

encamina

PIENSA EN COLORES

Gada-i Instalación y configuración



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

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

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

info@encamina.com
www.encamina.com

Historia del documento

Fecha	Status	Versión	Autor	Descripción
20/01/2020		1.0	Encamina	
22/06/2021		2.0	Encamina	
29/09/2021		3.0	Encamina	

Índice

1	Objetivo del documento	6
2	Pasos a realizar.....	6
2.1	Prerequisitos.....	6
2.2	Registro de las aplicaciones que consumen GADA en Azure Active Directory.....	6
2.2.1	Comprobación de permisos	6
2.2.2	Creación de las aplicaciones de Azure Active Directory	8
2.2.3	Obtención del Id. y la clave de autenticación de la aplicación.....	10
2.2.4	Registro de aplicaciones que debemos crear.....	11
2.2.4.1	Aplicación Web.....	11
2.2.4.2	Aplicación cliente.....	12
2.3	Cosmos DB	13
2.3.1	Creación de colecciones y bases de datos en una cuenta de Azure Cosmos DB.....	13
2.3.1.1	GadaCollection.....	15
2.3.2	Elementos de Cosmos DB necesarios en la aplicación web.	16
2.4	Storage Account.....	17
2.4.1	Creación de un Blob container.....	17

- 2.4.2 Creación de una Queue.....19
- 2.4.3 Elementos de la Queue necesarios en la aplicación web.20
- 2.5 Configuración a nivel organización.....21
 - 2.5.1 Configurar Azure App Configuration/App Settings de la aplicación web. 21
- 2.6 Azure Search y Cognitive Services.21
 - 2.6.1 Archivos Json generación infraestructura Azure Cognitive Search.21
 - 2.6.2 Creación de un índice.....22
 - 2.6.2.1 Colección de Cosmos DB para configuración.....22
 - 2.6.2.2 Creación de Servicios Cognitivos desde API autonoma.....24
 - 2.6.2.3 Generar el índice manual.....24
 - 2.6.3 Elementos necesarios en la aplicación web y en la Azure Function. ...26
- 2.7 Azure App Configuration.....27
 - 2.7.1 Creación de Azure App Configuration.....27
 - 2.7.2 Funcionalidad extra que aporta Gada-i con AppConfiguration.....28
 - 2.7.3 Elementos necesarios en la aplicación web y en la Azure Function. ...29
- 3 Configuración de Gada-i.30
 - 3.1 Configuración si usa Azure App Configuration30
 - 3.1.1 Configuración del Azure AppConfiguration Gada-i30
 - 3.1.2 Configuración de la Api Gada-i.....34
 - 3.2 Configuración sin Azure App Configuration.....34
 - 3.2.1 Configuración de la Api Gada-i.....34
 - 3.2.2 Configuración de las Azure Function Gada-i.....38
- 4 Permisos de Azure a otorgar a Api Gada-i.41
- Contacto42

Índice de imágenes

Ilustración 1- Microsoft Azure	6
Ilustración 2- Active Directory Role	7
Ilustración 3- Active Directory User Settings	7
Ilustración 4- Active Directory User Settings App Registrations	8
Ilustración 5- Microsoft Azure	8
Ilustración 6- Active Directory Menú	9
Ilustración 7- Active Registrations App registrations	9
Ilustración 8- Registro de aplicación.....	9
Ilustración 9- Listado de Apps Active Directory.....	10
Ilustración 10- Información de App registrada	10
Ilustración 11- Configuración App registrada	10
Ilustración 12- Key App registrada	11
Ilustración 13- Key generada de la App registrada	11
Ilustración 14- Permisos App recién creada.....	11
Ilustración 15- Información de la App.....	12
Ilustración 16- Permisos de la App-1.....	12
Ilustración 17- Permisos de la App-2.....	12
Ilustración 18- Permisos de la App-3.....	13
Ilustración 19- CosmosDB Menú	13
Ilustración 20- Menú de creación de Container CosmosDB	14
Ilustración 21- New Container CosmosDB.....	14
Ilustración 22- Database CosmosDB.....	15
Ilustración 23- Método Mangement/SetConfigFields de la Api GADA-i.....	15
Ilustración 24- Keys CosmosDB	16
Ilustración 25- EndPoint/AuthKey CosmosDB.....	16
Ilustración 26- Configuraciones CosmosDB.....	17
Ilustración 27- Storage Account	17
Ilustración 28- Storage Containers	17
Ilustración 29- New Container Storage.....	18
Ilustración 30- ConnectionString Storage.....	18
Ilustración 31- Configuraciones Storage	19
Ilustración 32- Storage Account	19
Ilustración 33- Queues Storage.....	19
Ilustración 34- New Queue Storage	20
Ilustración 35- ConnectionString Storage.....	20
Ilustración 36- Configuraciones Organization.....	21
Ilustración 37- Ficheros de Configuración	22

Ilustración 38- CosmosConfig de App en GadaCollection CosmosDB	23
Ilustración 39- Método Management/Warm de la Api GADA-i	24
Ilustración 40- Menú Azure Search.....	24
Ilustración 41- Postman petición Search-1.....	25
Ilustración 42- Postman petición Search-2	25
Ilustración 43- Postman petición Search archivo ejemplo	26
Ilustración 44- Creación de Azure App Configuration	27
Ilustración 45- Método AppConfiguration/newAppConfiguration de la Api GADA-i	28
Ilustración 46- Método AppConfiguration/newAppConfiguration de la Api Gada-i	29
Ilustración 47- ConnectionString Azure App Configuration en la Function GADA-i	30

1 Objetivo del documento

El objetivo del presente documento es definir los pasos a seguir para realizar una correcta instalación y configuración de Gada-i.

2 Pasos a realizar

2.1 Prerequisitos

Se presupone que toda la infraestructura de Azure necesaria para Gada-i ya se encuentra desplegada en su grupo de recursos.

Se procede a mostrar la configuración básica de Gada-i

2.2 Registro de las aplicaciones que consumen GADA en Azure Active Directory

2.2.1 Comprobación de permisos

Para seguir el documento, debe tener permisos suficientes para registrar una aplicación en su inquilino de Active Directory y asignar a la aplicación un rol en su suscripción de Azure.

Para comprobar qué permisos tiene:

1. Seleccione **Azure Active Directory**.

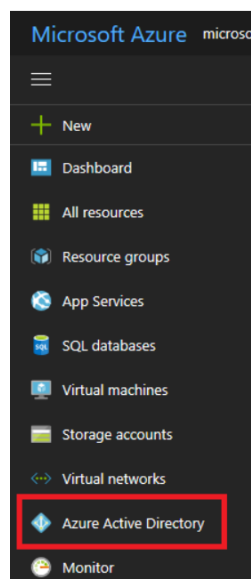


Ilustración 1- Microsoft Azure

2. Seleccione **Información general** y compruebe la información del usuario.

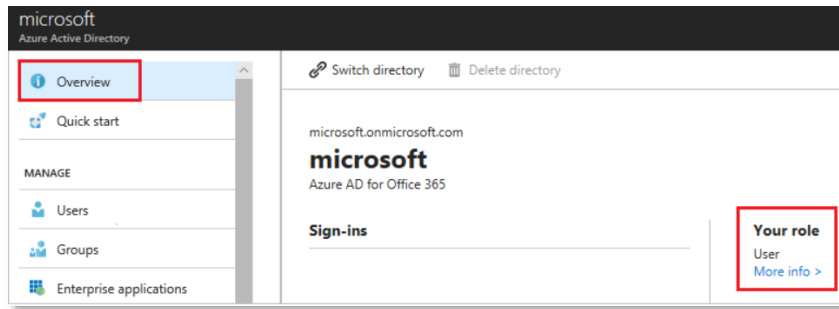


Ilustración 2- Active Directory Role

Si no tiene permisos de administrador global, pida al administrador que le asigne este rol o que permita a los usuarios registrar las aplicaciones. Para esto, deberá seguir los siguientes pasos:

1. En su instancia de Azure Active Directory, seleccione **Configuración de usuario**.

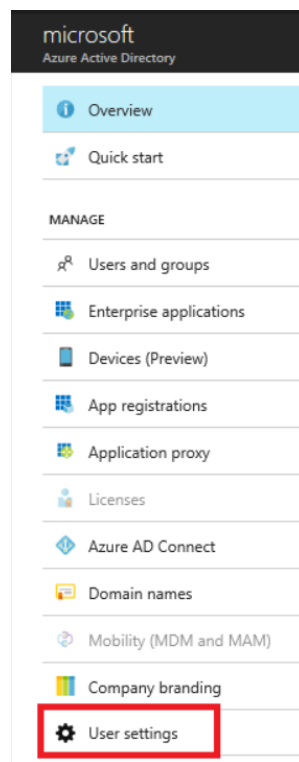


Ilustración 3- Active Directory User Settings

2. Compruebe la configuración de **App registrations** (Registro de aplicaciones). Si está establecida en **Sí**, los usuarios que no sean administradores podrán registrar aplicaciones de AD.

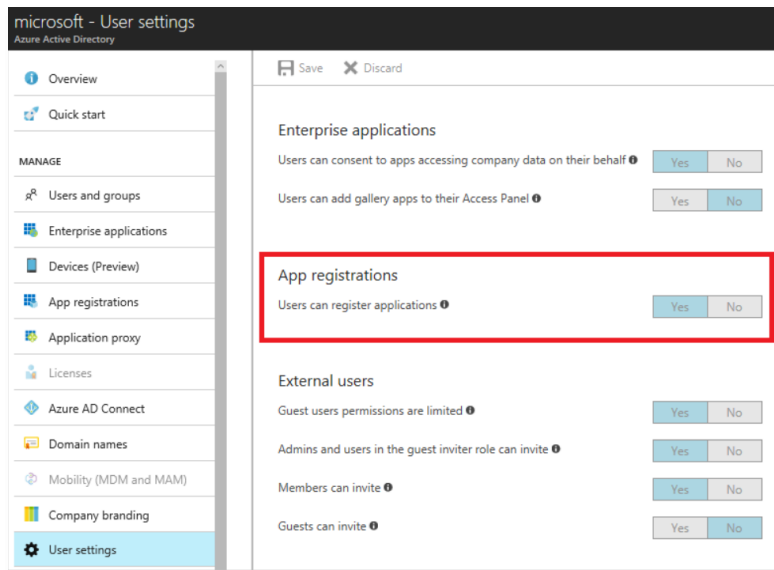


Ilustración 4- Active Directory User Settings App Registrations

2.2.2 Creación de las aplicaciones de Azure Active Directory

Para la creación de las aplicaciones, siga los siguientes pasos:

1. Seleccione **Azure Active Directory**.

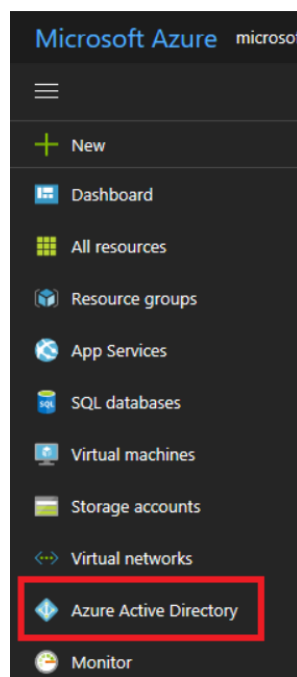


Ilustración 5- Microsoft Azure

2. Seleccione **App registrations** (Registros de aplicaciones).

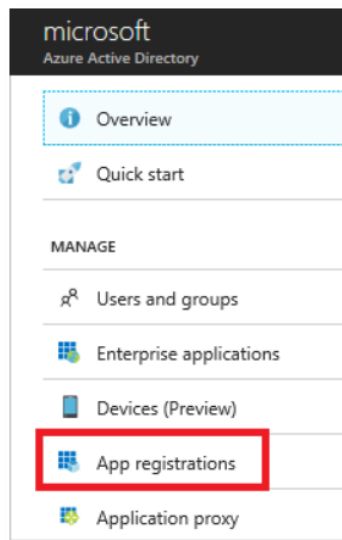


Ilustración 6- Active Directory Menú

3. Seleccione **Nuevo registro de aplicaciones**.



Ilustración 7- Active Registrations App registrations

4. Proporcione un nombre y una dirección URL para la aplicación. Seleccione **Aplicación web o API** para indicar el tipo de aplicación que desea crear. No se pueden crear credenciales para una aplicación *nativa*; por consiguiente, este tipo no funciona en una aplicación automatizada. Después de configurar los valores, seleccione **Crear**.

The image shows a form for creating a new application registration. It has three input fields: 'Nombre' with the value 'api-demo', 'Tipo de aplicación' with a dropdown menu set to 'Aplicación web o API', and 'URL de inicio de sesión' with the value 'https://localhost'. Each field has a green checkmark on the right. At the bottom of the form is a blue 'Crear' button.

Ilustración 8- Registro de aplicación

2.2.3 Obtención del Id. y la clave de autenticación de la aplicación

Para la obtención del identificador de la aplicación y la clave de autenticación es necesario seguir los siguientes pasos:

1. En **Registro de aplicaciones**, en Azure Active Directory, seleccione su nueva aplicación.

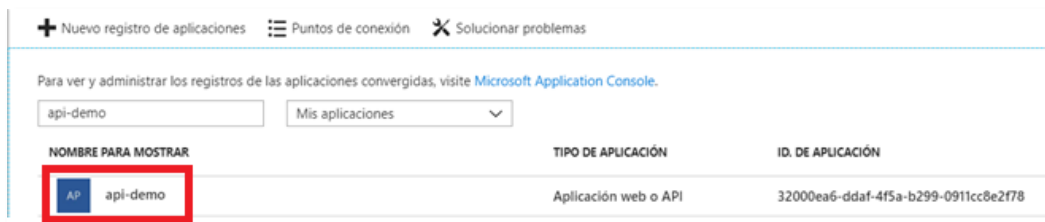


Ilustración 9- Listado de Apps Active Directory

2. Copie el **id. de aplicación** y guárdelo, será usado en los tags **Appid** de configuración para el correcto funcionamiento del blobStorage, search y cosmos DB.

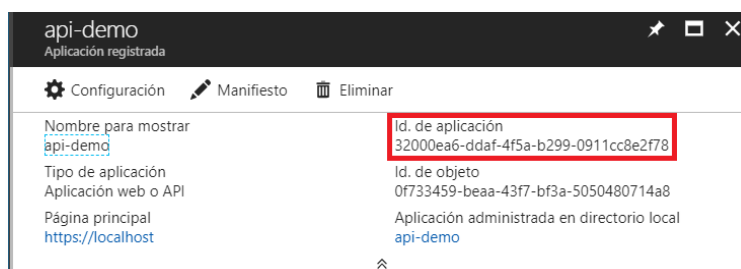


Ilustración 10- Información de App registrada

3. Para generar una clave de autenticación, seleccione **Configuración** y, después, seleccione el apartado **Claves**

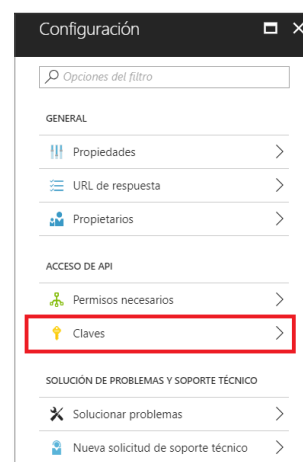


Ilustración 11- Configuración App registrada

- Proporcione una descripción de la clave y una duración. Cuando haya terminado, seleccione **Guardar**.

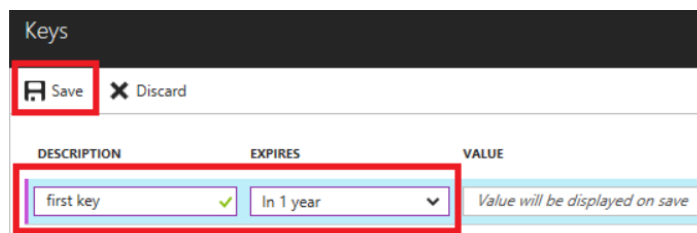


Ilustración 12- Key App registrada

Tras haber guardado la clave, se muestra el valor de esta. Es importante copiar este valor, ya que no podrá recuperarlo más adelante. Proporcione el valor de clave junto con el id. de aplicación para iniciar sesión con la aplicación.

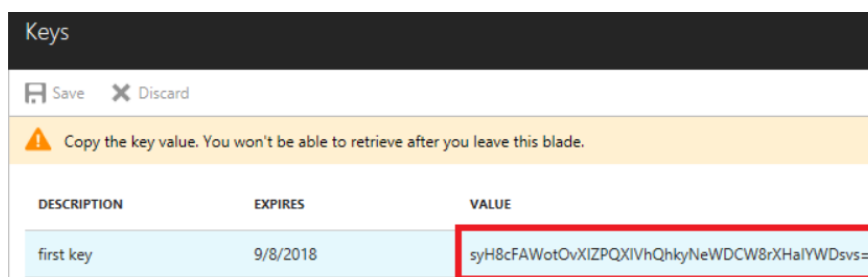


Ilustración 13- Key generada de la App registrada

2.2.4 Registro de aplicaciones que debemos crear

Debemos crear dos registros de aplicaciones.

2.2.4.1 Aplicación Web

Creamos la aplicación básica.

Como vemos, en ella no hace falta que configuremos ningún permiso necesario.



Ilustración 14- Permisos App recién creada

2.2.4.2 Aplicación cliente

Ahora crearemos la aplicación que necesitamos que tenga los permisos que veremos a continuación.

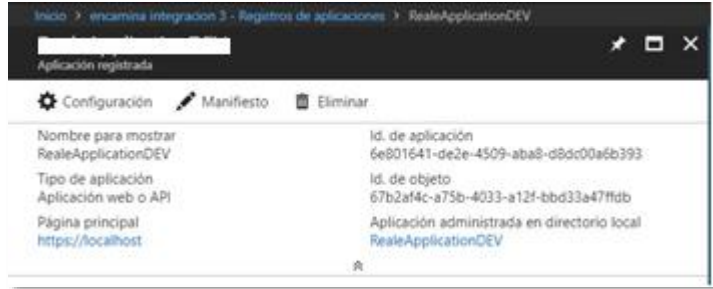


Ilustración 15- Información de la App

Como vemos, en los permisos necesarios le hemos dado acceso a nuestra app básica que hemos creado anteriormente.



Ilustración 16- Permisos de la App-1



Ilustración 17- Permisos de la App-2

También vemos los permisos que debe tener sobre nuestro Windows Azure Active Directory esta segunda aplicación.



Ilustración 18- Permisos de la App-3

2.3 Cosmos DB

En cuanto a Cosmos DB, se necesitará crear una colección para almacenar los metadatos de los documentos de la nueva aplicación.

2.3.1 Creación de colecciones y bases de datos en una cuenta de Azure Cosmos DB

Para empezar, se necesita acceder a la sección de "Data Explorer" de la cuenta de Azure Cosmos DB.

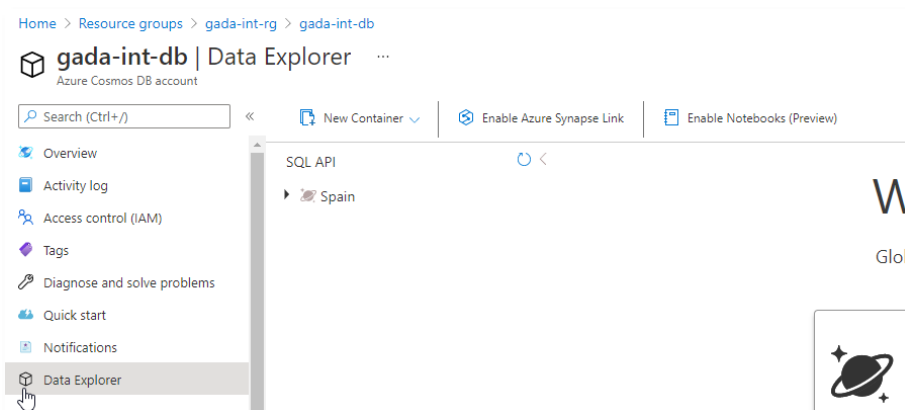


Ilustración 19- CosmosDB Menú

Pulsar en el botón "New Container" del menú.

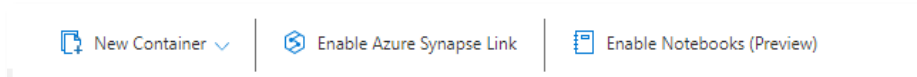


Ilustración 20- Menú de creación de Container CosmosDB

En el nuevo formulario que se abre hay que rellenar el nombre de la colección, los RUs (esto afectará tanto al rendimiento como al costo) y asignarlo, o crear, una base de datos.

Database id: Nombre de la base de datos.

Puede crear una nueva **Database Id** o usar una que se encuentre definida apareciendo en un desplegable para seleccionar.

Database throughput recomendamos manual y a 400 RU/s para un tema de costes. Es libre de elegir el gasto.

Container id: Identificador único para el contenedor y se usa para el enrutamiento basado en identificadores a través de REST y todos los SDK.

Partition key: * `/partitionKey` es **keySensitive importante no modificar este campo ponerlo tal cual se indica**

Ejemplo: (los nombres serán los indicados por el cliente)

Database id: SPAIN

Container id: Aire

Partition key: * `/partitionKey` **No cambia**

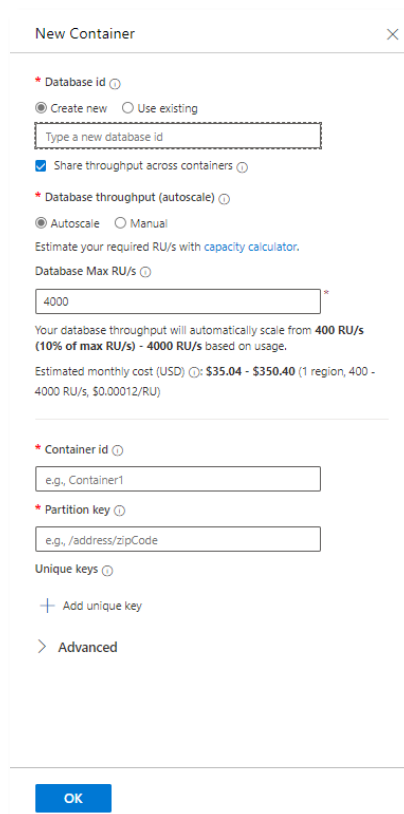


Ilustración 21- New Container CosmosDB

Después, pulsar en el botón "OK" y se comprobará que se ha creado la nueva colección en la base de datos especificada. Quedando cómo se indica en la imagen siguiente.



Ilustración 22- Database CosmosDB

2.3.1.1 GadaCollection

Es importante esta colección ya que obtiene los metadatos asociados a cada AppId en la generación del Search.

Realizar el proceso anterior de creación de una nueva colección llamada GadaCollection

Container id: GadaCollection

Partition key: * /configurationId **No cambia**

Ejemplo de valores en caso de no tener Search

```
"cosmosConfig": [], -> configuración de los metadatos del search  
"id":APPID,  
"configurationId": "1",
```

- Dicha configuración se crea desde la API un método POST /Management/SetConfigFields con el cuerpo del mensaje el listado de valores a ser configurados para el indexer del search.

POST /api/v{v}/Management/SetConfigFields Set Props/metadata configuration

Sample request:

```
POST api/v2.0/Management/SetConfigFields  
[  
  {  
    "metaData": "Header",  
    "searchProp": "AppId"  
  }  
]
```

Ilustración 23- Método Management/SetConfigFields de la Api GADA-i

2.3.2 Elementos de Cosmos DB necesarios en la aplicación web

En este paso se copiarán los valores necesarios de la configuración de Cosmos DB a la configuración de la aplicación de Gada-i. Para ello se necesita acceder a la pestaña de **"Keys"** de Cosmos DB.



Ilustración 24- Keys CosmosDB

En esta sección se encuentran los dos parámetros que se necesitan configurar en el Azure App Configuration. Son el **"Endpoint"** y el **"AuthKey"**.



Ilustración 25- EndPoint/AuthKey CosmosDB

En nuestro aplicativo, estableceremos la siguiente configuración en el Azure App Configuration o en nuestro AppsSettings en caso de no utilizar Azure App Configuration. Para ello se agregarán cuatro nuevas entradas en la sección **"Apps:X:CosmosDB"**:

- **Apps:X:CosmosDB:DB** – La base de datos donde reside la nueva colección de documentos.
- **Apps:X:CosmosDB:Collection** – La colección de documentos que se acaba de crear.
- **Apps:X:CosmosDB:Endpoint** – La url del servicio de Azure CosmosDB.
- **Apps:X:CosmosDB:AuthKey** – La AuthKey para acceder al servidor Azure CosmosDB .

El valor **"X"** será el índice de la nueva entrada en la configuración de la aplicación, por lo que antes de nada hay que verificar cuál es el mayor valor de este índice antes de agregar una configuración nueva.

También tenemos que agregar otras entradas en la sección de **"CosmosDB"**:

- **CosmosDB:AuthKey** – La AuthKey para acceder al servidor Azure CosmosDB.
- **CosmosDB:ConfigurationBD** – La base de datos de CosmosDB que consume Gada-i.
- **CosmosDB:ConfigurationCollection** – La colección donde se establece la configuración por appld de propiedades de documentos con campo de índice.
- **CosmosDB:Endpoint** – La url del servicio del Azure Cosmos DB.
- **CosmosDB:MaxRetriesOnThrottling** – El máximo número de reintentos en caso de que la solicitud falle a causa del límite de consultas impuesto por Azure CosmosDB.

- **CosmosDB:MaxRetryWaitTimeInSeconds** – El máximo tiempo entre reintentos en segundos

Application settings

APP SETTING NAME	VALUE	SLOT SETTING	DELETE
CosmosDB:Endpoint	https://cadocumentacion.documents.azure.com:443/	<input type="checkbox"/>	✕
CosmosDB:AuthKey	b7NGQ6HL0mmLsQoEOafTNDLxOUyT3vu80qwj5sl8OTzl5xVrU8CryzUQOMb7b6cMVjh5...	<input type="checkbox"/>	✕
CosmosDB:MaxRetriesOnThrottling	10	<input type="checkbox"/>	✕
CosmosDB:MaxRetryWaitTimeInSeconds	5	<input type="checkbox"/>	✕

Ilustración 26- Configuraciones CosmosDB

2.4 Storage Account

En esta sección se verá cómo crear un nuevo contenedor en una cuenta de almacenamiento de Azure cómo asociarla en la configuración de la aplicación de Gada-i.

2.4.1 Creación de un Blob container

En el portal de Azure, ir al grupo de recursos de Gada-i y localizar la cuenta de almacenamiento.

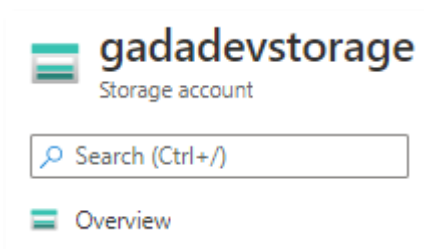


Ilustración 27- Storage Account

Acceder a la sección de "**Containers**".



Ilustración 28- Storage Containers

Una vez situados, se creará un nuevo contenedor. Establecer el nombre y el campo "Public access level" field to "Private".

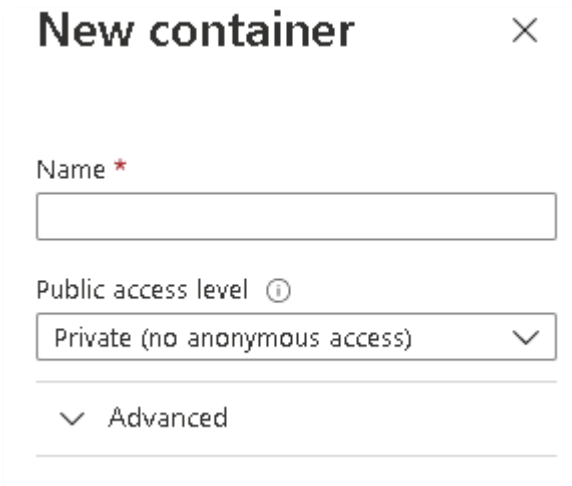


Ilustración 29- New Container Storage

Se han de crear dos Storages debido a las políticas de ciclo de vida para los documentos:

1. Storage en el que crear los contenedores de app para alojar documentos. Se le aplica el ciclo de vida.
 - a. **Apps:X:Container** – Nombre del container para los documentos
 - b. **Apps:X:ConnectionString** – La cadena de conexión.
2. Storage en el que tenemos que crear un contenedor dónde alojar los ficheros de recursos. No se le aplica el ciclo de vida.
 - a. **BlobConfiguration:ContainerResources** – Nombre del container dónde se alojarán los ficheros json.
 - b. **BlobConfiguration:ConnectionString** – La cadena de conexión.

Para configurar la cuenta de almacenamiento en la aplicación de Gada-I, se accede a la pestaña de "Access Keys" y se copia el campo "Connection String".

Ejemplo:

```
DefaultEndpointsProtocol=https;AccountName=storageName;AccountKey=KeyXXXXXXXXXXXXXXXXXXXXX==;EndpointSuffix=core.windows.net
```

Connection string

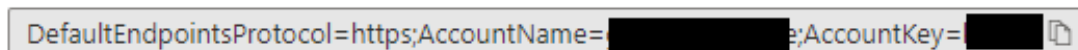


Ilustración 30- ConnectionString Storage

Para completar esta configuración se necesita agregar una entrada similar a la configuración realizada para Cosmos DB. Rellenar la siguiente información:

En la configuración de Azure App Configuration o la app settings de la API:

- **Apps:X:Container** – Nombre del contenedor que se ha creado anteriormente.
- **Apps:X:ConnectionString** – La cadena de conexión.
- **Apps:X:Temp** – Nombre de la carpeta temporal para el nuevo contenedor.

Key ↑↓	Value	Label
Apps:0:ConnectionString	(Hidden value)	(No label)
Apps:0:Container	(Hidden value)	(No label)
Apps:0:Temp	(Hidden value)	(No label)

Ilustración 31- Configuraciones Storage

2.4.2 Creación de una Queue

En el portal de Azure, ir al grupo de recursos de Gada-i y localizar la cuenta de almacenamiento.

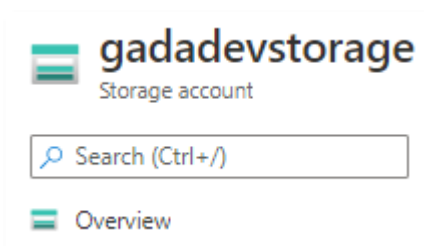


Ilustración 32- Storage Account

Acceder a la sección de "Queues".

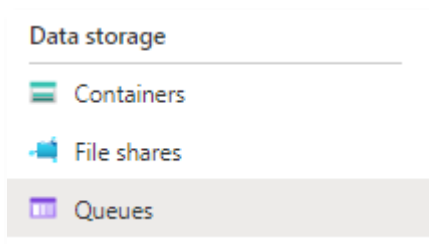


Ilustración 33- Queues Storage

Una vez situados, se creará una nueva cola que será la encargada de comunicar con las Azure Function que se explicarán más adelante para realizar la indexación del documento en el servicio de búsqueda. Establecer el nombre que ha de ser el mismo que el de la settings.

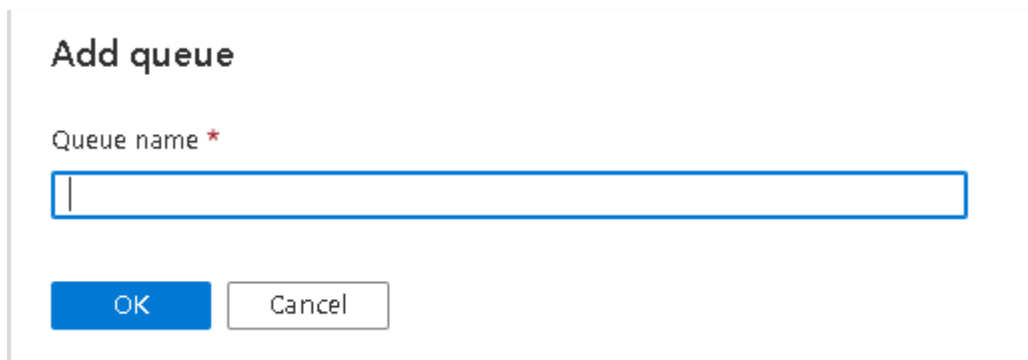


Ilustración 34- New Queue Storage

2.4.3 Elementos de la Queue necesarios en la aplicación web

Para configurar la cuenta de almacenamiento en la aplicación de Gada-I, se accede a la pestaña de "Access Keys" y se copia el campo "Connection String".

Ejemplo:

```
DefaultEndpointsProtocol=https;AccountName=storageName;AccountKey=KeyXXXXXXXXXXXXXXXXXXXXX==;EndpointSuffix=core.windows.net
```

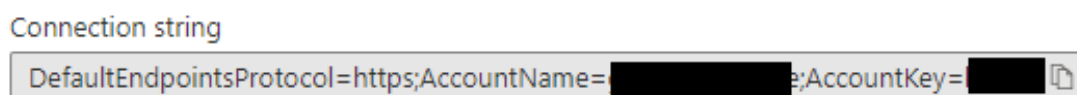


Ilustración 35- ConnectionString Storage

Para completar esta configuración se necesita agregar una entrada similar a la configuración realizada para Cosmos DB. Rellenar la siguiente información:

API En la configuración de Azure App Configuration o la app settings de la API y functions:

- **BlobConfiguration:ConnectionString** – La cadena de conexión.
- **BlobConfiguration:QueueName** – Nombre de la cola creada.

2.5 Configuración a nivel organización

En este paso se configura a nivel Organización, qué aplicaciones consumirán Gada-i y si requieren o no usar Azure Search y/o Cognitive Services para extraer información de los archivos manejados. Para establecerlo, se requiere ir al Azure App Configuration o a la sección “**Application Settings**” de la aplicación web.

2.5.1 Configurar Azure App Configuration/App Settings de la aplicación web

De la misma forma que las secciones anteriores, se necesita dar de alta una nueva entrada en la sección “**Apps:X:Organization**”:

- **Apps:X:Organization:Language** – Localización de la nueva aplicación.
- **Apps:X:Organization:CognitiveConfig** – Si la nueva aplicación que consume Gada-i requiere usar Azure Search y/o Cognitive Services.

Key ↑↓	Value	Label
Apps:0:Organization:Cognitive...	(Hidden value)	(No label)
Apps:0:Organization:Language	(Hidden value)	(No label)

Ilustración 36- Configuraciones Organization

2.6 Azure Search y Cognitive Services

En este punto se procede a explicar cómo configurar el aplicativo para usar el servicio de búsqueda de Azure y los servicios cognitivos.

2.6.1 Archivos Json generación infraestructura Azure Cognitive Search

Existen 3 archivos con extensión “json” que configuran la creación de la infraestructura del servicio de Azure Cognitive Search.

Estos archivos contienen marcadores para realizar la sustitución de valores obtenidos en la configuración.

Estos archivos se deben encontrar en el container del BlobStorage en la carpeta definida (ver: [2.4.1 Creación de un Blob container](#)) en el tag de configuración

BlobConfiguration:ContainerResources, que pueden variar el nombre son los siguientes:

- Gada_DataSource.json
- Gada_IndexerJsonName.json
- Gada_SkillSetJsonName.json

Nombres definidos en los tag:

Apps:X:SearchService:DataSourceJsonName

Apps:X:SearchService:IndexerJsonName

Apps:X:SearchService:SkillSetJsonName

Se adjuntan a este documento estos ficheros.



Ilustración 37- Ficheros de Configuración

2.6.2 Creación de un índice

Se procede a explicar cómo se crea un índice y porqué es necesario realizar ciertas configuraciones en la aplicación Gada-i.

2.6.2.1 Colección de Cosmos DB para configuración

Antes de comenzar con la creación de un índice, se debe tener en cuenta que Gada-i no procesa todas las propiedades del CosmosDB asociado automáticamente, a la hora de generar el servicio de búsqueda y su indexador, lo que hace es serializar las mismas en el campo "**Properties**".

Explicación:

Si se requiere buscar por un campo personalizado, por ejemplo, en la inserción a CosmosDB disponemos de un metadato que se llama ejemplo: "TipoX", esta propiedad en Azure solo aparecerá reflejada sobre el campo "**Properties**", impidiendo que se pueda hacer una búsqueda (que es el objetivo final).

Para solucionar este inconveniente desde Gada-i se creó la posibilidad de enriquecer las propiedades o campos de los que dispone el índice, y de esta manera permitir buscar por cualquier campo personalizado.

A continuación, explicaremos la forma de personalizar y unir los valores de estos.

Como primer punto disponemos de una colección en Gada-i, establecida en la configuración:

- **CosmosDB:ConfigurationBD** – Base de datos donde reside la configuración.
- **CosmosDB:ConfigurationCollection** – Colección donde reside la configuración.

Donde se debe crear una entrada donde el id, debe coincidir con el AppId (ClientId) de la aplicación que va a consumir Gada-i. Su contenido será de la siguiente manera y se muestra como ejemplo:

The screenshot shows a database query result on the left and a JSON configuration on the right. The query result is a table with columns 'id' and '/configurati...'. The JSON configuration is a list of objects under the key 'cosmosConfig'. Two objects are highlighted with red boxes: the first object has 'MetaData': 'IdExpediente' and 'PropsConfigSearch' with 'name': 'IdExpediente'; the second object has 'MetaData': 'FileLeafRef' and 'PropsConfigSearch' with 'name': 'FileLeafRef'.

id	/configurati...
c6a8860a-851c-48...	1
c6a8860a-851c-48...	1
0072be0a-4351-4...	1
6c28f6c1-5211-41...	1
370fdabd-cbc5-4f...	1

```

1  {
2    "cosmosConfig": [
3      {
4        "MetaData": "IdExpediente",
5        "PropsConfigSearch": {
6          "name": "IdExpediente",
7          "type": "Edm.String",
8          "facetable": true,
9          "filterable": true,
10         "key": false,
11         "retrievable": true,
12         "searchable": true,
13         "sortable": false,
14         "analyzer": "es.microsoft",
15         "indexAnalyzer": null,
16         "searchAnalyzer": null,
17         "synonymMaps": [],
18         "fields": []
19       }
20     },
21     {
22       "MetaData": "FileLeafRef",
23       "PropsConfigSearch": {
24         "name": "FileLeafRef",
25         "type": "Edm.String",
26         "facetable": true,
27         "filterable": true,
28         "key": false,
29         "retrievable": true,
30         "searchable": true,
31         "sortable": false,
32         "analyzer": "es.microsoft",
33         "indexAnalyzer": null,
34         "searchAnalyzer": null,
35         "synonymMaps": [],
36         "fields": []
37       }
38     }
39   ]
40 }

```

Ilustración 38- CosmosConfig de App en GadaCollection CosmosDB

En este ejemplo se le indica que busque en CosmosDb el metadato "IdExpediente", y lo asocie a la propiedad del índice de Azure Search "IdExpediente" que es de tipo String.

2.6.2.2 Creación de Servicios Cognitivos desde API autonoma

Para este sistema deberá tener configurado toda la aplicación de GADA-i ya que existe una API que gestiona la creación de los sistemas de Search y Cognitive y los ficheros .json de la colección del blob storage llamado resource.

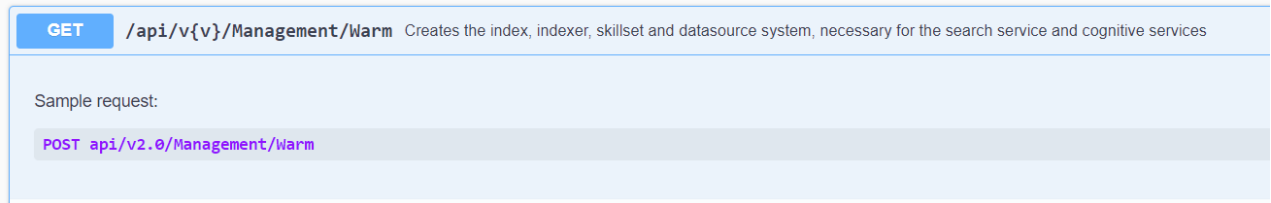


Ilustración 39- Método Management/Warm de la Api GADA-i

* Dicha API generará todo el indexes, Datasource y el indexer obteniendo los elementos de configuración de las settings:

- **Apps:X:SearchService:IndexName** – Nombre del index que se generará cuando se genere la infraestructura.
- **Apps:X:Container** – Nombre del contenedor que se usará en el datasource cuando se genere la infraestructura.
- **Apps:X:ConnectionString** – Cadena de conexión a la cuenta de storage del contenedor que se usará en el datasource cuando se genere la infraestructura.

2.6.2.3 Generar el índice manual

Para la creación de un índice es necesario el API Key del servicio de búsqueda, y haciendo uso de la colección proporcionada en Postman, buscaremos el método indexes:

La API Key la debemos obtener del Servicio de búsquedas tal que:

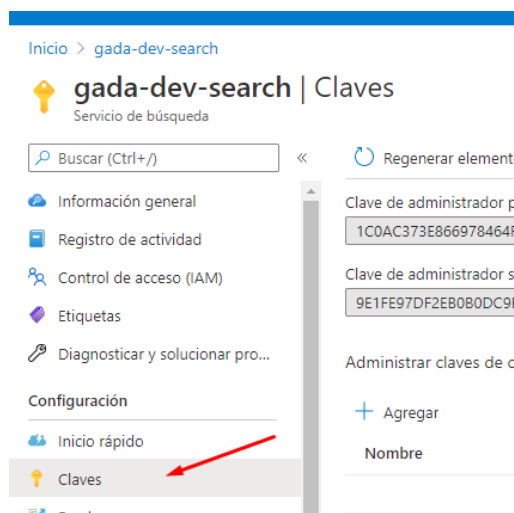


Ilustración 40- Menú Azure Search

Y debe ser remplazada en la cabecera de nuestra petición:

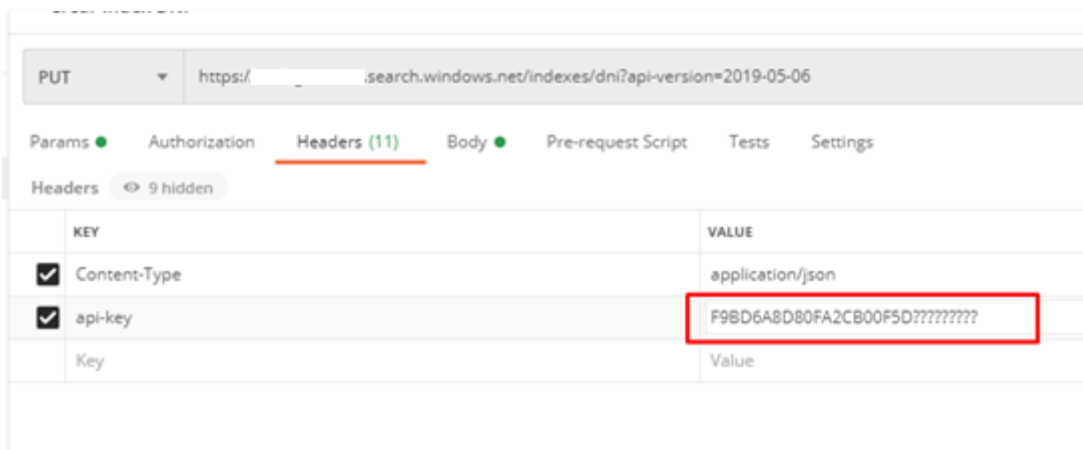


Ilustración 41- Postman petición Search-1

Procediendo a crear el índice:

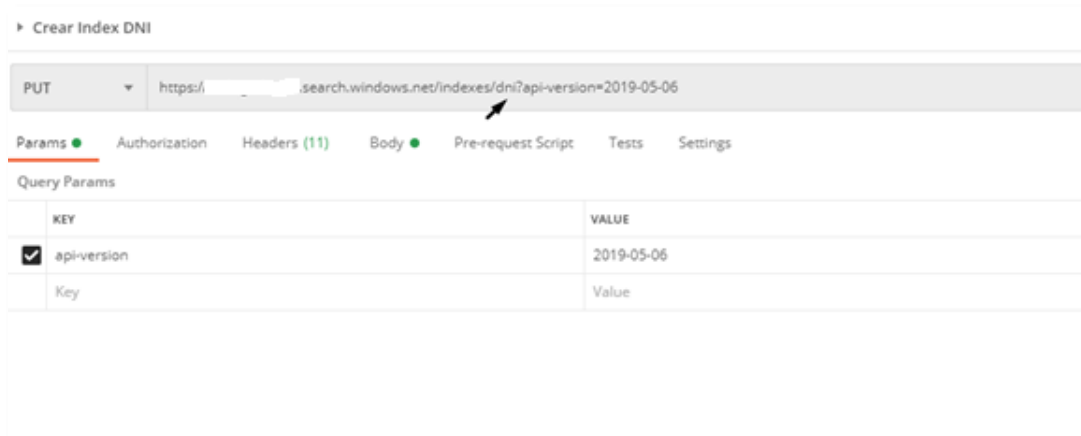


Ilustración 42- Postman petición Search-2

Para la creación de las otras partes del servicio de búsqueda como DataSource, Skillset o Indexer, se debe usar como cuerpo de la petición lo indicado en [los archivos Json del punto 2.6.1](#).

Documentación de Microsoft:

- [Creación SkillSet](#)
- [Creación de DataSource](#)
- [Creación Indexer](#)

Postman (**Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST**) de ejemplo para crear la infraestructura necesaria de Azure Search.



Ilustración 43- Postman petición Search archivo ejemplo

2.6.3 Elementos necesarios en la aplicación web y en la Azure Function

Para completar esta configuración se necesita agregar una entrada en el Azure App Configuration o "App Settings" similar a la configuración realizada para Cosmos DB. Rellenar la siguiente información:

- **SearchService:SearchUrl** – Url del servicio de Azure Cognitive Search
- **SearchService:ApiVersion** – Versión del api sobre la cual realizar las peticiones a Azure Cognitive Search.
- **SearchService:Name** – Nombre del servicio de Azure Cognitive Search.
- **SearchService:AppKey** – AppKey del servicio de Azure Cognitive Search.
- **Apps:X:SearchService:IndexName** – Nombre del índice sobre el que realizar operaciones para el cliente de la aplicación que consume Gada-i.
- **Apps:X:SearchService:IndexerJsonName** – Nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el indexer.
- **Apps:X:SearchService:SkillSetJsonName** – Nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el skillset. (Si no se desea usar Servicios Cognitivos se puede dejar en blanco).
- **Apps:X:SearchService:DataSourceJsonName** – Nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el datasource.
- **Apps:X:SearchService:CognitiveServicesKey** – Key para acceder al Cognitive Services.
- **Apps:X:SearchService:CognitiveServicesDescription** – Id. de recurso de las propiedades del Cognitive Services.

Estos dos últimos ya han tenido que ser agregados en el punto [2.4.1 Creación de un Blob container](#).

- **Apps:X:Container** – Nombre del contenedor que se usará en el datasource cuando se genere la infraestructura.
- **Apps:X:ConnectionString** – Cadena de conexión a la cuenta de storage del contenedor que se usará en el datasource cuando se genere la infraestructura.

2.7 Azure App Configuration

Gada-i da la posibilidad de centralizar su configuración en un Azure AppConfiguration. En el caso que queremos tener el máximo de sus funciones tendremos que crear uno.

2.7.1 Creación de Azure App Configuration

Crear configuración de aplicación ...

Datos básicos
Redes
Etiquetas
Revisar y crear

Azure App Configuration proporciona un servicio para administrar de forma centralizada la configuración de la aplicación y las marcas de características. Los programas modernos, especialmente los programas que se ejecutan en una nube, suelen tener muchos componentes que se distribuyen por naturaleza. La propagación de las opciones de configuración a través de estos componentes puede provocar errores difíciles de solucionar durante la implementación de una aplicación. Use App Configuration para almacenar toda la configuración de la aplicación y proteger sus accesos en un solo lugar. [Más información](#)

Detalles del proyecto

Suscripción *

Grupo de recursos * [Crear nuevo](#)

Detalles de la instancia

Nombre de recurso * ✓
Los nombres de recursos se reservan durante un período de tiempo después de la eliminación. [Más información](#)

Ubicación *

Plan de tarifa * [Ver todos los detalles de los precios](#)

Revisar y crear
< Anterior
Siguiente: Redes >

Ilustración 44- Creación de Azure App Configuration

Debe seleccionar el grupo de recursos en el que se encuentra alojada Gada-i. Dependiendo de sus preferencias podrá asignarle la tarifa "free" o "Estándar".

Una vez lo tenemos creado, tenemos que agregar todas las configuraciones que nos especifica el punto [3.1.1 Configuración del Azure AppConfiguration Gada-i](#).

2.7.2 Funcionalidad extra que aporta Gada-i con AppConfiguration

Funcionalidades extra:

- Agregación de toda la configuración de una app mediante un método de la API. Durante esta llamada también realiza la inserción de las propiedades necesarias en la Collection del CosmosDB del punto [2.3.1.1 GadaCollection](#).

POST /api/v{v}/AppConfiguration/newAppConfiguration Add AppConfigurations of a specific app.

This method has to be used in a unitary way

Sample request:

```

POST /api/v{v}/AppConfiguration/newAppConfiguration
{
  "appId": "string",
  "connectionString": "string",
  "container": "string",
  "minutesSasExpire": "string",
  "temp": "string",
  "cosmosDB": {
    "db": "string",
    "collection": "string",
    "endpoint": "string",
    "authKey": "string"
  },
  "organization":
  {
    "cognitiveConfig": true,
    "language": "string"
  },
  "searchService":
  {
    "indexName": "string",
    "dataSourceJsonName": "string",
    "indexerJsonName": "string",
    "skillSetJsonName": "string",
    "cognitiveServicesKey": "string",
    "cognitiveServicesDescription": "string"
  }
}
        
```

Ilustración 45- Método `AppConfiguration/newAppConfiguration` de la Api GADA-i

- Posibilidad de actualizar las configuraciones de una app mediante un método de la API.

PUT

/api/v{v}/AppConfiguration/updateAppConfiguration
Update AppConfigurations of a specific app.

This method has to be used in a unitary way

Sample request:

```

PUT /api/v{v}/AppConfiguration/newAppConfiguration
{
  "appId": "string",
  "connectionString": "string",
  "container": "string",
  "minutesSasExpire": "string",
  "temp": "string",
  "cosmosDB": {
    "db": "string",
    "collection": "string",
    "endpoint": "string",
    "authKey": "string"
  },
  "organization":
  {
    "cognitiveConfig": true,
    "language": "string"
  },
  "searchService":
  {
    "indexName": "string",
    "dataSourceJsonName": "string",
    "indexerJsonName": "string",
    "skillSetJsonName": "string",
    "cognitiveServicesKey": "string",
    "cognitiveServicesDescription": "string"
  }
}

```

Ilustración 46- Método AppConfiguration/newAppConfiguration de la Api Gada-i

Después de añadir la configuración se debe llamar al [Warm](#) para que complete la configuración de la app de forma automática.

2.7.3 Elementos necesarios en la aplicación web y en la Azure Function

Para completar esta configuración se necesita agregar una entrada en el "App Settings" de Api Gada-i. Rellenar la siguiente información:

- **AppConfiguration:Endpoint** – La url del servicio de Azure App Configuration
- **Settings:UseAppConfiguration** – Booleano que especifica si se utiliza Azure App Configuration.

En el "App Settings" de la function Gada-i. Rellenar la siguiente información:

- **Settings:UseAppConfiguration** – Booleano que especifica si se utiliza Azure App Configuration.
- En las Connection strings tendremos que añadir la ConnectionString del Azure **AppConfiguration**.

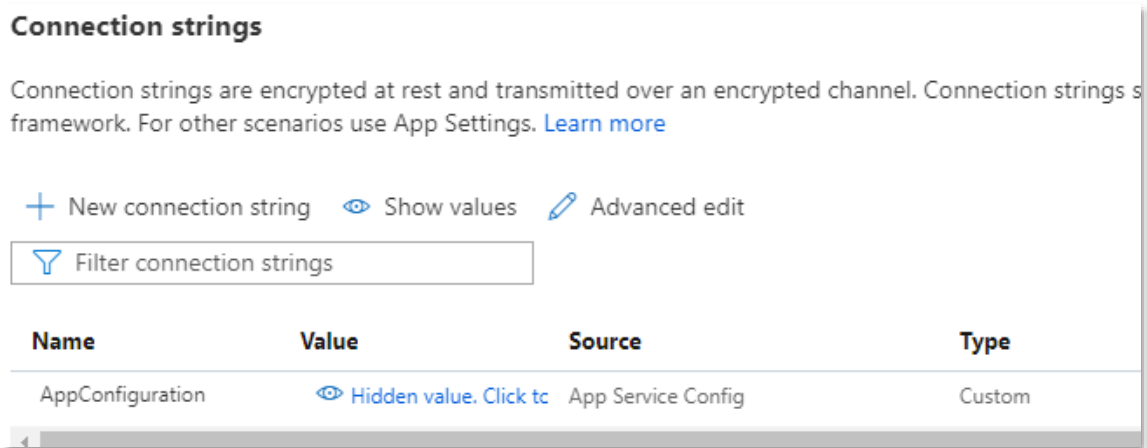


Ilustración 47- ConnectionString Azure App Configuration en la Funcion GADA-i

3 Configuración de Gada-i

Se muestran en las siguientes tablas la recopilación de claves de configuración usadas en la aplicación Gada-i.

Tanto las AppConfiguration y las Settings de GADA-i deben tener los mismos tags y estructura, ya que son análogos.

Indicar que el tag que contenga :X: indica la X el valor del índice al que corresponda por AppId.

Gada-i puede configurarse con Azure App Configuration o sin él.

3.1 Configuración si usa Azure App Configuration

A continuación, se mostrarán como se distribuyen las configuraciones en caso de utilizar Azure App Configuration la aplicación Gada-i.

3.1.1 Configuración del Azure AppConfiguration Gada-i

Se procede a mostrar los diferentes parámetros de la configuración de Gada-i.

CLAVE	VALOR
ApplicationInsights::InstrumentationKey	<clave de instrumentación del Application Insights asociado>
ApplicationInsights::ControlErrors	<array de códigos de error que Application Insights no mostrará como error> [404]

AzureAd:ClientId	<clientId configurado en la API en el Azure AD>
AzureAd:Domain	<dominio del Azure Active Directory de la empresa>
AzureAd:Instance	https://login.microsoftonline.com/
AzureAd:TenantId	<identificador del tenant de NN en el Azure AD>
Apps:X:AppId	<clientId de la aplicación que consume Gada-i>
Apps:X:ConnectionString	<cadena de conexión a la cuenta de storage>
Apps:X:Container	<nombre del contenedor a almacenar los archivos>
Apps:X:MinutesSasExpire	<número de minutos en el que un enlace Sas de acceso a un archivo en el blob storage es válido>
Apps:X:Temp	<nombre del contenedor de almacenamiento temporal>
Apps:X:CosmosDB:AuthKey	<AuthKey para acceder al servidor Azure CosmosDB>
Apps:X:CosmosDB:Endpoint	<url del servicio de Azure CosmosDB>
Apps:X:CosmosDB:DB	<base de datos que usa la aplicación que consume Gada-i>
Apps:X:CosmosDB:Collection	<colección que usa la aplicación que consume Gada-i>
Apps:X:Organization:CognitiveConfig	<valor que establece si se va a usar Azure Cognitive Search>
Apps:X:Organization:Language	<lenguaje por defecto de cada cliente que consume Gada-i>

Apps:X:SearchService:IndexName	<nombre del índice sobre el que realizar operaciones para el cliente de la aplicación que consume Gada-i>
Apps:X:SearchService:DataSourceJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el datasource>
Apps:X:SearchService:IndexerJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el indexer>
Apps:X:SearchService:SkillSetJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el skillset>
Apps:X:SearchService:CognitiveServicesKey	<Key para acceder al Cognitive Services>
Apps:X:SearchService:CognitiveServicesDescription	<Id. de recurso de las propiedades del Cognitive Services>
BlobConfiguration:ConnectionString	<cadena de conexión a la cuenta del storage>
BlobConfiguration:ContainerNotifications	<nombre del Container donde se almacena la configuración del servicio de notificaciones de Gada-i>
BlobConfiguration:ContainerResources	<nombre del Container donde se almacena los archivos de configuración>
BlobConfiguration:QueueName	<nombre de la cola>
CosmosDB:Endpoint	<url del servicio de Azure CosmosDB>
CosmosDB:AuthKey	<AuthKey para acceder al servidor Azure CosmosDB>

CosmosDB:MaxRetriesOnThrottling	<máximo número de reintentos en caso de que la solicitud falle a causa del límite de consultas impuesto por Azure CosmosDB>.
CosmosDB:MaxRetryWaitTimeInSeconds	<máximo tiempo entre reintentos en segundos>
CosmosDB:ConfigurationBD	<base de datos de CosmosDB que consume Gada-i>
CosmosDB:ConfigurationCollection	<colección donde se establece la configuración por appld de propiedades de documentos con campo de índice>
FormOptions:ValueLengthLimit	<límite de longitud en bytes por cada valor enviado individualmente>
FormOptions:MultipartBodyLengthLimit	<límite de longitud en bytes por cada multipart body>
FormOptions:MultipartHeadersLengthLimit	<límite de longitud en bytes para el total de claves y valores de cabeceras en cada sección multipart>
FormOptions:MultipartBoundaryLengthLimit	<límite de longitud en bytes del identificador de boundary>
Logging:Default	<tipo de traza del log>
Logging:LogLevel:Microsoft	<nivel información>
Logging:LogLevel:System	<nivel información>
SearchService:SearchUrl	<url del servicio de Azure Cognitive Search>
SearchService:ApiVersion	<versión del api sobre la cual realizar las peticiones a Azure Cognitive Search>
SearchService:Name	<nombre del servicio de Azure Cognitive Search>
SearchService:AppKey	<appKey del servicio de Azure Cognitive Search>
Settings:Origin	<origen de la configuración>

Settings:Sentinel

<cuando se produce un cambio de su valor actualiza la configuración de la Api>

3.1.2 Configuración de la Api Gada-i

Se procede a mostrar los diferentes parámetros de la configuración de la Api de Gada-i.

CLAVE	VALOR
AppConfiguration:Endpoint	<url del servicio de Azure App Configuration>

3.2 Configuración sin Azure App Configuration

A continuación, se mostrarán como se distribuyen las configuraciones en caso de **no** utilizar Azure App Configuration la aplicación Gada-i.

3.2.1 Configuración de la Api Gada-i

Se procede a mostrar los diferentes parámetros de la configuración de Gada-i.

CLAVE	VALOR
ApplicationInsights::InstrumentationKey	<clave de instrumentación del Application Insights asociado>
ApplicationInsights::ControlErrors	<array de códigos de error que Application Insights no mostrará como error> [404]
AzureAd:ClientId	<clientId configurado en la API en el Azure AD>
AzureAd:Domain	<dominio del Azure Active Directory de la empresa>
AzureAd:Instance	https://login.microsoftonline.com/

AzureAd:TenantId	<identificador del tenant de NN en el Azure AD>
Apps:X:AppId	<clientId de la aplicación que consume Gada-i>
Apps:X:ConnectionString	<cadena de conexión a la cuenta de storage>
Apps:X:Container	<nombre del contenedor a almacenar los archivos>
Apps:X:MinutesSasExpire	<número de minutos en el que un enlace Sas de acceso a un archivo en el blob storage es válido>
Apps:X:Temp	<nombre del contenedor de almacenamiento temporal>
Apps:X:CosmosDB:AuthKey	<AuthKey para acceder al servidor Azure CosmosDB>
Apps:X:CosmosDB:Endpoint	<url del servicio de Azure CosmosDB>
Apps:X:CosmosDB:DB	<base de datos que usa la aplicación que consume Gada-i>
Apps:X:CosmosDB:Collection	<colección que usa la aplicación que consume Gada-i>
Apps:X:Organization:CognitiveConfig	<valor que establece si se va a usar Azure Cognitive Search>
Apps:X:Organization:Language	<lenguaje por defecto de cada cliente que consume Gada-i>
Apps:X:SearchService:IndexName	<nombre del índice sobre el que realizar operaciones para el cliente de la aplicación que consume Gada-i>

Apps:X:SearchService:DataSourceJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el datasource>
Apps:X:SearchService:IndexerJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el indexer>
Apps:X:SearchService:SkillSetJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el skillset>
Apps:X:SearchService:CognitiveServicesKey	<Key para acceder al Cognitive Services>
Apps:X:SearchService:CognitiveServicesDescription	<Id. de recurso de las propiedades del Cognitive Services>
BlobConfiguration:ConnectionString	<cadena de conexión a la cuenta del storage>
BlobConfiguration:ContainerNotificarions	<nombre del Container donde se almacena la configuración del servicio de notificaciones de Gada-i>
BlobConfiguration:ContainerResources	<nombre del Container donde se almacena los archivos de configuración>
BlobConfiguration:QueueName	<nombre de la cola>
CosmosDB:Endpoint	<url del servicio de Azure CosmosDB>
CosmosDB:AuthKey	<AuthKey para acceder al servidor Azure CosmosDB>
CosmosDB:MaxRetriesOnThrottling	<máximo número de reintentos en caso de que la solicitud falle a causa del límite de consultas impuesto por Azure CosmosDB>.

CosmosDB:MaxRetryWaitTimeInSeconds	<máximo tiempo entre reintentos en segundos>
CosmosDB:ConfigurationBD	<base de datos de CosmosDB que consume Gada-i>
CosmosDB:ConfigurationCollection	<colección donde se establece la configuración por appld de propiedades de documentos con campo de índice>
FormOptions:ValueLengthLimit	<límite de longitud en bytes por cada valor enviado individualmente>
FormOptions:MultipartBodyLengthLimit	<límite de longitud en bytes por cada multipart body>
FormOptions:MultipartHeadersLengthLimit	<límite de longitud en bytes para el total de claves y valores de cabeceras en cada sección multipart>
FormOptions:MultipartBoundaryLengthLimit	<límite de longitud en bytes del identificador de boundary>
Logging:Default	<tipo de traza del log>
Logging:LogLevel:Microsoft	<nivel información>
Logging:LogLevel:System	<nivel información>
SearchService:SearchUrl	<url del servicio de Azure Cognitive Search>
SearchService:ApiVersion	<versión del api sobre la cual realizar las peticiones a Azure Cognitive Search>
SearchService:Name	<nombre del servicio de Azure Cognitive Search>
SearchService:AppKey	<appKey del servicio de Azure Cognitive Search>

Settings:Origin	<origen de la configuración>
Settings:UseAppConfiguration	<booleano que dice si se utiliza Azure App Configuration>

3.2.2 Configuración de las Azure Function Gada-i

Se procede a mostrar los diferentes parámetros de la configuración de la Azure Function de Gada-i.

CLAVE	VALOR
Apps:X:AppId	<clientId de la aplicación que consume Gada-i>
Apps:X:ConnectionString	<cadena de conexión a la cuenta de storage>
Apps:X:Container	<nombre del contenedor a almacenar los archivos>
Apps:X:MinutesSasExpire	<número de minutos en el que un enlace Sas de acceso a un archivo en el blob storage es válido>
Apps:X:Temp	<nombre del contenedor de almacenamiento temporal>
Apps:X:CosmosDB:AuthKey	<AuthKey para acceder al servidor Azure CosmosDB>
Apps:X:CosmosDB:Endpoint	<url del servicio de Azure CosmosDB>
Apps:X:CosmosDB:DB	<base de datos que usa la aplicación que consume Gada-i>
Apps:X:CosmosDB:Collection	<colección que usa la aplicación que consume Gada-i>

Apps:X:Organization:CognitiveConfig	<valor que establece si se va a usar Azure Cognitive Search>
Apps:X:Organization:Language	<lenguaje por defecto de cada cliente que consume Gada-i>
Apps:X:SearchService:IndexName	<nombre del índice sobre el que realizar operaciones para el cliente de la aplicación que consume Gada-i>
Apps:X:SearchService:DataSourceJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el datasource>
Apps:X:SearchService:IndexerJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el indexer>
Apps:X:SearchService:SkillSetJsonName	<nombre del archivo json a recuperar con la plantilla a rellenar para enviar a Azure Cognitive Search y generar el skillset>
Apps:X:SearchService:CognitiveServicesKey	<Key para acceder al Cognitive Services>
Apps:X:SearchService:CognitiveServicesDescription	<Id. de recurso de las propiedades del Cognitive Services>
BlobConfiguration:ConnectionString	<cadena de conexión a la cuenta del storage>
BlobConfiguration:ContainerNotificarions	<nombre del Container donde se almacena la configuración del servicio de notificaciones de Gada-i>
BlobConfiguration:ContainerResources	<nombre del Container donde se almacena los archivos de configuración>
BlobConfiguration:QueueName	<nombre de la cola>
	<nombre de la cola>

BlobConfiguration:QueueName	
CosmosDB:Endpoint	<url del servicio de Azure CosmosDB>
CosmosDB:AuthKey	<AuthKey para acceder al servidor Azure CosmosDB>
CosmosDB:MaxRetriesOnThrottling	<máximo número de reintentos en caso de que la solicitud falle a causa del límite de consultas impuesto por Azure CosmosDB>.
CosmosDB:MaxRetryWaitTimeInSeconds	<máximo tiempo entre reintentos en segundos>
CosmosDB:ConfigurationBD	<base de datos de CosmosDB que consume Gada-i>
CosmosDB:ConfigurationCollection	<colección donde se establece la configuración por appld de propiedades de documentos con campo de índice>
SearchService:SearchUrl	<url del servicio de Azure Cognitive Search>
SearchService:ApiVersion	<versión del api sobre la cual realizar las peticiones a Azure Cognitive Search>
SearchService:Name	<nombre del servicio de Azure Cognitive Search>
SearchService:AppKey	<appKey del servicio de Azure Cognitive Search>
Settings:Origin	<origen de la configuración>
Settings:UseAppConfiguration	<booleano que dice si se utiliza Azure App Configuration>

4 Permisos de Azure a otorgar a Api Gada-i

Se pasa a mostrar el listado de permisos que se le tienen que hacer otorgar a la Api de Gada-i. En el caso de no utilizar Azure App Configuration o KeyVault esos permisos no se tendrán que asignar. El resto de los que aparecen es obligatorio que los tenga otorgados.

TAGS	PERMISO
Gada Search	Search Index Data Contributor
Gada Storage	Owner
Gada CosmosDB	Owner
Gada AppConfiguration	App Configuration Data Reader
Gada AppConfiguration	App Configuration Data Owner
Gada KeyVault	App Configuration Data Reader
Gada KeyVault	Asignar todas las directivas de acceso

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

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



Documentos de interés

[Landing - GADA \(encamina.com\)](#)

[GADA-I GUÍA DE LA API.docx](#)

Docuberto: [GADA-i: La historia de un archivo inteligente, llamado Docuberto - YouTube](#)

Conoce Gada en 2 min. : <https://youtu.be/fFxPW1n-s-M>