

# Breaking Cloud Cognitive

## Lab 1: Procesamiento cognitivo del lenguaje natural



## Tabla de contenidos

---

<b>Introducción al laboratorio .....</b>	<b>3</b>
1- Creamos el servicio de Watson Natural Language Understanding.....	4
2- Analizamos textos con Watson NLU .....	7
3- Analizamos recetas con Watson NLU.....	13



## Introducción al laboratorio

En este laboratorio, adquirirás los conocimientos necesarios para utilizar Watson AI en el procesamiento del lenguaje natural. Para ello, utilizaremos en primer lugar el servicio de **Natural Language Understanding** (NLU) para analizar las características semánticas de los textos y más tarde, utilizaremos **Watson Knowledge Studio** para crear nuestros propios anotadores y entrenar nuestro modelo de **machine learning**.

### Requisitos:

- Tener cuenta de IBM Cloud
- Acceso a Internet

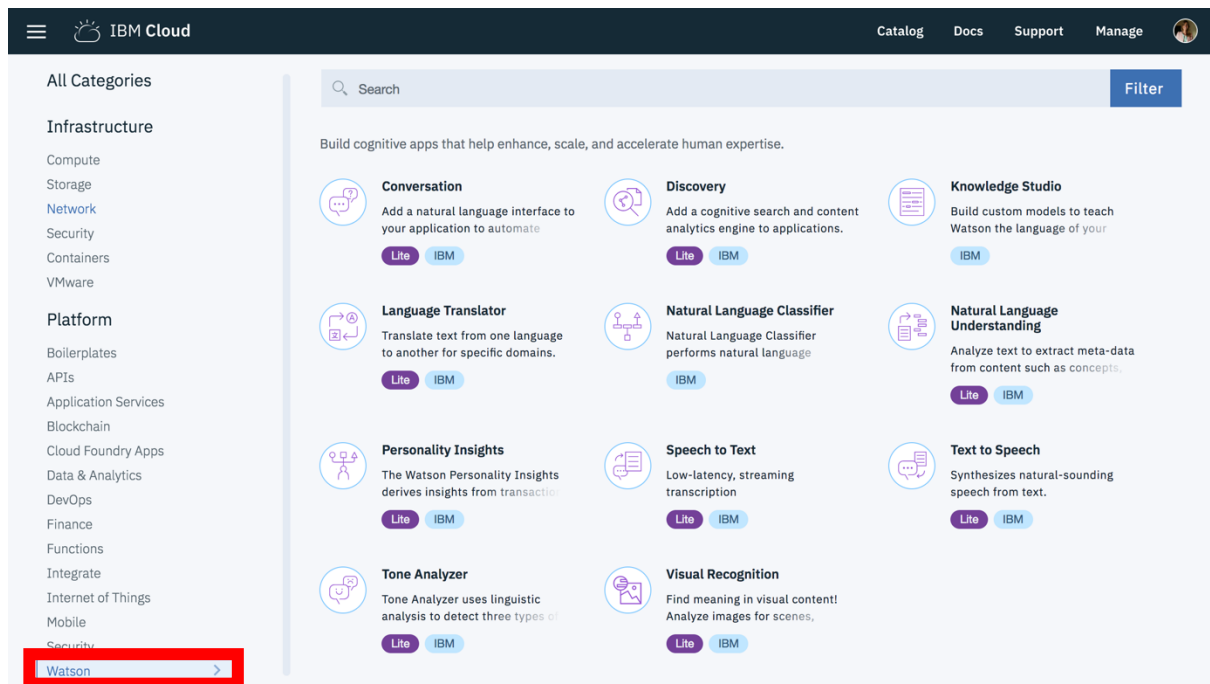
## 1- Creamos el servicio de Watson Natural Language Understanding

Para crear el servicio de Watson NLU, necesitamos acceder al catálogo de **IBM Cloud** desde la siguiente URL: [bluemix.net](https://bluemix.net)

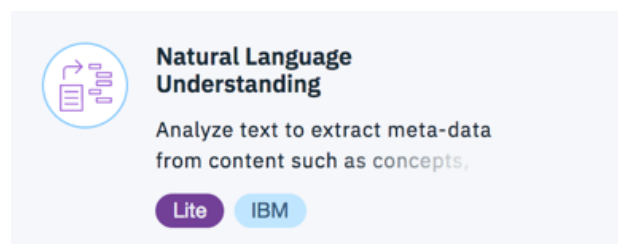
Una vez hemos accedido a **IBM Cloud**, hacemos click en catálogo (barra superior) como se muestra en la imagen:



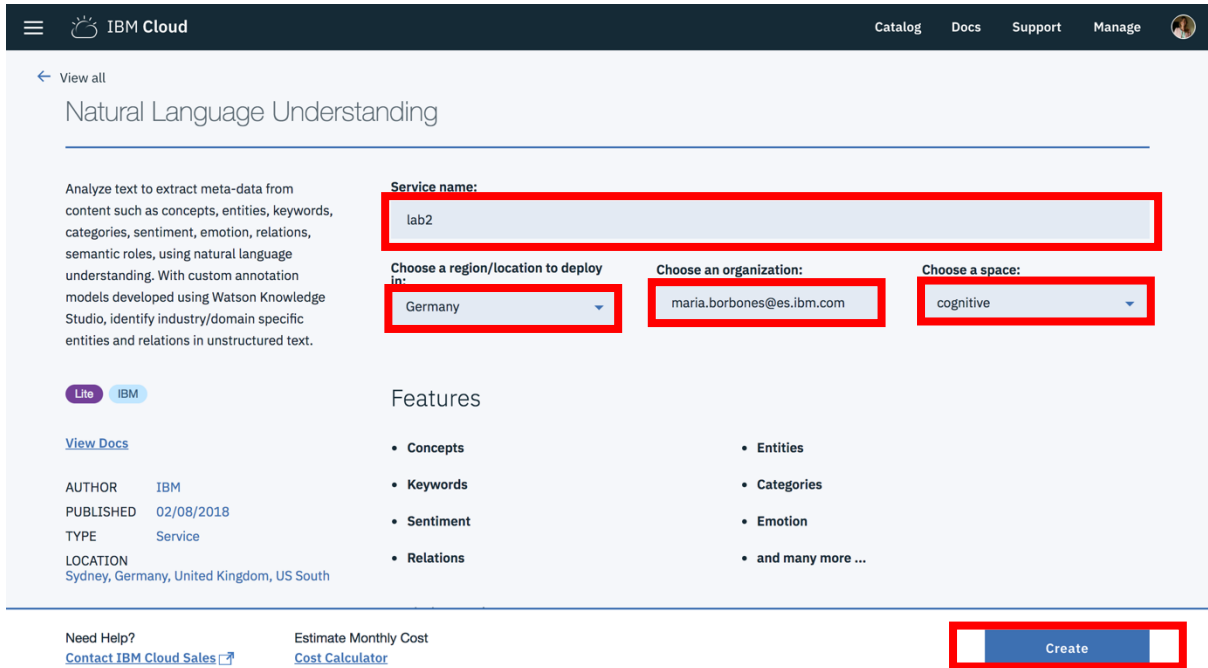
En el menú de la izquierda, donde se muestran todos los servicios de IBM Cloud, buscamos la categoría **Watson** debajo de **Plataforma** como se muestra en la imagen y hacemos click en ella:



En este caso, entre todo el conjunto de servicios, vamos a elegir desplegar Natural Language Understanding, así que lo buscamos y hacemos click sobre el mismo:



En la siguiente pantalla, debemos asignarle un nombre al servicio (por ejemplo: lab1), una región (recomendable EEUU), nuestra organización y el espacio de trabajo donde queremos desplegarlo. Hacemos click en **crear**.



IBM Cloud

View all

## Natural Language Understanding

Analyze text to extract meta-data from content such as concepts, entities, keywords, categories, sentiment, emotion, relations, semantic roles, using natural language understanding. With custom annotation models developed using Watson Knowledge Studio, identify industry/domain specific entities and relations in unstructured text.

Service name: lab2

Choose a region/location to deploy in: Germany

Choose an organization: maria.borbones@es.ibm.com

Choose a space: cognitive

Life IBM

View Docs

AUTHOR IBM  
PUBLISHED 02/08/2018  
TYPE Service  
LOCATION Sydney, Germany, United Kingdom, US South

### Features

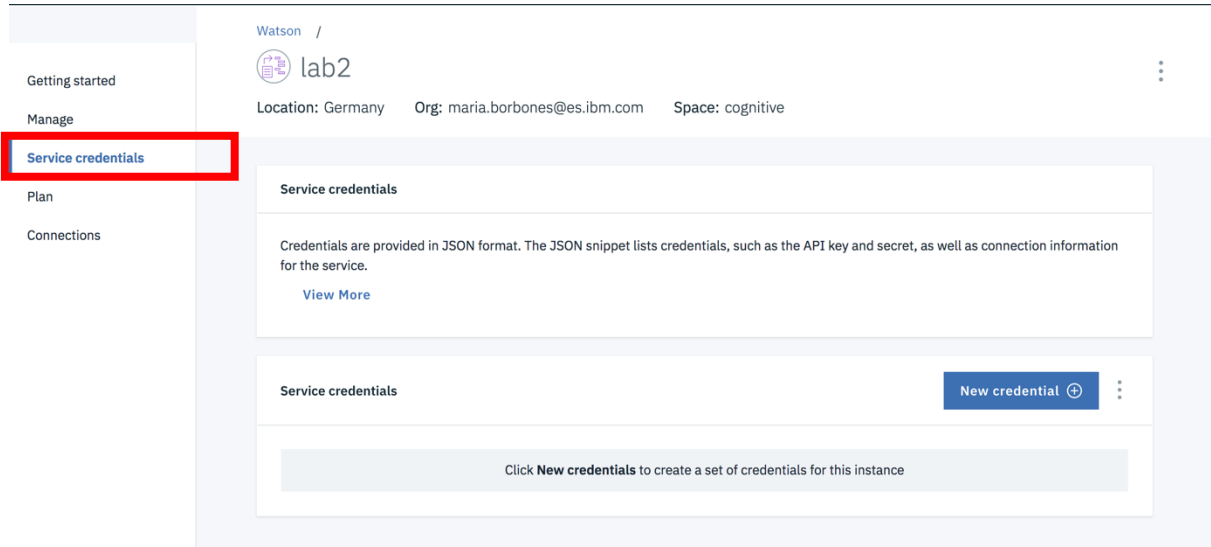
- Concepts
- Keywords
- Sentiment
- Relations
- Entities
- Categories
- Emotion
- and many more ...

Need Help? [Contact IBM Cloud Sales](#)

Estimate Monthly Cost [Cost Calculator](#)

Create

Una vez creado el servicio, accedemos a las credenciales ya que vamos a necesitarlas para poder invocar al servicio. Así que, desde el panel lateral izquierdo accedemos a las credenciales del servicio como se muestra en la imagen:



Watson / lab2

Location: Germany Org: maria.borbones@es.ibm.com Space: cognitive

Getting started

Manage

**Service credentials**

Plan

Connections

### Service credentials

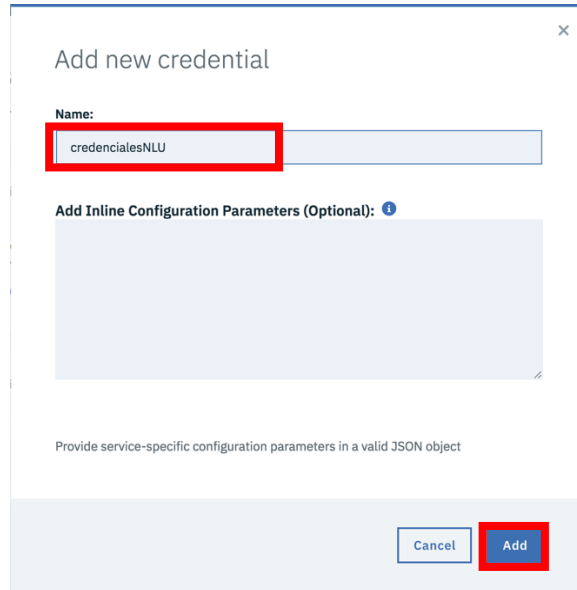
Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

[View More](#)

Service credentials [New credential](#)

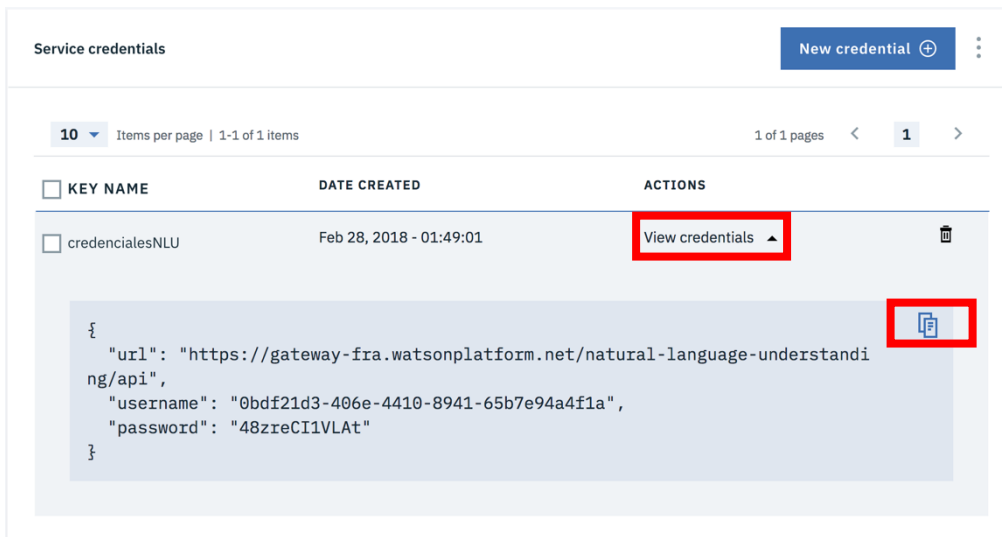
Click **New credentials** to create a set of credentials for this instance

Y hacemos click en nuevas credenciales. Asignamos un nombre (Por ejemplo: credencialesNLU) y le damos a añadir.



A dialog box titled "Add new credential" with a close button (X) in the top right corner. It contains a "Name:" label followed by a text input field containing "credencialesNLU", which is highlighted with a red rectangle. Below this is a section for "Add Inline Configuration Parameters (Optional):" with an information icon (i) and a large text area. At the bottom, there are "Cancel" and "Add" buttons, with the "Add" button highlighted by a red rectangle.

Una vez creadas, hacemos click en ver credenciales y las copiamos en cualquier editor de texto para poder acceder a ellas más tarde.



The "Service credentials" page shows a table with one item. The table has columns for "KEY NAME", "DATE CREATED", and "ACTIONS". The item "credencialesNLU" is listed with a creation date of "Feb 28, 2018 - 01:49:01". The "ACTIONS" column for this item contains a "View credentials" link, which is highlighted with a red rectangle. Below the table, the JSON credentials are displayed in a text area, and a copy icon (two overlapping sheets of paper) is highlighted with a red rectangle.

KEY NAME	DATE CREATED	ACTIONS
credencialesNLU	Feb 28, 2018 - 01:49:01	View credentials

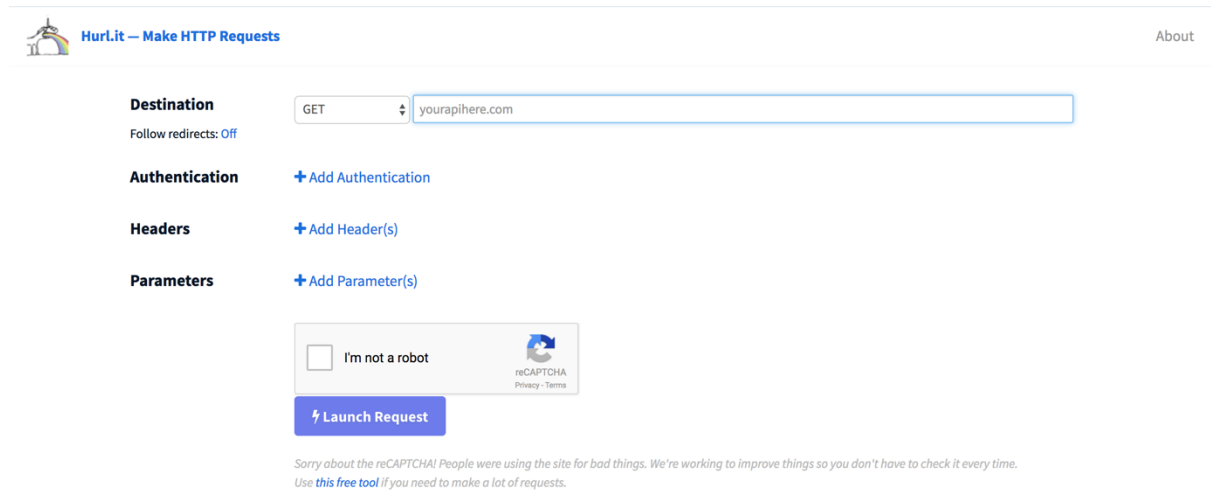
```
{
  "url": "https://gateway-fra.watsonplatform.net/natural-language-understandi
ng/api",
  "username": "0bdf21d3-406e-4410-8941-65b7e94a4f1a",
  "password": "48zreCI1VLAt"
}
```

*¡Enhorabuena! Has completado la primera parte del laboratorio. Ya sabes cómo desplegar un servicio de Watson en IBM Cloud.*

## 2- Analizamos textos con Watson NLU

Ahora nos ponemos manos a la obra y empezamos a trabajar con **Watson NLU**. Primero, como **Watson NLU** no tiene una interfaz web y sólo es consumible via API REST, necesitamos una herramienta que nos permita realizar peticiones HTTP a nuestro servicio de IBM Cloud.

Para ello, vamos a acceder en una nueva pestaña del navegador a <https://www.hurl.it/>, una herramienta que nos va a permitir simular peticiones HTTP desde nuestro propio navegador.



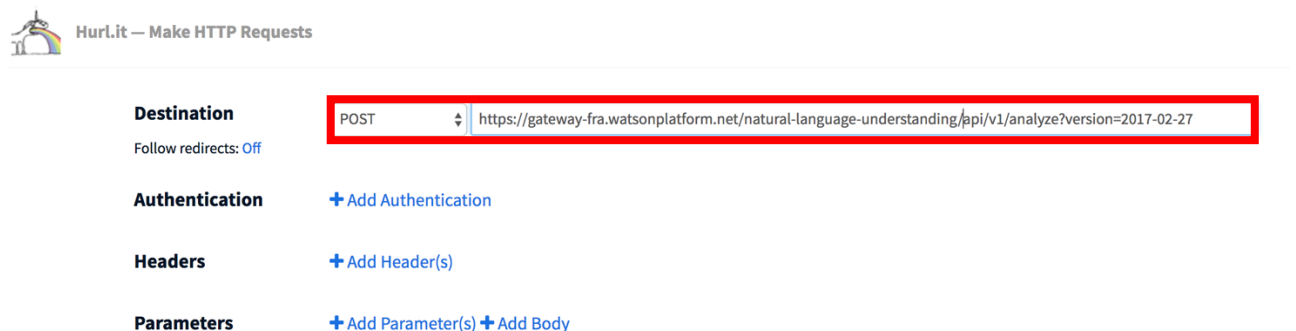
The screenshot shows the Hurl.it interface with the following configuration:

- Destination:** GET, yourapihere.com
- Follow redirects:** Off
- Authentication:** + Add Authentication
- Headers:** + Add Header(s)
- Parameters:** + Add Parameter(s)

Below the configuration fields is a reCAPTCHA widget with the text "I'm not a robot" and a "Launch Request" button. A small note at the bottom states: "Sorry about the reCAPTCHA! People were using the site for bad things. We're working to improve things so you don't have to check it every time. Use [this free tool](#) if you need to make a lot of requests."

Vamos a definir la petición HTTP para invocar a **Watson NLU**, así que vamos a elegir como método en este caso POST. Para la URL, vamos a utilizar la que hemos recogido en la credenciales del servicio (por ejemplo: <https://gateway-fra.watsonplatform.net/natural-language-understanding/api>) y vamos a añadirle el path `/v1/analyze?version=2017-02-27` como se muestra en la imagen:

**\*\*¡CUIDADO!**, la URL del servicio puede cambiar dependiendo de la región donde hayamos desplegado el servicio\*\*



The screenshot shows the Hurl.it interface with the following configuration:

- Destination:** POST, <https://gateway-fra.watsonplatform.net/natural-language-understanding/api/v1/analyze?version=2017-02-27>
- Follow redirects:** Off
- Authentication:** + Add Authentication
- Headers:** + Add Header(s)
- Parameters:** + Add Parameter(s) + Add Body

Añadimos la cabecer de autenticación básica para que la petición pueda ser autenticada correctamente por el servicio y nos de acceso seguro a la información. Hacemos click en **+Add Authentication**, elegimos Basic, y añadimos como usuario y contraseña las credenciales del servicio que nos habíamos guardado anteriormente:



**Destination** POST   
Follow redirects: Off

**Authentication** **Basic** OAuth 2 OAuth 1.0a Digest  
[remove](#)

[eye icon](#)

**Headers** [+ Add Header\(s\)](#)

**Parameters** [+ Add Parameter\(s\)](#) [+ Add Body](#)

Añadimos una cabecera, donde le vamos a indicar al servicio que el contenido que le enviamos está en formato JSON. Para ello, hacemos click en **+Add Header(s)** y añadimos como nombre **Content-Type** y como valor **application/json** como se muestra en la imagen:



**Destination** POST   
Follow redirects: Off

**Authentication** **Basic** OAuth 2 OAuth 1.0a Digest  
[remove](#)

[eye icon](#)

**Headers** **Content-Type** application/json [X](#)  
[remove all](#) [+ Add another header](#)

**Parameters** [+ Add Parameter\(s\)](#) [+ Add Body](#)

Ya tenemos el destino de nuestra petición y las credenciales necesarias para acceder de forma segura a nuestro servicio. Ahora vamos a definir el cuerpo de la petición, donde le vamos a indicar a Watson NLU que campos queremos obtener y qué texto queremos analizar (en este caso la página de noticias de Antena 3).

Para ello hacemos click en **+Add Body**, y añadimos el siguiente contenido en formato JSON:

```
{  
  "url": "http://www.antena3.com/noticias/",  
  "language": "es",  
  "features": {  
    "entities": {  
      "emotion": true,  
      "sentiment": true  
    }  
  },  
}
```



```
"keywords": {  
  
  "emotion": true,  
  
  "sentiment": true  
  
},  
  
"relations": {},  
  
"concepts": {},  
  
"semantic_roles": {  
  
  "entities": true  
  
},  
  
"categories": {},  
  
"emotion": {  
  
  "document": true  
  
},  
  
"metadata": {}  
  
}  
  
}
```

Analizamos en detalle la petición. Lo primero que incluimos es la URL del texto que queremos analizar, en este caso, la página web de Antena 3 noticias (<http://www.antena3.com/noticias/>) . También podríamos enviarle un documento HTML o un texto.

La siguiente línea, nos permite definir el idioma en el que queremos realizar el procesamiento del lenguaje natural, en nuestro caso el español.

Una vez definido el idioma y el texto a analizar, le indicamos las características que queremos analizar en el texto. Se describen a continuación las distintas posibilidades:

**Entidades:** Encuentra personas, lugares, eventos y otro tipo de entidades que se mencionan en tu contenido. Puedes ver una lista completa de tipos y subtipos de entidades [aquí](#)

#### Input

```
text: "IBM is an American multinational technology company headquartered in  
Armonk, New York, United States, with operations in over 170 countries."
```

#### Response

```
IBM: Company  
Armonk: Location
```

New York: Location  
United States: Location

**Keywords:** Busca palabras clave en el texto.

**Input**

url: "<http://www-03.ibm.com/press/us/en/pressrelease/51493.wss>"

**Response**

Australian Open  
Tennis Australia  
IBM SlamTracker analytics

**Relations:** Reconoce cuando dos entidades están relacionadas, e identifica el tipo de relación.

**Input**

text: "The Nobel Prize in Physics 1921 was awarded to Albert Einstein."

**Response**

"awardedTo" relation between "Noble Prize in Physics" and "Albert Einstein"  
"timeOf" relation between "1921" and "awarded"

**Concepts:** Identifica conceptos a alto nivel que necesariamente no tienen que estar directamente referenciados en el texto.

**Input**

text: "Natural Language Understanding uses natural language processing to analyze text."

**Response**

Linguistics  
Natural language processing  
Natural language understanding

**Semantic Roles:** Parsea las frases en sujeto-action-objeto, e identifica entidades y palabras clave que son sujetos o objetos de la acción.

**Input**

text: "In 2011, Watson competed on Jeopardy!"

**Response**

Subject: Watson  
Action: competed  
Object: on Jeopardy

**Categories:** Categoriza tu contenido utilizando una jerarquía de cinco niveles. Puedes consultar la lista de categorías [aquí](#)

#### Input

url: "www.cnn.com"

#### Response

/news  
/art and entertainment  
/movies and tv/television  
/news  
/international news

**Emotion:** Analiza la emoción de algunas frase o de la totalidad del documento. También, como se muestra en el JSON, se puede analizar para las entidades y palabras clave.

#### Input

text: "I love apples, but I hate oranges."  
targets: "apples", and "oranges"

#### Response

"apples": joy  
"oranges": anger

**Metadata:** Para páginas HTML, obtén el nombre del autor de la página web, el título de la página y la fecha de publicación.

#### Input

url: "<https://www.ibm.com/blogs/think/2017/01/cognitive-grid/>"

#### Response

Author: Stephen Callahan  
Title: Girding the Grid with Cognitive Computing - THINK Blog  
Publication date: January 31, 2017

Una vez definida nuestra consulta, seleccionamos **I'm not a robot** en el captcha y lanzamos la petición haciendo click en **Launch Request**.




Hurl.it — Make HTTP Requests

**Destination** POST   
Follow redirects: Off

**Authentication** Basic OAuth 2 OAuth 1.0a Digest  
remove

**Headers** Content-Type application/JSON  
remove all + Add another header

**Parameters**

☐ I'm not a robot  reCAPTCHA Privacy - Terms

**Launch Request**

Y el resultado aparecerá en la parte inferior de la pantalla en formato JSON, donde podemos ver todas las características semánticas que ha analizado el servicio de Watson NLU a partir de nuestro contenido. Tomaros vuestro tiempo para analizar el resultado e incluso cambiar los parámetros de la petición como la URL, para probar a analizar otros contenidos.

En la siguiente imagen se muestra un ejemplo de la respuesta:

```
{
  "usage": {
    "text_units": 1,
    "text_characters": 1969,
    "features": 7
  },
  "semantic_roles": [
    {
      "subject": {
        "text": "Prácticamente toda España"
      },
      "sentence": "Prácticamente toda España amanece este miércoles en aviso por lluvia, nieve o viento.",
      "object": {
        "text": "en aviso por lluvia, nieve o viento",
        "entities": [
          {
            "type": "Location",
            "text": "España"
          }
        ]
      },
      "action": {
        "text": "amanece"
      }
    },
    {
      "subject": {
        "text": "El temporal"
      },
      "sentence": "El temporal está siendo especialmente intenso en el tercio norte.",
      "object": {
        "text": "especialmente intenso en el tercio norte"
      },
      "action": {
        "text": "está siendo"
      }
    }
  ]
}
```

*¡Enhorabuena! Has completado la segunda parte del laboratorio. Ahora sabes invocar el servicio de Watson NLU y procesar textos en lenguaje natural.*

Copyright IBM Corporation 2013-2017. All rights reserved.

### 3- Analizamos recetas con Watson NLU

Como paso final, y para ayudarnos a entender los próximos laboratorios, vamos a analizar una receta pública en internet (Karlos Arguiñano) para ver cuál es el análisis de Watson NLU.

Para poder realizar el análisis, simplemente vamos al cambiar el cuerpo de la petición por el siguiente (el resto de campos los mantenemos) :

```
{  
  
  "url": "https://www.hogarmania.com/cocina/recetas/ensaladas-verduras/201802/ensalada-pasta-aguacate-rucula-39340.html",  
  
  "language": "es",  
  
  "features": {  
  
    "entities": {},  
  
    "keywords": { },  
  
    "relations": {},  
  
    "concepts": {},  
  
    "semantic_roles": {  
  
      "entities": true  
  
    },  
  
    "categories": {},  
  
    "emotion": {  
  
      "document": true  
  
    },  
  
    "metadata": {}  
  
  }  
}
```

Y volvemos a seleccionar **I'm not a robot** en el captcha y lanzamos la petición haciendo click en **Launch Request**.

Analizamos el resultado:

```
},
"retrieved_url": "https://www.hogarmania.com/cocina/recetas/ensaladas-verduras/201802/ensalada-pasta-aguacate-rucula-39340.html",
"relations": [],
"metadata": {
  "title": "Receta de Ensalada de pasta, aguacate y rúcula - Karlos Arguiñano",
  "publication_date": "",
  "image": "https://www.hogarmania.com/archivos/201802/6128-2-ensalada-de-pasta-aguacate-y-tomate-xl-668x400x80xX.jpg",
  "feeds": [],
  "authors": [
    {
      "name": "Karlos Arguiñano"
    }
  ]
},
"language": "es",
"keywords": [],
"entities": [],
"concepts": [
  {
    "text": "Ensalada",
    "relevance": 0.966978,
    "dbpedia_resource": "http://es.dbpedia.org/resource/Ensalada"
  },
  {
    "text": "Vinagre",
    "relevance": 0.867372,
    "dbpedia_resource": "http://es.dbpedia.org/resource/Vinagre"
  }
]
```

Vemos que no es capaz de identificar relaciones ni entidades en este tipo de texto. Pero, por ejemplo, si es capaz de obtener conceptos, metadatos o roles semánticos.

A continuación, a modo demo vamos a ver como gracias a Knowledge Studio, vamos a ser capaces de crear nuestros propios anotadores para detectar entidades y relaciones y generar nuestro propio modelo de machine learning para analizar recetas culinarias.

*¡Enhorabuena! Has completado el paso 3 del laboratorio y ya eres todo un experto en Watson Natural Language Understanding.*

