



Tecnológico de Monterrey

Campus Querétaro

Entendimiento de los datos

Gamaliel Marines Olvera	A01708746
Uri Jared Gopar Morales	A01709413
José Antonio Miranda Baños	A01611795
María Fernanda Moreno Gómez	A01708653
Oskar Adolfo Villa López	A01275287
Luis Ángel Cruz García	A01736345

Inteligencia artificial avanzada para la ciencia de datos II
Grupo 501

Introducción

En este documento se detallarán las herramientas tecnológicas que utilizamos para procesar y analizar los datos proporcionados por nuestro socio formador, CAETEC. Explicaremos los métodos de almacenamiento que empleamos, el proceso de limpieza y análisis de las imágenes, así como las ventajas y desventajas de trabajar con big data en el contexto de este proyecto. El objetivo principal es desarrollar un sistema eficiente para clasificar imágenes de camas de vacas utilizando técnicas de machine learning, aprovechando plataformas como Google Drive para el manejo colaborativo de los datos y AWS S3 para la demostración de soluciones escalables de almacenamiento.

Herramientas y tecnologías

- *Google Drive*: Es una plataforma gratuita que permite almacenar archivos en la nube, por lo que puedes acceder a ellos en cualquier momento en cualquier parte del mundo. Utilizamos Drive para subir las imágenes elegidas del dataset de modo que, nos las pudiéramos repartir entre los miembros de los dos equipos de camas y poder clasificarlas. Una vez que las imágenes estuvieran clasificadas, estas se subían al Drive para que todos tuviéramos acceso a ellas.
- *Python*: Es un lenguaje de programación que, entre muchas cosas, es utilizado en machine learning (ML). Usamos python debido a sus bibliotecas (que hemos visto reiteradamente y de las cuales, estamos familiarizados) y así crear, de primera instancia, el clasificador de las imágenes de vacas en sus 3 estados posibles (cama vacía/cama con vaca acostada/cama con vaca de pie). Además, python tiene una gran variedad de bibliotecas para la visualización de datos y de ML, por lo que es una gran herramienta de trabajo en este contexto.
- *AWS S3*: Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento. Almacena datos como objetos dentro de buckets, donde un objeto es entendido como un archivo y cualquier metadato que describa el archivo (es un contenedor de objetos en pocas palabras). Aunque vamos a correr el modelo en un futuro de manera local (aprovechando la GPU de nuestras computadoras), usamos AWS S3 en manera de demostración para cargar y almacenar las imágenes por lotes, ya que nos ofrece escalabilidad y seguridad, atributos que son muy importantes para nosotros.

Colecta inicial de los datos

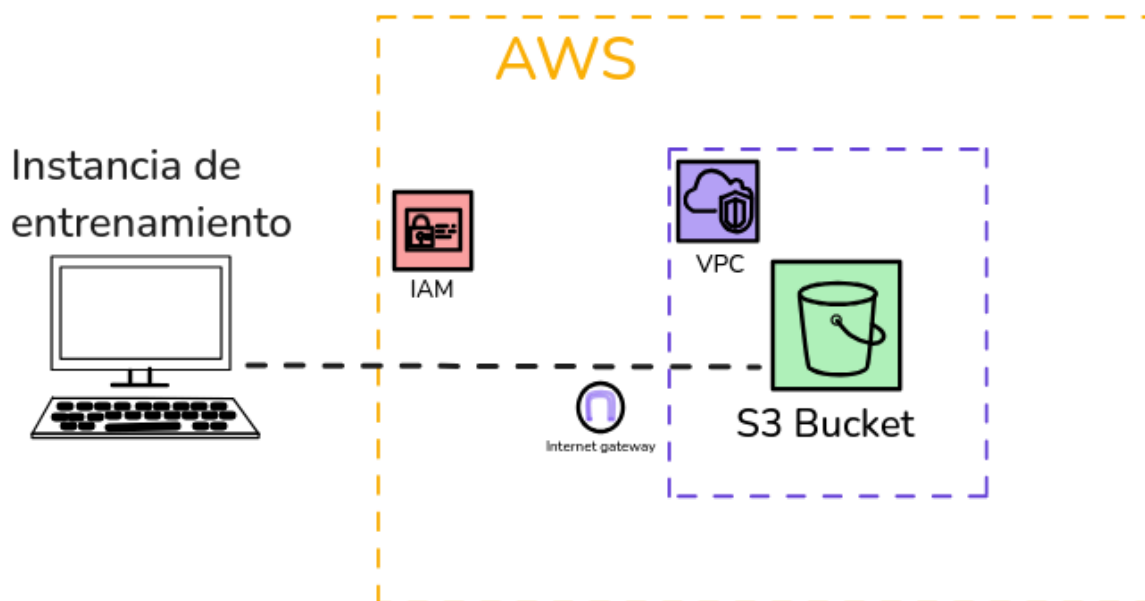
Los datos que en este caso fueron las imágenes de 3 de las camas de los espacios de “Camas” de las vacas fueron proporcionados por el Dr. Ivo Ayala por medio de una unidad compartida de One Drive que contiene 9,634 imágenes, que son de tipo JPG y con dimensiones de 1920 x 1080 px. Cada imagen tiene un tamaño de al menos 150 KB hasta 750 KB aproximadamente, dando un total de 12.4 GB en total por todo el dataset de “Beds”, fotos que fueron tomadas por medio de una cámara web marca Logitech. De las 9,634 disponibles del dataset, utilizamos aproximadamente 4,004 de las imágenes para clasificarlas. El proceso de transformación de las imágenes para hacer nuestro nuevo dataset fue el siguiente:

- Elegimos 4,004 imágenes de las 9,634 disponibles para hacer su clasificación (cama vacía/cama con vaca parada/cama con vaca acostada).
- Repartimos la cantidad de imágenes elegida entre los miembros de los 2 equipos de camas en partes equitativas para clasificarlas.
- Se usa el script para cortar las imágenes entre las 3 camas y se clasifica cada cama tomando en cuenta cada uno de los estados posibles de las camas.
- Individualmente, se suben las carpetas de las imágenes de cada estado de las camas en un Drive para que estén disponibles para todos los miembros de los dos equipos de camas y poder armar nuestro dataset.

Modelo de almacenamiento

Debido a la cantidad y naturaleza de los datos actuales, se consideró que un almacenamiento local es más adecuado, en conjunto con Google Drive para compartir las imágenes entre miembros del equipo.

Sin embargo, se realizó un plan de almacenamiento en caso de que se necesiten más datos. Este consiste en el uso de la herramienta S3 de AWS, un servicio administrado de almacenamiento de objetos, el cual es una opción segura y que no requiere de configuraciones complejas.



Arquitectura de almacenamiento.

S3 permite consultar los datos con facilidad, ya que cuenta con APIs y SDKs con los que se puede consultar la información directamente desde el dispositivo que realiza el cómputo. Cuenta con el servicio IAM para la autenticación de usuarios, y una Virtual Private Cloud, que protege a los datos de accesos de distintas redes.

Flujo de datos

El flujo de los datos se divide en distintas etapas:

- Recolección.

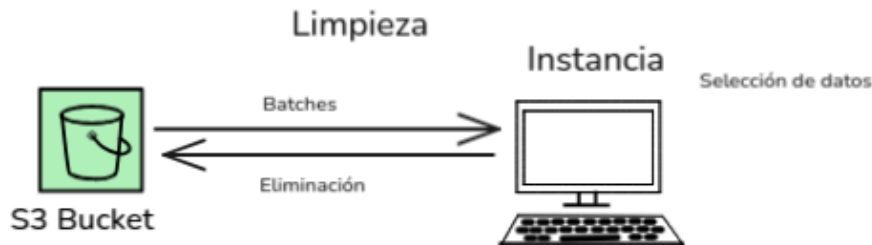
La recolección requiere un cargado inicial desde la instancia que es la fuente de los datos hacia el bucket de S3, en donde serán almacenadas. En el caso específico, ésta instancia es la raspberry encargada de tomar las fotos. Sin embargo, esta instancia puede ser una computadora o un servidor. La instancia puede realizar cargas posteriores, siguiendo el mismo flujo.



Flujo de datos en recolección.

- Limpieza.

La limpieza de los datos se realiza en una estancia de procesamiento. Las imágenes se leen y se utiliza algún proceso para eliminar las fotos que no son útiles, ya sea manual o automáticamente. Cuando se encuentra una imagen que no es útil, se elimina la imagen del bucket de S3.



Flujo de datos de limpieza.

- Procesamiento y entrenamiento.

A pesar de que son dos procesos distintos, ambos se realizan en serie, por lo que es conveniente condensar el flujo de datos en uno solo para aprovechar el procesamiento y reducir los costos. La instancia de entrenamiento puede obtener las imágenes por batches, los cuales son conjuntos de datos de un tamaño definido menor al tamaño total. Los batches son solicitados al bucket de S3. Esto se realiza de esta manera para evitar descargar todo el dataset en un solo movimiento, lo cual puede ser muy lento y superar la capacidad de memoria de la instancia de entrenamiento. La instancia realiza las transformaciones o aumentaciones de manera local aplicadas solo al batch.

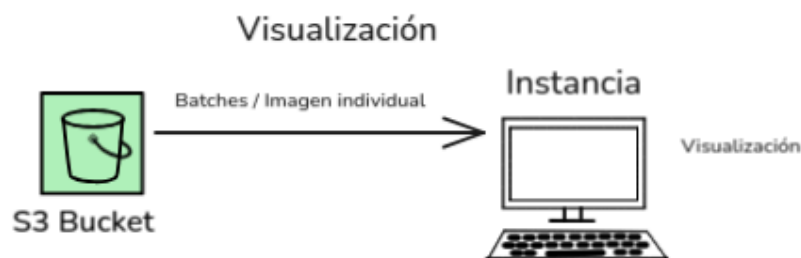
Posteriormente se alimentan las imágenes transformadas al modelo. Una vez que se utiliza el batch, se borra de la memoria y se descarga el siguiente.



Flujo de datos de procesamiento y entrenamiento.

- Visualización.

La visualización de las imágenes se puede realizar individualmente o por batches, para evitar descargar todos los datos en un solo proceso. La instancia solicita las imágenes al bucket de S3 en donde están almacenadas.



Flujo de datos de visualización.

Limpieza y clasificación

El enfoque del modelo planteado es clasificar las imágenes de cada cama en 3 categorías:

- Cama vacía.

- Vaca acostada.
- Vaca de pie.

Para ello, es necesario obtener imágenes de las camas individuales.

Las imágenes originales comprenden 3 camas y media. Se decidió descartar la media cama porque la información que otorga es incompleta. Cada imagen se separó en 3 fotos del mismo tamaño: 450 x 950 px. Cada imagen separada contiene una sola cama.



Foto original.

Para la separación, se creó un programa de Python. Antes de separar la imagen, se aplica un proceso de la librería de OpenCV para reducir la curvatura que causa la perspectiva de la cámara. Posteriormente, las imágenes se cortan y se presentan al usuario, el cual puede utilizar las teclas del teclado para indicar a qué categoría pertenece cada imagen recortada. Una vez que se selecciona la categoría, el programa mueve la imagen a su carpeta correspondiente, y muestra la siguiente. Si hay imágenes corruptas, se da clic a la tecla espacio para omitirlas, y no se incluyen en las carpetas de imágenes clasificadas.



Imagen recortada clasificada como “Cama vacía”.

La clasificación debe ser realizada por el usuario, por lo que se definieron los siguientes criterios:

- “Vaca de pie”:
 - Hay una vaca de pie dentro de la cama, con por lo menos dos patas sobre la cama.
- “Vaca acostada”:
 - Hay una vaca acostada dentro de la cama.
- “Cama vacía”
 - No hay una vaca en la imagen.
 - Hay partes de vacas, pero pertenecen a vacas en las camas contiguas.
 - Hay partes de una vaca sobre la cama, pero no tiene por lo menos dos patas sobre la cama. (La vaca se está asomando o aún no se sube a la cama).

Separación de datos en train / test

Una vez contando con los datos clasificados, lo siguiente que realizamos fue el código necesario para implementar la técnica split que nos permite trabajar con sets de entrenamiento, validación y prueba, lo que permite principalmente evitar overfitting en nuestro modelo, es decir, evitar que memorice información respecto a los datos de entrenamiento y después no pueda responder de forma correcta a datos nuevos.

Ahora bien, el set train es el que naturalmente cuenta con un mayor número de elementos, debido a que, como lo dice el nombre, será con el que se llevará a cabo todo el proceso de entrenamiento. Por otra parte, el set de validación se suele tomar como parte del set de entrenamiento, por lo que en este paso nos concentramos específicamente en train y test, que será el subconjunto de datos que se utilizará a modo de prueba para verificar el funcionamiento del modelo con nuevos datos.

Es importante mencionar que existe una especie de “estándar” respecto al porcentaje de datos que componen cada uno de estos sets, asignando normalmente el 80% de los datos a train y el 20% restante a test. Esto es debido a que en la mayoría de los casos son los valores con los que se ha obtenido mejor rendimiento en diferentes modelos, sin embargo, puede haber ocasiones en los que no sea lo mejor utilizar estos porcentajes, por ejemplo en casos en los que se cuente una cantidad muy diferente en las proporciones de datos o cuando se tiene una cantidad demasiado grande de datos.

Es por eso que para poder validar y entrenar un modelo existen técnicas como implementar k-fold cross validation que permite una mayor precisión en la forma de evaluar el modelo. Este es un algoritmo que no implementa de forma directa un porcentaje para partir el dataset en train y test, sino que cuenta con un parámetro k que define un número de particiones y de iteraciones con las que se entrenará el modelo.

Por ejemplo, al definir $k = 5$, el modelo se dividirá en 5 subsets, se entrenará y se evaluará 5 veces, siendo el accuracy el promedio del obtenido en esas 5 iteraciones.

El proceso consiste en que cada iteración 4 de esos subsets se utilizarán para entrenar el modelo, mientras que uno se excluirá y se utilizará para validación, y así se repetirá con cada iteración, permitiendo evaluar y entrenar con todos los datos.

Una de las desventajas de este proceso es puede ser muy costoso, como lo es en nuestro caso, ya que contamos con alrededor de 10,000 imágenes, lo cual se puede volver un modelo que tarde un tiempo considerable en entrenarse y que por lo

mismo, también complicaría el ajuste de otros hiperparámetros de importancia como lo es el learning rate.

Actualmente no contamos con el modelo final, por lo que la técnica no podrá ser implementada en su totalidad para esta entrega, sin embargo, se realizó un código en el cual se utiliza una red neuronal convolucional simple que permite mostrar el funcionamiento de esta técnica, al no contar con nuestro modelo, tampoco nos será posible definir si este proceso se aplicará o no en nuestro desarrollo, esto más que nada debido al tiempo que requiere, sin embargo sí será discutido una vez que se tenga dicho modelo, aunque la decisión hasta el momento es que no se utilizará, ya que la prioridad será ajustar y probar con el número de épocas y el learning rate..

Por otra parte, también se realizó un código que separa los datos en carpetas de train y test. Debido a que en dado caso de que no nos decidamos por utilizar k-fold cross validation debido al tiempo de ejecución, ya estamos más acostumbrados a contar con el dataset separado, por lo que sería más fácil implementar nuestro modelo.

Códigos para split

[SPLIT DATASET](#)

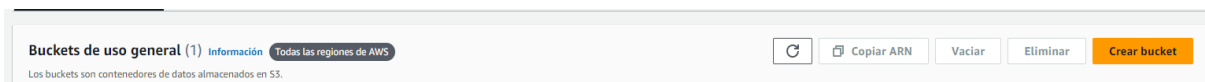
Cargado de datos

Para la utilización de S3 en Amazon debemos contar con una cuenta, en este caso esta cuenta es proporcionada por el Tecnológico de Monterrey Campus Querétaro, en ella contamos con un saldo de 50 dólares, los cuales son de uso libre para realizar prácticas dentro del entorno de Amazon.

Paso uno al momento de Iniciar con el proceso de utilizar S3, debemos de inicializar nuestra consola, una vez hecho esto en la parte de servicios buscaremos "S3".



Una vez dentro seleccionaremos la opción de crear bucket, está bucket será nuestro contenedor virtual en el que almacenaremos nuestras imágenes del dataset, las cuales fueron proporcionadas por nuestro socio formador CAETEC, se usarán posteriormente para poder entrenar nuestro modelo de inteligencia artificial.



Al seleccionar crear nos direccionará a esta pestaña:

En la cual le podremos ver la región en la que estamos trabajando, cambiar el nombre de nuestra bucket en este caso fue "vacas_bucket" y la propiedad del contenido de la Bucket, en este caso decimos que es de uso personal y que no se compartan.

Crear bucket [Información](#)

Los buckets son contenedores de datos almacenados en S3.

Configuración general

Región de AWS

EE. UU. Este (Norte de Virginia) us-east-1

Tipo de bucket [Información](#)

☒ **Uso general**

Recomendado para la mayoría de los casos de uso y patrones de acceso. Los buckets de uso general son del tipo de bucket de S3 original. Permiten una combinación de clases de almacenamiento que almacenan objetos de forma redundante en múltiples zonas de disponibilidad.

☐ **Directorio**

Recomendado para casos de uso de baja latencia. Estos buckets utilizan únicamente la clase de almacenamiento S3 Express One Zone, que proporciona un procesamiento más rápido de los datos dentro de una única zona de disponibilidad.

Nombre del bucket [Información](#)

vacas_bucket

El nombre del bucket debe ser único dentro del espacio de nombres global y seguir las reglas de nomenclatura del bucket. [Consulte las reglas para la asignación de nombres de buckets](#)

Copiar la configuración del bucket existente: *opcional*

Solo se copia la configuración del bucket en los siguientes ajustes.

Elegir el bucket

Formato: s3://bucket/prefijo

Propiedad de objetos [Información](#)

Controle la propiedad de los objetos escritos en este bucket desde otras cuentas de AWS y el uso de listas de control de acceso (ACL). La propiedad de los objetos determina quién puede especificar el acceso a los objetos.

☒ **ACL deshabilitadas (recomendado)**

Todos los objetos de este bucket son propiedad de esta cuenta. El acceso a este bucket y sus objetos se especifica solo mediante políticas.

☐ **ACL habilitadas**

Los objetos de este bucket pueden ser propiedad de otras cuentas de AWS. El acceso a este bucket y sus objetos se puede especificar mediante ACL.

Propiedad del objeto

Bloquearemos el acceso público

Configuración de bloqueo de acceso público para este bucket

Se concede acceso público a los buckets y objetos a través de listas de control de acceso (ACL), políticas de bucket, políticas de puntos de acceso o todas las anteriores. A fin de garantizar que se bloquee el acceso público a todos sus buckets y objetos, active Bloquear todo el acceso público. Esta configuración se aplica exclusivamente a este bucket y a sus puntos de acceso. AWS recomienda activar Bloquear todo el acceso público, pero, antes de aplicar cualquiera de estos ajustes, asegúrese de que las aplicaciones funcionarán correctamente sin acceso público. Si necesita cierto nivel de acceso público a los buckets u objetos, puede personalizar la configuración individual a continuación para adaptarla a sus casos de uso de almacenamiento específicos. [Más información](#)

☒ **Bloquear todo el acceso público**

Activar esta configuración equivale a activar las cuatro opciones que aparecen a continuación. Cada uno de los siguientes ajustes son independientes entre sí.

☒ **Bloquear el acceso público a buckets y objetos concedido a través de nuevas listas de control de acceso (ACL)**
S3 bloqueará los permisos de acceso público aplicados a objetos o buckets agregados recientemente, y evitará la creación de nuevas ACL de acceso público para buckets y objetos existentes. Esta configuración no cambia los permisos existentes que permiten acceso público a los recursos de S3 mediante ACL.

☒ **Bloquear el acceso público a buckets y objetos concedido a través de cualquier lista de control de acceso (ACL)**
S3 ignorará todas las ACL que conceden acceso público a buckets y objetos.

☒ **Bloquear el acceso público a buckets y objetos concedido a través de políticas de bucket y puntos de acceso públicas nuevas**
S3 bloqueará las nuevas políticas de buckets y puntos de acceso que concedan acceso público a buckets y objetos. Esta configuración no afecta a las políticas ya existentes que permiten acceso público a los recursos de S3.

☒ **Bloquear el acceso público y entre cuentas a buckets y objetos concedido a través de cualquier política de bucket y puntos de acceso pública**
S3 ignorará el acceso público y entre cuentas en el caso de buckets o puntos de acceso que tengan políticas que concedan acceso público a buckets y objetos.

También nos permite tener un control de versiones, sin embargo, como esta cuenta es limitada, no activaremos esta opción.

Control de versiones de buckets

El control de versiones es una forma de mantener múltiples variantes de un objeto dentro del mismo bucket. Puede utilizar el control de versiones para conservar, recuperar y restaurar todas las versiones de los objetos almacenados en su bucket de Amazon S3. Con el control de versiones, puede recuperarse con facilidad de las acciones involuntarias de los usuarios y de los errores en las aplicaciones. [Más información](#)

Control de versiones de buckets

- ☒ Desactivar
☐ Habilitar

Dejaremos el cifrado predeterminado, al igual que la configuración avanzada, una vez hecho esto le podemos dar en crear Bucket.

Cifrado predeterminado [Información](#)
El cifrado del lado del servidor se aplica automáticamente a los nuevos objetos almacenados en este bucket.

Tipo de cifrado [Información](#)
☒ Cifrado del servidor con claves administradas de Amazon S3 (SSE-S3)
☐ Cifrado del servidor con claves de AWS Key Management Service (SSE-KMS)
☐ Cifrado de doble capa del servidor con claves de AWS Key Management Service (DSSE-KMS)
Proteja sus objetos con dos capas de cifrado independientes. Para obtener más información sobre los precios, consulte [DSSE-KMS pricing](#) (Precios de DSSE-KMS) en la pestaña **Storage** (Almacenamiento) de la [página de precios de Amazon S3](#).
Clave de bucket
El uso de una clave de bucket de S3 para SSE-KMS reduce los costos de cifrado al reducir las llamadas a AWS KMS. Las claves de bucket de S3 no son compatibles con DSSE-KMS. [Más información](#)
☐ Desactivar
☒ Habilitar

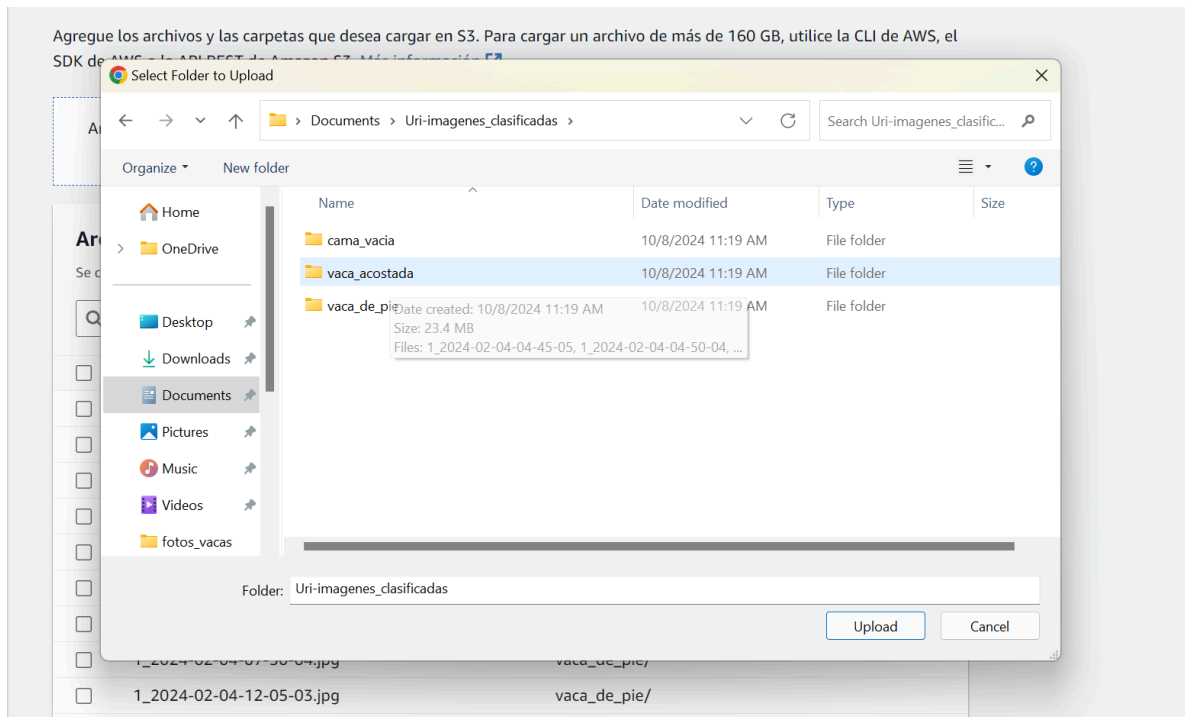
▼ Configuración avanzada
Bloqueo de objetos
Almacene objetos mediante un modelo de escritura única, lectura múltiple (WORM, write-once-read-many) para evitar que se eliminen o sobrescriban objetos durante un periodo de tiempo fijo o de manera indefinida. El bloqueo de objetos solo funciona en buckets con control de versiones. [Más información](#)
☒ Desactivar
☐ Habilitar
Permitir permanentemente bloquear los objetos de este bucket. Después de la creación del bucket, se requiere una configuración adicional de bloqueo de objetos en los detalles del bucket para proteger sus objetos y que no se eliminen o sobrescriban.

i El bloqueo de objetos solo funciona en buckets con control de versiones. Al habilitar el bloqueo de objetos, se habilita automáticamente el control de versiones.

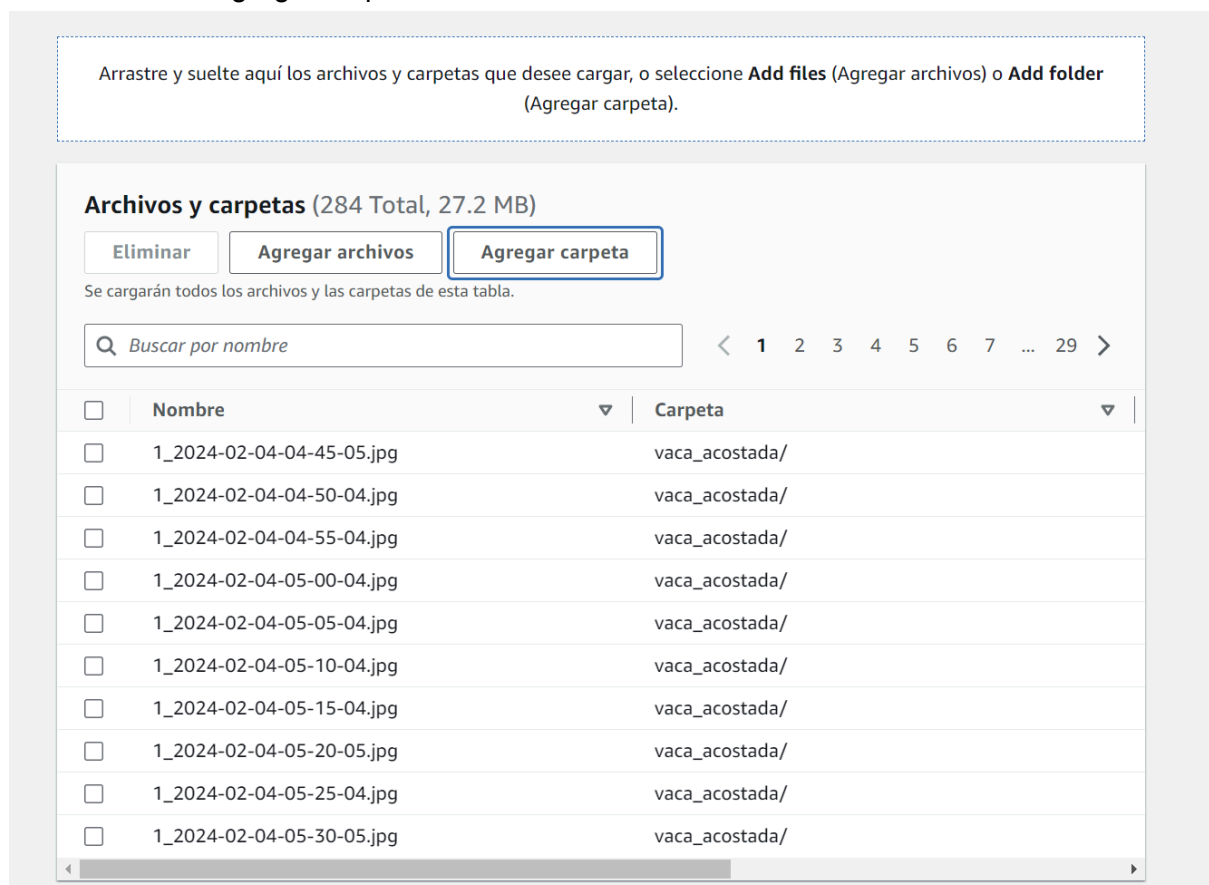
Se debe de visualizar de la siguiente manera

Buckets de uso general (1) Información Todas las regiones de AWS				Copiar ARN	Vaciar	Eliminar	Crear bucket
Los buckets son contenedores de datos almacenados en S3.							
<input type="text" value="Buscar buckets por nombre"/>							
<div><div><div>Nombre</div><div>Región de AWS</div><div>Analizador de acceso de IAM</div><div>Fecha de creación</div></div><div><div><input type="radio"/> vacas-bucket</div><div>EE. UU. Este (Norte de Virginia) us-east-1</div><div>Ver analizador para us-east-1</div><div>10 Oct 2024 9:56:33 AM CST</div></div></div>							

Le daremos clic a nuestra bucket creada, y seleccionaremos la opción de cargar, en este caso seleccionamos la opción de subir carpeta.



Seleccionamos Agregar carpeta



Veremos cómo es que todo el contenido de esta carpeta se está subiendo a AWS S3.

Cargando

Total restante: 276 archivos: 26.4 MB (97.03%)

Tiempo restante estimado: 4 minutos

Velocidad de transferencia: 110.9 KB/s

Cancelar

Destino

s3://vacas-bucket

Realizado correctamente

8 archivos, 828.5 KB (2.97%)

Con errores

0 archivos, 0 B (0%)

Archivos y carpetas

Configuración

Archivos y carpetas (284 Total, 27.2 MB)

< 1 2 3 4 5 6 7 ... 29 >

Nombre	Carpeta	Tipo	Tamaño	Estado	Error
1_2024-02-0...	vaca_acostada/	image/jpeg	94.0 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	93.2 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	93.4 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	92.0 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	90.9 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	91.4 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	91.7 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	91.5 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	90.5 KB	Realizado corre	-
1_2024-02-0...	vaca_acostada/	image/jpeg	90.4 KB	Pendiente	-

Y si entramos otra vez a nuestra Bucket de “vacas_bucket” podemos ver como la estructura cambió y ahora nos aparecen todas las carpetas que hemos subido, y si le damos clic a esas carpetas podemos visualizar su contenido.

vacas-bucket

Información

Objetos

Propiedades

Permisos

Métricas

Administración

Puntos de acceso

Objetos (2)

Información

< 1 >

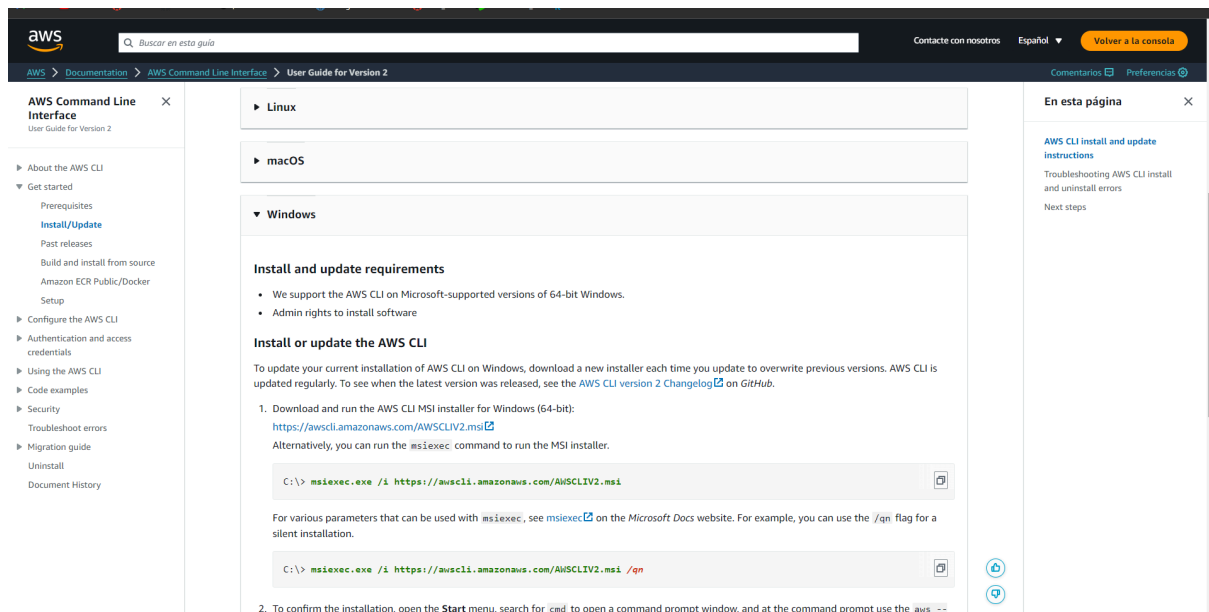
	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	vaca_acostada/	Carpeta	-	-	-
<input type="checkbox"/>	vaca_de_pie/	Carpeta	-	-	-

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	1_2024-02-04-04-45-05.jpg	jpg	10 Oct 2024 10:03:02 AM CST	94.0 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-04-50-04.jpg	jpg	10 Oct 2024 10:03:03 AM CST	93.2 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-04-55-04.jpg	jpg	10 Oct 2024 10:03:03 AM CST	93.4 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-00-04.jpg	jpg	10 Oct 2024 10:03:04 AM CST	92.0 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-05-04.jpg	jpg	10 Oct 2024 10:03:05 AM CST	90.9 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-10-04.jpg	jpg	10 Oct 2024 10:03:05 AM CST	91.4 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-15-04.jpg	jpg	10 Oct 2024 10:03:06 AM CST	91.7 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-20-05.jpg	jpg	10 Oct 2024 10:03:07 AM CST	91.5 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-25-04.jpg	jpg	10 Oct 2024 10:03:07 AM CST	90.5 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-30-05.jpg	jpg	10 Oct 2024 10:03:08 AM CST	90.4 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-35-05.jpg	jpg	10 Oct 2024 10:03:08 AM CST	89.5 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-40-05.jpg	jpg	10 Oct 2024 10:03:09 AM CST	90.2 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-45-04.jpg	jpg	10 Oct 2024 10:03:09 AM CST	92.3 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-50-05.jpg	jpg	10 Oct 2024 10:03:09 AM CST	89.9 KB	Estándar
<input type="checkbox"/>	1_2024-02-04-05-55-04.jpg	jpg	10 Oct 2024 10:03:10 AM CST	90.8 KB	Estándar

De esta manera ya contamos con un almacenamiento en la nube el cual nos ayudará para gestionar nuestra información de una mejor manera, de igual forma podremos acceder a

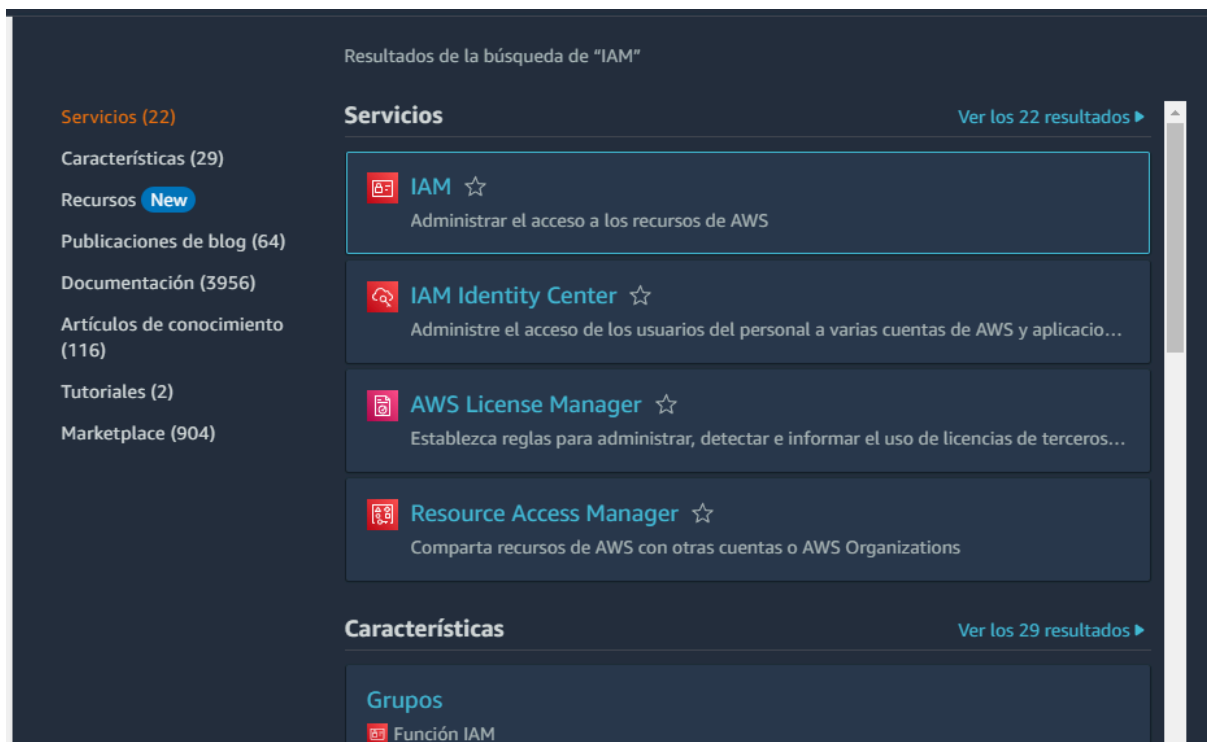
ella en cualquier momento. Pero ahora como es que podemos tomar estas imágenes a nuestra computadora local para entrenar a nuestro modelo.

Es esencial que para realizar esta actividad descarguemos [AWSCLIV2.msi](#), donde instalaremos en nuestra computadora la terminal de Amazon.



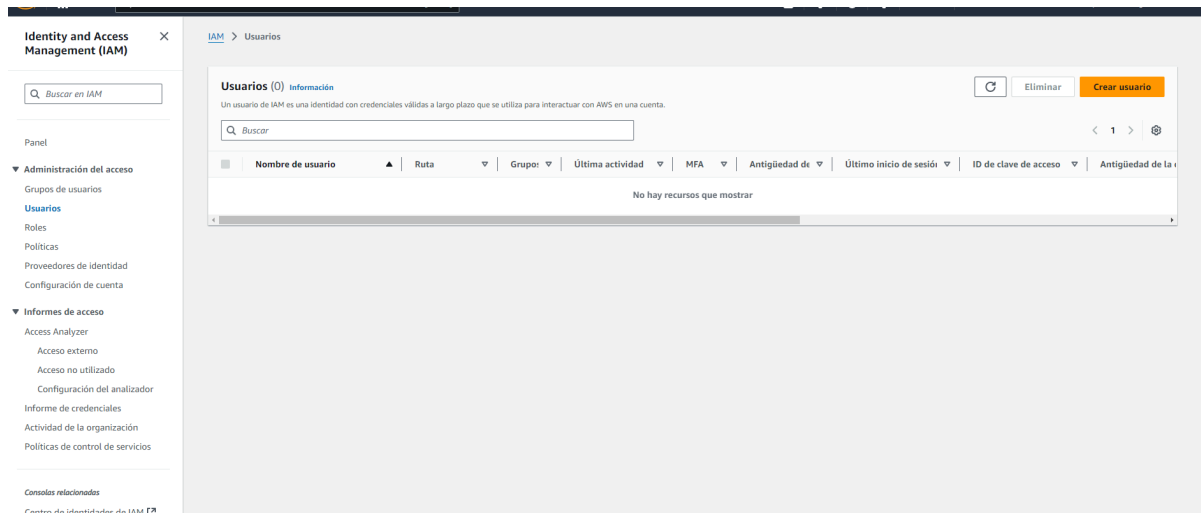
The screenshot shows the AWS Command Line Interface User Guide for Version 2. The left sidebar contains a navigation menu with sections like 'About the AWS CLI', 'Get started', 'Prerequisites', 'Install/Update', 'Past releases', 'Build and install from source', 'Amazon ECR Public/Docker', 'Setup', 'Configure the AWS CLI', 'Authentication and access credentials', 'Using the AWS CLI', 'Code examples', 'Security', 'Troubleshoot errors', 'Migration guide', 'Uninstall', and 'Document History'. The main content area is titled 'Linux' and 'macOS', with a section for 'Windows'. Under 'Windows', there is a section 'Install and update requirements' with bullet points: 'We support the AWS CLI on Microsoft-supported versions of 64-bit Windows.' and 'Admin rights to install software'. Below this is a section 'Install or update the AWS CLI' with instructions: 'To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the [AWS CLI version 2 Changelog](#) on GitHub.' The instructions include a numbered list: '1. Download and run the AWS CLI MSI installer for Windows (64-bit):' followed by the URL <https://awscli.amazonaws.com/AWSCLIV2.msi>. It also mentions an alternative command: 'Alternatively, you can run the `msiexec` command to run the MSI installer.' Below this is a code block showing the command: `C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi`. A note follows: 'For various parameters that can be used with `msiexec`, see [msiexec](#) on the Microsoft Docs website. For example, you can use the `/qn` flag for a silent installation.' Another code block shows the command: `C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn`. At the bottom, there is a numbered list: '2. To confirm the installation, open the **Start** menu, search for `cmd` to open a command prompt window, and at the command prompt use the `aws --`'.

Retomaremos este paso en breve, porque antes debemos de crear un Usuario, por lo que en Servicios de Amazon donde estaba nuestra bucket ahora buscaremos IAM.

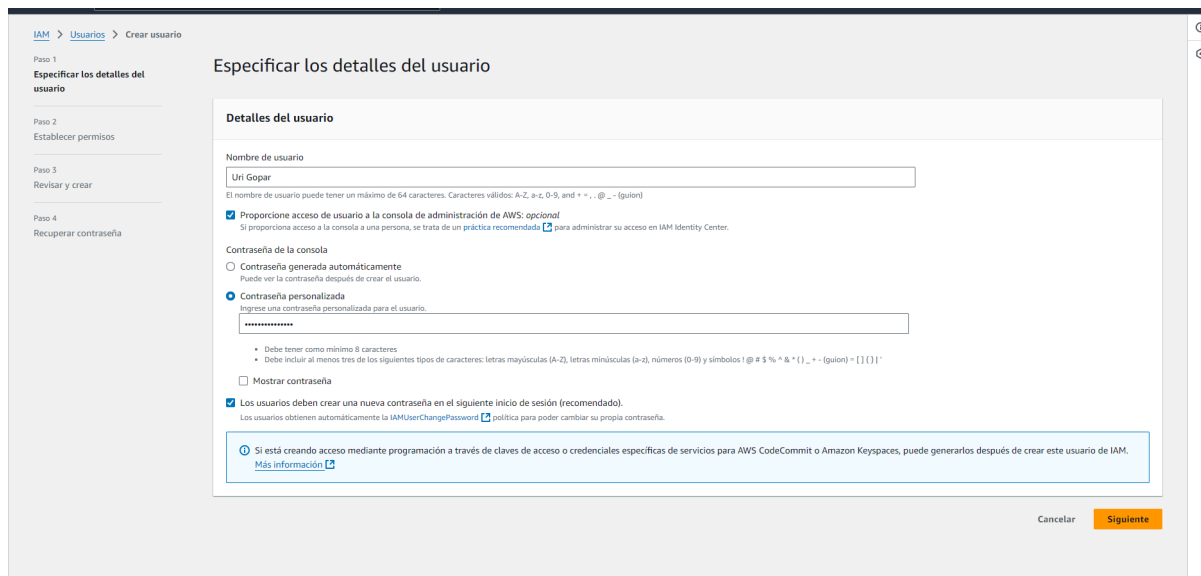


The screenshot shows the AWS IAM service search results page. The top section is titled 'Resultados de la búsqueda de "IAM"'. Below this is a sidebar with navigation links: 'Servicios (22)', 'Características (29)', 'Recursos **New**', 'Publicaciones de blog (64)', 'Documentación (3956)', 'Artículos de conocimiento (116)', 'Tutoriales (2)', and 'Marketplace (904)'. The main content area is titled 'Servicios' and shows a list of services: 'IAM' (Administrar el acceso a los recursos de AWS), 'IAM Identity Center' (Administre el acceso de los usuarios del personal a varias cuentas de AWS y aplicacio...), 'AWS License Manager' (Establezca reglas para administrar, detectar e informar el uso de licencias de terceros...), and 'Resource Access Manager' (Comparta recursos de AWS con otras cuentas o AWS Organizations). Below this is a section titled 'Características' and shows a list of features: 'Grupos' and 'Función IAM'.

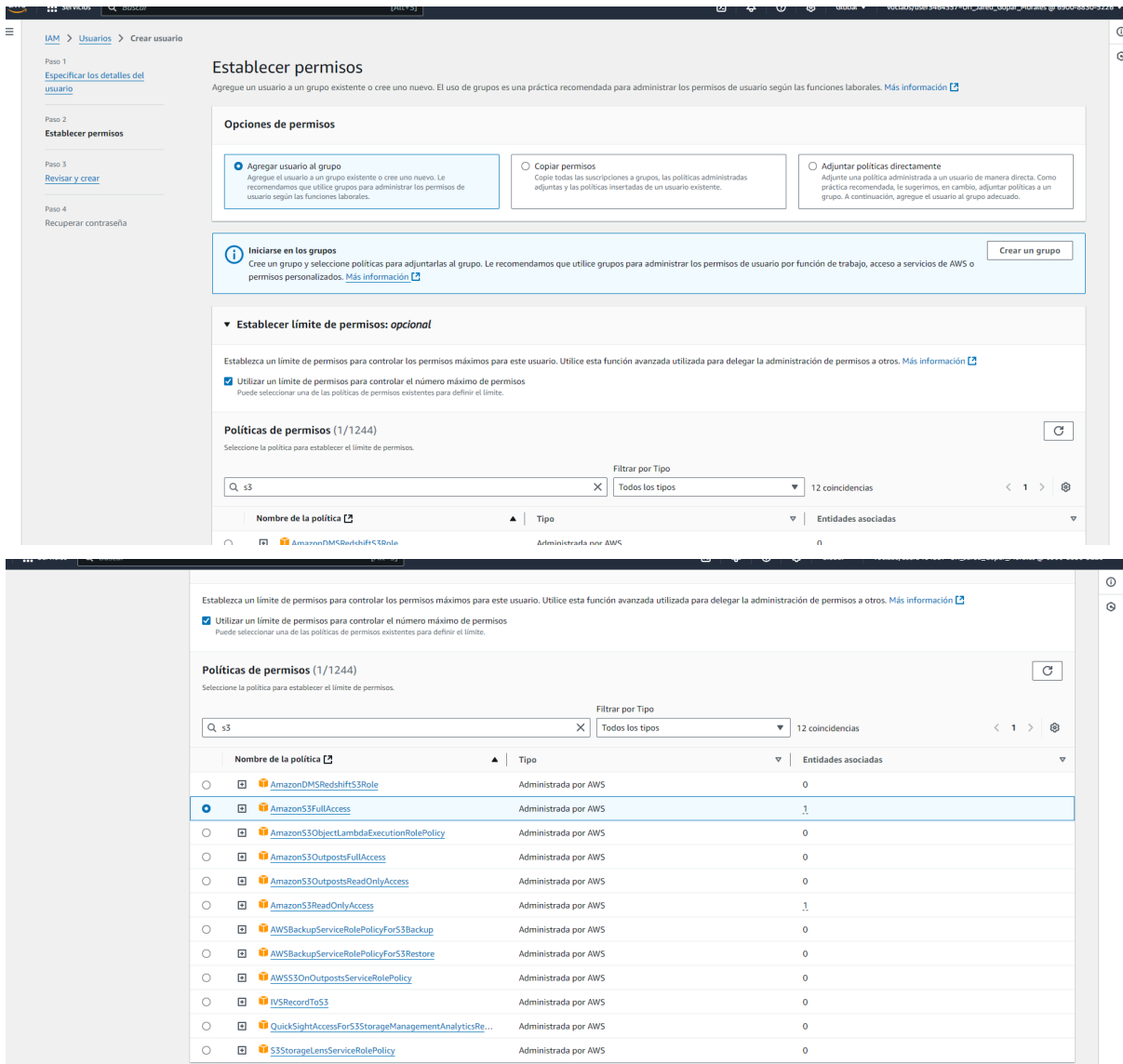
Entraremos y seleccionaremos crear Usuario



Nosotros le pusimos contraseña personalizada y que este usuario pueda acceder a la consola de administración.



En siguiente nos aparecerá la pestaña de establecer permisos, abriremos la pestaña de establecer límites de permisos y buscaremos en “S3”, en este caso quiero que sea un superusuario por lo que le daré acceso a total al S3, esto quiere decir que puede borrar, descargar o editar, los archivos de la bucket.



Establecer permisos

Agregue un usuario a un grupo existente o cree uno nuevo. El uso de grupos es una práctica recomendada para administrar los permisos de usuario según las funciones laborales. [Más información](#)

Opciones de permisos

- ☒ **Agregar usuario al grupo**
Agregue el usuario a un grupo existente o cree uno nuevo. Le recomendamos que utilice grupos para administrar los permisos de usuario según las funciones laborales.
- ☐ **Copiar permisos**
Copie todas las suscripciones a grupos, las políticas administradas adjuntas y las políticas insertadas de un usuario existente.
- ☐ **Adjuntar políticas directamente**
Adjunte una política administrada a un usuario de manera directa. Como práctica recomendada, le sugerimos, en cambio, adjuntar políticas a un grupo. A continuación, agregue el usuario al grupo adecuado.

Comenzar en los grupos
Cree un grupo y seleccione políticas para adjuntarlas al grupo. Le recomendamos que utilice grupos para administrar los permisos de usuario por función de trabajo, acceso a servicios de AWS o permisos personalizados. [Más información](#) [Crear un grupo](#)

Establecer límite de permisos: opcional

Establezca un límite de permisos para controlar los permisos máximos para este usuario. Utilice esta función avanzada utilizada para delegar la administración de permisos a otros. [Más información](#)

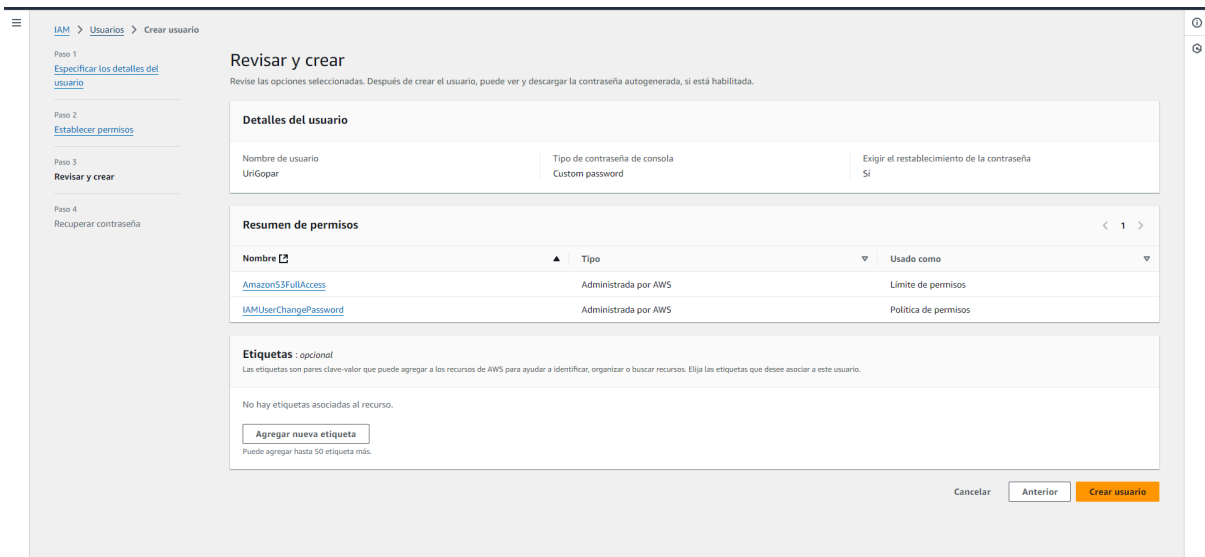
☒ **Utilizar un límite de permisos para controlar el número máximo de permisos**
Puede seleccionar una de las políticas de permisos existentes para definir el límite.

Políticas de permisos (1/1244)
Seleccione la política para establecer el límite de permisos.

Filtrar por Tipo: Todos los tipos 12 coincidencias

Nombre de la política	Tipo	Entidades asociadas
<input type="radio"/> AmazonDMSRedshiftS3Role	Administrada por AWS	0
<input checked="" type="radio"/> AmazonS3FullAccess	Administrada por AWS	1
<input type="radio"/> AmazonS3ObjectLambdaExecutionRolePolicy	Administrada por AWS	0
<input type="radio"/> AmazonS3OutpostsFullAccess	Administrada por AWS	0
<input type="radio"/> AmazonS3OutpostsReadOnlyAccess	Administrada por AWS	0
<input type="radio"/> AmazonS3ReadOnlyAccess	Administrada por AWS	1
<input type="radio"/> AWSBackupServiceRolePolicyForS3Backup	Administrada por AWS	0
<input type="radio"/> AWSBackupServiceRolePolicyForS3Restore	Administrada por AWS	0
<input type="radio"/> AWSSS3OnOutpostsServiceRolePolicy	Administrada por AWS	0
<input type="radio"/> IVSRecordToS3	Administrada por AWS	0
<input type="radio"/> QuickSightAccessForS3StorageManagementAnalyticsRe...	Administrada por AWS	0
<input type="radio"/> S3StorageLensServiceRolePolicy	Administrada por AWS	0

Al darle siguiente podemos ver el resumen del Usuario creado, así como sus permisos.



Revisar y crear

Revise las opciones seleccionadas. Después de crear el usuario, puede ver y descargar la contraseña autogenerada, si está habilitada.

Detalles del usuario

Nombre de usuario UriGopar	Tipo de contraseña de consola Custom password	Exigir el restablecimiento de la contraseña Sí
-------------------------------	--	---

Resumen de permisos

Nombre	Tipo	Usado como
AmazonS3FullAccess	Administrada por AWS	Límite de permisos
IAMUserChangePassword	Administrada por AWS	Política de permisos

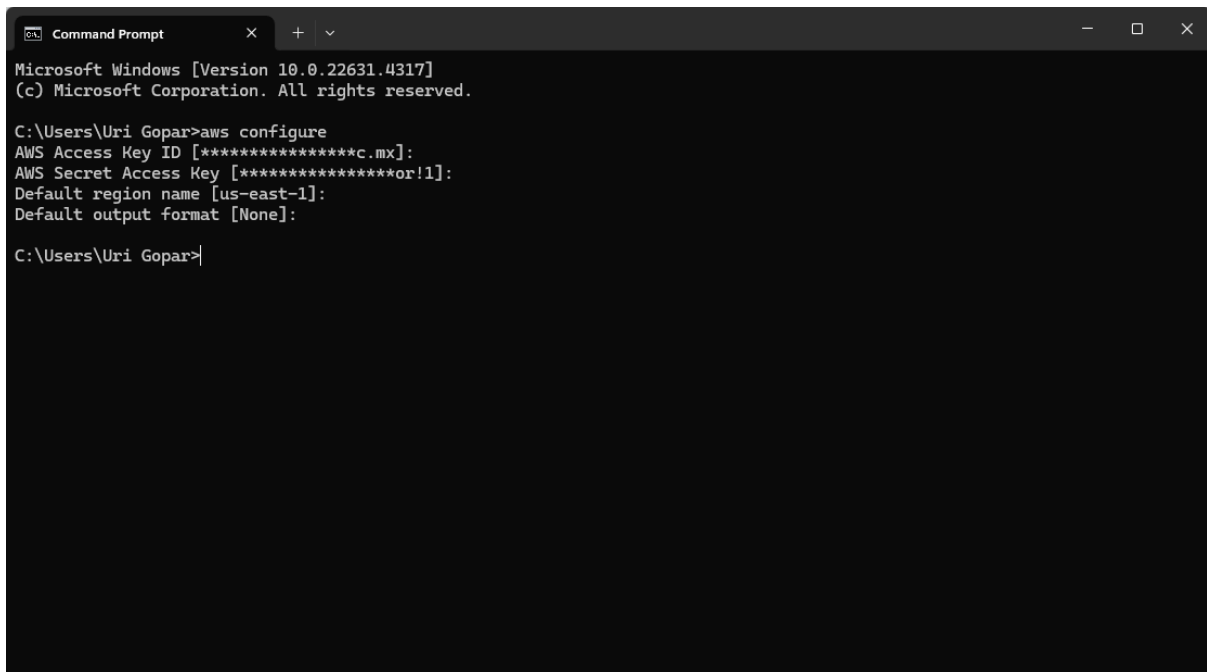
Etiquetas : opcional
Las etiquetas son pares clave-valor que puede agregar a los recursos de AWS para ayudar a identificar, organizar o buscar recursos. Elija las etiquetas que desee asociar a este usuario.

No hay etiquetas asociadas al recurso.

[Agregar nueva etiqueta](#)
Puede agregar hasta 50 etiquetas más.

[Cancelar](#) [Anterior](#) [Crear usuario](#)

Le damos crear. Una vez creado nos regresamos a nuestra cmd y escribimos el siguiente código:aws configure

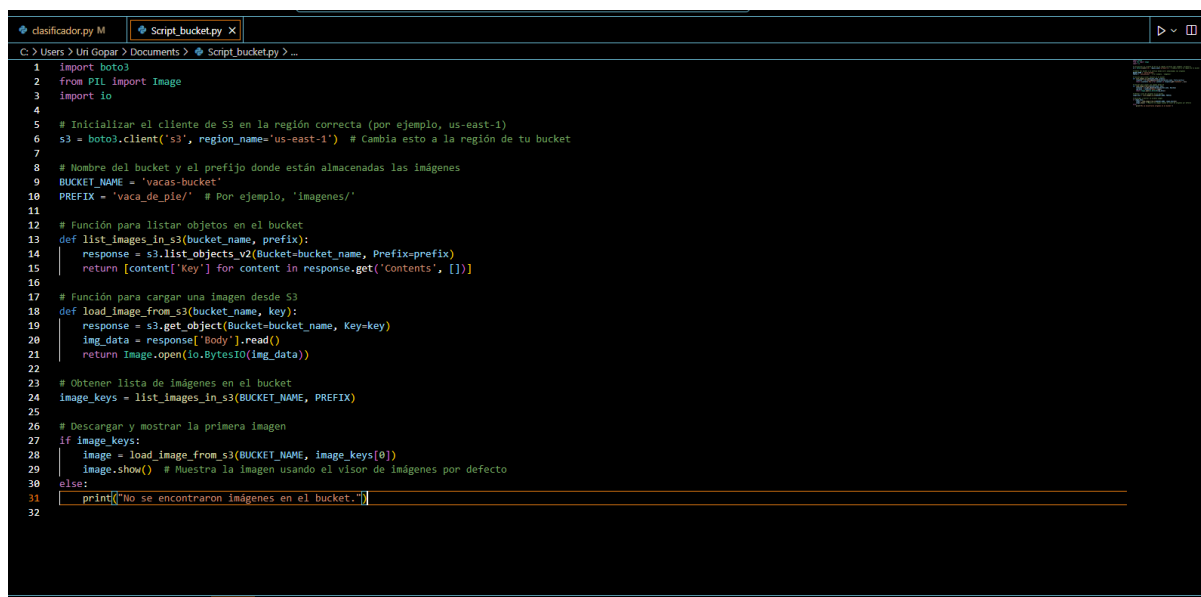


```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Uri Gopar>aws configure
AWS Access Key ID [*****c.mx]:
AWS Secret Access Key [*****or!1]:
Default region name [us-east-1]:
Default output format [None]:

C:\Users\Uri Gopar>
```

Pondremos nuestra información del Usuario que creamos y ya seremos capaces de poder descargar las imágenes de nuestro bucket, únicamente tendremos que correr este código, en python desde nuestro editor de texto de confianza.



```
1 import boto3
2 from PIL import Image
3 import io
4
5 # Inicializar el cliente de S3 en la región correcta (por ejemplo, us-east-1)
6 s3 = boto3.client('s3', region_name='us-east-1') # Cambia esto a la región de tu bucket
7
8 # Nombre del bucket y el prefijo donde están almacenadas las imágenes
9 BUCKET_NAME = 'vacas-bucket'
10 PREFIX = 'vaca_de_pie/' # Por ejemplo, 'imagenes/'
11
12 # Función para listar objetos en el bucket
13 def list_images_in_s3(bucket_name, prefix):
14     response = s3.list_objects_v2(Bucket=bucket_name, Prefix=prefix)
15     return [content['key'] for content in response.get('Contents', [])]
16
17 # Función para cargar una imagen desde S3
18 def load_image_from_s3(bucket_name, key):
19     response = s3.get_object(Bucket=bucket_name, Key=key)
20     img_data = response['Body'].read()
21     return Image.open(io.BytesIO(img_data))
22
23 # Obtener lista de imágenes en el bucket
24 image_keys = list_images_in_s3(BUCKET_NAME, PREFIX)
25
26 # Descargar y mostrar la primera imagen
27 if image_keys:
28     image = load_image_from_s3(BUCKET_NAME, image_keys[0])
29     image.show() # Muestra la imagen usando el visor de imágenes por defecto
30 else:
31     print("No se encontraron imágenes en el bucket.")
32
```

El cual lo que hace es descargar las imágenes por batches, ya que estas se descargan por lotes, que es una muy buena opción si no quieres saturar tu internet o almacenamiento, de igual forma esta opción es más rápida que descargar todas las imágenes.

Análisis de uso de big data

Big Data se refiere a conjuntos de datos extremadamente grandes y complejos que son difíciles de gestionar, procesar y analizar utilizando las herramientas y técnicas tradicionales de procesamiento de datos. Este tipo de datos tiene generalmente características conocidas como las "5 V's":

1. Volumen: Cantidad masiva de datos generados por diversas fuentes, como redes sociales, sensores, transacciones comerciales, etc.
2. Velocidad: La rapidez con la que los datos se generan, almacenan y analizan.
3. Variedad: Diversidad en los tipos de datos (estructurados, no estructurados, semiestructurados).
4. Veracidad: La calidad y precisión de los datos, lo cual es crucial en su análisis.
5. Valor: El valor que se puede extraer del análisis de esos datos para tomar decisiones.

Las técnicas de Big Data se utilizan en contextos donde los datos cumplen con una o más de estas características, como:

- Enormes cantidades de información de clientes, productos o servicios.
- Sistemas de sensores o dispositivos IoT que generan datos en tiempo real.
- Análisis científicos donde se necesitan procesar y analizar grandes volúmenes de datos.

Estas técnicas permiten encontrar patrones, realizar predicciones y tomar decisiones estratégicas a gran escala.

Entre las herramientas para Big Data se pueden encontrar las siguientes:

1. Procesamiento y análisis distribuido

- Apache Hadoop: Framework que permite el procesamiento distribuido de grandes conjuntos de datos a través de clusters. Su componente principal es Hadoop Distributed File System (HDFS) para almacenar datos.
- Apache Spark: Una plataforma de procesamiento distribuido que realiza tareas en memoria, lo que lo hace mucho más rápido que Hadoop para ciertos tipos de procesamiento.
 - PySpark: Es la API de Python para trabajar con Apache Spark.
- Apache Flink: Framework de procesamiento distribuido en tiempo real, diseñado para procesar flujos de datos continuos.
- Dask: Una biblioteca en Python que permite el procesamiento paralelo y distribuido de datos que no caben en memoria.

2. Almacenamiento distribuido

- Hadoop HDFS: Sistema de archivos distribuido de Hadoop que permite almacenar grandes volúmenes de datos distribuidos en varios nodos.
- Amazon S3: Almacenamiento escalable en la nube que se usa para almacenar grandes cantidades de datos.
- Apache Cassandra: Base de datos NoSQL distribuida diseñada para manejar grandes volúmenes de datos en varios servidores.
- Google Bigtable: Base de datos NoSQL de Google para aplicaciones que necesitan almacenar y consultar datos a gran escala.

3. Bases de datos NoSQL

- MongoDB: Base de datos NoSQL orientada a documentos, que es ideal para manejar grandes volúmenes de datos no estructurados.
- Apache HBase: Una base de datos NoSQL distribuida que trabaja sobre HDFS, ideal para el almacenamiento en tiempo real de datos de grandes cantidades.

- Couchbase: Base de datos NoSQL orientada a documentos, usada para aplicaciones que requieren alta disponibilidad y escalabilidad.

4. Procesamiento de datos en tiempo real

- Apache Kafka: Plataforma de mensajería y procesamiento de flujos de datos en tiempo real, utilizada para construir pipelines de datos distribuidos.
- Apache Storm: Herramienta de procesamiento en tiempo real para analizar datos en flujos continuos.

5. Ingesta de datos

- Apache NiFi: Herramienta para automatizar el flujo de datos entre sistemas, que permite la ingesta, transformación y análisis de datos de manera sencilla.
- Logstash: Herramienta de ingesta y procesamiento de logs y datos en tiempo real, que forma parte del stack ELK (Elasticsearch, Logstash, Kibana).

6. Almacenamiento y consulta de datos estructurados

- Apache Hive: Un sistema de data warehousing que corre sobre Hadoop, permite consultas tipo SQL (HQL) sobre grandes conjuntos de datos almacenados en HDFS.
- Presto: Motor de consultas distribuidas que permite realizar consultas SQL sobre datos en Hadoop y otros sistemas.
- Google BigQuery: Data warehouse en la nube de Google para realizar análisis de grandes volúmenes de datos de manera rápida utilizando SQL.

7. Visualización y análisis de datos

- Tableau: Herramienta de visualización de datos que permite conectar y analizar grandes conjuntos de datos.
- Apache Superset: Plataforma de visualización y análisis de datos que permite crear dashboards a partir de grandes volúmenes de datos.
- Power BI: Plataforma de Microsoft para visualización de datos que soporta la ingesta y análisis de grandes volúmenes de datos desde diversas fuentes.

8. Machine Learning en Big Data

- Apache Mahout: Librería de machine learning para crear algoritmos escalables sobre datos distribuidos.
- H2O.ai: Plataforma de machine learning escalable para grandes volúmenes de datos.

- **MLlib (Apache Spark):** Librería de Apache Spark para machine learning, que permite realizar algoritmos de aprendizaje automático sobre datos distribuidos.

9. Orquestación y gestión de flujos de trabajo

- *Apache Airflow:* Plataforma de orquestación de flujos de trabajo que permite automatizar pipelines de procesamiento de datos en Big Data.
- *Luigi:* Herramienta de orquestación de flujos de trabajo en Python, que permite la creación de pipelines de datos y su seguimiento.

El uso de tecnologías y técnicas de Big Data en escenarios con pocos datos puede ser innecesario y contraproducente por varias razones:

1. *Costos:* Las herramientas de Big Data (Hadoop, Spark, etc.) y la infraestructura necesaria para gestionarlas suelen ser costosas y requieren de recursos significativos, como servidores o almacenamiento en la nube.
2. *Complejidad:* El diseño e implementación de sistemas de Big Data puede ser complicado y requiere expertos en la materia. Si los datos son manejables con técnicas tradicionales, es innecesario introducir esta complejidad.
3. *Sub utilización:* Muchas herramientas de Big Data están diseñadas para manejar grandes volúmenes de datos de forma distribuida, lo que puede ser un desperdicio de recursos si se aplican a conjuntos de datos pequeños.
4. *Alternativas más simples:* Con pocos datos, las bases de datos relacionales o incluso un procesamiento en un solo servidor pueden ser más eficientes, fáciles de usar y económicos.

Por lo tanto, conviene utilizar Big Data únicamente cuando los datos y la situación lo justifiquen.

Referencias

Amazon Web Services. (n.d.). *¿Qué es Python?*. AWS. <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes>.

Amazon Web Services. (n.d.). *Guía del usuario de Amazon S3*. AWS. https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/Welcome.html