

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE OCCIDENTE

CÓMPUTO EN LA NUBE



REPORTE DE PRACTICA DE LABORATORIO

“Práctica 1 - Cloud storage”

I. Introducción

La práctica tiene como objetivo poner en práctica el uso de KVS, adicionalmente la práctica permitirá poner en práctica el uso del SDK de Amazon Web Services tanto en Java como en JavaScript.

La práctica consta de cuatro fases y el entregable final es crear un buscador de imágenes utilizando DynamoDB como storage en la nube así como implementar una interfaz gráfica web para realizar consultas a la base de datos.

II. Marco Teórico

Berkeley DB es una familia de bibliotecas de bases de datos de valor clave incorporadas que proporcionan servicios de gestión de datos escalables de alto rendimiento para aplicaciones. Los productos de Berkeley DB utilizan API de llamada a funciones simples para el acceso y la administración de datos.

Amazon DynamoDB es una base de datos de documentos y clave valor que ofrece un rendimiento de milisegundos de un solo dígito a cualquier escala. Se trata de una base de datos multirregional y totalmente administrada con seguridad, copia de seguridad y restauración incorporadas, así como almacenamiento en caché en memoria para aplicaciones escalables de Internet. DynamoDB puede manejar más de 10 billones de solicitudes por día y soportar picos de más de 20 millones de solicitudes por segundo.

AWS SDK permite client-Side Data Encryption for Amazon S3, Amazon DynamoDB Object Mapper, Amazon S3 Transfer Manager, Amazon SQS Client-Side Buffering.

[1]O. Berkeley DB, "Oracle Berkeley DB", *Oracle.com*, 2019. [Online]. Available: <https://www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html>. [Accessed: 24- Feb- 2019]

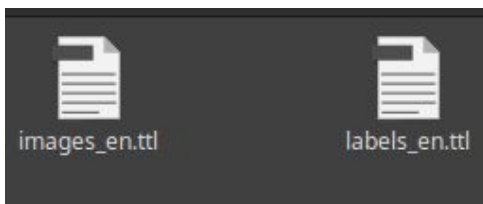
[3]A. DynamoDB, "Amazon DynamoDB - Overview", *Amazon Web Services, Inc.*, 2019. [Online]. Available: <https://aws.amazon.com/dynamodb/>. [Accessed: 24- Feb- 2019]

[4]A. SDK, "Uso de elementos: Java - Amazon DynamoDB", *Docs.aws.amazon.com*, 2019. [Online]. Available: https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/JavaDocumentAPIItemCRUD.html#PutDocumentAPIJava. [Accessed: 24- Feb- 2019]

[5]A. SDK, "AWS SDK for Java", *Amazon Web Services, Inc.*, 2019. [Online]. Available: <https://aws.amazon.com/sdk-for-java/>. [Accessed: 24- Feb- 2019]

III. Desarrollo de la Práctica.

Para comenzar se descargó el material indicado en las instrucciones, que es el esqueleto del proyecto con código fuente en java, así como un dataset de labels o etiquetas relacionadas a las imágenes.



En el archivo IndexImages hace la relación de las imagenes de tipo <http://xmlns.com/foaf/0.1/depiction>

```
// Imágenes
Triple tripleImages = images.getNextTriple();
String[] partsImages;
String imageSubject, imageObject, lowerPartsImage;

while(tripleImages != null){
    if(tripleImages.getPredicate().equals("http://xmlns.com/foaf/0.1/depiction")){
        imageSubject = tripleImages.getSubject().toLowerCase();
        partsImages = imageSubject.split("/");
        lowerPartsImage = partsImages[4].toLowerCase();
        if(lowerPartsImage.matches("^"+Config.filter.toLowerCase()+".*$")){
            imageObject = tripleImages.getObject().toLowerCase();
            imageStore.put(imageSubject, imageObject);
        }
    }
    tripleImages = images.getNextTriple();
}
```

De igual modo para el caso de las etiquetas nos interesa el tipo:

<http://www.w3.org/2000/01/rdf-schema#label>

```
ParseTriples labels = new ParseTriples("./dbd/" + titleFileName);
Triple tripleLabels = labels.getNextTriple();
String labelSubject, labelObject, labelStemmer;
String[] partsLabels, partsLabelsObject;

// Labels
while (tripleLabels != null) {

    if(tripleLabels.getPredicate().equals("http://www.w3.org/2000/01/rdf-schema#label")){

        labelObject = tripleLabels.getObject();
        if (labelObject.matches("^"+Config.filter+".*")) {
            labelSubject = tripleLabels.getSubject().toLowerCase();
            partsLabels = labelObject.split("/");
            partsLabelsObject = partsLabels[0].split(" ");

            for (int i = 0; i < partsLabelsObject.length; i++) {
                labelStemmer = PorterStemmer.stem(partsLabelsObject[i].toLowerCase());
                titleStore.put(labelStemmer, labelSubject);
            }
        }
        tripleLabels = labels.getNextTriple();
    }
}
```

La función de close, unicamente cierra la conexión a la base de datos

```
public void close() {
    imageStore.close();
    titleStore.close();
}
```

La clase QueryImages la cual toma uno o más palabras clave desde la línea de comandos, es decir como parámetros al mandar llamar el programa e imprime en consola la lista de URLs de imágenes relacionadas.

El método query de tipo Set regresa un Set de paths o url y el label ingresado por argumento.

```
public Set<String> query(String word)
{
    HashSet<String> imgQuerySet = new HashSet<String>();

    titleStore.get(PorterStemmer.stem(word.toLowerCase())).forEach((imgLabel) ->
        imageStore.get(imgLabel).forEach((path) ->
            imgQuerySet.add(path)));

    return imgQuerySet;
}
```

En el main basta con asociar los IKeyValuesStorage

CÓMPUTO EN LA NUBE
REPORTE DE PRÁCTICA DE LABORATORIO

```
try {  
    imageStore = KeyValueStoreFactory.getNewKeyValueStore(Config.storeType, dbName: "images");  
    titleStore = KeyValueStoreFactory.getNewKeyValueStore(Config.storeType, dbName: "terms");  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

Con ayuda del IDE al pasar los argumentos, la clase QueryImages muestra un listado de las coincidencias de la label o labels proporcionadas como argumentos, a continuación la salida:

Program arguments:

*** Alumno: _____ (Exp: _____)

APPLE:

- http://commons.wikimedia.org/wiki/special:filepath/apple_a8x_system-on-a-chip.jpg
- http://commons.wikimedia.org/wiki/special:filepath/apple_lisa.jpg

ORANGE:

- http://commons.wikimedia.org/wiki/special:filepath/acridine_orange.png
- http://commons.wikimedia.org/wiki/special:filepath/accama_anna_van_hannover.jpg
- http://commons.wikimedia.org/wiki/special:filepath/apple_and_orange_-_they_do_not_com

SALAD:

- <http://commons.wikimedia.org/wiki/special:filepath/abdiqasimsalad.jpg>
- <http://commons.wikimedia.org/wiki/special:filepath/afghan-salad.jpg>
- <http://commons.wikimedia.org/wiki/special:filepath/arabsalad.jpg>

GREEN:

- <http://commons.wikimedia.org/wiki/special:filepath/ashbelgsimonton.jpg>
- http://commons.wikimedia.org/wiki/special:filepath/apsey_green_-_geograph.org.uk_-_178460.jpg
- http://commons.wikimedia.org/wiki/special:filepath/alice_green.jpg
- [http://commons.wikimedia.org/wiki/special:filepath/african_green_pineapple_\(treron_calvus\).ind](http://commons.wikimedia.org/wiki/special:filepath/african_green_pineapple_(treron_calvus).ind)

Subiendo los datos a dynamodb

CÓMPUTO EN LA NUBE

REPORTE DE PRÁCTICA DE LABORATORIO

```
*** Alumno: _____ (Exp: _____ )
Creating table images
Table already exists
Creating table terms
Table already exists
ImgSubj: http://dbpedia.org/resource/aristotle
ImgObjc: http://commons.wikimedia.org/wiki/special:filepath/aristotle\_altemps

uploaded 0
ImgSubj: http://dbpedia.org/resource/aruba
ImgObjc: http://commons.wikimedia.org/wiki/special:filepath/flag\_of\_aruba.svg

uploaded 1
ImgSubj: http://dbpedia.org/resource/arthur\_schopenhauer
ImgObjc: http://commons.wikimedia.org/wiki/special:filepath/schopenhauer.jpg

uploaded 2
ImgSubj: http://dbpedia.org/resource/articles\_of\_confederation
ImgObjc: http://commons.wikimedia.org/wiki/special:filepath/articles\_pagel.jpg

uploaded 3
ImgSubj: http://dbpedia.org/resource/art
ImgObjc: http://commons.wikimedia.org/wiki/special:filepath/art-portrait-colla
```

Tablas en DynamoDB.

Imágenes.

Keyword ⓘ	inx	Value
http://dbpedia.org/resource/araang_league	13814	http://commons
http://dbpedia.org/resource/araby_(mason's_springs,_maryland)	8722	http://commons
http://dbpedia.org/resource/aragua	688	http://commons
http://dbpedia.org/resource/arakamani	13592	http://commons
http://dbpedia.org/resource/arakan_liberation_army	13879	http://commons
http://dbpedia.org/resource/araneus_marmoreus	7343	http://en.wikipe
http://dbpedia.org/resource/arantva	7242	http://commons

Labels.

Keyword ⓘ	inx	Value
of	299	http://dbpedia.org/re
of	303	http://dbpedia.org/re
revolut	54	http://dbpedia.org/re
score	520	http://dbpedia.org/re
screw	248	http://dbpedia.org/re
screw	634	http://dbpedia.org/re
uzbekistan	464	http://dbpedia.org/re

Después de modificar config para comunicarnos con AWS, pasamos a modificar el archivo de DynamoDBStorage, para crear la tabla, recuperar e insertar elementos.

Después de clonar el repositorio de la fase 3, se modificó el index.jade para presentar las fotografías, hacer que los botones next y previous sólo aparezcan en caso de ser necesarios en el app.js y la funcionalidad de estos mismos botones, así como validar que el tamaño de la data recibida en app.js (root) no sea nulo.

```
var index = 0;
var data = [];
$(document).ready(function() {
  $("#version").html("v0.14");

  $("#searchbutton").click(function(e) {
    displayModal();
  });

  $("#searchfield").keydown(function(e) {
    if (e.keyCode == 13) {
      displayModal();
    }
  });

  function displayModal() {
    $("#myModal").modal("show");
    $("#status").html("Searching...");
    $("#dialogtitle").html("Search for: " + $("#searchfield").val());
    $("#previous").hide();
    $("#next").hide();
    $.getJSON("/search/" + $("#searchfield").val(), function(data) {
      renderQueryResults(data);
    });
  }

  $("#next").click(function(e) {
    if (index + 1 > data.length / 4) {
      console.log(index);
    } else {
      index++;
      $("#pic0").attr("src", data.results[index * 4]);
      $("#pic1").attr("src", data.results[1 + index * 4]);
      $("#pic2").attr("src", data.results[2 + index * 4]);
      $("#pic3").attr("src", data.results[3 + index * 4]);
    }
  });

  $("#previous").click(function(e) {
    if (index - 1 < 0) {
      console.log(index);
    } else {
      index--;
      $("#pic0").attr("src", data.results[index * 4]);
      $("#pic1").attr("src", data.results[1 + index * 4]);
      $("#pic2").attr("src", data.results[2 + index * 4]);
      $("#pic3").attr("src", data.results[3 + index * 4]);
    }
  });
});
```

```
function renderQueryResults(data) {
  $("#pic0").attr("src", "");
  $("#pic1").attr("src", "");
  $("#pic2").attr("src", "");
  $("#pic3").attr("src", "");
  $("#pic0").hide();
  $("#pic1").hide();
  $("#pic2").hide();
  $("#pic3").hide();
  $("#previous").hide();
  $("#next").hide();

  if (data.error != undefined) {
    $("#status").html("Error: " + data.error);
    $("#pic0").attr("src", "");
    $("#pic1").attr("src", "");
    $("#pic2").attr("src", "");
    $("#pic3").attr("src", "");
  } else {
    this.data = data;
    $("#status").html("" + data.num_results + " result(s)");

    if (data.results.length > 4) {
      $("#next").show();
      $("#previous").show();
    }

    if (data.results.length >= 1) {
      $("#pic0").show();
      $("#pic0").attr("src", data.results[0]);
    }

    if (data.results.length >= 2) {
      $("#pic1").show();
      $("#pic1").attr("src", data.results[1]);
    }

    if (data.results.length >= 3) {
      $("#pic2").show();
      $("#pic2").attr("src", data.results[2]);
    }

    if (data.results.length >= 4) {
      $("#pic3").show();
      $("#pic3").attr("src", data.results[3]);
    }
  }
}
```

Archivo Jade index para pintar imágenes en la tabla.

```
table
  tr
    td#photo0
      img.movie-poster(src="", width='250px' height='250px' alt='1 ')#pic0
    td(width='20')
      img.movie-poster(src="", width='250px' height='250px' alt='2 ')#pic1
    td#photo1
  tr(height='20')
    td colspan='2'
  tr
    td#photo2
      img.movie-poster(src="", width='250px' height='250px' alt='3 ')#pic2
    td(width='20')
      img.movie-poster(src="", width='250px' height='250px' alt='4 ')#pic3
    td#photo3
  span#status Status
```

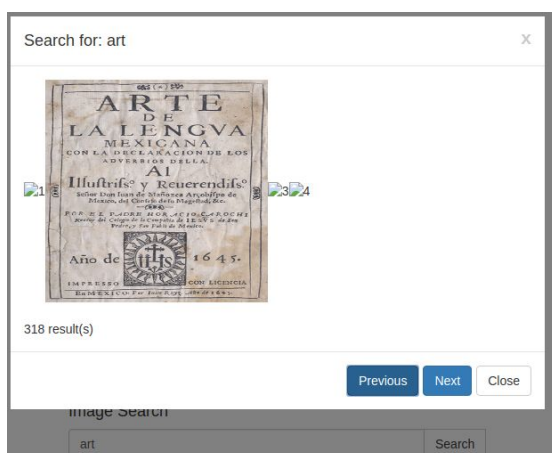
Keyvalues.js

```
db.describeTable({ TableName: tableName }, function(err, data) {
  if (err) {
    var params = {
      AttributeDefinitions: [
        {
          AttributeName: "Keyword",
          AttributeType: "S"
        },
        {
          AttributeName: "inx",
          AttributeType: "N"
        }
      ],
      KeySchema: [
        {
          AttributeName: "Keyword",
          KeyType: "HASH"
        },
        {
          AttributeName: "inx",
          KeyType: "RANGE"
        }
      ],
      ProvisionedThroughput: {
        ReadCapacityUnits: 1,
        WriteCapacityUnits: 1
      },
      TableName: tableName
    };
    db.createTable(params, function(err, data) {
      if (err) console.log(err, err.stack);
      // an error occurred
      else console.log(data); // successful response
    });
  } else {
    console.log(data);
  }
});

var searchValues = search.split(" ");
var items = [];
searchValues.forEach(function(value) {
  var params = {
    ExpressionAttributeValues: {
      ":v1": {
        S: value
      }
    },
    KeyConditionExpression: "Keyword = :v1",
    TableName: self.tableName
  };

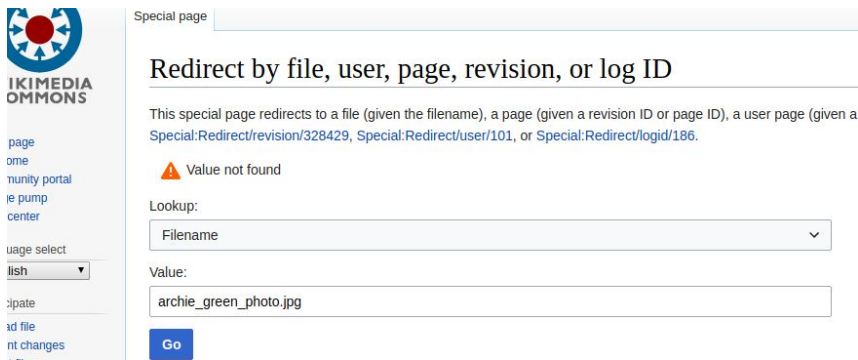
  db.query(params, function(err, data) {
    if (err) {
      console.log(err);
    } else {
      data.Items.forEach(function(item) {
        items.push({
          inx: item.inx.N,
          value: item.Value.S,
          key: item.Keyword.S
        });
      });
      self.cache.set(search, items);
      console.log(items);
      callback(err, items);
    }
  });
});
```

A continuación se muestra la salida montada en el local host al buscar la palabra 'art'.



Algunas imágenes no se muestran debido a que han sido movidas o eliminadas por Wikipedia.

A continuación se presenta una captura del navegador intentando acceder a un recurso imagen que ha sido movido o eliminado.



Conectando a la instancia creada en aws para posteriormente instalar las dependencias y correr express.

```
ubuntu@ip-172-31-28-228:~$ ssh ubuntu@ec2-52-33-49-31.us-west-2.compute.amazonaws.com
The authenticity of host 'ec2-52-33-49-31.us-west-2.compute.amazonaws.com (52.33.49.31)' can't be established.
ECDSA key fingerprint is SHA256:BnMar9AYmTud6YN12BgeVvZR0cwi0aU0I2Cd00rXX9g.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'ec2-52-33-49-31.us-west-2.compute.amazonaws.com,52.33.49.31' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Feb 25 01:40:50 UTC 2019

System load:  0.17               Processes:    88
Usage of /:   13.3% of 7.69GB    Users logged in: 0
Memory usage: 14%               IP address for eth0: 172.31.28.228
Swap usage:   0%

Get cloud support with Ubuntu Advantage Cloud Guest:
```

Dependencias de node en el server.

```
ubuntu@ip-172-31-28-228:~$ sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libhttp-parser2.7.1 libuv1 nodejs-doc
The following NEW packages will be installed:
  libc-ares2 libhttp-parser2.7.1 libuv1 nodejs nodejs-doc
0 upgraded, 5 newly installed, 0 to remove and 154 not
Need to get 5670 kB of archives.
After this operation, 24.8 MB of additional disk space
```

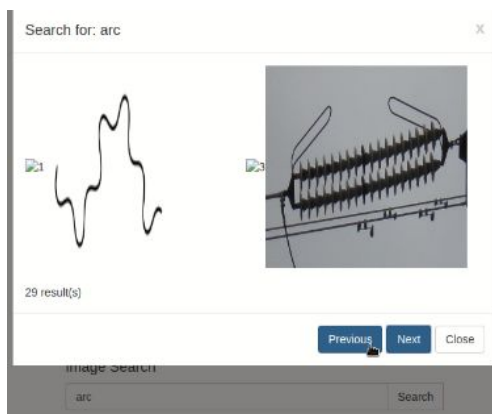
Dependencias de npm en el server.

```
ubuntu@ip-172-31-28-228:~$ sudo apt-get install npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu
```

Una vez instaladas las dependencias, se clona el proyecto de git y se ejecuta, para después poder acceder mediante la ipv4 pública al recurso buscador de imágenes.



Y al hacer las pruebas la búsqueda de las imágenes que se encuentra en video adjunto se puede ver cómo el funcionamiento es similar al del local host.



IV. Problemas y Soluciones

Los problemas principales fueron los siguientes:

Para insertar datos en la base de datos de Dynamo, tuve que verificar la referencia para Java que se encuentra en

https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/JavaDocumentAPIItemCRUD.html#PutDocumentAPIJava

Además de que se tuvo que hacer algunas modificaciones en el archivo de configuración debido que el entorno de trabajo fue basado en ubuntu, por lo que cambiar el path de la base de datos fue necesario.

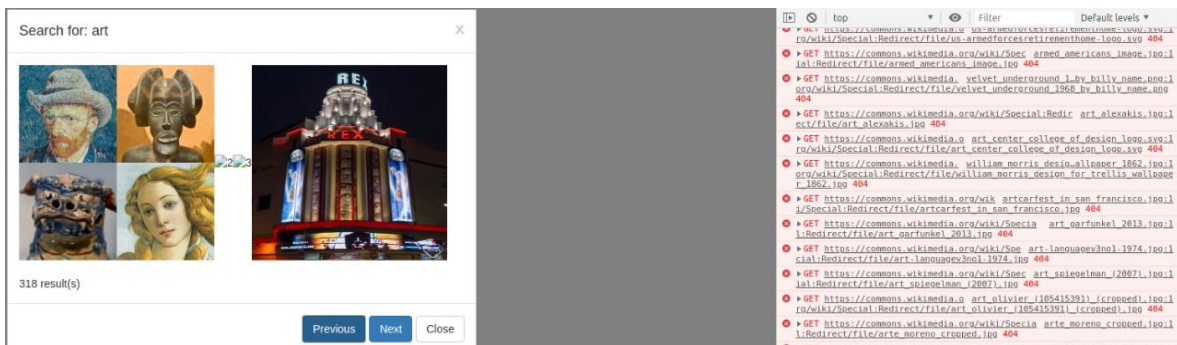
En el javascript al correr el servidor marcaba un error e indicaba que faltaba agregar la palabra new en keyvaluestore.js

Otro problema fue que a veces se reciben datos vacíos y el servidor no sabía cómo proceder, por lo que fue necesario agregar una validación y verificar que el tamaño de la data fuera mayor que 0 en el app.js.

Además de los detalles anteriores, otro inconveniente fue que se subieron muchos datos a la base de datos, por lo que se hizo muy tardada la carga de documentos y detuve manualmente la carga.

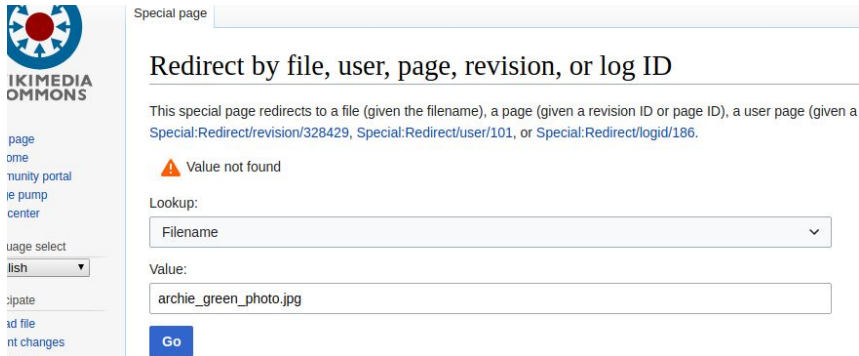
V. Experimentos y Resultados.

Uno de los experimentos realizados fue en el navegador inspeccionando para verificar la causa que no se pintaban todas las imágenes.



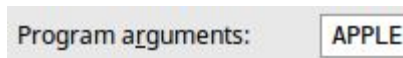
Fue entonces que al acceder a uno de los links marcados de color rojo por la consola del navegador, mostró un mensaje de que el recurso no estaba disponible.

CÓMPUTO EN LA NUBE
REPORTE DE PRÁCTICA DE LABORATORIO



The image shows the Wikimedia Commons 'Special page' interface. On the left is a sidebar with the Commons logo and various navigation links. The main content area is titled 'Special page' and 'Redirect by file, user, page, revision, or log ID'. It contains a message: 'This special page redirects to a file (given the filename), a page (given a revision ID or page ID), a user page (given a Special:Redirect/revision/328429, Special:Redirect/user/101, or Special:Redirect/logid/186.' Below this is a warning icon and the text 'Value not found'. There are two input fields: 'Lookup:' with a dropdown menu currently showing 'Filename', and 'Value:' with the text 'archie_green_photo.jpg'. A blue 'Go' button is at the bottom.

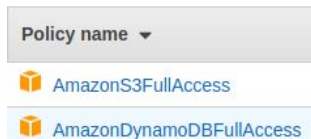
Otro de los experimentos realizados fue en la base de datos local haciendo consultas para verificar que se obtiene los datos solicitados.



A screenshot of a 'Program arguments' input field. The text 'APPLE' is entered into the field.

```
*** Alumno: _____ (Exp: _____ )
APPLE:
- http://commons.wikimedia.org/wiki/special:filepath/apple\_a8x\_system-on-a-chip.jpg
- http://commons.wikimedia.org/wiki/special:filepath/apple\_lisa.jpg
```

Además de los problemas anteriores a la hora de correr el servidor, no tenía permisos el rol para acceder a DynamoDB full access, lo que se soluciono agregando permisos desde la consola de amazon.



The image shows the AWS IAM console 'Policy name' dropdown menu. Two policies are visible: 'AmazonS3FullAccess' and 'AmazonDynamoDBFullAccess', which is currently selected.

VI. Conclusiones

La práctica fue retadora pues tuve que investigar información a parte de lo que vimos en clase debido que los ejemplos del profesor fueron algo simples pero interesantes pues con eso simple se pueden hacer cosas más grandes basándose en esa funcionalidad.

Hay mucha información de SDK de aws con los lenguajes de programación más famosos por lo que es una gran ayuda para las personas que quieren utilizar los servicios de amazon web services.