

encamina

PIENSA EN COLORES

Gada-i Guía de la API



Valencia (sede central)
Av. Jerónimo Roure, 49
46520, Sagunto (Valencia)
Telf. +34 962 698 064

Madrid
C/ O'Donell, 34
28009, Madrid
Telf. +34 917 893 823

Murcia
C/ del Ángel, 32
30202 Cartagena (Murcia)

Tenerife
Av. Trinidad, 61. Campus Central ULL
38200, S.Cristóbal de La Laguna
(Tenerife)

info@encamina.com
www.encamina.com

Historia del documento

Fecha	Status	Versión	Autor	Descripción
27/09/2021		3.0	Encamina	

Índice

1	Objetivo del documento.....	6
2	Métodos de la API.....	6
2.1	AppConfiguration.....	6
2.1.1	/newAppConfiguration POST.....	6
	Request body	6
	Responses	7
	Errores	7
2.1.2	/updateAppConfiguration PUT.....	7
	Request body	7
	Responses	7
	Errores	8
2.2	Document.....	8
2.2.1	/Document GET.....	8
	Parameters	8
	Responses	8
	Errores	9
2.2.2	/uri GET.....	9
	Parameters	9
	Responses	9

- Errores** 9
- 2.2.3 /content GET 9
 - Parameters** 9
 - Responses** 9
 - Errores** 10
- 2.2.4 /upload POST 10
 - Request body** 10
 - Responses** 10
 - Errores** 10
- 2.2.5 /index POST 10
 - Request body** 10
 - Responses** 11
 - Errores** 11
- 2.2.6 /indexbinary POST 11
 - Request body** 11
 - Responses** 12
 - Errores** 12
- 2.2.7 /delete DELETE 12
 - Request body** 12
 - Responses** 12
 - Errores** 13
- 2.2.8 /legalhold DELETE 13
 - Request body** 13
 - Responses** 13
 - Errores** 13
- 2.2.9 /update PUT 13
 - Request body** 13
 - Responses** 14



Errores	14
2.2.10 /revertDelete PUT	14
Request body	14
Responses	15
Errores	15
2.2.11 /reindex PUT	15
Request body	15
Responses	15
Errores	16
2.2.12 /query GET	16
Request body	16
Responses	16
Errores	17
2.2.13 /querySql GET	17
Request body	17
Responses	17
Errores	18
2.2.14 /DocumentClass/documents GET	18
Parameters	18
Responses	19
Errores	19
2.2.15 /DocumentClass/delete DELETE	20
Request body	20
Responses	20
Errores	20
2.2.16 /DocumentClass/delete DELETE	20
Request body	20
Responses	20

- Errores**21
- 2.3 Management..... 21
 - 2.3.1 /ConfigInfo GET..... 21
 - Responses**21
 - Errores**21
 - 2.3.2 /Warm GET..... 21
 - Responses**21
 - Errores**21
 - 2.3.3 /SetConfigFields POST..... 21
 - Request body**22
 - Responses**22
 - Errores**22
 - 2.3.4 /GetConfigFieldsGET 22
 - Responses**23
 - Errores**23
 - 2.3.5 /CreateIndexDataSource POST 23
 - Responses**23
 - Errores**23
- 2.4 Search. 24
 - 2.4.1 /documents/filter POST..... 24
 - Request body**24
 - Responses**24
 - Errores**25
- 3 Glosario de conceptos.....26
- Contacto27



1 Objetivo del documento.

El objetivo del presente documento es definir la función que realiza cada método de la API de Gada-i.

2 Métodos de la API.

Todos los métodos de la API tienen parámetros obligatorios, son los siguientes:

- v : string.
- x-user-name: string.

2.1 AppConfiguration.

2.1.1 /newAppConfiguration POST

Una vez registremos una aplicación tendremos el Id de la aplicación registrada. Con dicho Id configuraremos el resto de las configuraciones de esa app con este método. Debemos rellenar todos los campos que lo contiene. Todas estas configuraciones se añadirán a nuestro Azure App Configuration y a nuestro collection "GadaCollection".

2.1.1.1 Request body

```
{
  "appId": "appId",
  "connectionString": "cadena de conexion de storage",
  "container": "nombre del contenedor",
  "minutesSasExpire": "minutos de expiracion",
  "temp": "nombre de la carpeta temporal",
  "cosmosDB": {
    "db": "nombre de la CosmosDb",
    "collection": "nombre de la collection",
    "endpoint": "endpoint de la CosmosDb",
    "authKey": "AuthKey de la CosmosDb"
  },
  "organization": {
    "cognitiveConfig": true, //dependiendo de si tiene servicios cognitivos o no
    "language": "lenguaje" //es-ES lenguaje español (ejemplo)
  },
  "searchService": {
    "indexName": "nombre del index",
    "dataSourceJsonName": "Gada_DataSource.json",
    "indexerJsonName": "Gada_Indexer.json",
    "skillSetJsonName": "Gada_SkillSet.json",
    "cognitiveServicesKey": "Key del Servicio Cognitivo",
    "cognitiveServicesDescription": "cadena de conexión de los cognitive services"
  }
}
```

Responses

Una vez se ha ejecutado nos devolverá un listado de configuraciones que se deben enviar a IT para que se configuren con Key Vault para otorgarles mayor seguridad.

```
String
```

Errores

- 400: El JSON del request es nulo o no es válido.

2.1.2 /updateAppConfiguration PUT

Método que actualiza las configuraciones de una app con Id específico. Las configuraciones que se actualizaran son las que están contenidas en nuestro Azure App Configuration.

Request body

```
{
  "appId": "appId",
  "connectionString": "cadena de conexion de storage",
  "container": "nombre del contenedor",
  "minutesSasExpire": "minutos de expiracion",
  "temp": "nombre de la carpeta temporal",
  "cosmosDB": {
    "db": "nombre de la CosmosDb",
    "collection": "nombre de la collection",
    "endpoint": "endpoint de la CosmosDb",
    "authKey": "AuthKey de la CosmosDb"
  },
  "organization": {
    "cognitiveConfig": true, //dependiendo de si tiene servicios cognitivos o no
    "language": "lenguaje" //es-ES lenguaje español (ejemplo)
  },
  "searchService": {
    "indexName": "nombre del index",
    "dataSourceJsonName": "Gada_DataSource.json",
    "indexerJsonName": "Gada_IndexerJsonName.json",
    "skillSetJsonName": "Gada_SkillSetJsonName.json",
    "cognitiveServicesKey": "Key del Servicio Cognitivo",
    "cognitiveServicesDescription": "cadena de conexión de los cognitive services"
  }
}
```

Responses

Una vez se ha ejecutado nos devolverá un listado de configuraciones que se deben enviar a IT para que se configuren con Key Vault para otorgarles mayor seguridad.

```
String
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El apId no existe.

2.2 Document.

2.2.1/Document GET

Método que devuelve un documento específico a partir del "Id" y el "DocumentClassId" que le enviamos.

Parameters

- Id: string.
- DocumentClassId: string.
- User: string.

Responses

```
{
  "id": "string",
  "created": "2021-09-26T09:11:03.972Z",
  "modified": "2021-09-26T09:11:03.972Z",
  "createdUser": "string",
  "lastModifiedUser": "string",
  "absoluteUri": "string",
  "documentClass": "string",
  "documentClassId": "string",
  "documentType": "string",
  "fileName": "string",
  "contentType": "string",
  "size": 0,
  "partitionKey": "string",
  "container": "string",
  "folderName": "string",
  "blobName": "string",
  "properties": {
    "additionalProp1": [
      null
    ],
    "additionalProp2": [
      null
    ],
    "additionalProp3": [
      null
    ]
  }
}
```


Errores

- 400: Algún parámetro es nulo o no es válido.
- 404: No encuentra un documento con esos parámetros.

2.2.2/uri GET

Método que devuelve la Uri de un documento específico a partir del "Id" y el "DocumentClassId" que le enviamos.

Parameters

- Id: string.
- DocumentClassId: string.
- User: string.

Responses

```
{  
  "documentId": "string",  
  "uri": "string"  
}
```

Errores

- 400: Algún parámetro es nulo o no es válido.
- 404: No encuentra un documento con esos parámetros.

2.2.3 /content GET

Método que devuelve el contenido de un documento específico a partir del "Id" y el "DocumentClassId" que le enviamos.

Parameters

- Id: string.
- DocumentClassId: string.
- User: string.

Responses

```
String
```

Errores

- 400: Algún parámetro es nulo o no es válido.
- 404: No encuentra un documento con esos parámetros.

2.2.4 /upload POST

Método que sube un archivo a la carpeta temporal. Devuelve el nombre del archivo que se le ha asignado.

Request body

```
Body: form-data
  File: File to upload
User { "User" : "username" }
```

Responses

```
String
```

Errores

- 400: El JSON del request es nulo o no es válido.

2.2.5/index POST

Método que indexa un archivo que previamente se ha subido. Devuelve el documento con todas sus propiedades.

Request body

```
{
  "User": "userName",
  "DocumentClassId": DocumentClassId,
  "DocumentClass": "documentClass",
  "DocumentType": "documentType",
  "Filename": "filename",
  "Properties": {
    "NameProperty": "value"
  }
}
```

Responses

```
{
  "id": "string",
  "created": "2021-09-27T11:50:48.245Z",
  "modified": "2021-09-27T11:50:48.245Z",
  "createdUser": "string",
  "lastModifiedUser": "string",
  "absoluteUri": "string",
  "documentClass": "string",
  "documentClassId": "string",
  "documentType": "string",
  "fileName": "string",
  "contentType": "string",
  "size": 0,
  "partitionKey": "string",
  "container": "string",
  "folderName": "string",
  "blobName": "string",
  "properties": {
    "additionalProp1": [
      null
    ],
    "additionalProp2": [
      null
    ],
    "additionalProp3": [
      null
    ]
  }
}
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El archivo no es encontrado.
- 409: El archivo ya ha sido indexado.

2.2.6 /indexbinary POST

Método que sube un archivo y lo indexa. Devuelve el documento con todas sus propiedades.

Request body

```
Body: form-data
  File: File to upload
  Metadata:
    {
      "User": "userName",
      "DocumentClassId": DocumentClassId,
      "DocumentClass": "documentClass",
      "DocumentType": "documentType",
      "Properties": {
        "NameProperty": "value"
      }
    }
}
```

Responses

```
{
  "id": "string",
  "created": "2021-09-27T11:50:48.245Z",
  "modified": "2021-09-27T11:50:48.245Z",
  "createdUser": "string",
  "lastModifiedUser": "string",
  "absoluteUri": "string",
  "documentClass": "string",
  "documentClassId": "string",
  "documentType": "string",
  "fileName": "string",
  "contentType": "string",
  "size": 0,
  "partitionKey": "string",
  "container": "string",
  "folderName": "string",
  "blobName": "string",
  "properties": {
    "additionalProp1": [
      null
    ],
    "additionalProp2": [
      null
    ],
    "additionalProp3": [
      null
    ]
  }
}
```

Errores

- 400: El JSON del request es nulo o no es válido.

2.2.7 /delete DELETE

Método que cambia la propiedad de borrado de un documento específico. Devuelve un booleano en función de si ha borrado el documento o no.

Request body

```
{
  "user": "string",
  "id": "string",
  "documentClassId": "string"
}
```

Responses

```
true
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El documento no es encontrado.

2.2.8 /legalhold DELETE

Método que cambia la propiedad de borrado de un documento específico. Devuelve un booleano en función de si ha borrado el documento o no.

Request body

```
{  
  "user": "string",  
  "id": "string",  
  "documentClassId": "string"  
}
```

Responses

```
True
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El documento no es encontrado.

2.2.9 /update PUT

Método que actualiza las propiedades de un documento específico. Devuelve el documento con todas sus propiedades actualizadas.

Request body

```
{  
  "user": "string",  
  "id": "string",  
  "documentClassId": "string",  
  "documentType": "string",  
  "properties": {  
    "additionalProp1": [  
      null  
    ],  
    "additionalProp2": [  
      null  
    ],  
    "additionalProp3": [  
      null  
    ]  
  }  
}
```

Responses

```
{
  "id": "string",
  "created": "2021-09-27T12:19:07.593Z",
  "modified": "2021-09-27T12:19:07.593Z",
  "createdUser": "string",
  "lastModifiedUser": "string",
  "absoluteUri": "string",
  "documentClass": "string",
  "documentClassId": "string",
  "documentType": "string",
  "fileName": "string",
  "contentType": "string",
  "size": 0,
  "partitionKey": "string",
  "container": "string",
  "folderName": "string",
  "blobName": "string",
  "properties": {
    "additionalProp1": [
      null
    ],
    "additionalProp2": [
      null
    ],
    "additionalProp3": [
      null
    ]
  }
}
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El documento no es encontrado.

2.2.10 /revertDelete PUT

Método que cambia la propiedad de borrado de un documento específico a "false". Devuelve un booleano en función de si ha recuperado el documento o no.

Request body

```
{
  "user": "string",
  "id": "string",
  "documentClassId": "string"
}
```

Responses

```
true
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El documento no es encontrado.

2.2.11 /reindex PUT

Método que borra un documento específico (incluyendo sus registros) y lo vuelve a subir indexándolo (nuevos registros). Devuelve el documento con todas sus propiedades actualizadas.

Request body

```
{
  "user": "string",
  "destinationDocumentClassId": "string",
  "destinationDocumentClass": "string",
  "documentId": "string",
  "documentClassId": "string"
}
```

Responses

```
{
  "id": "string",
  "created": "2021-09-27T08:05:43.318Z",
  "modified": "2021-09-27T08:05:43.318Z",
  "createdUser": "string",
  "lastModifiedUser": "string",
  "absoluteUri": "string",
  "documentClass": "string",
  "documentClassId": "string",
  "documentType": "string",
  "fileName": "string",
  "contentType": "string",
  "size": 0,
  "partitionKey": "string",
  "container": "string",
  "folderName": "string",
  "blobName": "string",
  "properties": {
    "additionalProp1": [
      null
    ],
    "additionalProp2": [
      null
    ]
  }
}
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El documento no es encontrado.

2.2.12 /query GET

Método que busca documentos en función de la consulta en formato JSON introducida. Devuelve la colección de documentos encontrados en la consulta.

Request body

```
{
  "user": "string",
  "conditions": {
    "additionalProp1": [
      null
    ],
    "additionalProp2": [
      null
    ],
    "additionalProp3": [
      null
    ]
  },
  "fields": [
    "string"
  ]
}
```

Responses

```
[
  {
    "id": "string",
    "created": "2021-09-27T12:30:55.361Z",
    "modified": "2021-09-27T12:30:55.361Z",
    "createdUser": "string",
    "lastModifiedUser": "string",
    "absoluteUri": "string",
    "documentClass": "string",
    "documentClassId": "string",
    "documentType": "string",
    "fileName": "string",
    "isDeleted": true,
    "hasLegalHold": true,
    "contentType": "string",
    "size": 0,
    "blobName": "string",
    "properties": {
      "additionalProp1": [
        null
      ],
      "additionalProp2": [
        null
      ]
    }
  },
]
```



```

"content": "string",
"people": [
  "string"
],
"organizations": [
  "string"
],
"places": [
  "string"
],
"metadataStorageSize": 0,
"metadataAuthor": "string",
"metadataCharacterCount": "string",
"metadataCreationDate": "2021-09-27T12:30:55.361Z",
"metadataLastModified": "2021-09-27T12:30:55.361Z",
"metadataPageCount": "string",
"metadataWordCount": "string",
"partitionKey": "string",
"container": "string",
"folderName": "string"
}
]

```

Errores

- 400: El JSON del request es nulo o no es válido.

2.2.13 /querySql GET

Método que busca documentos en función de la consulta en formato SQL introducida. Devuelve la colección de documentos encontrados en la consulta.

Request body

```

{
  "user": "string",
  "sqlQuery": "string"
}

```

Responses

```

[
  {
    "id": "string",
    "created": "2021-09-27T12:33:07.436Z",
    "modified": "2021-09-27T12:33:07.436Z",
    "createdUser": "string",
    "lastModifiedUser": "string",
    "absoluteUri": "string",
    "documentClass": "string",
    "documentClassId": "string",
    "documentType": "string",
    "fileName": "string",
    "isDeleted": true,
    "hasLegalHold": true,
    "contentType": "string",
  }
]

```

```

"size": 0,
"blobName": "string",
"properties": {
  "additionalProp1": [
    null
  ],
  "additionalProp2": [
    null
  ],
  "additionalProp3": [
    null
  ]
},
"content": "string",
"people": [
  "string"
],
"organizations": [
  "string"
],
"places": [
  "string"
],
"metadataStorageSize": 0,
"metadataAuthor": "string",
"metadataCharacterCount": "string",
"metadataCreationDate": "2021-09-27T12:33:07.436Z",
"metadataLastModified": "2021-09-27T12:33:07.436Z",
"metadataPageCount": "string",
"metadataWordCount": "string",
"partitionKey": "string",
"container": "string",
"folderName": "string"
}
]

```

Errores

- 400: El JSON del request es nulo o no es válido.

2.2.14 /DocumentClass/documents GET

Método que busca documentos en función del "DocumentClassId" y el "DocumentClass" introducidos. Devuelve la colección de documentos encontrados en la consulta.

Parameters

- DocumentClassName: string.
- DocumentClassId: string.
- FieldsToRetrieve: array[string]. (Opcional)
- User: string.

Responses

```
[
  {
    "id": "string",
    "created": "2021-09-27T08:21:25.167Z",
    "modified": "2021-09-27T08:21:25.167Z",
    "createdUser": "string",
    "lastModifiedUser": "string",
    "absoluteUri": "string",
    "documentClass": "string",
    "documentClassId": "string",
    "documentType": "string",
    "fileName": "string",
    "isDeleted": true,
    "hasLegalHold": true,
    "contentType": "string",
    "size": 0,
    "blobName": "string",
    "properties": {
      "additionalProp1": [
        null
      ],
      "additionalProp2": [
        null
      ],
      "additionalProp3": [
        null
      ]
    },
    "content": "string",
    "people": [
      "string"
    ],
    "organizations": [
      "string"
    ],
    "places": [
      "string"
    ],
    "metadataStorageSize": 0,
    "metadataAuthor": "string",
    "metadataCharacterCount": "string",
    "metadataCreationDate": "2021-09-27T08:21:25.167Z",
    "metadataLastModified": "2021-09-27T08:21:25.167Z",
    "metadataPageCount": "string",
    "metadataWordCount": "string",
    "partitionKey": "string",
    "container": "string",
    "folderName": "string"
  }
]
```

Errores

- 400: Algún parámetro es nulo o no es válido.
- 404: No encuentra un documento con esos parámetros.

2.2.15 /DocumentClass/delete DELETE

Método que busca documentos en función del "DocumentClassId" y el "DocumentClass" introducidos y cambia la propiedad de borrado de los documentos a "true". Devuelve un booleano en función de si ha borrado los documentos o no.

Request body

```
{
  "user": "string",
  "documentClassName": "string",
  "documentClassId": "string"
}
```

Responses

```
true
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El "DocumentClass" no es encontrado.

2.2.16 /DocumentClass/delete DELETE

Método que busca documentos en función del "DocumentClassId" y el "DocumentClass" introducidos y cambia la propiedad de borrado de los documentos a "true". Devuelve un booleano en función de si ha borrado los documentos o no.

Request body

```
{
  "user": "string",
  "documentClassName": "string",
  "documentClassId": "string"
}
```

Responses

```
true
```

Errores

- 400: El JSON del request es nulo o no es válido.
- 404: El "DocumentClass" no es encontrado.

2.3 Management.

2.3.1/ConfigInfo GET

Método que devuelve el origen de la configuración.

Responses

```
string
```

Errores

- 400: Error mientras recupera el origen de la configuracion.

2.3.2/Warm GET

Método que crea el CosmosDb, el BlobContainer, el índice, el indexador, el conjunto de habilidades y el sistema de origen de datos, necesarios para el servicio de búsqueda y los servicios cognitivos.

Responses

```
true
```

Errores

- 400: Durante la creación del CosmosDb, el BlobContainer, el índice, el indexador, el conjunto de habilidades y el sistema de origen de datos, no se ha realizado correctamente .
- 404: Algun elemento a crear no ha sido encontrado.

2.3.3/SetConfigFields POST

Método que añade nuevas propiedades de configuracion a la app (con la que nos hemos logueado con el token) en el GadaCollection del CosmosDb. Devuelve las propiedades de configuraciones que tiene la app en el GadaCollection.

Request body

```
[  
  {  
    "metaData": "string",  
    "searchProp": "string"  
  }  
]
```

Responses

```
{  
  "metaData": "string",  
  "propsConfigSearch": {  
    "isKey": true,  
    "isRetrievable": true,  
    "isSearchable": true,  
    "isFilterable": true,  
    "isSortable": true,  
    "isFacetable": true,  
    "name": "string",  
    "type": {},  
    "analyzer": {},  
    "searchAnalyzer": {},  
    "indexAnalyzer": {},  
    "synonymMaps": [  
      "string"  
    ],  
    "fields": [  
      null  
    ]  
  }  
}
```

Errores

- 400: Si durante la inserción de un nuevo campo de configuración se produce un error.

2.3.4 /GetConfigFieldsGET

Método que busca las propiedades de la app (con la que nos hemos logueado con el token) en el GadaCollection del CosmosDb y las devuelve.

Responses

```
{
  "id": "string",
  "configurationId": "string",
  "cosmosConfig": [
    {
      "metaData": "string",
      "propsConfigSearch": {
        "isKey": true,
        "isRetrievable": true,
        "isSearchable": true,
        "isFilterable": true,
        "isSortable": true,
        "isFacetable": true,
        "name": "string",
        "type": {},
        "analyzer": {},
        "searchAnalyzer": {},
        "indexAnalyzer": {},
        "synonymMaps": [
          "string"
        ],
        "fields": [
          null
        ]
      }
    }
  ]
}
```

Errores

- 400: Si se produce un error mientras recupera la configuración.

2.3.5/CreateIndexDataSource POST

Método que crea el indexador, el conjunto de habilidades y el sistema de origen de datos, necesarios para el servicio de búsqueda y los servicios cognitivos.

Responses

```
true
```

Errores

- 400: Durante la creación del indexador, el conjunto de habilidades y el sistema de origen de datos, se ha producido un error.

2.4 Search.

2.4.1/documents/filter POST

Método que busca documentos a través de los servicios cognitivos. Se puede buscar en el por el texto a buscar y los filtros que le queremos aplicar. Si queremos ver los filtros por los que podemos filtrar debemos tener las "facets" a "true". El total de los elementos que encuentra serán paginados, solo nos mostrara el número de elementos que le hayamos indicado y de la página en la queremos que nos mostrara la búsqueda.

Request body

```
{
  "textSearch": "string",
  "filter": "string",
  "facets": true,
  "page": 0,
  "numberElements": 0
}
```

Responses

```
{
  "docs": [
    {
      "id": "string",
      "created": "2021-09-27T14:33:34.601Z",
      "modified": "2021-09-27T14:33:34.601Z",
      "createdUser": "string",
      "lastModifiedUser": "string",
      "absoluteUri": "string",
      "documentClass": "string",
      "documentClassId": "string",
      "documentType": "string",
      "fileName": "string",
      "isDeleted": true,
      "hasLegalHold": true,
      "contentType": "string",
      "size": 0,
      "blobName": "string",
      "properties": {
        "additionalProp1": [
          null
        ],
        "additionalProp2": [
          null
        ],
        "additionalProp3": [
          null
        ]
      }
    },
    "content": "string",
    "people": [
      "string"
    ]
  ],
}
```



```

    "organizations": [
      "string"
    ],
    "places": [
      "string"
    ],
    "metadataStorageSize": 0,
    "metadataAuthor": "string",
    "metadataCharacterCount": "string",
    "metadataCreationDate": "2021-09-27T14:33:34.601Z",
    "metadataLastModified": "2021-09-27T14:33:34.601Z",
    "metadataPageCount": "string",
    "metadataWordCount": "string"
  }
],
"facets": {
  "additionalProp1": [
    {
      "count": 0
    }
  ],
  "additionalProp2": [
    {
      "count": 0
    }
  ],
  "additionalProp3": [
    {
      "count": 0
    }
  ]
}
},
"total": 0,
"page": 0
}

```

Errores

- 400: El JSON del request es nulo o no es válido.

3 Glosario de conceptos.

Parte de la guía donde se explica el significado de las propiedades de un documento.

- Id: partitionKey por defecto que asigna el CosmosDb.
- DocumentClassId: partitionKey que se le asigna para tenerlo identificado cuando realiza particiones lógicas el CosmosDb para mejorar su rendimiento. Es de tipo string, por lo que puede contener más valores aparte de números.
- DocumentClass: clase en la que se agrupa el documento.
- DocumentType: tipo de archivo del documento.
- Filename: nombre que se le asigna al archivo que pertenece al documento.
- Properties: son todas las propiedades personalizadas que le podemos asignar a un documento. Como objetivo es que aporte mayor información sobre el documento.
- Created: fecha de creación del documento.
- Modified: fecha de la última modificación del documento.
- CreatedUser: nombre del usuario que subió el documento.
- LastModifiedUser: nombre del último usuario que modifico el documento.
- AbsoluteUri: Uri del documento.
- IsDelete: especifica si el documento esta borrado o no.
- HasLegalHold: especifica si el documento esta borrado o no.
- Size: tamaño del documento.
- BlobName: nombre del Blob del documento.
- Container: nombre del Container del Storage donde esta almacenado el documento.
- FolderName: nombre del FolderName dentro del Container donde esta almacenado el documento.

En referencia con las particiones del CosmosDb se adjunta la siguiente documentación:

<https://docs.microsoft.com/es-es/azure/cosmos-db/partitioning-overview>

Contacto

Para localizar o contactar con ENCAMINA puedes:



Enviar un **mail** a:

encamina@encamina.com

info@encamina.com



Llamar al

962 698 064

917 893 823



Visitarnos en:

C/ O'Donnell, 34. 3ª planta

28009 , Madrid, Madrid

Jerónimo Roure, 49

46520 Puerto de Sagunto, Valencia

Calle del Ángel, 32

30202 Cartagena, Murcia

Av. Trinidad, 61. Campus Central ULL

38200, S.Cristóbal de La Laguna, Tenerife



Enviar un **fax** al 962 698 063



O hablar personalmente con:

- Hugo de Juan, CEO