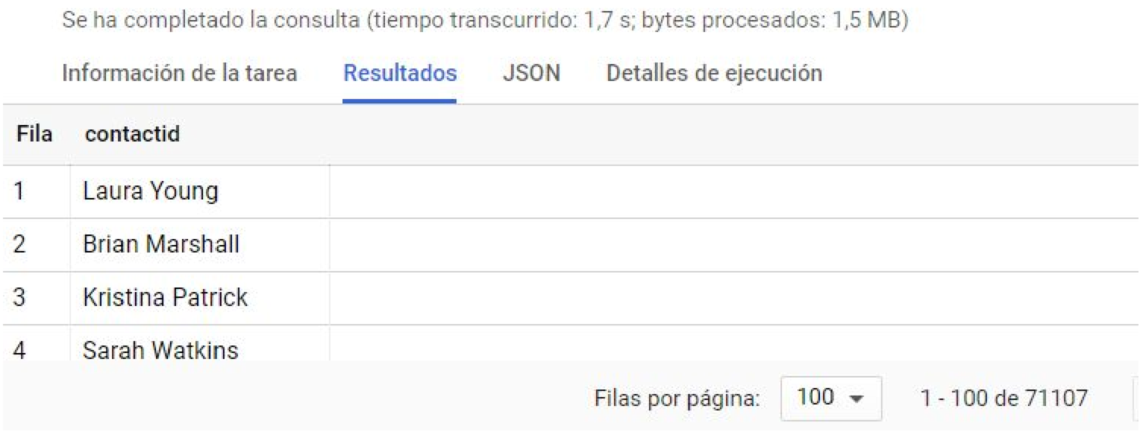


	Fases		Espera	Lectura	Cálculo	Escritura		Filas
✓	S00: Input	Prom.:	1 ms	5 ms	6 ms	52 ms	Entrada:	1
		Máx.:	1 ms	5 ms	6 ms	52 ms	Salida:	1
✓	S01: Output	Prom.:	3 ms	0 ms	5 ms	21 ms	Entrada:	1
		Máx.:	3 ms	0 ms	5 ms	21 ms	Salida:	1

e) Escriba la Query que uso para responder el punto 3C, así como el tiempo que tardo y los megabytes usados ?

Query:

```
SELECT DISTINCT contactid FROM `dataset.tabla`
```

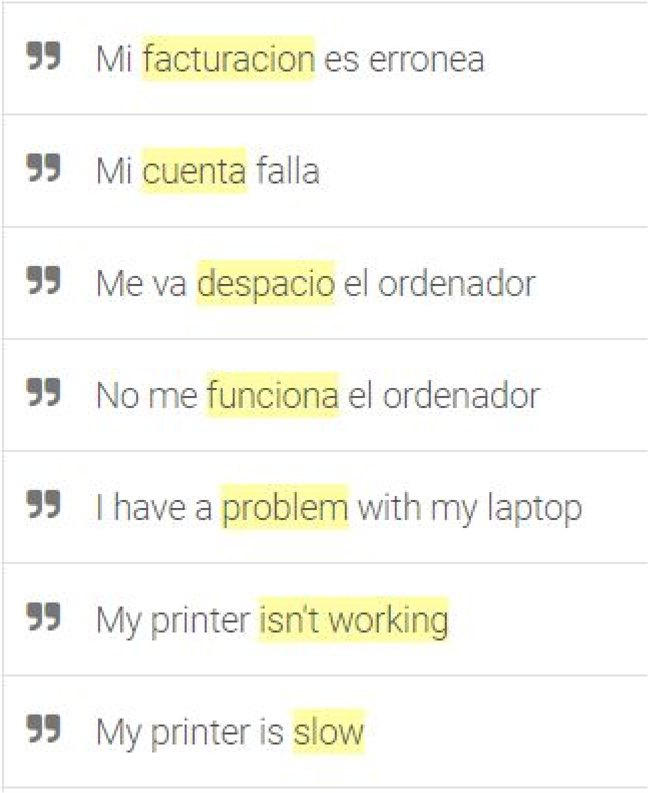


Se ha completado la consulta (tiempo transcurrido: 1,7 s; bytes procesados: 1,5 MB)

Fila	contactid
1	Laura Young
2	Brian Marshall
3	Kristina Patrick
4	Sarah Watkins

Filas por página: 100 1 - 100 de 71107

PREGUNTA 7:



Mi facturacion es erronea

Mi cuenta falla

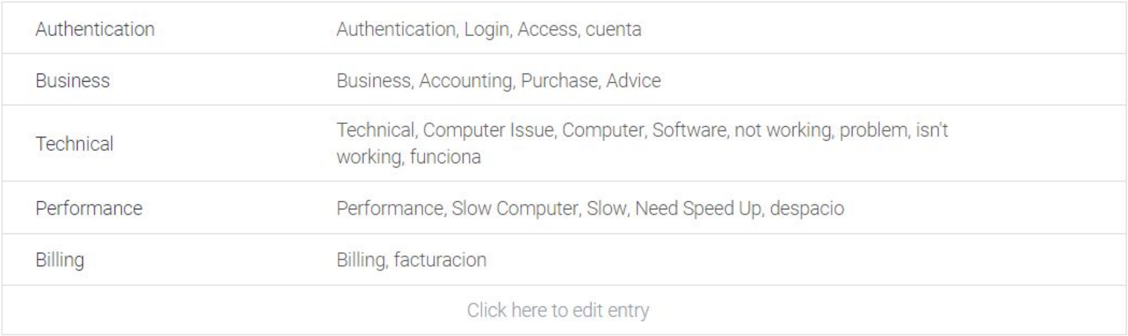
Me va despacio el ordenador

No me funciona el ordenador

I have a problem with my laptop

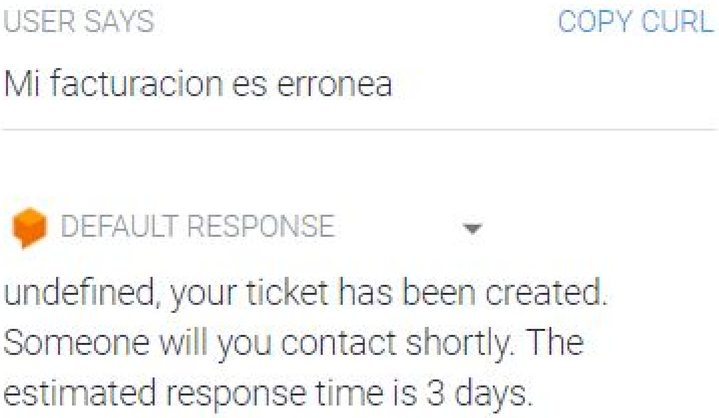
My printer isn't working

My printer is slow



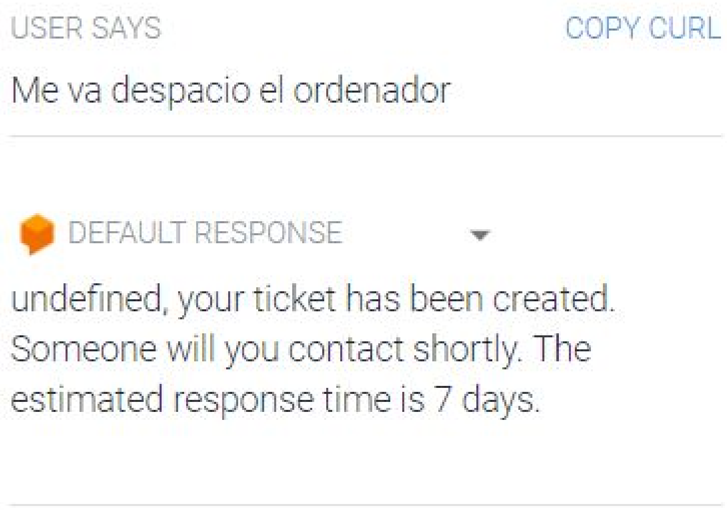
Authentication	Authentication, Login, Access, cuenta
Business	Business, Accounting, Purchase, Advice
Technical	Technical, Computer Issue, Computer, Software, not working, problem, isn't working, funciona
Performance	Performance, Slow Computer, Slow, Need Speed Up, despacio
Billing	Billing, facturacion

RESPONDER : CUANTO TIEMPO TARDA UNA PREDICCION PARA UN EXPERTO NIVEL 7.(Estas respuestas debe obtenerlas del diálogo con el chat Bot)



Mi facturacion es erronea

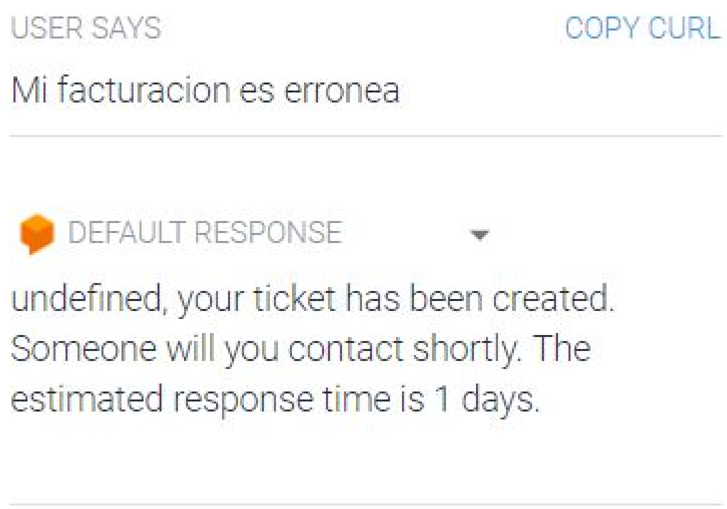
undefined, your ticket has been created. Someone will you contact shortly. The estimated response time is 3 days.



Me va despacio el ordenador

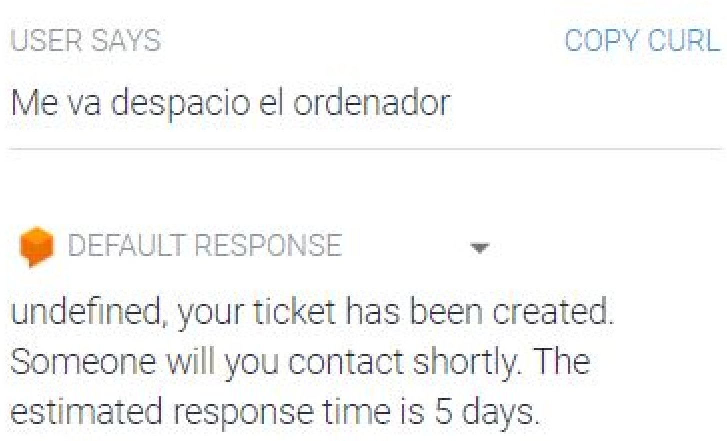
undefined, your ticket has been created. Someone will you contact shortly. The estimated response time is 7 days.

RESPONDER : CUANTO TIEMPO TARDA UNA PREDICCION PARA UN EXPERTO NIVEL 13, y categoría TECHNICAL . (Estas respuestas debe obtenerlas del diálogo con el chat Bot)



Mi facturacion es erronea

undefined, your ticket has been created. Someone will you contact shortly. The estimated response time is 1 days.



Me va despacio el ordenador

undefined, your ticket has been created. Someone will you contact shortly. The estimated response time is 5 days.

Los tiempos en este apartado mejoran, ya que estamos en nivel experto 13

Index.js

```
// See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
// for Dialogflow fulfillment library docs, samples, and to report issues
'use strict';


const functions = require('firebase-functions');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Card, Suggestion} = require('dialogflow-fulfillment');
const BIGQUERY = require("@google-cloud/bigquery");
const BIGQUERY_CLIENT = new BIGQUERY({projectId: "examensofia"});


process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements




exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
  const agent = new WebhookClient({ request, response });
  console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));


  function welcome(agent) {
    agent.add(`Welcome to my agent!`);
  }
```

```
function fallback(agent) {
  agent.add(`I didn't understand`);
  agent.add(`I'm sorry, can you try again?`);
}


function ticketCollection(agent) {
  // Capture Parameters from the Current Dialogflow Context
  console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
  const OUTPUT_CONTEXTS = request.body.queryResult.outputContexts;
  const EMAIL = OUTPUT_CONTEXTS[OUTPUT_CONTEXTS.length - 1].parameters["email.original"];
  const ISSUE_CATEGORY = OUTPUT_CONTEXTS[OUTPUT_CONTEXTS.length - 1].parameters.category;
  const ISSUE_TEXT = request.body.queryResult.queryText;



  // The SQL Query to Run
  //PREGUNTA 8
//const SQLQUERY = `WITH eval_table AS ( SELECT  7 as seniority,'3-Advanced'as experience,
  //@category as category, 'Request' as type)
//SELECT cast(predicted_label as INT64) as predicted_label
//FROM ML.PREDICT(MODEL dataset.predicion, TABLE eval_table)`;
```

```
 //PREGUNTA 9


  const SQLQUERY = `WITH eval_table AS ( SELECT  13 as seniority,'3-Advanced'as experience,
 @category as category, 'Request' as type)
SELECT cast(predicted_label as INT64) as predicted_label
FROM ML.PREDICT(MODEL dataset.predicion, TABLE eval_table)`;


 const OPTIONS = {
  query: SQLQUERY,
  // Location must match that of the dataset(s) referenced in the query.
  location: "US",
  params: {
   category: ISSUE_CATEGORY
  }
 };
 return BIGQUERY_CLIENT.query(OPTIONS)
  .then(results => {
   //Capture results from the Query
   console.log(JSON.stringify(results[0]));
   const QUERY_RESULT = results[0];
   const ETA_PREDICTION = QUERY_RESULT[0].predicted_label;


   //Format the Output Message
```

```
    agent.add( EMAIL + ", your ticket has been created. Someone will you contact shortly. " +
      " The estimated response time is " + ETA_PREDICTION  + " days."
    );


    agent.setContext({
     name: "submitticket-collectname-followup",
     lifespan: 2
    });
   })
   .catch(err => {
    console.error("ERROR:", err);
   });
}


// // Uncomment and edit to make your own intent handler
// // uncomment `intentMap.set('your intent name here', yourFunctionHandler);`
// // below to get this function to be run when a Dialogflow intent is matched
// function yourFunctionHandler(agent) {
//   agent.add(`This message is from Dialogflow's Cloud Functions for Firebase editor!`);
//   agent.add(new Card({
//      title: `Title: this is a card title`,
//      imageUrl:
'https://developers.google.com/actions/images/badges/XPM_BADGING_GoogleAssistant_VER.png',
//      text: `This is the body text of a card.  You can even use line\n  breaks and emoji! 💁`,

//      buttonText: 'This is a button',

//      buttonUrl: 'https://assistant.google.com/'

//    })

//  );

//   agent.add(new Suggestion(`Quick Reply`));

//   agent.add(new Suggestion(`Suggestion`));

//   agent.setContext({ name: 'weather', lifespan: 2, parameters: { city: 'Rome' }});

// }



// // Uncomment and edit to make your own Google Assistant intent handler

// // uncomment `intentMap.set('your intent name here', googleAssistantHandler);`

// // below to get this function to be run when a Dialogflow intent is matched

// function googleAssistantHandler(agent) {

//   let conv = agent.conv(); // Get Actions on Google library conv instance

//   conv.ask('Hello from the Actions on Google client library!') // Use Actions on Google library

//   agent.add(conv); // Add Actions on Google library responses to your agent's response

// }

// // See https://github.com/dialogflow/fulfillment-actions-library-nodejs

// // for a complete Dialogflow fulfillment library Actions on Google client library v2 integration sample



// Run the proper function handler based on the matched Dialogflow intent name

// Run the proper function handler based on the matched Dialogflow intent name
```

```
  let intentMap = new Map();

  intentMap.set("Default Welcome Intent", welcome);

  intentMap.set("Default Fallback Intent", fallback);

  intentMap.set("Submit Ticket - Issue Category", ticketCollection);

  agent.handleRequest(intentMap);



});
```

Packaje.json

```
{
  "name": "dialogflowFirebaseFulfillment",
  "description": "This is the default fulfillment for a Dialogflow agents using Cloud Functions for Firebase",
  "version": "0.0.1",
  "private": true,
  "license": "Apache Version 2.0",
  "author": "Google Inc.",
  "engines": {
    "node": "10"
  },
  "scripts": {
    "start": "firebase serve --only functions:dialogflowFirebaseFulfillment",
    "deploy": "firebase deploy --only functions:dialogflowFirebaseFulfillment"
  },
  "dependencies": {
    "actions-on-google": "^2.2.0",
    "firebase-admin": "^5.13.1",
    "firebase-functions": "^2.0.2",
    "dialogflow": "^0.6.0",
    "dialogflow-fulfillment": "^0.5.0",
    "@google-cloud/bigquery": "^1.3.0"
  }

}
```