

1. Create a static website using AWS S3 and explain in great technical detail what you did. 30 points.

Para crear un website estático usando el servicio del almacenamiento S3 de AWS, se tienen que seguir los siguientes pasos:

1. Crear un Bucket S3

El siguiente comando te permite crear un crear un bucket, sin embargo requieres tener un AWS Access Key ID valida para que tu solicitudes sean autorizadas. Este comando toma dos argumentos: - - bucket , es el nombre que le asignas a tu S3 y - - region , se define la región del bucket. Por defecto la región del bucket se define en US East, aún así, se puede especificar la región donde quieres que opere el bucket.

Dentro de AWS CLI, se puede utilizar `s3` o `s3api`, siendo que el primero ofrece comandos de alto nivel para el manejo simple de instancias S3 con un enfoque amigable con el usuario, mientras que el segundo te permite acceder al conjunto de comandos de alto nivel que ofrece un enfoque más granular, y da más libertad para configurar el bucket.

```
aws s3api create-bucket --bucket agraz.cetystijuana.com --region us-east-1
```

```
kekaz16@Daniels-MacBook-Air ~ % aws s3api create-bucket --bucket agraz.cetystijuana.com --region us-east-1
{
  "Location": "/agraz.cetystijuana.com"
}
```

Para confirmar que tu bucket ha sido creado con éxito:

```
aws s3 ls
```

Este comando regresa una lista de todas las instancias S3 registradas.

```
[kekaz16@Daniels-MacBook-Air agraz.cetystijuana.com % aws s3 ls
2023-02-04 12:07:08 abner.cetystijuana.com
2023-02-03 18:16:09 agraz.cetystijuana.com
2023-02-03 18:23:29 angel.cetystijuana.com
2023-02-03 18:17:23 annie.cetystijuana.com
2023-02-03 18:16:23 aortega.cetystijuana.com
```

2. Habilitar tu Bucket para acceso público

Una vez creado el bucket se tiene que crear una política que permite que cualquier usuario pueda obtener los archivos de tu bucket para visualizarlos en su browser, ya que por defecto, S3 no está configurado para el acceso público.

Según la página de AWS (2023) una política es “un objeto de AWS que, cuando se asocia a una entidad o recurso, define sus permisos. AWS evalúa estas políticas cuando un principal, como un usuario, realiza una solicitud. Los permisos de las políticas determinan si se permite o deniega la solicitud. La mayoría de las políticas se almacenan en AWS como documentos JSON.” Entonces una política es un conjunto de permisos que describen la versión de la política y una lista de declaraciones.

Existen varios tipos de políticas, sin embargo, en este caso se utilizará la “S3 Bucket Policy” que forma parte de las “**Resource-based policies**” que engloba aquellas políticas que se enfocan a un recurso de AWS.

Para crear la política necesitas crear un archivo JSON con la siguiente estructura:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "PublicReadGetObject",
      "Effect" : "Allow",
      "Principal" : "*",
      "Action" : "s3:GetObject",
      "Resource" : "arn:aws:s3:::agraz.cetystijuana.com"
    }
  ]
}
```

Ahora se presentará cada una de las llaves declaradas en la poliza:

- **Version:** Esta llave hace referencia al versionado de los Buckets S3. El versionado te permite almacenar múltiples versiones de un objeto en un solo bucket S3. En esta política se utiliza la versión más actual que es la del 2012.
- **Statement:** Es la declaración principal que contiene un conjunto de especificaciones
 - **Sid:** Esta llave hace referencia al “Statement ID” del bucket, es el nombre que define el propósito o uso de la declaración.
 - **Effect:** Es el efecto que tendrá cuando un usuario solicite una acción específica, esto puede ser tanto “Allow” como “Deny”.
 - **Principal:** Es la cuenta o el usuario que se le permite el acceso a las acciones y recursos de la declaración. En una política S3, el principal puede ser una cuenta, servicio o usuario. En la política del bucket se utilizó el asterisco (*) para indicar que cualquier usuario puede acceder al bucket.

- **Action:** Para cada recurso, Amazon S3 permite un conjunto de acciones que le aceptan o niegan al usuario la capacidad de realizar una operación con respecto a un recurso de AWS. En esta política se establece que se va a usar la acción **GetObject**, la cual permite extraer objeto de un S3.
- **Resource:** Esta llave hace referencia al Bucket al que se le está aplicando la política, por lo que se requiere el ARN (Amazon Resource Name) para identificar el recurso. Por lo que en este apartado se puso:
arn:aws:s3:::agraz.cetystijuana.com. Cabe destacar que esta llave no está sujeta a solo, sino que también se puede hacer referencia a objetos, “access points” y trabajos que forman parte de los recursos de S3 a los cuales se les puede aceptar o denegar permisos.

Habiendo explicado cada una de las llaves. En resumen, la política que se definió garantiza que cualquier usuario puede obtener los archivos de nuestro sitio web estático.

3. Aplicar la política al bucket.

Una vez establecida la política del bucket, para poner la política en nuestro bucket, se ejecutó el siguiente comando:

```
aws s3api put-bucket-policy --bucket agraz.cetystijuana.com  
--policy file:///Users/kekaz16/Desktop/bucket_policy.json
```

Este comando aplica una política S3 a un Bucket S3, y se especifica con los siguientes argumentos: - - bucket , indica el nombre del bucket al que se le va a aplicar la política y - - policy , describe la ruta del archivo JSON donde está la política.

Para revisar si el ingreso de la política fue exitosa se puede confirmar con el siguiente comando:

```
aws s3api get-bucket-policy --bucket agraz.cetystijuana.com
```

Este comando regresa el archivo JSON minimizado de la política que se aplicó al bucket.

```
kekaz16@Daniels-MacBook-Air agraz.cetystijuana.com % aws s3api get-bucket-policy --bucket agraz.cetystijuana.com
{
  "Policy": "{\n\"Version\": \"2012-10-17\", \"Statement\": [\n{\n\"Sid\": \"PublicReadGetObject\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::agraz.cetystijuana.com/*\" }\n]\n}"
}
```

4. Configurar la Estructura del Bucket S3

Ya que los archivos de la página web serán alojados en el bucket, se tiene que configurar la estructura del S3 para que sepa lo primero que busque para cargar la página, para esto se define un indexDocument y un errorDocument, por tanto se hace uso del siguiente comando.

```
aws s3 website s3://agraz.cetystijuana.com --index-document index.html
--error-document error.html
```

Donde `aws s3 website` se utiliza para definir la configuración del sitio estático. Después de eso se indica el nombre del bucket a configurar para luego establecer que el `-index-document` se debe encontrar dentro del bucket como `index.html`. Este `indexDocument` es el archivo HTML que se va a cargar si alguien visita tu página y `-error-document` es el archivo que se va a cargar cuando la ruta de tu archivo `index.html` no se encuentre o esté mal puesta.

Ya declarada la estructura principal del sitio web, se prosiguió a crear la página web desde 0 respetando el IndexDocument y ErrorDocument.

Estructura completa de la página:



5. Cargar tu sitio web al S3

Ahora teniendo todos los archivos de la página, el siguiente paso es subir el directorio. Para esto se emplea este comando:

```
aws s3 sync . s3://agraz.cetystijuana.com
```

`aws s3 sync` sincroniza los archivos del directorio en el que estes con el bucket establecido, de modo que, al ejecutar este comando se hace el deposito de los archivos del proyecto en el bucket.

Para verificar que los archivos se cargaron al bucket, usa este comando:

```
kekaz16@Daniels-MacBook-Air agraz.cetystijuana.com % aws s3 ls s3://agraz.cetystijuana.com
PRE Assignments/
PRE css/
PRE html/
PRE imgs/
2023-02-04 14:43:07      6148 .DS_Store
2023-02-03 19:20:15         39 error.html
2023-02-04 14:43:07     2168 index.html
```

Y ahora ya debes de poder acceder a tu sitio web, bajo el siguiente enlace:

[http://\[BUCKET NAME\].s3-website-\[REGION\].amazonaws.com](http://[BUCKET NAME].s3-website-[REGION].amazonaws.com). En mi caso este sería mi enlace <http://agraz.cetystijuana.s3-website-us-east-1.amazonaws.com> . (Al momento de acceder a este link mostrará un error, ya que el sitio web ya habrá enlazado a un subdominio).

2. Link your website to a subdomain of cetystijuana.com and explain in great technical detail what you did. **30 points.**

Para enlazar un sitio web al subdominio de cetystijuana.com se hace uso de Amazon Route 53, que es un servicio web de DNS que permite registrar un dominio, dirigir el tráfico de internet a los recursos del dominio y comprobar la salud de los recursos de AWS que administre.

1. Crea una host zone

En nuestro caso es paso ya esta hecho, porque la zona de alojamiento fue creada por el profesor al registrar el dominio de cetystijuana.com, aún así, con fines demostrativos el comando a utilizar para crear un “host zone” es este:

```
aws route53 create-hosted-zone --name cetystijuana.com --caller-reference 1
```

Para ejecutar este comando se requiere de dos parámetros: - - name, pide que escribas el dominio de la página y - - caller-reference es un comentario opcional para el programador.

2. Crea un registro para el subdominio

Al final de la pregunta 1. se finalizó que para acceder al sitio web se tiene que hacer bajo este modelo de enlace: `http://[BUCKET NAME].s3-website-[REGION].amazonaws.com`. En vez de tener que hacer referencia al recurso S3 y a la región de AWS, se requiere crear un registro en formato JSON que tenga la siguiente estructura:

```
1  {
2    "Changes": [
3      {
4        "Action": "UPSERT",
5        "ResourceRecordSet": {
6          "Name": "agraz.cetystijuana.com",
7          "Type": "CNAME",
8          "TTL": 600,
9          "ResourceRecords": [
10           {
11             "Value": "agraz.cetystijuana.com.s3-website-us-east-1.amazonaws.com"
12           }
13         ]
14       }
15     ]
16   }
17 }
```

Brevemente, aquí se define un alias para la página web, para que en vez de usar `http://agraz.cetystijuana.s3-website-us-east-1.amazonaws.com` para acceder a los objetos del bucket, sea suficiente poner `agraz.cetystijuana.com`.

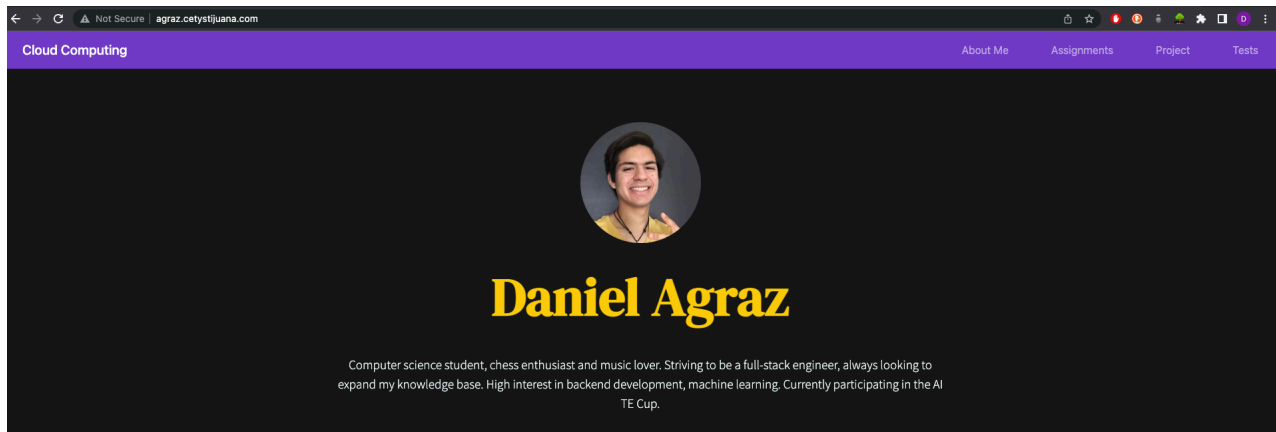
Una vez creado el archivo del registro se necesita ejecutar este comando, para crear el conjunto de recursos, que contiene información DNS para el subdominio especificado.

```
aws route53 change-resource-record-sets --hosted-zone-id Z03346142C3RKH191036Y
--change-batch file:///Users/kekaz16/Documents/Semestre6/CloudComputing/
subdomainLink.json
```


El comando contiene dos argumentos: `--hosted-zone-id` hace referencia al identificador de la zona del dominio, y `--change-batch`, indica la ruta al archivo JSON, que solicita el registro del subdominio.

```
{
  "ChangeInfo": {
    "Id": "/change/C04183354GZQPY16H1N6",
    "Status": "PENDING",
    "SubmittedAt": "2023-02-04T23:06:16.669000+00:00"
  }
}
```

Después de unos minutos, tu solicitud se procesa y ahora puedes acceder al contenido del bucket bajo `agraz.cetystijuana.com`.



3. Add yourself to the list of students in cetystijuana.com. 15 points

Para agregarse a la lista de estudiantes del dominio cetystijuana.com . Solo se tuvo que descargar los archivos del bucket que le pertenece al dominio con el comando **aws s3 sync s3://cetystijuana.com** . En mi caso cree un nuevo directorio para almacenar todos los archivos del sitio web.

```
kekaz16@Daniels-MacBook-Air cetystijuana.com % aws s3 sync s3://cetystijuana.com .
download: s3://cetystijuana.com/.idea/.gitignore to .idea/.gitignore
download: s3://cetystijuana.com/.idea/workspace.xml to .idea/workspace.xml
download: s3://cetystijuana.com/.idea/aws.xml to .idea/aws.xml
download: s3://cetystijuana.com/.DS_Store to ./.DS_Store
download: s3://cetystijuana.com/assets/img/.DS_Store to assets/img/.DS_Store
download: s3://cetystijuana.com/.idea/modules.xml to .idea/modules.xml
download: s3://cetystijuana.com/assets/.DS_Store to assets/.DS_Store
download: s3://cetystijuana.com/.idea/cetyswebsite.iml to .idea/cetyswebsite.iml
download: s3://cetystijuana.com/assets/img/apple-touch-icon.png to assets/img/apple-touch-icon.
download: s3://cetystijuana.com/assets/img/clients/client-2.png to assets/img/clients/client-2.
download: s3://cetystijuana.com/assets/css/style.css to assets/css/style.css
download: s3://cetystijuana.com/assets/img/clients/client-3.png to assets/img/clients/client-3.
download: s3://cetystijuana.com/assets/img/clients/client-4.png to assets/img/clients/client-4.
download: s3://cetystijuana.com/assets/img/favicon.png to assets/img/favicon.png
download: s3://cetystijuana.com/assets/img/clients/client-1.png to assets/img/clients/client-1.
download: s3://cetystijuana.com/assets/img/clients/client-5.png to assets/img/clients/client-5.
download: s3://cetystijuana.com/assets/img/clients/client-6.png to assets/img/clients/client-6.
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-3.jpg to assets/img/portfolio/po
olio-3.jpg
```

Una vez ejecutado el comando, te carga todos los archivos del bucket al directorio que se indico como el destino.

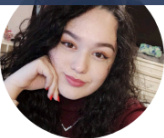
Luego edite el `index.html` y cree un nuevo `<div>` en la sección de estudiantes en la que puse mis datos, el enlace a mi sitio y una frase.

```
550 <div class="col-lg-6 mt-4">
551   <div class="member d-flex align-items-start" data-aos="zoom-in" data-aos-delay="400">
552     <div class="pic"></div>
553     <div class="member-info">
554       <h4>Daniel Agraz</h4>
555       <span><a href="//agraz.cetystijuana.com"> Agraz's Website </a></span>
556       <p>"Pero que reviente la bocina compadre"</p>
557       <div class="social">
558         <a href="https://www.linkedin.com/in/daniel-agraz-vallejo"> <i class="ri-linkedin-box-fill"></i> </a>
559       </div>
560     </div>
561   </div>
562 </div>
```

Después se uso `aws s3 sync . s3://cetystijuana.com`. Este comando indica que del directorio en el que se encuentra actualmente va a subir todos los archivos al bucket S3 del domino `cetystijuana.com`. Para comprobar que se agregó correctamente solo accedí al sitio del dominio.


ICE191 - SOFTWARE ARCHITECTURE

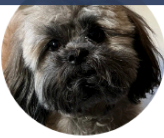
[Home](#)[Teams](#)[Students](#)[Teacher](#)[Need Help?](#)



Stephanie Gonzalez
[Stephanie - Personal Website](#)


""






Hugo Martin
[Hugo - Personal Website](#)


"Guau Guau". - Rocky






Carlos Rincon
[Cornito's - Personal Website](#)


"OK, let's go"






Ulises Aviles
[Ulises - Personal Website](#)


Vamos a Bucaramanga





Daniel Agraz
[Agraz's Website](#)

"Pero que revienta la bocina compadre"



4. Create an entry in the Students DynamoDB table using the cli with the following model: **15 points.**

```
{
  "id" : "matricula"
  "full_name" : "YYYYY"
  "personal_website" : "s3_url"
}
```

De acuerdo con AWS, para escribir un registro a una tabla de DynamoDB se tiene que hacer a partir de un archivo JSON o escribir toda la estructura al llamar el comando. Dentro del JSON para especificar que el dato es un String, se tiene que especificar con “S” para después colocar el valor de la llave.

DynamoDB API	PartiQL for DynamoDB
<pre>aws dynamodb put-item \ --table-name Music \ --item \ '{"Artist": {"S": "No One You Know"}, "SongTitle": {"S": "Call Me Today"}}</pre>	

Este el archivo JSON que cree con mis datos.

```
{ } AgrazRecord.json ×
Users > kekaz16 > Documents > Semestre6 > CloudComputing > { } AgrazRecord.json >
1  {
2    "id" : {"S" : "33293"},
3    "full_name" : {"S" : "Daniel Agraz Vallejo"},
4    "personal_website": {"S" : "agraz.cetystijuana.com"}
5  }
```

Para saber el nombre de la tabla a la que tengo que cargar este registro, utilice el comando.

```
aws dynamodb list-tables
```

```
{
  "TableNames": [
    "Students"
  ]
}
```

Conociendo esta información ahora se puede proceder con la inserción del registro en la tabla Students en DynamoDB. Con el comando:

```
aws dynamodb put-item --table-name Students --item
file:///Users/kekaz16/Documents/Semestre6/CloudComputing/AgrazRecord.json
```

Donde `aws dynamodb put-item` hace referencia a la operación de inserción, que te permite crear un nuevo ítem o reemplazar ítem viejo con uno nuevo. Este comando tiene los argumentos de: `-table-name` que te pide el nombre de la tabla a la que quieres insertar el ítem y `--item` que es la ruta al archivo JSON que contiene el registro.

Una vez hecho el registro para checar que se insertó correctamente. Usa el comando

```
aws dynamodb scan --table-name Students
```

Que te permite listar todas las entradas que tiene la tabla **Students**.

```
{
  {
    "full_name": {
      "S": "Daniel Armando Velazquez Caña"
    },
    "id": {
      "S": "29896"
    },
    "personal_website": {
      "S": "velazquez.cetystijuana.com"
    }
  },
  {
    "full_name": {
      "S": "Jorge Axel Cruz Jimenez"
    },
    "id": {
      "S": "031973"
    },
    "personal_website": {
      "S": "jaxelc.cetystijuana.com"
    }
  },
  {
    "full_name": {
      "S": "Daniel Agraz Vallejo"
    },
    "id": {
      "S": "33293"
    },
    "personal_website": {
      "S": "agraz.cetystijuana.com"
    }
  },
  {
    "full_name": {
      "S": "Alberto Ortega Morales"
    },
    "id": {
      "S": "18821"
    },
    "personal_website": {
      "S": "aortega.cetystijuana.com"
    }
  },
}
```

5. Watch the Not Just Code Monkeys by Martin Fowler and write your comments in your personal static website. **10 points**

Esta plática de Martin Fowler se me hizo muy interesante porque busca replantear el papel del Programador en el mundo globalizado que vivimos e impulsar varios valores que nos hagan respetar por lo que hacemos.

La sociedad nos ha catalogado bajo el término de “Code Monkeys” y parte de la comunidad se ha tomado muy enserio en seguir ese rol. Tanto que existen casos donde el único propósito del programador es construir lo que le pidan sin cuestionamiento alguno y sin la oportunidad de aportar ideas que sean consideradas dentro del proyecto.

Los proyectos de Software se suelen regir bajo dos equipos uno de Programadores y otro de Inversionistas o Analistas, que funge ser el cliente que busca que su idea se convierta en software. Con la metodología en Cascada, el resultado de estos proyectos era muy probable que fuera desastroso o que tuviera pocas cosas buenas pero que no fuera flexible. Por ello, el surgimiento de la metodología AGI llegó para ser uno de los marcos de referencia populares que promovería la creación de historias de usuarios, siendo que dichas historias serían la unidad fundamental para construir proyectos en fragmentos. Si bien esto llegó a traer mucho beneficios al desarrollo de proyectos de software, Fowler indica que el proceso clave de convertir la historia a código es uno de los problemas en los que el programador se le ha quitado la libertad sobre su profesión.

A la fecha en la que fue grabada el video, la programación no se consideraba una profesión real, era algo que siempre ha estado al alcance de todo aquel que tenga una computadora y que no requiere de un título para saber programar. No se le daba la seriedad que merecía. Considero que el programador debe de tener el mismo derecho que el cliente de proponer ideas e involucrarse en todos los aspectos que respecten al proyecto de software y no solo ser la mente genial que pasa las ideas a código. En

parte el propósito de AGI era que fuera un acercamiento conversacional, que hubiera un pensamiento bilateral entre programadores y cliente. Actualmente se suele tener un Project Manager que dirige al equipo de programadores y es el encargado de interactuar con el cliente, y si bien esta persona no suele tener un vasto conocimiento en programación se le asigna a la tarea de dirigir un proyecto de este tipo. En lo personal estoy en desacuerdo de este tipo de organización, es muy claro que debe haber un líder en un equipo, pero que ese líder sea capaz de participar también como miembro del equipo también es importante.

Otra idea que se me pareció buena, fue la de reforzar el interés del programador por dar un esfuerzo en aprender y expandir nuestro conocimiento sobre el área de la que somos parte. Es muy grato saber que como programador se puede trabajar se en la industria médica, financiera, científica, farmacéutica, moda e incluso entretenimiento. Es un mundo de posibilidades donde un programador puede formar parte desarrollando un proyecto de software. Y es importante que cuando participemos en un proyecto, seamos curiosos sobre la atmósfera que nos rodea, que investiguemos sobre aquellas cosas que no sepamos para conocer más sobre el área en la que estamos trabajando.

También se propone que un programador no solo escribe código para cumplir con una historia. El programador debe de ser consciente que la calidad y desempeño del código que produzca es en gran parte su responsabilidad. Esta idea debe ser un pensamiento recurrente en la comunidad, ya que el bienestar del usuario debe ser nuestra prioridad.

Considero fundamental hablar sobre el concepto de Alienating Atmosphere del cual Fowler considera que es algo preocupante que la comunidad de desarrolladores tiene que mejorar. Este concepto busca la inclusión de mujeres y grupos de minorías, a que sean recibidos de buena manera por el mundo del software, para que la programación sea representada por varios grupos y que las ideas trasciendan en diversidad, y no sea algo que solo un sector de la población acapare.

A lo largo de la historia de la programación, el éxito se ha basado en la méritocracia. Si vienes de una familia acomodada, o de la clase trabajadora, la programación ha sido indistinto con cualquier persona que se sumerge en su viaje, ya que solo importa “¿Puedo producir buen código?”. Y eso significa que la programación recompensa el talento y ser programador te da la oportunidad de decidir en donde quieres que tu talento tenga un impacto en el mundo.

Referencias Bibliográficas.

AWS (2023). *What are policies?* https://docs.aws.amazon.com/robomaker/latest/dg/auth_access_what-are-policies.html

AWS (2023). *AWS CLI Command Reference*. <https://awscli.amazonaws.com/v2/documentation/api/latest/index.html>

AWS (2023). *Step 2: Write data to a table using the console or AWS CLI*. <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/getting-started-step-2.html>

Meech-Ward S. (2022). *AWS CLI | Setup an S3 Static Website*. <https://www.sammeechward.com/aws-cli-s3-static-website>

GOTO Conferences (2015). *Not Just Code Monkeys - Martin Fowler - GOTO 2014*. https://www.youtube.com/watch?v=4E3xfR6IBII&ab_channel=GOTOConferences