

Librería AWS S3 .NET

Contenido

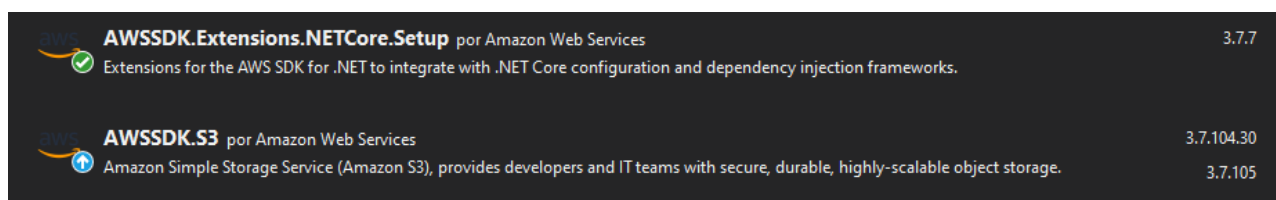
1.	RESUMEN DE LA LIBRERÍA.....	2
2.	RECURSOS DE MATERIALES.....	3
2.1	Análisis de requisitos.....	3
2.1.1	Objetivos y alcance.....	3
2.2	Diseño	3
2.2.1	Diseño Técnico de la solución	3
2.2.2	Funciones.....	3
2.2.2.1	addFile	3
2.2.2.2	deleteFile	4
2.2.2.3	downloadFile	4
2.2.2.4	downloadAllFilesFromFolder	5
2.2.2.5	getFilename	6
2.2.2.6	getAllFileNamesFromFolder.....	6
2.2.2.7	downloadAllFilesWith.....	7
2.2.2.8	getAllFilesURLWith	8
2.2.2.9	createFolder	8
2.2.2.10	deleteFolder.....	9
2.2.2.11	existFolder.....	10
2.2.2.12	getBytes.....	10
3.	INSTALACION EN OTROS PROYECTOS	11
4.	RECURSOS DE MATERIALES.....	12
5.1	Implementación.....	12

1. RESUMEN DE LA LIBRERÍA

El objetivo de la librería es poder utilizar los diferentes servicios de Amazon Web Services en cualquiera de nuestros proyectos. Inicialmente la librería se va a crear para poder utilizar los servicios de Amazon S3 para el almacenamiento de archivos, pero se puede actualizar para introducir nuevos servicios de AWS que se quieran incluir.

Para poder utilizar los servicios de AWS en nuestra propia librería necesitaremos instalar previamente en Visual Studio Code los siguientes paquetes NuGet:

- AWSSDK.S3
- AWSSDK.Extensions.NETCore.Setup



Para poder entender cómo funciona la librería es necesario entender los diferentes componentes básicos de Amazon S3 que se utilizan en el código de nuestra librería:

- Bucket: contenedor de almacenamiento en la nube que permite almacenar y organizar datos de forma segura. Es similar a un directorio o carpeta en un sistema de archivos tradicional. Cada "bucket" tiene un nombre único globalmente y se utiliza para almacenar y recuperar objetos, como archivos, imágenes o documentos.
- Región AWS: una ubicación geográfica física donde se encuentran los centros de datos de Amazon Web Services (AWS) que almacenan y operan los servicios de Amazon S3.
- El AccessKey y SecretAccessKey son credenciales de seguridad utilizadas para autenticar y autorizar el acceso de un usuario a los servicios de Amazon S3. Estas credenciales de acceso son proporcionadas por AWS cuando se crea un usuario en el servicio IAM y se pueden crear diferentes usuarios con diferentes permisos.

2. RECURSOS DE MATERIALES

2.1 Análisis de requisitos

2.2 Diseño

A continuación, se pasa a detallar el diseño tanto técnico como funcional de la librería

2.2.1 Diseño Técnico de la solución

Se van a explicar a nivel técnico las tablas implementadas en la base de datos, que son las que almacenan la información, así como las relaciones entre ellas.

2.2.2 Funciones

2.2.2.1 addFile

Esta función se va a utilizar para poder subir un archivo a un bucket de Amazon S3. Esta función devolverá un string con el nombre completo del archivo.

El archivo que se subirá al contenedor contendrá al principio del nombre del archivo un long referente a la fecha y hora de subida, con esto evitamos que se sobrescriban archivos que sean distintos y se llaman igual.

CABECERA DE LA FUNCION:

```
public async static Task<int> addFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, HttpPostedFile file)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.

- Folder: string que referencia la ruta donde se quiere subir el archivo dentro del bucket
- File: HttpPostedFile que contiene el archivo con toda la información

2.2.2.2 deleteFile

Esta función se va a utilizar para poder eliminar un archivo de un bucket de Amazon S3 y almacenarlo en una carpeta de seguridad (backup). Esta función devolverá un entero indicando si la operación se ha realizado correctamente o no.

CABECERA DE LA FUNCION:

```
public async static Task<int> deleteFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- Folder: string que referencia la ruta donde se quiere subir el archivo dentro del bucket
- fileName: string que contiene el nombre del archivo que se desea eliminar (ej: archivo.docx)

2.2.2.3 downloadFile

Esta función se va a utilizar para poder descargar un archivo de un bucket de Amazon S3. Esta función devolverá un diccionario con una única posición, cuya clave es el nombre del archivo y el valor es el contenido del fichero.

CABECERA DE LA FUNCION:

```
public async static Task<Dictionary<string,Stream>> downloadFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- Folder: string que referencia la ruta donde se quiere subir el archivo dentro del bucket
- fileName: string que contiene el nombre del archivo que se desea descargar (ej: archivo.docx)

2.2.2.4 downloadAllFilesFromFolder

Esta función se va a utilizar para poder descargar todos los archivos contenidos en una carpeta dentro del bucket de Amazon S3. Esta función devolverá un diccionario cuyas claves son los nombres de los archivos, en dichos nombres si aparecerá la fecha y hora que tienen los archivos al principio del nombre, y el valor es el contenido de dichos archivos.

CABECERA DE LA FUNCION:

```
public async static Task<Dictionary<string, Stream>> downloadAllFilesFromFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- Folder: string que referencia la ruta donde se quiere subir el archivo dentro del bucket

2.2.2.5 getFilename

Esta función se va a utilizar para poder obtener el nombre completo de un archivo, incluido el long que se les asigna a los archivos al inicio del nombre, cuando solo se envía el nombre del archivo sin el long.

Esta función devolverá el nombre completo del fichero en caso correcto, y un null en caso contrario.

CABECERA DE LA FUNCION:

```
public async static Task<string> getFilename(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- Folder: string que referencia la ruta donde se quiere subir el archivo dentro del bucket
- fileName: string que contiene el nombre del archivo sin el long

2.2.2.6 getAllFileNamesFromFolder

Esta función se va a utilizar para poder obtener todos los nombres de los archivos contenidos en una carpeta dentro del bucket de Amazon S3. Esta función devolverá una lista de strings con los nombres de los archivos, incluyendo la fecha y hora que tienen al principio del nombre.

CABECERA DE LA FUNCION:

```
public async static Task<List<string>> allFileNamesFromFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- Folder: string que referencia la ruta donde se quiere subir el archivo dentro del bucket

2.2.2.7 downloadAllFilesWith

Esta función se va a utilizar para poder buscar un archivo que este dentro del bucket pero sin indicar la carpeta donde se encuentra. Esta función devolverá un diccionario con una única posición, cuya clave es el nombre del archivo, en dichos nombres si aparecerá la fecha y hora que tienen los archivos al principio del nombre, y el valor es el contenido.

CABECERA DE LA FUNCION:

```
public async static Task<Dictionary<string,Stream>> searchFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string fileName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- Filename: string que referencia al nombre del archivo que se desea buscar

2.2.2.8 getAllFilesURLWith

Esta función se va a utilizar para poder obtener la ruta de un archivo sin indicar la carpeta dentro del bucket de Amazon S3. Esta función devolverá un string con la ruta.

CABECERA DE LA FUNCION:

```
public async static Task<string> searchFileURL(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string fileName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- FileName: string al nombre del archivo

2.2.2.9 createFolder

Esta función se va a utilizar para crear una carpeta dentro del bucket de Amazon S3. Esta función devolverá un entero indicando el estado de la operación

CABECERA DE LA FUNCION:

```
public async static Task<int> createFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string FolderName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- FolderName: string del nombre del archivo

2.2.2.10 deleteFolder

Esta función se va a utilizar para eliminar una carpeta dentro del bucket de Amazon S3. Esta función devolverá un entero indicando el estado de la operación

CABECERA DE LA FUNCION:

```
public async static Task<int> deleteFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string FolderName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket.
- FolderName: string del nombre del archivo

2.2.2.11 existFolder

Esta función se va a utilizar para buscar una carpeta dentro del bucket de Amazon S3. Esta función devolverá un booleano indicando el estado de la operación

CABECERA DE LA FUNCION:

```
public async static Task<bool> existFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string FolderName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario
- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:

<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

Región de AWS

UE (Irlanda) eu-west-1

- Bucket: string que contiene el nombre del bucket. Se obtiene de:
- FolderName: string del nombre del archivo

2.2.2.12 getBytes

Esta función se va a utilizar para poder obtener el tamaño en bytes de un archivo que este dentro del bucket. Esta función devolverá un entero indicando los bytes.

CABECERA DE LA FUNCION:

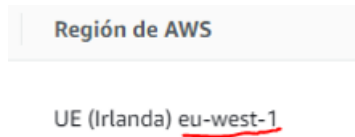
```
public async static Task<double> getBytes(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName)
```

PARÁMETROS DE LA FUNCION:

- AccessKey : string que hace referencia a la credencial publica del usuario
- SecretAccessKey: string que hace referencia a la credencial secreta del usuario

- ServiceURL: string que contiene la región AWS donde se encuentre. Las url siguen este formato:
<https://s3.<region>.amazonaws.com>

Donde región se obtiene de:

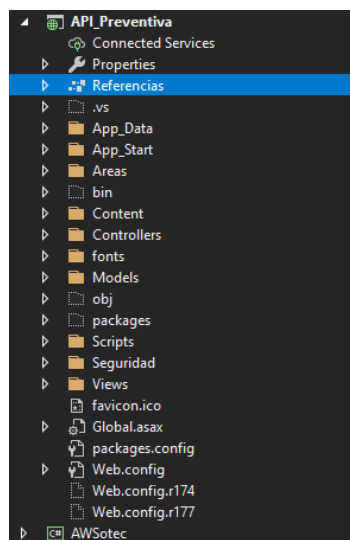


- Bucket: string que contiene el nombre del bucket.
- Folder: string que referencia a la ruta de la carpeta donde se encuentra el archivo
- Filename: string que referencia al nombre del archivo que se desea buscar

3. INSTALACION EN OTROS PROYECTOS

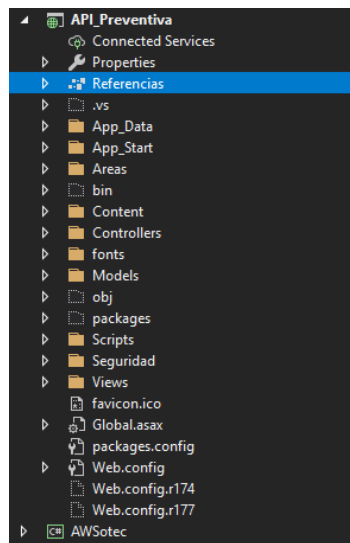
Para instalar la librería en nuestro proyectos podremos utilizar 2 implementaciones diferentes:

- Para la **primera opción** deberemos seguir los siguientes pasos:
 1. Abrir el proyecto donde queramos utilizar la librería
 2. Incluir la librería 'AWS3' dentro de nuestro proyecto.
 3. Dentro de nuestro proyecto nos dirigimos al apartado de 'Referencias':



4. Pulsaremos click derecho y seleccionaremos la opción 'Agregar referencia'

5. Se desplegará un mensaje un menú, pulsamos sobre la opción 'Proyectos' en la parte izquierda, seleccionamos la librería 'AWS3' y aceptamos.
 6. Nos aparecerá la librería en la pestaña 'Referencias' y ya podremos usarla.
- Para la **segunda opción** deberemos:
 1. Deberemos guardar/copiar el archivo 'AWS3.dll' que se encuentra dentro del proyecto de la librería, en la ruta: AWS3\bin\Release
 2. Abrir el proyecto donde queramos utilizar la librería
 3. Dentro de nuestro proyecto nos dirigimos al apartado de 'Referencias':



4. Pulsaremos click derecho y seleccionaremos la opción 'Agregar referencia'
5. Se desplegará un mensaje un menú, pulsamos sobre la opción 'Examinar', seleccionamos el archivo 'AWS3.dll' que hemos guardado previamente y aceptamos.
6. Nos aparecerá la librería en la pestaña 'Referencias' y ya podremos usarla

4. RECURSOS DE MATERIALES

5.1 Implementación

5.1.1. S3.cs

- **public async static Task<int> addFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, HttpPostedFileBase file):** función encargada para subir un archivo (parámetro file) a un bucket cuyo nombre se ha especificado en el parámetro 'Bucket' en la carpeta 'Folder'. Se subirá con los permisos

del usuario que se especifica con los parámetros 'AccessKey' y 'SecretAccessKey'. La función nos devolverá un entero indicando el estado de la operación.

- **public async static Task<int> deleteFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName):** función encargada para eliminar un archivo cuyo nombre se indica en el parámetro fileName a un bucket cuyo nombre se ha especificado en el parámetro 'Bucket' en la carpeta 'Folder'. Se eliminará con los permisos del usuario que se especifica con los parámetros 'AccessKey' y 'SecretAccessKey'. La función nos devolverá un entero indicando el estado de la operación.
- **public async static Task<Dictionary<string,Stream>> downloadFile(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName):** función encargada para descargar un archivo cuyo nombre se indica en el parámetro fileName de un bucket cuyo nombre se ha especificado en el parámetro 'Bucket' en la carpeta 'Folder'. Se descargará con los permisos del usuario que se especifica con los parámetros 'AccessKey' y 'SecretAccessKey'. La función nos devolverá un diccionario con una única posición, en el cual la clave hace referencia al nombre del archivo y el valor al contenido del mismo.
- **public async static Task<Dictionary<string, Stream>> downloadAllFilesFromFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder):** función muy similar a la anterior, cuya diferencia es que no indicaremos el nombre del archivo, solo indicaremos la carpeta de la que nos queremos descargar todos los archivos. La función nos devolverá un diccionario cuyas claves hacen referencia a los nombres de los archivos y el valor al contenido de los mismos.
- **public async static Task<List<string>> getAllFileNamesFromFolder (string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder):** función igual que la anterior pero en vez de obtener el contenido de todos los ficheros, obtendremos el nombre de los archivos contenidos en esa carpeta. La función devolverá una lista con el nombre de todos los archivos.
- **public async static Task<Dictionary<string,Stream>> downloadAllFilesWith (string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string fileName):** función encargada de obtener todos los archivos sin indicar su ruta (la carpeta) que contengan el string fileName. La función devolverá un diccionario, donde la clave es el nombre del fichero y el valor el contenido del mismo.
- **public async static Task<string> getAllFilesURLWith (string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string fileName):** función encargada de devolver la ruta de todos los

archivos cuyo nombre contenga el string que se pasa por el parámetro `fileName`. La función devolverá una lista con los strings de la ruta de los archivos.

- **`public async static Task<int> createFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string FolderName)`**: función encargada de crear una carpeta dentro de un bucket. La función devolverá un entero indicando con un 0 que se ha creado la carpeta, 1 para indicar que ya existe la carpeta y 2 en caso de error
- **`public async static Task<int> deleteFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string FolderName)`**: función encargada de eliminar una carpeta y todo su contenido. La función devolverá un entero indicando el estado de la operación.
- **`public async static Task<bool> existFolder(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string FolderName)`**: función que nos indica si una carpeta existe en el bucket o no.
- **`public async static Task<double> getBytes(string AccessKey, string SecretAccessKey, string ServiceURL, string Bucket, string Folder, string fileName)`**: función para obtener el tamaño en bytes del archivo llamado `fileName` que se encuentra en la ruta 'Folder' del bucket.

5.2. Referencias

- **AWSSDK.S3**: biblioteca proporcionada por Amazon Web Services (AWS) que permite interactuar con el servicio de almacenamiento en la nube Amazon S3 que facilita la integración de funcionalidades de Amazon S3 en tus aplicaciones .NET. Proporciona una serie de clases y métodos que permiten realizar operaciones comunes, como subir y descargar archivos, administrar permisos, trabajar con "buckets" (contenedores de almacenamiento) y realizar operaciones de bajo nivel con objetos almacenados en Amazon S3.
- **AWSSDK.Extensions.NETCore.Setup**: extensión de la biblioteca "AWSSDK" proporcionada por Amazon Web Services (AWS) para facilitar la configuración de servicios de AWS. Al utilizar la librería "AWSSDK.Extensions.NETCore.Setup" en Visual Studio, se proporciona una forma fácil y conveniente de configurar servicios de AWS en una aplicación .NET Core. La librería incluye extensiones y métodos de configuración que se integran con la funcionalidad de configuración estándar de .NET Core permitiendo una configuración flexible y modular de los servicios de AWS.