



# COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

ICADE

CIHS

## Sprint 1

**Grupo 2: María Carreño Nin de Cardona, Raquel  
Fernández Esquinas, Teresa X. Garvía Gallego e  
Isabel V. Morell Maudes**

Tecnologías de Procesamiento Big Data  
3º Grado en Ingeniería Matemática e Inteligencia Artificial

# Índice

INTRODUCCIÓN .....	3
METODOLOGÍA .....	4
RESULTADOS .....	5
CONCLUSIÓN .....	8

# Introducción

Este sprint marca el inicio del proyecto de Big Data sobre criptomonedas. Antes de poder realizar cualquier análisis, es fundamental la recopilación de los datos necesarios. Para ello, descargaremos información de TradingView y la almacenaremos utilizando Amazon S3, un servicio de AWS diseñado para el almacenamiento escalable y seguro de datos.

El objetivo principal de este sprint es desarrollar un script en Python que nos permita obtener, organizar y almacenar estos datos de manera eficiente. Tomar decisiones adecuadas sobre la estructura y almacenamiento de los datos es clave, ya que impactará directamente en la velocidad de descarga, acceso y procesamiento en futuras fases del proyecto.

Este proceso es crucial porque una gestión eficiente de los datos nos permitirá optimizar el rendimiento del sistema, garantizar la integridad de la información y facilitar su análisis en los próximos sprints.

# Metodología

Durante el desarrollo de este sprint, se ha optado por Python como lenguaje de programación debido a la disponibilidad de una librería denominada `TradingViewData`, que facilita la descarga de los datos de `TradingView` necesarios para el proyecto. En cuanto al almacenamiento de los datos, se ha utilizado `Amazon S3` de `AWS`, creando un bucket donde se alojan todos los archivos resultantes de la división de nuestros datos.

La arquitectura de la solución se organiza en dos bloques principales: la descarga de los datos y su posterior almacenamiento. Para optimizar el acceso y la organización de la información, hemos decidido estructurar los datos en función de tres parámetros: tipo de criptomoneda, año y mes. Esta estrategia permite que la recuperación de los datos sea más eficiente y rápida. La estructura adoptada es la siguiente:

```
Criptomoneda/  
|-- Año/  
    |-- Mes/  
        |-- Archivo.csv
```

En lo que respecta a la validación del proceso, se llevaron a cabo pruebas para asegurar que los datos se subieron correctamente al bucket de `S3`. Para ello, se accedió a una selección de archivos almacenados y se verificó que la estructura jerárquica se mantenía durante la transferencia. Además, para comprobar el correcto funcionamiento del script, se realizó una revisión del código en ejecución, donde no se generaron excepciones, lo que indicó que el proceso se llevó a cabo sin errores.

Para la automatización del proceso de carga de datos, se emplearon las librerías `io` y `boto3`. La librería `io` facilita la manipulación de archivos en memoria antes de su carga, mientras que `boto3` permite tanto la interacción con los servicios de `AWS`, como la creación de buckets y la subida de archivos a `S3`. De este modo, se asegura un flujo de trabajo eficiente y completamente automatizado en la gestión de datos financieros.

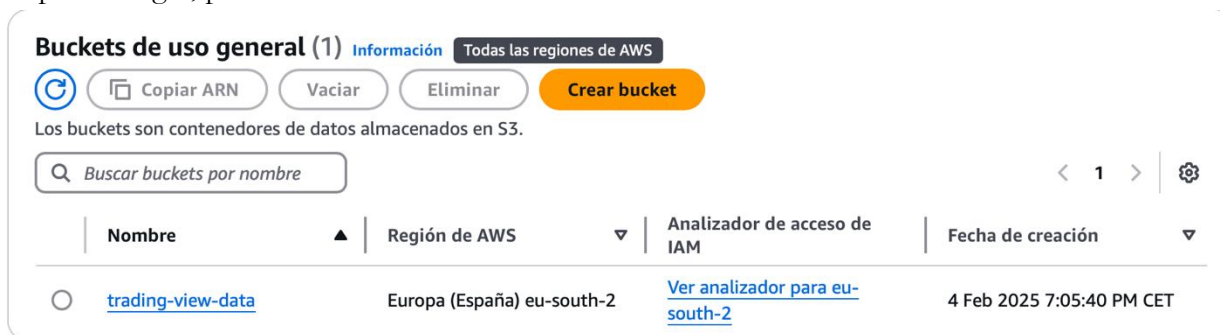
# Resultados

Durante este sprint, logramos establecer una estrategia eficiente para el almacenamiento de datos en Amazon S3, centralizando toda la información en un único bucket. Creemos que con esta decisión permitiremos optimizar el acceso a los datos, reducir la latencia y mejorar la organización del almacenamiento.

Además, implementamos un script en Python que nos permite automatizar la descarga de datos desde TradingView y los almacena en S3 siguiendo una estructura modular. Los datos se guardan en formato CSV, organizados por criptomoneda y período histórico, facilitando su acceso y análisis en futuras etapas del proyecto.

Siguiendo la jerarquía que hemos explicado en la sección anterior, en s3 podemos ver nuestro bucket que tiene esta forma:

En primer lugar, podemos observar el bucket creado:

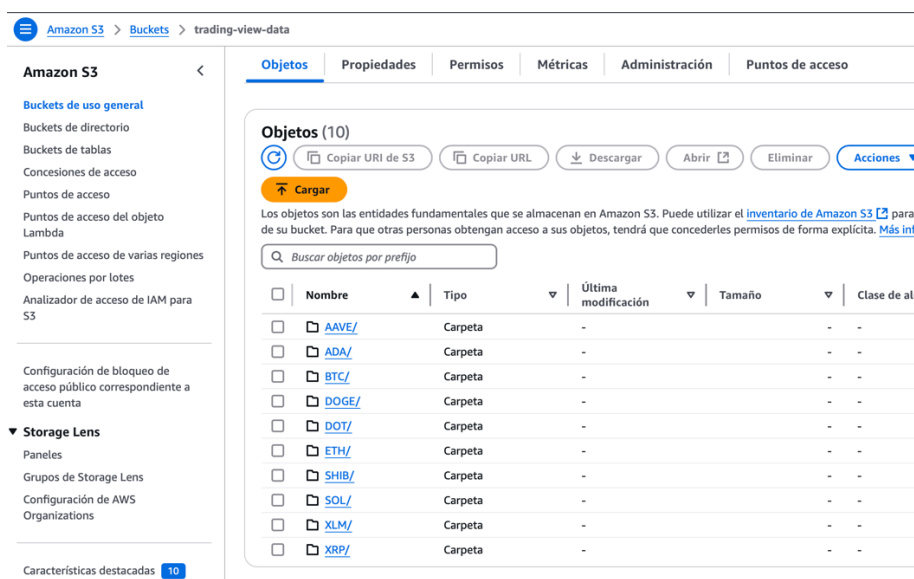


**Buckets de uso general (1)** Información Todas las regiones de AWS

Los buckets son contenedores de datos almacenados en S3.

Nombre	Región de AWS	Analizador de acceso de IAM	Fecha de creación
<a href="#">trading-view-data</a>	Europa (España) eu-south-2	<a href="#">Ver analizador para eu-south-2</a>	4 Feb 2025 7:05:40 PM CET

Dentro del bucket principal, encontramos una carpeta correspondiente a cada tipo de moneda:

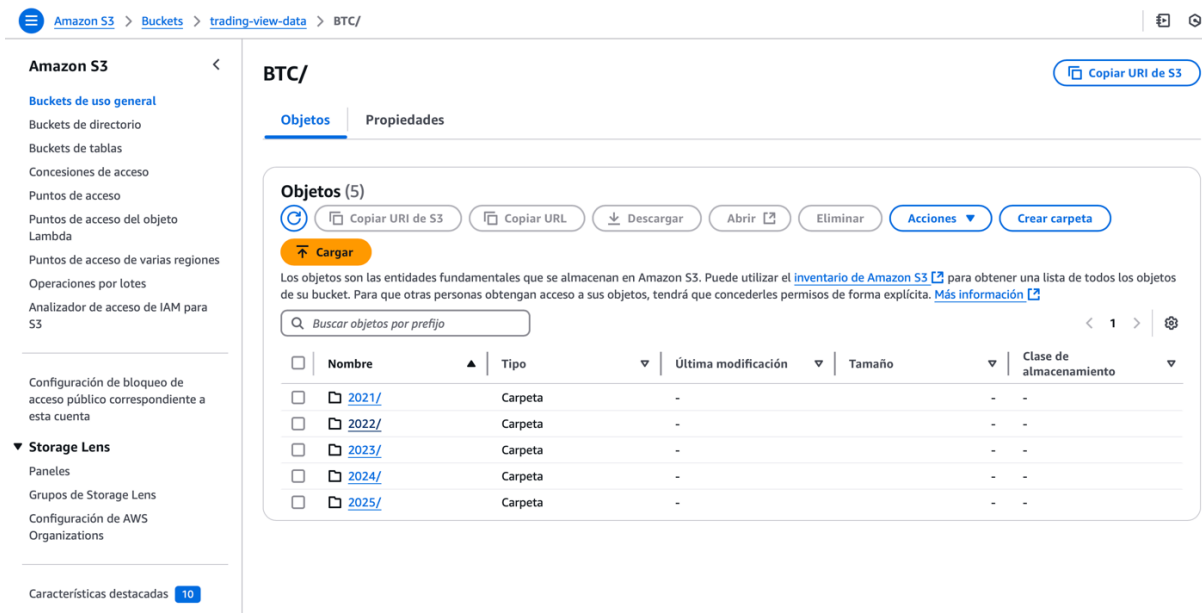


Amazon S3 > Buckets > trading-view-data

**Objetos (10)**

Nombre	Tipo	Última modificación	Tamaño	Clase de alm
<a href="#">AAVE/</a>	Carpeta	-	-	-
<a href="#">ADA/</a>	Carpeta	-	-	-
<a href="#">BTC/</a>	Carpeta	-	-	-
<a href="#">DOGE/</a>	Carpeta	-	-	-
<a href="#">DOT/</a>	Carpeta	-	-	-
<a href="#">ETH/</a>	Carpeta	-	-	-
<a href="#">SHIB/</a>	Carpeta	-	-	-
<a href="#">SOL/</a>	Carpeta	-	-	-
<a href="#">XLM/</a>	Carpeta	-	-	-
<a href="#">XRP/</a>	Carpeta	-	-	-

Dentro de cada moneda, podemos ver varias carpetas de años donde se encuentran los diferentes datos:



Amazon S3 > Buckets > trading-view-data > BTC/

**Amazon S3**

- Buckets de uso general
  - Buckets de directorio
  - Buckets de tablas
  - Concesiones de acceso
  - Puntos de acceso
  - Puntos de acceso del objeto
  - Lambda
  - Puntos de acceso de varias regiones
  - Operaciones por lotes
  - Analizador de acceso de IAM para S3
- ▼ **Storage Lens**
  - Paneles
  - Grupos de Storage Lens
  - Configuración de AWS Organizations
- Características destacadas 10

**BTC/** Copiar URI de S3

**Objetos** Propiedades

**Objetos (5)**

Copiar URI de S3 Copiar URL Descargar Abrir Eliminar Acciones Crear carpeta

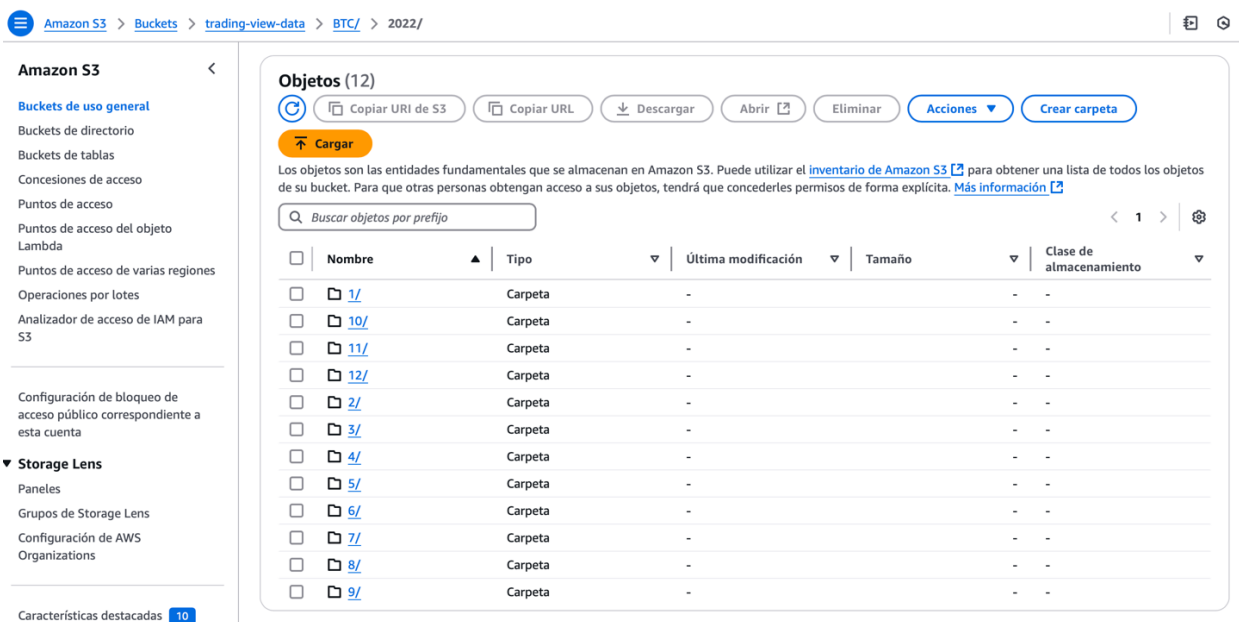
Cargar

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	2021/	Carpeta	-	-	-
<input type="checkbox"/>	2022/	Carpeta	-	-	-
<input type="checkbox"/>	2023/	Carpeta	-	-	-
<input type="checkbox"/>	2024/	Carpeta	-	-	-
<input type="checkbox"/>	2025/	Carpeta	-	-	-

Dentro de cada carpeta de año, se encuentra cada carpeta correspondiente con un mes, donde se almacena los datos:



Amazon S3 > Buckets > trading-view-data > BTC/ > 2022/

**Amazon S3**

- Buckets de uso general
  - Buckets de directorio
  - Buckets de tablas
  - Concesiones de acceso
  - Puntos de acceso
  - Puntos de acceso del objeto
  - Lambda
  - Puntos de acceso de varias regiones
  - Operaciones por lotes
  - Analizador de acceso de IAM para S3
- ▼ **Storage Lens**
  - Paneles
  - Grupos de Storage Lens
  - Configuración de AWS Organizations
- Características destacadas 10

**Objetos (12)**

Copiar URI de S3 Copiar URL Descargar Abrir Eliminar Acciones Crear carpeta

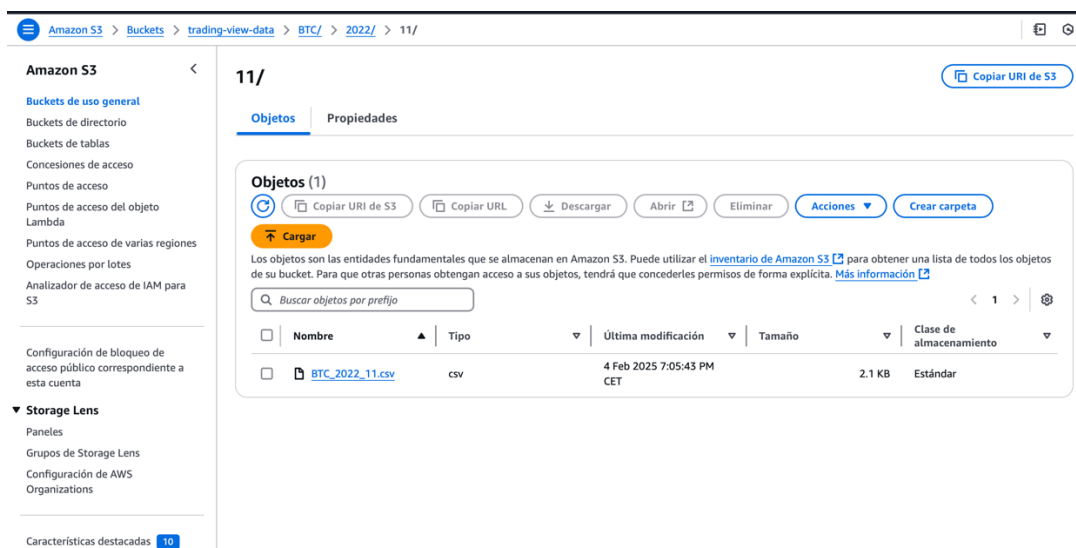
Cargar

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	1/	Carpeta	-	-	-
<input type="checkbox"/>	10/	Carpeta	-	-	-
<input type="checkbox"/>	11/	Carpeta	-	-	-
<input type="checkbox"/>	12/	Carpeta	-	-	-
<input type="checkbox"/>	2/	Carpeta	-	-	-
<input type="checkbox"/>	3/	Carpeta	-	-	-
<input type="checkbox"/>	4/	Carpeta	-	-	-
<input type="checkbox"/>	5/	Carpeta	-	-	-
<input type="checkbox"/>	6/	Carpeta	-	-	-
<input type="checkbox"/>	7/	Carpeta	-	-	-
<input type="checkbox"/>	8/	Carpeta	-	-	-
<input type="checkbox"/>	9/	Carpeta	-	-	-

Por último, si nos metemos en una carpeta de un mes, podemos observar el fichero CSV donde se almacenan los datos:



Después de probar el script y de añadir los datos al bucket de s3, el sistema se comportó según lo esperado, cumpliendo con los objetivos del sprint. Se logró:

- Automatizar la descarga y almacenamiento de datos en el sistema sin intervención manual
- Organizar los datos de manera eficiente dentro de Amazon S3, lo que nos facilitará su uso en análisis posteriores.
- Reducir la latencia en el acceso a los datos, gracias a la estrategia de almacenamiento centralizado.

Sin embargo, durante la implementación surgieron algunos desafíos. Por un lado, el tiempo de carga de grandes volúmenes de datos fue mayor de lo esperado, que lo resolvimos mediante la modificación de la configuración de concurrencia en las solicitudes. Por otro lado, tuvimos que validar que el formato CSV es el más adecuado para nuestro flujo de trabajo, asegurando su compatibilidad con herramientas de análisis.

En general, los resultados obtenidos han sentado una base sólida para los próximos Sprints, donde se enfocará en mejorar el rendimiento del almacenamiento y optimizar el procesamiento de datos.

## Conclusión

En este primer sprint hemos sido capaces de extraer una serie de datos de la página web “Trading View”. En concreto hemos podido extraer los datos relacionados con las siguientes criptomonedas:

De cada criptomoneda hemos extraído la información por años, y dentro de cada año por meses. De esta forma, al tener mayor división en los datos, tenemos un acceso más eficiente. Una vez hemos extraído y dividido los datos correctamente, hemos pasado a crear los csv con la información. Todos los datos los hemos almacenado en un bucket en AWS, para después poder guardar la información en una base de datos. Esto nos permitirá visualizar información y obtener diferentes conclusiones gracias a la tecnología de la nube.

En este sprint hemos logrado comprender el proyecto más a fondo, así como la página de donde proviene la información de las criptomonedas. Ahora entendemos mejor como extraer conocimiento de la página web, y conocemos más sobre la estructura de los datos, ya que hemos visualizado los csv para comprobar que se haya descargado todo correctamente.