

# Sistema para el intercambio de información a través de servicios web

## 1. Objetivo

El presente protocolo define las características técnicas que deben tener en cuenta los prestadores y EAPB que decidan usar los servicios web ofrecidos por la cuenta de alto costo dentro del módulo de interoperabilidad SISCAC, permite a las Entidades publicar API's de modo seguro, confiable y a escala. Se utiliza para controlar el uso de servicios entre los equipos internos, asociados y desarrolladores mientras se aprovechan las ventajas que ofrecen los análisis de registros.

## 2. Alcance

Los Prestadores y Aseguradores se les asignara una clave específica para el uso de los modelos de servicios web, previa capacitación y aprobación por parte de la CAC para el correcto funcionamiento del modelo Web Service.

## 3. API Management

Permite a las Entidades publicar API's de modo seguro, confiable y a escala. Se utiliza para controlar el uso de servicios entre los equipos internos, asociados y desarrolladores mientras se aprovechan las ventajas que ofrecen los análisis de registros.

Este servicio proporciona las herramientas que la organización necesita para la administración de API de un extremo a otro, desde el aprovisionamiento de roles de usuario y la creación de planes de uso y cuotas hasta la aplicación de directivas para la transformación de cargas, limitaciones, análisis y alertas.

Método de Transferencia REST diseñado para ser liviano, fácil de usar y adecuado para servicios de gran volumen, protocolo web seguro HTTPS en formato JSON, amigable para el equipo de desarrollo como funcional para la máquina. Los desarrolladores pueden leer y escribir como un lenguaje de programación normal y es fácil de analizar.

## 4. Portal del Desarrollador

El portal para desarrolladores actúa como la presencia web principal para desarrolladores, donde estos pueden:

- probar una API a través de la consola interactiva
- obtener acceso las reglas y variables de los instructivos de la cuenta de alto costo.
- Indicar problemas o dudas relacionados con alguna API.



## 5. Códigos de coberturas y periodos.

Los códigos de cobertura hacen alusión al código de la enfermedad a cargar o probar en las API's disponibles, este código es fijo y no cambia a través el tiempo.

**Códigos de coberturas:**

idCobertura	nemonico	legislacion
5	CÁNCER	RESOLUCION 0247 DE 2014
6	ERC	Resolución Número 004700 de 2008
7	HEMOFILIA	Resolución 123/2015
8	ARTRITIS	RESOLUCION 1393/2015
22	PoblacionSinVih	Resolución 273 de 2019
23	PoblacionConVIH	Resolución 273 de 2019

Los códigos de categorías y periodos hacen alusión al código de la enfermedad correspondiente a la fecha de cohorte a cargar, cuando existan los nuevos códigos del periodo de cohorte 2021, se enviarán a los equipos de desarrollo para poder hacer uso de las API.

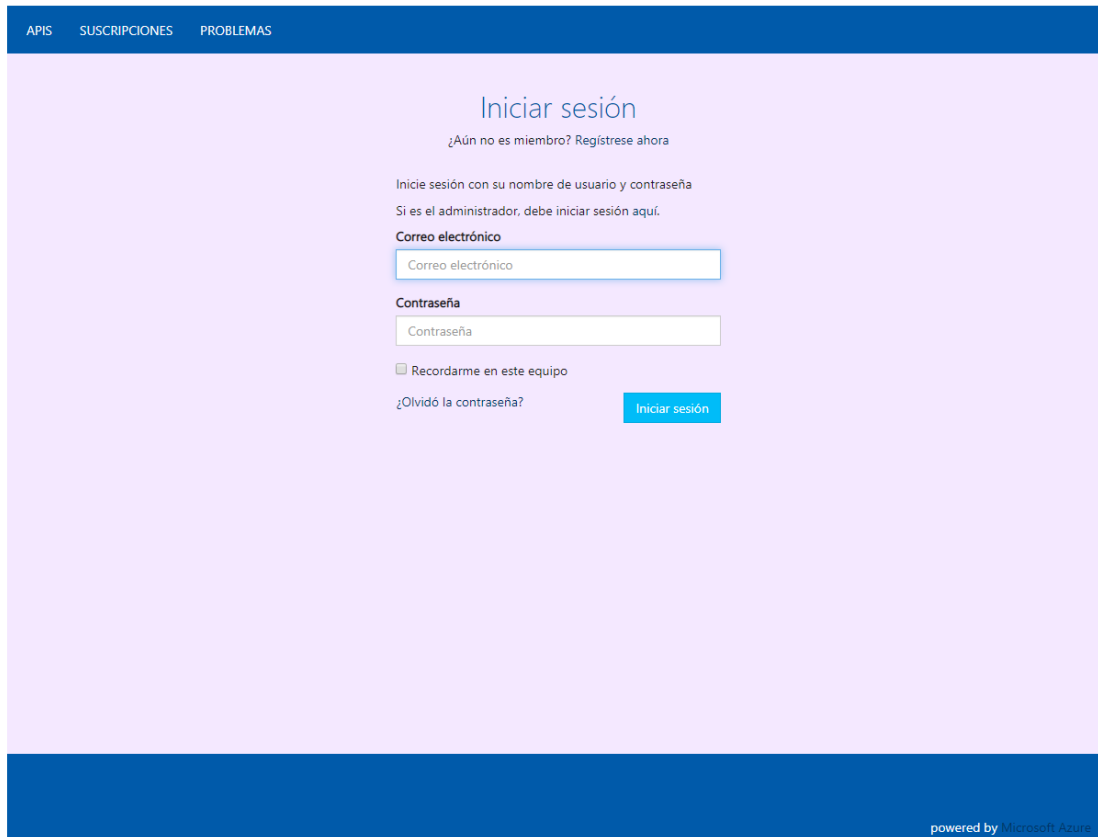
**Códigos de categoría y periodos:**

idperiodo	idCoberturaCAC	nombre	fechaCorte
81	7	Hemofilia Medicion 31-01-2021	31/01/2021
82	22	Personas SIN VIH Medicion 31-01-2021	31/01/2021
83	23	Personas CON VIH Medicion 31-01-2021	31/01/2021
84	5	Cancer Medicion 2021-01-01	1/01/2021
88	6	ERC Medicion 30-06-2021	30/06/2021
89	8	Artritis Medición 2021-06-30	30/06/2021



## 6. Ingreso portal API CAC:

URL del Portal: <https://cac.portal.azure-api.net> para ingresar al sitio Cuenta de Alto Costo le suministrara un usuario y contraseña, una vez permitido el acceso tenemos disponibles diferentes métodos para probar el uso de las API por medio de la consola interactiva.



Una vez adentro podrá ver las API'S publicadas para el uso de las entidades que son:



## API

### Entidades - Cargue de pacientes

Servicio que se encarga de realizar el cargue de información no por aplicación si no directamente consumiendo el servicio

### Entidades - Datos referenciales

Servicio para obtener las tablas de referencia

### Entidades - Reglas de validación

Apis desde donde se exponen las reglas de validación.

## Entidades - Reglas de validación.

Este método nos permite obtener las variables de las diferentes patologías, como sus datos referenciales y tipos de dato.

al dar clic sobre el método, en la parte inferior encontraremos ejemplos en los diferentes lenguajes de programación para poder hacer uso de esta API.

### Ejemplos de código

Curl C# Java JavaScript ObjC PHP Python Ruby

@ECHO OFF

```
curl -v -X POST "https://cac.azure-api.net/cargueExterno/cargarExterna?correoElectronico={string}"
-H "Authorization: Basic dGVzdDp0ZXN0"
-H "Content-Type: application/json"
-H "Ocp-Apim-Subscription-Key: {subscription key}"

--data-ascii "{body}"
```



## Entidades - Reglas de validacion

### Obtener variables tipo A

Api para la obtención de reglas de validación tipo A

Parámetros de consulta

idCategoria

Valor



+ Agregar parámetro

Como parámetro recibe el código de la categoría el cual esta en la tabla anteriormente mencionada.

Una vez ingresado el código de la categoría y dando clic en enviar, el servicio nos retornara un JSON con la información de la variables y tablas referenciales y tipos de dato.

Authorization: Basic dGVzdDp0ZXN0

Ocp-Apim-Subscription-Key: .....

+ Agregar encabezado

Authorization

Subscription key: Principal-c41f... x v

Request URL

https://cac.azure-api.net/reglas/obtener\_a?idCategoria=77

HTTP request

```
GET https://cac.azure-api.net/reglas/obtener_a?idCategoria=77 HTTP/1.1
Host: cac.azure-api.net
Authorization: Basic dGVzdDp0ZXN0
Ocp-Apim-Subscription-Key: .....
```

Enviar



La cantidad de variables depende de la patología.

#### Contenido de la respuesta

Transfer-Encoding: chunked  
Date: Wed, 12 Feb 2020 13:49:36 GMT  
X-Powered-By: ASP.NET  
Content-Type: application/json; charset=utf-8

```
[{
  "orden": 1,
  "nombre": "V1CodigoEAPB",
  "nemonico": "idEPS",
  "descripcion": "Identificador EPS",
  "tipo": "varchar",
  "tipoAlternativo": null,
  "longitud": 10,
  "decimales": 0,
  "formato": "",
  "formatoAlternativo": "",
  "tablaReferencial": "cacEPS",
  "campoReferencial": "idEPS",
  "permiteVacio": "No",
  "identificadorRegistro": "No",
  "clavePrimaria": "Si",
  "sistema": "No",
  "exportable": "No",
  "enmascarado": "No",
  "tipoAnálisisEpidemiológico": null,
  "mensajeErrorTipo": "No cumple con la definicion de V1Codigo de la EAPB o de la entidad territorial AR",
  "mensajeErrorReferencial": "No cumple con la definicion de V1Codigo de la EAPB o de la entidad territorial AR",
  "mensajeErrorVacio": "No cumple con la definicion de V1Codigo de la EAPB o de la entidad territorial AR"
}, {
  "orden": 2,
  "nombre": "V2Regimen",
  "nemonico": "Regimen",
  "descripcion": "Régimen",
  "tipo": "varchar",
  "tipoAlternativo": null,
  "longitud": 1,
  "decimales": 0
}]
```

Con este JSON tenemos que tener presente dos campos:



### Contenido de la respuesta

Transfer-Encoding: chunked  
Date: Wed, 12 Feb 2020 13:49:36 GMT  
X-Powered-By: ASP.NET  
Content-Type: application/json; charset=utf-8

```
[{
  "orden": 1,
  "nombre": "V1CodigoEAPB",
  "nemonico": "idEPS",
  "descripcion": "Identificador EPS",
  "tipo": "varchar",
  "tipoAlternativo": null,
  "longitud": 10,
  "decimales": 0,
  "formato": "",
  "formatoAlternativo": "",
  "tablaReferencial": "cacEPS",
  "campoReferencial": "idEPS",
  "permiteVacio": "No",
  "identificadorRegistro": "No",
  "clavePrimaria": "Si",
  "sistema": "No",
  "exportable": "No",
  "enmascarado": "No",
  "tipoAnálisisEpidemiológico": null,
  "mensajeErrorTipo": "No cumple con la definicion de V1Codigo de la EAPB o de la entidad territorial AR",
  "mensajeErrorReferencial": "No cumple con la definicion de V1Codigo de la EAPB o de la entidad territorial AR",
  "mensajeErrorVacio": "No cumple con la definicion de V1Codigo de la EAPB o de la entidad territorial AR"
}, {
  "orden": 2,
  "nombre": "V2Regimen",
  "nemonico": "Regimen",
  "descripcion": "Régimen",
  "tipo": "varchar",
  "tipoAlternativo": null,
  "longitud": 1,
  "decimales": 0
}
```

- Orden: cuando se intente cargar un paciente por la API [Entidades - Cargue de pacientes](#), se debe tener presente el código del orden para subir la variable a la que corresponde.
- TablaReferencial: cuando se intente cargar un paciente por la API [Entidades - Cargue de pacientes](#), se debe tener presente el nombre de la tabla referencial, con este nombre hacemos uso de la API [Entidades - Datos referenciales](#), la cual explicaremos a continuación.



## Entidades - Datos referenciales

### ObtenerPorTablaReferencial

Parámetros de consulta

TablaReferencial

Valor

✕ Quitar parámetro

+ Agregar parámetro

Encabezados

Authorization

Basic dGVzdDp0ZXN0

Ocp-Apim-Subscription-Key

.....

+ Agregar encabezado

- Entidades - Datos referenciales

Con este servicio podemos obtener en formato JSON los códigos habilitados que se deben cargar por variable. Si colocamos el nombre de la tabla referencial que encontramos en la API [Entidades - Reglas de validación](#), por ejemplo cacEPS :

## Entidades - Datos referenciales

### ObtenerPorTablaReferencial

Parámetros de consulta

TablaReferencial

cacEPS

✕ Quitar parámetro

+ Agregar parámetro

Encabezados

Authorization

Basic dGVzdDp0ZXN0

Ocp-Apim-Subscription-Key

.....

+ Agregar encabezado

Authorization

Subscription key

Principal-c41f...





Como respuesta nos retorna los códigos de EPS habilitados:

Estado de respuesta

200 OK

Latencia de respuesta

211 ms

Contenido de la respuesta

```
Date: Wed, 12 Feb 2020 15:21:14 GMT
X-Powered-By: ASP.NET
Content-Length: 11652
Content-Type: application/json; charset=utf-8

{
  "id": "5da89a08743d4e7f62623ff9",
  "idTablaReferencial": "cacEPS",
  "datosTablaReferencial": [{
    "id": "05000",
    "texto":
  }, {
    "id": "08000",
    "texto":
  }, {
    "id": "08001",
    "texto":
  }, {
    "id": "11001",
    "texto":
  }, {
    "id": "13000",
    "texto":
  }, {
    "id": "13001",
    "texto":
  }, {
    "id": "15000",
    "texto":
  }, {
    "id": "17000",
    "texto":
  }
}
```

En este caso estos códigos deben ser validados antes de ser utilizados en la API de [Entidades - Cargue de pacientes.](#)



## Entidades - Cargue de pacientes

### Cargue

Método que se encarga de cargar los pacientes directamente al servicio, se validan reglas tipo A y tipo B asincrónicamente con notificación al correo

Parámetros de consulta

correoElectronico

Valor



+ Agregar parámetro

Encabezados

Authorization

Basic dGVzdDp0ZXN0

Content-Type

application/json

✕ Quitar encabezado

Ocp-Apim-Subscription-Key

.....



+ Agregar encabezado

## Entidades - Cargue de pacientes

<https://cac.portal.azure-api.net/docs/services/entidades-administracion-cargue-externo/operations/cargue>

al dar clic sobre el método, en la parte inferior encontraremos ejemplos en los diferentes lenguajes de programación para poder hacer uso de esta API.

### Ejemplos de código

Curl C# Java JavaScript ObjC PHP Python Ruby

@ECHO OFF

```
curl -v -X POST "https://cac.azure-api.net/cargueExterno/cargarExterna?correoElectronico={string}"
-H "Authorization: Basic dGVzdDp0ZXN0"
-H "Content-Type: application/json"
-H "Ocp-Apim-Subscription-Key: {subscription key}"

--data-ascii "{body}"
```

Este método se encarga de cargar los pacientes directamente al servicio, se validan reglas tipo A y tipo B asincrónicamente con notificación al correo.



[https://cac.azure-api.net/cargueExterno/cargarExterna\[?correoElectronico\]](https://cac.azure-api.net/cargueExterno/cargarExterna[?correoElectronico])

Parámetros de la solicitud por QueryString: correoElectronico

Ha este correo electrónico se le notificarán los errores al momento de cargar los pacientes, como también los cargues exitosos, esta notificación es por paciente.

### Encabezados de la solicitud

Authorization

**String**

credenciales dadas para la autenticación de la entidad

Content-Type (opcional)

**String**

Tipo de medio del cuerpo enviado a la API.

Ocp-Apim-Subscription-Key

**String**

Clave de suscripción que proporciona acceso a esta API. Se encuentra en su perfil de.

Para el cargue de información debemos crear una estructura en formato JSON, con la siguiente información.

Este JSON se compone de Encabezado y variables a cargar.

En el encabezado

### Cuerpo de la solicitud

```
{  
  "codigoEntidadRegistra": "AUDITA",  
  "id": "null",  
  "codigoEntidadPertenece": "AUDITA",  
  "tipoDocumento": "CC",  
  "numeroDocumento": "123456",  
  "idCobertura": 8,  
  "idPeriodo": 77,  
  "idCategoria": 77,  
  "idEstadoValidacion": 1,
```



**CodigoEntidadRegistro:** Debe ir el numero del NIT en el caso de los prestadores y el código de EPS en el caso de aseguradores.

**Id:** Se envía siempre null.

**CodigoEntidadPertenece:** Debe enviarse el código del Asegurador donde está habilitado el paciente.

**TipoDocumento:** Tipo de identificación del paciente.

**numeroDocumento:** número de identificación del paciente

**idCobertura:** Código cobertura mencionadas en la table de códigos de periodos y coberturas.

**idPeriodo:** El código del periodo y la cobertura es el mismo.

**idCategoria:** El código del periodo y la cobertura es el mismo.

**idEstadoValidacion:** Se envía siempre 1 para quedar en estado guardado.

**idEstadoProceso:** Se envía siempre 1 para quedar en estado guardado.

## VARIABLES

En el JSON enviado se debe generar un Array con las variables a cargar, el código del orden, el nombre del campo y en el caso de las variables que utilicen datos referenciales, el cambio obligatorio se puede enviar como false.

```
"variables":[
  {
    "orden": "1",
    "campo": "idEPS",
    "valor": "ESS207",
    "obligatorio": true
  },
  {
    "orden": "2",
    "campo": "tipoidentificacion",
    "valor": "TI",
    "obligatorio": true
  },
]
```



```
{
  "orden": "3",
  "campo": "identificacion",
  "valor": "22363206",
  "obligatorio": true
},
{
  "orden": "4",
  "campo": "Fecha realizacion actividad",
  "valor": "2017-07-14",
  "obligatorio": false
},
{
  "orden": "5",
  "campo": "Codigo de la actividad",
  "valor": "890237"
},
{
  "orden": "6",
  "campo": "Nombre actividad",
  "valor": " VEZ POR ESPECIALISTA EN CIRUGIA ONCOLOGICA"
},
{
  "orden": "7",
  "campo": "Tipo actividad",
  "valor": "373"
},
{
  "orden": "8",
  "campo": "CIE10 principal actividad",
  "valor": "R190"
```



```
    },  
    {  
      "orden": "9",  
      "campo": "CIE10secundarioactividad",  
      "valor": "999"  
    },  
    {  
      "orden": "10",  
      "campo": "Costoatencion",  
      "valor": "14637"  
    }  
  ]  
}
```

## JSON COMPLETO

```
{  
  "codigoEntidadRegistra": "AUDITA",  
  "id": "null",  
  "codigoEntidadPertenece": "AUDITA",  
  "tipoDocumento": "CC",  
  "numeroDocumento": "123456",  
  "idCobertura": 8,  
  "idPeriodo": 65,  
  "idCategoria": 65,  
  "idEstadoValidacion": 1,  
  "idEstadoProceso": 1,  
  "variables": [  
    {  
      "orden": "1",  
      "campo": "idEPS",
```



```
"valor": "ESS111",
"obligatorio": true
},
{
  "orden": "2",
  "campo": "tipoidentificacion",
  "valor": "CC",
  "obligatorio": true
},
{
  "orden": "3",
  "campo": "identificacion",
  "valor": "123456",
  "obligatorio": true
},
{
  "orden": "4",
  "campo": "Fecha realizacion actividad",
  "valor": "2017-07-14",
  "obligatorio": false
},
{
  "orden": "5",
  "campo": "Codigo de la actividad",
  "valor": "890237"
},
{
  "orden": "6",
  "campo": "Nombre actividad",
  "valor": " VEZ POR ESPECIALISTA EN CIRUGIA ONCOLOGICA"
},
}
```



```
{
  "orden": "7",
  "campo": "Tipoactividad",
  "valor": "373"
},
{
  "orden": "8",
  "campo": "CIE10principalActividad",
  "valor": "R190"
},
{
  "orden": "9",
  "campo": "CIE10secundarioactividad",
  "valor": "999"
},
{
  "orden": "10",
  "campo": "Costoatencion",
  "valor": "14637"
}
]
}
```

Si intentamos cargar un paciente con algún error referencial, este no se carga en el módulo de pacientes y el servicio nos devuelve el error correspondiente.





#### Estado de respuesta

200 OK

#### Latencia de respuesta

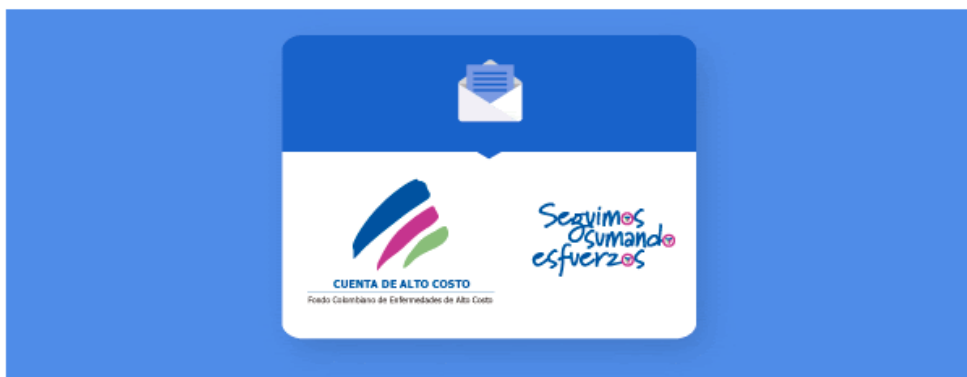
9300 ms

#### Contenido de la respuesta

Transfer-Encoding: chunked  
Date: Wed, 12 Feb 2020 19:02:02 GMT  
X-Powered-By: ASP.NET  
Content-Type: application/json; charset=utf-8

```
{  
  "nemonico": "tipoidentificacion",  
  "posicion": "2",  
  "mensajeError": "No cumple con la definicion de V8Tipo de identificacion del paciente AR"  
}
```

Y al correo electrónico Ingresado Como parámetro llega la respectiva notificación.



## Cargue de Pacientes

Se informa que el proceso de **Validación Reglas Tipo A** obtuvo las siguientes alertas:

tipoidentificacion - No cumple con la definicion de V8Tipo de identificacion del paciente AR - 2



Si el JSON cumple con las reglas de validación este se carga en SISCAC en el módulo de pacientes.

#### Estado de respuesta

201 Created

#### Latencia de respuesta

4763 ms

#### Contenido de la respuesta

Transfer-Encoding: chunked  
Date: Wed, 12 Feb 2020 19:48:20 GMT  
Location: http://40.78.45.251:81/api/CargaPaginada/uno?id=5e445683b6764722d49c9964  
X-Powered-By: ASP.NET  
Content-Type: application/json; charset=utf-8

```
{
  "_id": "5e445683b6764722d49c9964",
  "id": "null",
  "codigoEntidadRegistra": "AUDITA",
  "codigoEntidadPertenece": "AUDITA",
  "tipoDocumento": "CC",
  "numeroDocumento": "123456",
  "fechaRegistro": null,
  "fechaActualizacion": "2020-02-12T14:48:16.6817684-05:00",
  "idCobertura": 8,
  "idPeriodo": 77,
  "idCategoria": 77,
  "idEstadoValidacion": 1,
  "idEstadoProceso": 1,
  "variables": [{
    "orden": 1,
    "campo": "idEPS",
    "valor": "ESS207",
    "valorNumerico": 0,
    "valorFecha": null,
    "type": null,
    "obligatorio": true
  }, {
    "orden": 2,
    "campo": "tipoidentificacion",
```





## Cargue de Pacientes

Se informa que el proceso de **Finalización cargue de pacientes** obtuvo las siguientes alertas:

Su archivo se cargo Exitosamente



Nota: En todas las API de la cuenta se pide en el encabezado de conexión dos tokens, para el caso de Autorization, se debe ingresar de la siguiente forma al realizar la conexión en el desarrollo:

```
string authInfo = "123456789: DesarrolladorPass ";
```

reemplazar 123456789 por el NIT en el caso de prestador o por el IDEPS en el caso de aseguradores.

Reemplazar DesarrolladorPass por la clave del portal CAC-API.

Ejemplo de código C#, API reglas de validación:

```
0 referencias
public class Reglas
{
    0 referencias
    public async void ReglasSISCAC()
    {
        var client = new HttpClient();
        string authInfo = "123456789:DesarrolladorPass";
        authInfo = Convert.ToBase64String(Encoding.Default.GetBytes(authInfo));
        // Request headers
        client.DefaultRequestHeaders.Add("Authorization", "Basic " + authInfo);
        client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", "564564564dffgdfgdfgdg");

        // Request parameters
        var uri = "https://cac.azure-api.net/reglas/obtener_a?idCategoria=74";
        (variable local) string uri

        var response = await client.GetAsync(uri);
        Console.Write(response.Content.ReadAsStringAsync().Result);
    }
}
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleAppServiciosSISCAC
{
    public class Reglas
    {
        public async void ReglasSISCAC()
        {
            var client = new HttpClient();
            string authInfo = "12345678910:Desarrolador123";
            authInfoConvert.ToBase64String(Encoding.Default.GetBytes(authInfo));
            // Request headers
            client.DefaultRequestHeaders.Add("Authorization", "Basic " +
authInfo);
            client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key",
"23434234234324");

            // Request parameters
            var uri = "https://cac.azure-
api.net/reglas/obtener_a?idCategoria=74";

            var response = await client.GetAsync(uri);
            Console.Write(response.Content.ReadAsStringAsync().Result);
        }
    }
}
```

