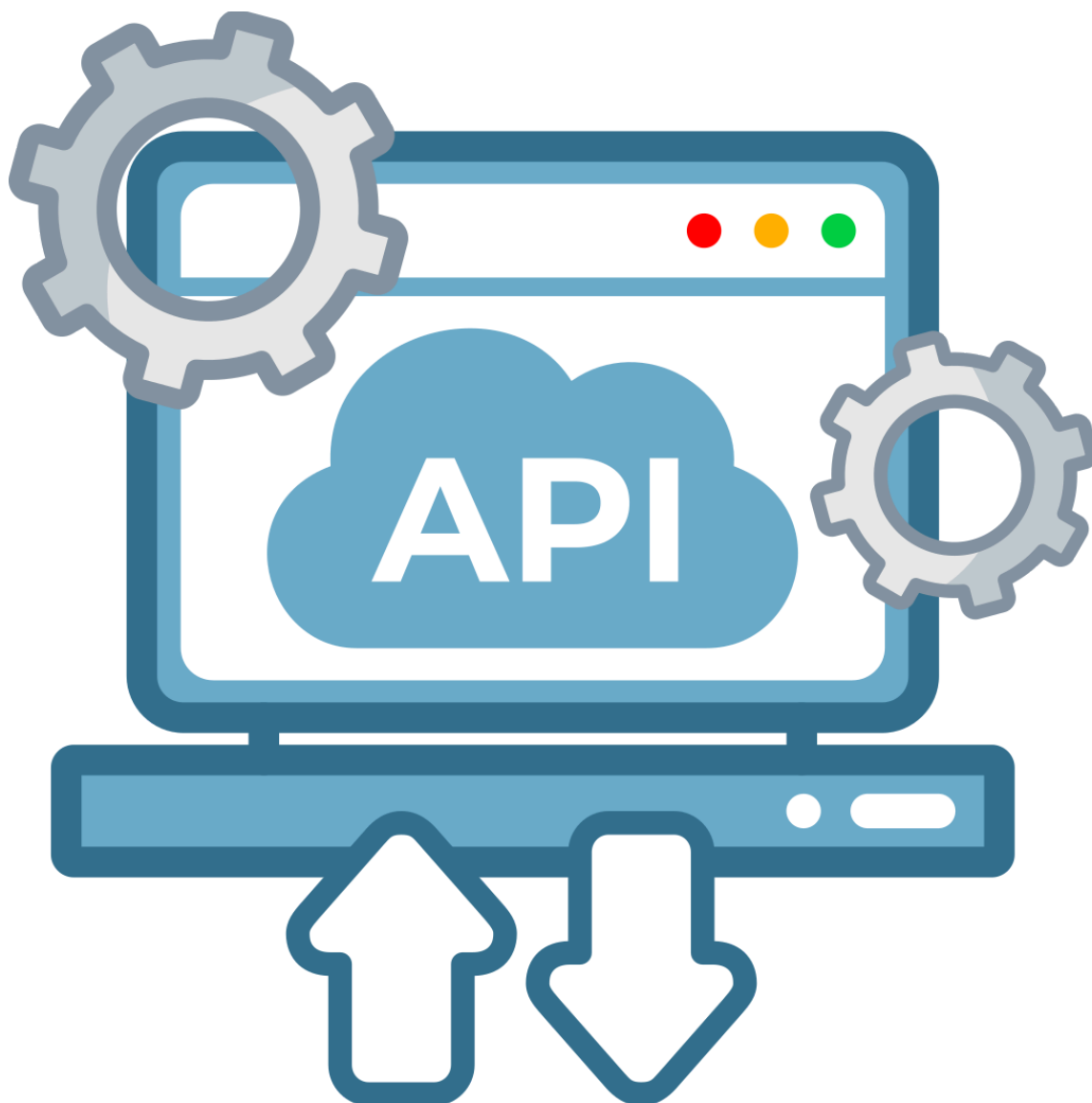


## Digitaldocu API

Versión 2.0

Última actualización del documento: 20/10/2025

# Documentación API General



# Índice de contenidos

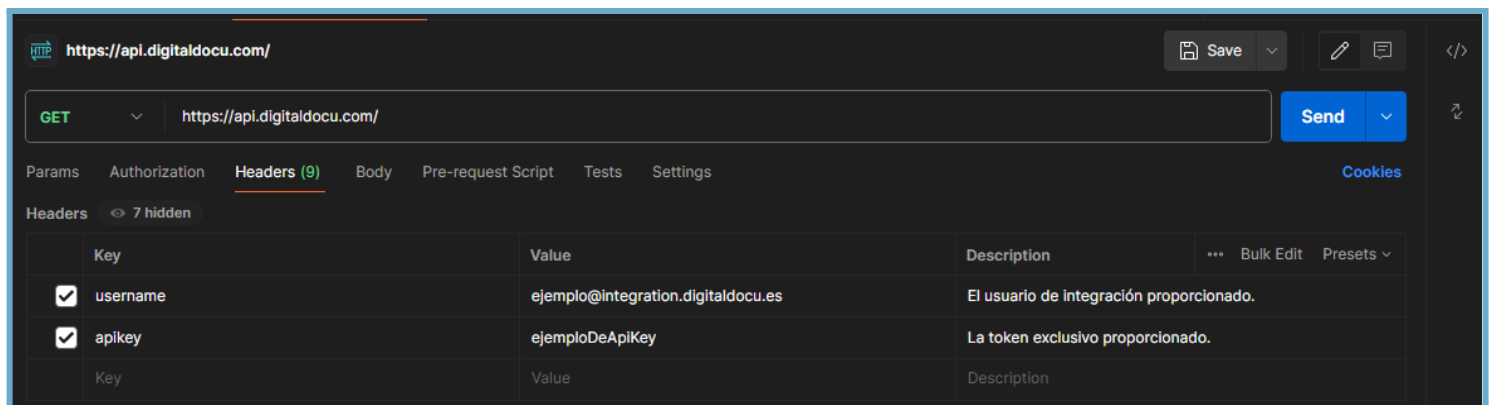
<b>INTRODUCCIÓN</b>	<b>3</b>
<b>ENDPOINTS DE LA API</b>	<b>3</b>
<b>DOCUMENTOS</b>	<b>3</b>
POST - UploadAndSave	3
PUT - Update	7
DELETE - Document	8
GET - Info	9
GET - Keywords	10
GET - Related	12
POST - Related	13
POST - Concatenate	13
GET - DocumentSameName	14
GET - ShareLink	14
POST - Share	15
GET - View	16
GET - Attached	16
PUT - Attached	17
GET - GetInboxFiles	17
POST - UploadDocument	18
POST - Archive	19
PUT - Restart Workflow	20
POST - Create Version	20
GET - WorkflowStatus	21

## INTRODUCCIÓN

La dirección de la API es: <https://api.digitaldocu.com>

Cada cliente recibirá un token exclusivo denominado "API Key", el cual otorgará acceso a los métodos de la API, así como también un usuario de integración. Para efectuar llamadas a la API, será necesario incluir los encabezados "apikey" y "username" en la solicitud.

A continuación se muestra un ejemplo de cómo agregar las cabeceras en una llamada utilizando el programa Postman.



## ENDPOINTS DE LA API

### DOCUMENTOS

Para la gestión de documentos, la API proporciona los siguientes endpoints.

#### POST - UploadAndSave

El endpoint `"/api/Document/UploadAndSave"` permite crear un nuevo archivo en Digitaldocu y requiere un único objeto de tipo `"UploadAndSaveDto"` como parámetro en el cuerpo de la petición. La estructura de este objeto es la siguiente (los campos marcados con \* son requeridos):

- ❖ **DocumentName\***: Indica el nombre del documento que se quiere archivar junto con su extensión. Por ejemplo: "Documento.pdf".
- ❖ **DocumentBase64\***: Representa el contenido del documento que se desea archivar, codificado como una cadena de texto en base64. Para codificar el documento y convertirlo a base64, se pueden utilizar herramientas en línea como <https://www.base64encode.org/> o <https://base64.guru/>.
- ❖ **IntegrationDocumentLocation\***: Representa la ruta de archivado del documento que se quiere subir. Este campo está compuesto por los códigos de integración de las entidades de Archivador, Categoría, Tipo y Subtipo. Además, permite indicar el año de archivado.

Su estructura es la siguiente:

```
"integrationDocumentLocation": {
  "folderIntegrationCode": "string",
  "categoryIntegrationCode": "string",
  "typeIntegrationCode": "string",
  "subTypeIntegrationCode": "string",
  "year": 0
}
```

**NOTA:** En caso de no proporcionar un año de archivado, el documento se archivará en el año actual.

- ❖ **ExternalCSV:** Representa un Código Seguro de Verificación que identifica de forma única un documento dentro de una integración. Este campo es de tipo **GUID**, y permite al integrador visualizar el documento una vez archivado en Digitaldocu.
- ❖ **DocumentKeywords:** Es un conjunto de metadatos vinculados al documento. Se compone de dos listas: "keywords" y "groupKeywords". La lista "keywords" contiene metadatos simples, cada uno formado por un par clave-valor que incluye el código de integración del metadato (keywordIntegrationCode) y su valor asociado (value).

La estructura de cada elemento en la lista "keywords" es la siguiente:

```
"keywords": [
  {
    "keywordIntegrationCode": "string",
    "value": "string"
  }
]
```

**NOTA:** En caso de que se requieran múltiples valores para un mismo metadato, se deberá duplicar el elemento de manera que se tengan dos elementos de pares clave-valor con la misma "keywordIntegrationCode" pero con valores diferentes.

En el caso de los metadatos de tipo grupo, se incluyen en la lista "groupKeywords". Esta última sigue una estructura similar a la lista de "keywords", pero cada elemento está compuesto por el código de integración del metadato padre y una lista de metadatos simples. Cada metadato dentro de esta lista de metadatos de grupo también sigue la misma estructura de clave-valor con el código de integración y el valor asociado.

La estructura de cada elemento en la lista "keywordsGroup" es la siguiente:

```
"keywordsGroup": [  
  {  
    "keywordIntegracionCode": "string",  
    "keywords": [  
      {  
        "keywordIntegracionCode": "string",  
        "value": "string"  
      }  
    ]  
  }  
]
```

**NOTA:** Al igual que con los metadatos simples, los metadatos de tipo grupo que requieran múltiples valores deberán duplicarse, de modo que haya dos elementos de pares clave-valor con el mismo "keywordIntegracionCode" pero con valores diferentes para cada uno.

Por lo tanto, la estructura general del campo "documentKeywords", que incluye tanto metadatos simples como de grupo, es la siguiente:

```
"documentKeywords": {  
  "keywords": [  
    {  
      "keywordIntegracionCode": "string",  
      "value": "string"  
    }  
  ],  
  "keywordsGroup": [  
    {  
      "keywordIntegracionCode": "string",  
      "keywords": [  
        {  
          "keywordIntegracionCode": "string",  
          "value": "string"  
        }  
      ]  
    }  
  ]  
}
```

A continuación se proporciona un ejemplo práctico del cuerpo de una llamada POST al endpoint UploadAndSave.

```
{
  "documentName": "UploadAndSave.txt",
  "documentBase64": "RXN0byB1cyB1bmEgcHJ1ZWJhIGR1bCB1bmRwb2ludCBVcGxvYWRBbmRTYXZl",
  "integrationDocumentLocation": {
    "folderIntegrationCode": "Documentos_de_prueba",
    "categoryIntegrationCode": "PDFs",
    "typeIntegrationCode": "Prueba_endpoint",
    "subTypeIntegrationCode": "",
    "year": 2024
  },
  "externalCSV": "162bfde3-9d4d-46bf-a64c-26435e4517ea",
  "documentKeywords": {
    "keywords": [
      {
        "keywordIntegrationCode": "Integracion_Empresa",
        "value": "Ofimática Digital"
      },
      {
        "keywordIntegrationCode": "Integracion_Creador",
        "value": "Ana García"
      },
      {
        "keywordIntegrationCode": "Integracion_Creador",
        "value": "Juan Pérez"
      }
    ]
  },
  "keywordsGroup": [
    {
      "keywordIntegrationCode": "Integracion_Iva",
      "keywords": [
        {
          "keywordIntegrationCode": "Integracion_Porcentaje",
          "value": "21"
        },
        {
          "keywordIntegrationCode": "Integracion_Cuota",
          "value": "2"
        },
        {
          "keywordIntegrationCode": "Integracion_BaseImponible",
          "value": "50"
        }
      ]
    }
  ]
}
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá el identificador del documento como respuesta.

## PUT - Update

El endpoint “[/api/Document/{documentId}/Update](#)” permite modificar tanto el fichero físico como los metadatos asociados a un documento existente en Digitaldocu. Para invocar este método, se debe proporcionar tanto el identificador del documento deseado a través de la URL como un objeto de tipo “[UpdateDocumentDto](#)” en el cuerpo de la petición.

Un ejemplo práctico de la URL para realizar una solicitud PUT a este endpoint sería el siguiente.

```
/api/Document/98152/Update
```

La estructura del objeto a pasar por el cuerpo de la petición es la siguiente (los campos marcados con \* son requeridos):

- ❖ **DocumentKeywords:** Representa el conjunto de metadatos asociados al documento. Su formato es idéntico al utilizado en la subida de un documento, tal como se describe en el endpoint “[UploadAndSave](#)”. Sin embargo, en el caso de una actualización, solo se requieren los metadatos que se deseen modificar, evitando la necesidad de enviar todas las palabras clave existentes. Es importante tener en cuenta que, para los metadatos múltiples, se deben proporcionar todos los valores nuevamente.
- ❖ **DocumentBase64:** Representa el contenido físico del documento que se desea actualizar, codificado como una cadena de texto en base64. Para codificar el documento y convertirlo a base64, se pueden utilizar herramientas en línea como <https://www.base64encode.org/> o <https://base64.guru/>.
- ❖ **ExternalCSV:** Representa un Código Seguro de Verificación que identifica de forma única un documento dentro de una integración. Este campo es de tipo **GUID**, y permite al integrador visualizar el documento una vez archivado en Digitaldocu. En el caso de la actualización, solo se permitirá asignar un nuevo CSV a los documentos que no tengan uno previamente asignado, es decir, no es posible reemplazar un CSV existente por otro.

A continuación se proporciona un ejemplo práctico del cuerpo de una llamada PUT al endpoint “Update” para actualizar los metadatos del documento subido mediante el endpoint “UploadAndSave”.

```
{
  "documentBase64": "RXN0byB1cyB1bmEgcHJ1ZWJhIGR1bCB1bmRwb2ludCBVcGxvYWRBbmRTYXZl",
  "externalCSV": "262bfde3-9d4d-46bf-a64c-26435e4517ea",
  "documentKeywords": {
    "keywords": [
      {
        "keywordIntegrationCode": "Integracion_Empresa",
        "value": "Google LLC"
      }
    ]
  }
}
```

```
{
  "keywordIntegrationCode": "Integracion_Creador",
  "value": "Laura Iborra"
},
{
  "keywordIntegrationCode": "Integracion_Creador",
  "value": "Pedro Juárez"
}
],
"keywordsGroup": [
  {
    "keywordIntegrationCode": "Integracion_Iva",
    "keywords": [
      {
        "keywordIntegrationCode": "Integracion_Porcentaje",
        "value": "10"
      },
      {
        "keywordIntegrationCode": "Integracion_Cuota",
        "value": "10"
      },
      {
        "keywordIntegrationCode": "Integracion_BaseImponible",
        "value": "100"
      }
    ]
  }
]
}
}
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta el identificador del documento.

## DELETE - Document

El endpoint `"/api/Document/{documentId}"` permite eliminar un documento existente en Digitaldocu. Para invocar este método, se debe proporcionar el identificador del documento deseado a través de la URL.

Un ejemplo práctico de la URL para realizar una solicitud POST a este endpoint sería el siguiente.

```
/api/Document/98152
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un "True".

## GET - Info

El endpoint `"/api/Document/{documentId}/Info"` permite obtener información de un documento existente en Digitaldocu. Para invocar este método, se debe proporcionar el identificador del documento deseado a través de la URL.

Un ejemplo práctico de la URL de la petición GET a este endpoint es el siguiente.

```
/api/Document/98150/Info
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto JSON con el siguiente formato.

```
{
  "id": 0,
  "name": "string",
  "extension": 0,
  "size": 0,
  "inboxIntegrationCode": "string",
  "folderIntegrationCode": "string",
  "categoryIntegrationCode": "string",
  "typeIntegrationCode": "string",
  "subtypeIntegrationCode": "string",
  "year": 0
}
```

Un ejemplo práctico de respuesta del servidor a la petición GET es el siguiente.

```
{
  "id": 98150,
  "name": "Documento.pdf",
  "extension": 82,
  "size": 131375,
  "inboxIntegrationCode": null,
  "folderIntegrationCode": "ARCHIVADOR",
  "categoryIntegrationCode": "CATEGORIA",
  "typeIntegrationCode": null,
  "subtypeIntegrationCode": null,
  "year": 2025
}
```

## GET - Keywords

El endpoint “[/api/Document/{documentId}/Keywords](#)” permite obtener los metadatos de un documento existente en Digitaldocu. Para invocar este método, se debe proporcionar el identificador del documento deseado a través de la URL.

Un ejemplo práctico de la URL de la petición GET a este endpoint es el siguiente.

```
/api/Document/98150/Keywords
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto JSON con el siguiente formato.

```
{
  "keywords": [
    {
      "keywordIntegrationCode": "string",
      "value": "string"
    }
  ],
  "keywordsGroup": [
    {
      "keywordIntegrationCode": "string",
      "keywords": [
        {
          "keywordIntegrationCode": "string",
          "value": "string"
        }
      ]
    }
  ]
}
```

Un ejemplo práctico de respuesta del servidor a la petición GET es el siguiente.

```
{
  "keywords": [
    {
      "keywordIntegrationCode": "INV_NUMEROFACTURA",
      "value": "#0015"
    },
    {
```

```
    "keywordIntegrationCode": "INV_TIPOFACTURA",
    "value": "Estándar"
  },
  {
    "keywordIntegrationCode": "INV_TOTAL",
    "value": "116.40"
  },
],
"keywordsGroup": [
  {
    "keywordIntegrationCode": "INV_VENCIMIENTOS",
    "keywords": [
      {
        "keywordIntegrationCode": "INV_IMPORTEVENCIMIENTO",
        "value": "116.40"
      },
      {
        "keywordIntegrationCode": "INV_FECHAVENCIMIENTO",
        "value": "9/6/2025 12:00:00 AM"
      }
    ]
  },
],
{
  "keywordIntegrationCode": "INV_IVA",
  "keywords": [
    {
      "keywordIntegrationCode": "INV_CUOTAIVA",
      "value": "4.656"
    },
    {
      "keywordIntegrationCode": "INV_TIPOIVA",
      "value": "4.00"
    },
    {
      "keywordIntegrationCode": "INV_BASEIMPONIBLEIVA",
      "value": "116.40"
    }
  ]
}
]
```

## GET - Related

El endpoint “[/api/Document/{documentId}/Related](#)” permite obtener la lista de documentos relacionados de un documento existente en Digitaldocu. Para invocar este método, se debe proporcionar el identificador del documento deseado a través de la URL.

Un ejemplo práctico de la URL de la petición GET a este endpoint es el siguiente.

```
/api/Document/98152/Related
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto JSON con el siguiente formato.

```
[
  {
    "documentName": "string",
    "documentId": 0,
    "directRelation": true,
    "hasKeywords": true,
    "hasAttachments": true,
    "location": "string"
  }
]
```

Un ejemplo práctico de respuesta del servidor a la petición GET es el siguiente.

```
[
  {
    "documentName": "Related.txt",
    "documentId": 98153,
    "directRelation": true,
    "hasKeywords": false,
    "hasAttachments": false,
    "location": "archivador1/categoria1/tipo2/2024"
  }
]
```

## POST - Related

El endpoint “[/api/Document/{documentId}/Related](#)” permite relacionar una lista de documentos existentes en Digitaldocu a un documento específico. Para utilizar este método, es necesario proporcionar tanto el identificador del documento principal a través de la URL como una lista de identificadores de documentos en el cuerpo de la petición.

Un ejemplo práctico de la URL para realizar una solicitud POST a este endpoint sería el siguiente.

```
/api/Document/98152/Related
```

En el caso del cuerpo de una llamada POST al endpoint, un ejemplo práctico sería el siguiente.

```
[  
  "3118",  
  "3117"  
]
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un “True”.

## POST - Concatenate

El endpoint “[/api/Document/{documentId}/Concatenate](#)” permite la concatenación física de una lista de documentos PDF existentes en Digitaldocu con un documento PDF específico. Al llamar a este endpoint, los documentos especificados se fusionan en el documento principal, creando un único documento que contiene la información de todos los documentos concatenados. Para utilizar este método, es necesario proporcionar tanto el identificador del documento principal a través de la URL como una lista de identificadores de documentos en el cuerpo de la petición.

Un ejemplo práctico de la URL para realizar una solicitud POST a este endpoint sería el siguiente.

```
/api/Document/98152/Concatenate
```

En el caso del cuerpo de una llamada POST al endpoint, un ejemplo práctico sería el siguiente.

```
[  
  "3118",  
  "3117"  
]
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un “True”.

## GET - DocumentSameName

El endpoint “[/api/Document/SameName](#)” permite comprobar la existencia de un documento con el mismo nombre para una ruta en concreto. Para utilizar este método, es necesario proporcionar una serie de parámetros como consulta en la URL descritos a continuación (los campos marcados con \* son requeridos):

- ❖ **DocName\***: Indica el nombre del documento del cual se desea verificar la existencia junto con su extensión. Por ejemplo: "Documento.pdf".
- ❖ **iic**: Indica el código de integración de la entidad Inbox.
- ❖ **fic**: Indica el código de integración de la entidad Archivador.
- ❖ **cic**: Indica el código de integración de la entidad Categoría.
- ❖ **tic**: Indica el código de integración de la entidad Tipo.
- ❖ **sic**: Indica el código de integración de la entidad Subtipo.
- ❖ **year**: Indica el año de archivado. En caso de no especificarse, se buscarán documentos que no tengan año de archivado.

Un ejemplo práctico de la URL de la petición GET a este endpoint es el siguiente.

```
/api/Document/SameName?docName=Documento%20de%20prueba%202.pdf&fic=Archivador1&cic=Categor  
ia1&tic=Tipo1&year=2024
```

**NOTA:** Se ha omitido la inclusión de los campos "iic" y "sic" en este ejemplo, ya que la ruta en la que se busca no tiene asignado un tipo ni un inbox.

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta el identificador del documento en el caso de existir y un 0 si no existe.

## GET - ShareLink

El endpoint “[/api/Document/{documentId}/ShareLink](#)” permite obtener el enlace a un documento compartido en Digitaldocu. Para utilizar este método, es necesario proporcionar el identificador del documento compartido como parámetro en la URL.

Un ejemplo práctico de la URL para realizar una solicitud GET a este endpoint sería el siguiente.

```
/api/Document/98152/ShareLink
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta el link de visualización del documento compartido.

## POST - Share

El endpoint “[/api/Document/{documentId}/Share](#)” permite activar o desactivar un enlace para compartir un documento en Digitaldocu. Para utilizar este método, se debe proporcionar el identificador del documento a través de la URL y un objeto de tipo “[ShareDocumentRequestDto](#)” en el cuerpo de la petición.

Un ejemplo práctico de la URL para realizar una solicitud PUT a este endpoint sería el siguiente.

```
/api/Document/98152/Share
```

La estructura del objeto a pasar por el cuerpo de la petición es la siguiente (los campos marcados con \* son requeridos):

- ❖ **ExternalCSV:** Representa un Código Seguro de Verificación que identifica de forma única un documento dentro de una integración. Este campo es de tipo **GUID**, y permite al integrador visualizar el documento una vez archivado en Digitaldocu. En el caso de la actualización, solo se permitirá asignar un nuevo CSV a los documentos que no tengan uno previamente asignado, es decir, no es posible reemplazar un CSV existente por otro.
- ❖ **ExpiredShareDate:** Define la fecha y hora de expiración del enlace compartido. Si este campo se deja vacío, el enlace no tendrá fecha de expiración y el acceso será indefinido.
- ❖ **ShareActive\*:** Indica si el enlace que permite el acceso al documento está activo o no. Es un campo de tipo booleano que permite habilitar o deshabilitar el acceso compartido.

A continuación se proporciona un ejemplo práctico del cuerpo de una llamada al endpoint para compartir un documento generando un CSV de manera automática.

```
{  
  "externalCSV": "",  
  "expiredShareDate": "2024-09-19T19:40:33.326Z",  
  "shareActive": true  
}
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto de tipo “**ShareDocumentResponseDto**” en formato JSON con la siguiente estructura:

```
{
  "externalCSV": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "shareDate": "2024-09-10T19:40:33.326Z",
  "expiredShareDate": "2024-09-19T19:40:33.326Z",
  "link": "https://api.digitaldocu.com/api/Document/View/162bfde3-9d4d-46bf-a64c-26435e4517ea/Factura1.pdf",
  "shareActive": true
}
```

## GET - View

El endpoint “**/api/Document/View/{csv}/{documentName}**” permite visualizar un documento existente compartido en Digitaldocu. Para utilizar este método, es necesario proporcionar tanto el CSV Externo del documento como el nombre del documento como parámetros en la URL.

Un ejemplo práctico de la URL para realizar una solicitud GET a este endpoint sería el siguiente.

```
/api/Document/View/162bfde3-9d4d-46bf-a64c-26435e4517ea/Factura1.pdf
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta el documento a visualizar.

## GET - Attached

El endpoint “**/api/Document/{documentId}/Attached**” permite obtener la lista de documentos adjuntos de un documento existente en Digitaldocu. Para invocar este método, se debe proporcionar el identificador del documento deseado a través de la URL.

Un ejemplo práctico de la URL de la petición GET a este endpoint es el siguiente.

```
/api/Document/98152/Attached
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto JSON con una lista de identificadores de documentos adjuntos.

```
[
  190,
  191,
  425
]
```

## PUT - Attached

El endpoint “[/api/Document/{documentId}/Attached/maintainLocation](#)” permite adjuntar un documento existente en Digitaldocu a otro documento existente. Para utilizar este método, es necesario proporcionar el identificador del documento principal a través de la URL y un valor booleano que indica si se desea mantener la ubicación del documento en Digitaldocu. Además,

Al adjuntar un documento a otro, existen dos opciones:

- ❖ **Mantener la ubicación:** Esto creará una copia del documento adjunto y lo asociará al documento principal, manteniendo su ubicación original en Digitaldocu.
- ❖ **No mantener la ubicación:** Esto moverá el documento adjunto al documento principal sin mantener su ubicación original. Esto significa que el documento ya no será visible en Digitaldocu y que los metadatos del documento se eliminarán. Para visualizar el documento, se habrá de visualizar el documento principal y acceder al apartado de adjuntos.

Un ejemplo práctico de la URL de la petición PUT a este endpoint es el siguiente.

```
/api/Document/98152/Attached/false
```

En el caso del cuerpo de una llamada POST al endpoint, un ejemplo práctico sería el siguiente.

```
[  
  "3118",  
  "3117"  
]
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un “True”.

## GET - GetInboxFiles

El endpoint “[/api/Document/GetInboxFiles/{inboxIntegrationCode}](#)” permite obtener todos los documentos existentes dentro de un inbox. Para utilizar este método, es necesario proporcionar el código de integración del inbox a través de la URL.

Un ejemplo práctico de la URL de la petición GET a este endpoint es el siguiente.

```
/api/Document/GetInboxFiles/TEST
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto JSON con una lista de documentos.

```
[
  {
    "id": 123,
    "name": "Test.xlsx",
    "extension": "xlsx",
    "creationDate": "2024-05-29T09:55:00.148535Z"
  },
  {
    "id": 124,
    "name": "Prueba.pdf",
    "extension": "pdf",
    "creationDate": "2024-06-12T07:45:52.193434Z"
  }
]
```

## POST - UploadDocument

El endpoint `/api/Inbox/UploadDocument` permite subir un documento nuevo a un Inbox existente en Digitaldocu. Para utilizar este método, se debe proporcionar un objeto de tipo `UploadDocumentDto` en el cuerpo de la petición.

La estructura del objeto a pasar por el cuerpo de la petición es la siguiente (los campos marcados con \* son requeridos):

- ❖ **DocumentName\***: Representa el nombre que recibirá el documento en Digitaldocu.
- ❖ **InboxIntegrationCode\***: Indica el código de integración del inbox al cual se quiere subir el documento.
- ❖ **DocumentBase64\***: Representa el documento físico codificado como Base64.

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un objeto JSON con los datos del documento subido.

```
{
  "id": 123,
  "name": "Test.pdf",
  "size": 453601,
  "mimeType": "application/pdf",
  "inboxName": "TEST"
}
```

## POST - Archive

El endpoint “[/api/Document/Archive](#)” permite archivar un documento en una ruta determinada a partir de un documento ubicado en un inbox. Si la ruta de destino contiene una plantilla de keywords, estas se asignan automáticamente al documento según la configuración definida en la plantilla.

Para utilizar este método, se debe proporcionar un objeto de tipo “[ArchiveRequestDto](#)” en el cuerpo de la petición.

La estructura del objeto a pasar por el cuerpo de la petición es la siguiente (los campos marcados con \* son requeridos):

- ❖ **DocumentId\***: Representa el identificador del documento que se quiere archivar.
- ❖ **DocumentNewName**: Representa el nombre que recibirá el documento una vez se archive. Este parámetro es opcional, por lo que si no se define un valor, el nombre del documento se mantendrá al archivar.
- ❖ **IntegrationDocumentLocation\***: Representa la ruta de archivado del documento que se quiere archivar. Este campo está compuesto por los códigos de integración de las entidades de Archivador, Categoría, Tipo y Subtipo. Además, permite indicar el año de archivado.

Su estructura es la siguiente:

```
"integrationDocumentLocation": {
  "folderIntegrationCode": "string",
  "categoryIntegrationCode": "string",
  "typeIntegrationCode": "string",
  "subTypeIntegrationCode": "string",
  "year": 0
}
```

**NOTA:** En caso de no proporcionar un año de archivado, el documento se archivará en el año actual.

- ❖ **DocumentKeywords**: Representa el conjunto de metadatos asociados al documento. Su formato es idéntico al utilizado en la subida de un documento, tal como se describe en el endpoint “[UploadAndSave](#)”.
- ❖ **CreateVersion**: Especifica si se debe crear una nueva versión en caso de que ya exista un documento con el mismo nombre en la ruta de archivado indicada.
  - Si se establece en false, se generará una excepción si el documento ya existe.
  - Si se establece en true, se creará automáticamente una nueva versión del documento.

## PUT - Restart Workflow

El endpoint `"/api/Document/{documentId}/Workflow/Restart"` permite reiniciar el flujo de trabajo (workflow) asociado a un documento existente en Digitaldocu, independientemente de si este ha finalizado o no. Es importante tener en cuenta que para que estas acciones se puedan llevar a cabo, el usuario de integración que haga la llamada ha de tener asignado el rol configurado para ello en el Workflow.

Para utilizar este método, es necesario proporcionar el identificador del documento a través de la URL.

```
/api/Document/123456/Workflow/Restart
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un booleano indicando "True" si el workflow se ha reiniciado y "False" si no se ha podido reiniciar.

## POST - Create Version

El endpoint `"/api/Document/CreateVersion"` permite subir una nueva versión de un documento previamente archivado en Digitaldocu. Al crear una nueva versión, el sistema conserva un historial completo de versiones del documento original, facilitando así el control de cambios y auditorías.

Para utilizar este método, se debe proporcionar un objeto de tipo `"CreateVersionDto"` en el cuerpo de la petición.

La estructura del objeto a pasar por el cuerpo de la petición es la siguiente (los campos marcados con \* son requeridos):

- ❖ **OriginalDocumentId\***: Representa el identificador del documento original que se quiere versionar.
- ❖ **DocumentBase64\***: Es el contenido del archivo que se desea subir, codificado en base64. Este archivo representa la nueva versión del documento.
- ❖ **ContentType**: Tipo MIME del archivo (application/pdf, image/png, etc.). Este campo solo debe incluirse si el tipo MIME del nuevo archivo es distinto al del documento original. Si la nueva versión mantiene el mismo tipo MIME, este campo puede omitirse. Por ejemplo, si el documento original es un PDF y se sube una nueva versión en formato PNG, se debe especificar "image/png".
- ❖ **Commentary\***: Es el comentario descriptivo de la versión.

A continuación se proporciona un ejemplo práctico del cuerpo de una llamada al endpoint para crear una versión.

```
{
  "originalDocumentId": 73998,
  "documentBase64": "",
  "contentType": "image/png",
  "commentary": "Cambiado a imagen"
}
```

Si la llamada se ha realizado correctamente, el servidor nos devolverá como respuesta un booleano indicando “True” si la versión se ha creado y “False” si no se ha podido crear.

## GET - WorkflowStatus

El endpoint “[/api/Document/{documentId}/WorkflowStatus](#)” permite consultar el estado actual del workflow de un documento existente en Digitaldocu. Para utilizar este método, es necesario proporcionar el identificador del documento (documentId) como parámetro en la URL.

Un ejemplo práctico de la URL para realizar una solicitud GET a este endpoint sería el siguiente.

```
/api/Document/3241/WorkflowStatus
```

Si la llamada se ha realizado correctamente, el servidor devolverá como respuesta un objeto con el estado actual del workflow del documento.

Esta respuesta incluirá:

- ❖ **Status:** Estado del flujo de trabajo, que puede ser:
  - **WithoutWorkflow** si el documento no tiene workflow.
  - **InProgress** si el workflow está activo.
  - **Completed** si el workflow ha finalizado.
- ❖ **StageName:** Nombre del procedimiento actual o finalizado.
- ❖ **Action:** Nombre de la acción realizada más reciente.
- ❖ **Comment:** Comentario registrado en la última acción significativa del workflow.

```
{  
  "status": "Completed",  
  "stageName": "Validación final",  
  "action": "Aprobación",  
  "comment": "Documento aprobado por el responsable legal"  
}
```

## POST - SearchByKeywords

El endpoint “[/api/Document/SearchByKeywords](#)” permite realizar búsquedas avanzadas de documentos utilizando filtros de keywords con operadores específicos. Este endpoint es ideal para consultas complejas que requieren múltiples criterios de filtrado basados en las propiedades (keywords) de los documentos.

Para utilizar este método, se debe proporcionar un objeto de tipo "**KeywordSearchRequestDto**" en el cuerpo de la petición. Un ejemplo práctico de la URL para realizar una solicitud POST a este endpoint sería el siguiente:

```
POST /api/Document/Search/ByKeywords
```

La estructura del objeto a pasar por el cuerpo de la petición es la siguiente (los campos marcados con \* son requeridos):

- ❖ **IntegrationDocumentLocation\***: Representa la ubicación del documento en la estructura documental. Este campo está compuesto por los códigos de integración de las entidades de Archivador, Categoría, Tipo y Subtipo. Además, permite indicar el año de archivado.

```
"integrationDocumentLocation": {
  "folderIntegrationCode": "string",
  "categoryIntegrationCode": "string",
  "typeIntegrationCode": "string",
  "subTypeIntegrationCode": "string",
  "year": 0
}
```

- ❖ **Filters\***: Representa la lista de filtros de keywords a aplicar. Cada filtro debe contener:
  - **keywordIntegrationCodes\***: Código de integración de la keyword a filtrar.
  - **operator\***: Operador de comparación (eq, neq, like, not like, gt, lt, gte, lte)
  - **value\***: Valor a comparar (puede ser string, número o booleano)
- ❖ **Page**: Número de página (por defecto: 1)
- ❖ **PageSize**: Tamaño de página (por defecto: 50)

Ejemplo de solicitud completa:

```
{
  "integrationDocumentLocation": {
    "inboxIntegrationCode": "INBOX001",
    "folderIntegrationCode": "FOLDER001",
    "categoryIntegrationCode": "CAT001",
    "typeIntegrationCode": "TYPE001",
    "subTypeIntegrationCode": "SUBTYPE001",
    "year": 2024,
  },
  "filters": [
    {
```

```
"keywordIntegrationCode": "CLIENT_CODE",
"operator": "eq",
"value": "CLIENT123",
},
{
"keywordIntegrationCode": "AMOUNT",
"operator": "gt",
"value": 1000,
},
{
"keywordIntegrationCode": "DOCUMENT_TYPE",
"operator": "like",
"value": "CONTRACT",
},
],
"page": 1,
"pageSize": 50,
}
```

Si la llamada se ha realizado correctamente, el servidor devolverá como respuesta un objeto con los documentos encontrados e información de paginación.

```
{
"documentIds": [123, 456, 789, 1011, 1213],
"totalCount": 150,
"page": 1,
"pageSize": 50,
"totalPages": 3
}
```

Esta respuesta incluirá:

- ❖ **documentIds**: Lista de identificadores de documentos que cumplen los filtros
- ❖ **totalCount**: Número total de documentos encontrados (limitado a 1000 para optimización)
- ❖ **page**: Página actual
- ❖ **pageSize**: Tamaño de página utilizado
- ❖ **totalPages**: Número total de páginas disponibles