

Project scope

1. Make an accurate estimate of hosting costs
2. Host Quick, Draw! database on Google Cloud Platform
3. Provide installation manual and use guide for Google Cloud database
4. Create API for inquiries to the self-hosted database
5. Provide API use manual.

Project requirements

The project must include a RESTful API, Database that includes the drawings of the Quickdraw project (which will be hosted on google cloud), which will be used to guess what drawing the user makes, the endpoint must be hosted on our linux centos server.

Service need:

Currently the demo https://gameconnections.net/demos_juegoscr/dibuja/inicio.php is directly connected to the quick draw api. You need to have your own API installed on our server and that this api directly consult a database. Example of the data:

<https://github.com/googlecreativelab/quickdraw-dataset> hosted on google cloud see recommendations on how to obtain your own quick draw database

<https://github.com/googlecreativelab/quickdraw-component/blob/master/api/README.md> ..

Purpose and scope of the project:

The purpose of this development is to provide its own restful API and have its own base in google cloud, which will serve as a connection, with the development that will be carried out in the gift factory portal, in order to guess What drawing the user makes on a digital whiteboard.

Obstacles and technical challenges:

- Not having knowledge of database management in google cloud specifically in the management of the quick draw database installed in an own hosting service.
- Not having the necessary knowledge to develop an api that can be installed on a local server and that it can receive parameters to connect with the google cloud database and return the result of what drawing the user is making.

Requirements:

Functional requirements:

Primary

1. Restful API receives POST JSON data and returns JSON response
2. Restful API must point to database hosted on google cloud (quick draw database copy).
3. It is expected to send 1500 requests per minute and must interact with the database that has an approximate weight of 191.63 GB

Technical requirements:

Primary

4. API developed in Python / Tensorflow installed on a local server.
5. The database must be the one offered by the Quick Draw game, it must be installed in your own hosting account in google cloud.

Secondary

6. Based on the API that is currently being used.
7. Currently the api response is returned in English, see the possibility that the response is returned in Spanish.

Regulatory Requirements:

Primary

8. We must not store any user generated images.

Usability requirements:

Primary

9. Manual for use and detail of the API in Spanish.
10. Google cloud base user, installation and configuration manual in Spanish.

Project limitations : Time limitations: Final delivery date: Friday, August 20, 2021

Cost limitations: This development must not exceed 100 hours of development.

Documentation:

Technical documentation:

- * API documentation: The documentation must include, use and management of API, as well as details of the processes carried out within the API in case any subsequent modification is required.
- * Database Documentation: The documentation must include a list of drawings that can be consulted, installation information, details of use within Google Cloud and a manual of possible errors and how to solve them so that the database is always working correctly.

Documentation for the end user:

- * Include a description of the terms of use or legal description of the use of data that is pertinent to share with the end user.

References and related documents:

Summary of what is needed:

- 1.- A copy of this REST POST api

<https://inputtools.google.com/requesttime=handwriting&app=quickdraw&dbg=1&cs=1&oe=UTF-8>

This should be installed on our local server.

- 2.- Upload a copy of the Quick draw database to our google cloud account.
3. The API must consult the database that will be uploaded in our google cloud account.

We currently have a development that is based on the following:

Currently REST api is consulted via POST

data is sent via JSON

443237304688,203.60903930664062,205.40451049804688,206.21337890625,206.21337890625,206.21337890625,206.21337890625,205.82846069335938,204.923095703125,204.2403564453125,201.69784545898438,199.99853515625,198.17138671875,196.0472412109375,193.2352294921875,191.86981201171875,188.89599609375,184.74398803710938,181.77020263671875,178.14862060546875,173.82333374023438,169.498046875,165.17276000976562,160.94293212890625,152.04110717773438,149.65921020507812,145.81451416015625,143.1014404296875,140.55245971679688,138.27313232421875,136.2882080078125,134.86575317382812,132.75143432617188],
[0,163.09999990463257,170.89999985694885,179.79999995231628,191.89999985694885,196.89999985694885,207.89999985694885,223.89999985694885,229.09999990463257,236.89999985694885,249,261.7999999523163,269.89999985694885,277.7999999523163,286.7999999523163,297.89999985694885,311.2999999523163,323,338.89999985694885,355.09999990463257,360.19999980926514,367.89999985694885,377,388.09999990463257,404.2999999523163,418,428.7999999523163,441.89999985694885,449.7999999523163,458.7999999523163,469.89999985694885,486.19999980926514,491.89999985694885,498.89999985694885,507.7999999523163,518.7999999523163,523.7999999523163,531.7999999523163,539.8999998569489,549,560.2999999523163,573,581.8999998569489,590,601.0999999046326,613.8999998569489,622,634.1999998092651,650.0999999046326,667,683,695.7999999523163,704.7999999523163,711.7999999523163,723.7999999523163,728.7999999523163,736.8999998569489,744.8999998569489,753,765,770.0999999046326,780.8999998569489,786,793.8999998569489,802.8999998569489,811,818.7999999523163,826.7999999523163,834.8999998569489,843.0999999046326,855.0999999046326,868,875.7999999523163,884.1999998092651,895.8999998569489,901.0999999046326,911.8999998569489,918,929.0999999046326,944.8999998569489]]]]}}

Response received via json

```
[
  "SUCCESS",
  [
    [
      "89e8e01d30b39225",
      [
        "circle",
        "moon",
        "blueberry",
        "planet",
        "potato",
        "hula hoop",
        "comet",
        "watermelon",
      ]
    ]
  ]
]
```

```

        "necklace",
        "ring",
        "bracelet",
        "cookie",
        "frying pan",
        "hockey puck",
        "fan",
        "unicycle",
        "panda",
        "basketball",
        "soccer ball",
        "computer mouse"
    ],
    [],
    {
        "debug_info": "SCORESINKS: [[\" circle \", 0.0457668], [\" moon \",
3.13367], [\" blueberry \", 7.08955], [\" planet \", 8.96333], [\" potato \", 9.7525], [\"
hula hoop \", 9.80241], [\" comet \", 10.881], [\" watermelon \", 11.2959], [\" necklace
\", 11.5493], [\" ring \", 11.9423], [\" bracelet \", 12.0853], [\" cookie \", 12.3329], [\"
frying pan \", 12.5161], [\" hockey puck \", 12.5342] , [\" fan \", 12.7543], [\" unicycle
\", 12.7924], [\" panda \", 12.9175], [\" basketball \", 12.9587], [\" soccer ball \", 13.113
], [\" computer mouse \", 13.4308]] Combiner: 4.7ms Service_Recognize: 4.9ms ",
        " is_html_escaped ":false
    }
]
],
"",
{
    "app_info": "quickdraw",
    "bns": "/bns/vj/borg/vj/bns/i18n-input-prod/input-frontend.ife/3",
    "decode_latency_in_ms": 5.554,
    "engine_info": [
        {
            "engine_call_latency_msec": 5,
            " engine_compute_latency_msec ": 5,
            " engine_name ":" handwriting | / bns / wn / borg / wn / bns / handwriting
/ recoservice, / bns / is / borg / is / bns / handwriting / recoservice, / bns / ld / borg / ld
/ bns / handwriting / recoservice, / bns / pg / borg / pg / bns / handwriting /
recoservice, / bns / th / borg / th / bns / handwriting / recoservice "
        }
    ],
    " experiment_id ": [],
    "is_valid_request_data ": true,
    "original_input_method_name": "handwriting",

```

```
    "plugin_name": "handwriting_plugin",  
    "request_serve_latency_in_ms": 5,  
    "rewritten_input_method_name": "und-t-i0-handwrit"  
  }  
]
```

What interests us is to have the api locally and the api Consult our google cloud service directly with a copy of the quick draw database.

We are interested in making it as similar as possible so that when integrating the api into our development, many changes do not have to be made.

database reference document: <https://github.com/googlecreativelab/quickdraw-component/blob/master/api/README.md>

api reference

<https://github.com/googlecreativelab/quickdraw-component/tree/master/api>