



# Bluemix Bootcamp

Luis Gerardo Baeza Castro  
IBM Digital Business Group  
[lgbaeza@mx1.ibm.com](mailto:lgbaeza@mx1.ibm.com)

## Introducción

Bluemix es la plataforma innovadora de cómputo en la nube de IBM, en la cual se combina infraestructura como servicio (IaaS), Plataforma como Servicio (PaaS) y Software como Servicio (SaaS) permitiendo a las organizaciones construir aplicaciones de negocio rápidamente. Los más de 200 servicios de Bluemix, permiten cubrir todas las fases del ciclo de vida de una aplicación, incluyendo: construcción, ejecución, despliegue y administración.

## Objetivo

El objetivo de este workshop consiste en brindar a los participantes las habilidades para utilizar, los distintos servicios existentes en Bluemix de tal manera que puedan construir aplicaciones innovadoras rápidamente. Después de tomar este taller, el participante contará con un panorama general de los servicios de Bluemix, y habrá desarrollado algunas aplicaciones utilizando la plataforma de innovación digital, Bluemix.

# Agenda

---

## **1. Fundamentos de Bluemix**

1. Estándares abiertos: Cloud Foundry, Docker y Open Stack
2. Organizaciones, regiones y espacios en Bluemix

## **2. Un mundo conectado por APIs**

1. Consulta del clima
2. Resumen de Wikipedia
3. Envío de SMS

## **3. Un asistente cognitivo para cada industria**

1. Asistente de viaje
2. Telecomunicaciones
3. Educación e inclusión social
4. Integración en una aplicación web

## **4. Siguiendo pasos**

1. Enlaces útiles

# Módulo 1: Fundamentos de Bluemix

Duración: 30 min

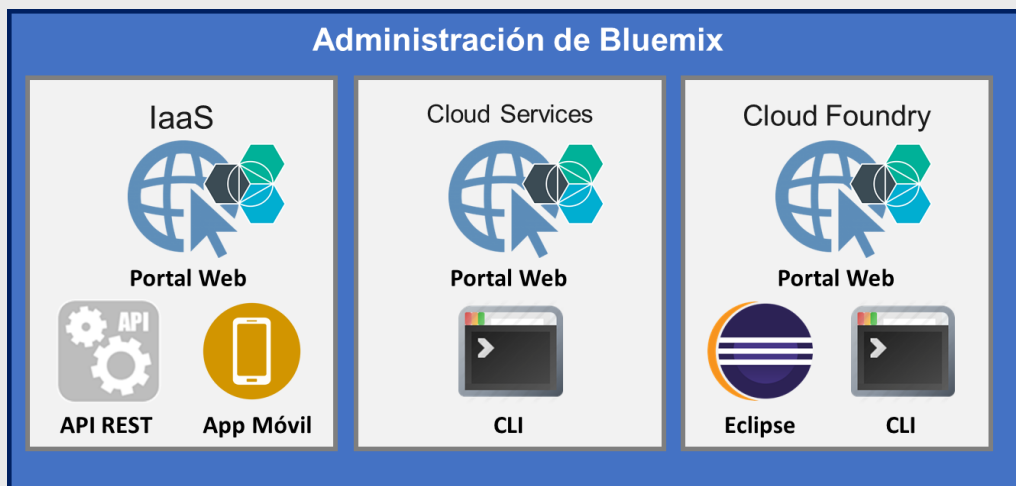
Bluemix es la plataforma innovadora de cómputo en la nube de IBM que permite a las organizaciones construir aplicaciones de negocio rápidamente. Bluemix contiene más de 120 servicios en los cuales combina:

- Infraestructura como servicio (IaaS)
- Plataforma como Servicio (PaaS)
- Software como Servicio (SaaS)

El catálogo de servicios que pueden ser integrado para cubrir todas las fases del ciclo de vida de una aplicación, incluyendo: construcción, ejecución y administración. Los servicios proveen una interfaz REST para interactuar con el mismo. Algunos proveen otras interfaces como ODBC o SDK (varía en función del servicio).

Con Bluemix ya no es necesario hacer grandes inversiones en hardware para probar o desplegar una nueva aplicación, en lugar de eso, IBM se encarga de la administración y su empresa solamente para por lo que utilice. Gracias a eso, tanto grandes empresas globales como pequeñas startups pueden aprovechar el poder de Bluemix para llevar la innovación en su negocio al siguiente nivel.

El esquema de autoservicio de Bluemix permite administrar infraestructura y servicios a través del portal web, CLI y app móvil. Además, puede obtener asistencia del equipo de soporte 24x7 disponible a través de ticket, chat, y teléfono.



## Watson Developer Cloud

Agregue características de cómputo cognitivo a sus aplicaciones usando IBM Watson Speech, Conversation, Lenguaje, entre otras más de las 30 disponibles. Extienda sus aplicaciones con capacidades cognitivas tal como lo vio en el [famoso concurso de Jeopardy de Watson](#).

Más adelante exploraremos el servicio de *Watson Conversation*.

## IBM Líder en estrategia de Hybrid cloud

Debido a que Hybrid cloud se ha posicionado como la estrategia de nube que ha probado ser el pilar de la innovación y crecimiento de los negocios (cerca de la mitad de las empresas utilizan algún esquema de Hybrid Cloud) en Bluemix contamos con servicios que permiten comunicar servicios y aplicaciones en la nube, con los sistemas de registros empresariales, así como cualquier ambiente externo a la organización de forma segura y confiable.

Más adelante exploraremos los servicios de *Secure Gateway* y *API Connect*.

## Seguridad

IBM® Bluemix® provee una plataforma en la nube segura en la que se puede confiar. Bluemix está diseñado con prácticas líderes en la industria, utilizando controles de seguridad a través de las diferentes capas como infraestructura y redes. Algunos ejemplos incluyen:

- ISO/IEC 27017:2015 Controles de seguridad de la información aplicables al aprovisionamiento de servicios en la nube
- ISO 27018:2014 Controles para la implementación de medidas de protección de información de identificación personal (PII)
- ISO/IEC 27001
- Payment Card Industry Data Security Standards (PCI DSS),

IBM cuenta con un equipo global de expertos (PSIRT) dedicado a gestionar la recepción, investigación y seguimiento interno de información de vulnerabilidades de seguridad referentes a productos de IBM.

Conozca más de Bluemix en la documentación: <https://console.ng.bluemix.net/docs/>

### 1.1 Estándares abiertos: Cloud Foundry, Docker y Open Stack

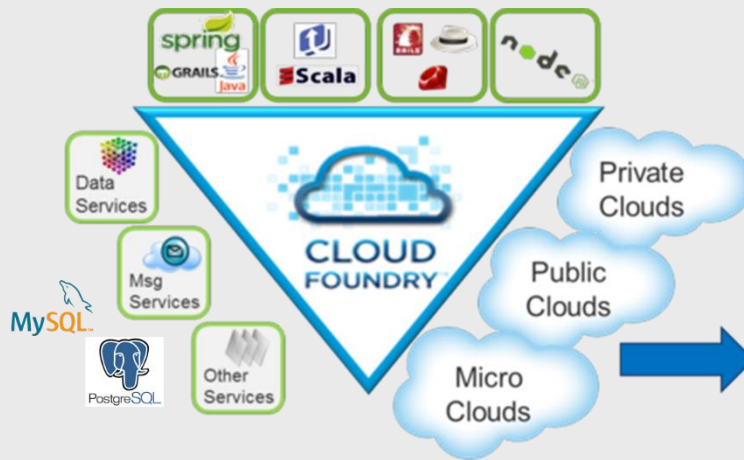
Los proyectos open source son impulsados por intereses creativos, independientemente de los intereses comerciales. Esta es la razón por la cual Bluemix está basada en estándares abiertos, ya que el principal motivador es brindar con herramientas de innovación a nuestros clientes.

Bluemix refleja los resultados prácticos de proyectos open source líderes.



## Cloud Foundry

Bluemix es la implementación más grande de Cloud Foundry, una plataforma abierta para aplicaciones en la nube. Cloud Foundry abstrae la necesidad inherente a ejecutar una aplicación en la nube y al ser abierto, no obliga a las empresas a utilizar software o infraestructura propietaria.



## Docker

Docker es la plataforma de “contenedores” más popular. Los contenedores son objetos de software que incluyen todos los elementos que una aplicación necesita para su ejecución. Un contenedor provee aislamiento y distribución de recursos, sin embargo, es más portable y eficiente que una máquina virtual.

Gracias a que Bluemix está basado en Docker, es posible hostear contenedores existentes on-premise sin requerir modificaciones en las aplicaciones.

## Open Stack

Tecnología abierta detrás de los servicios de IaaS disponibles en Bluemix. Provee aprovisionamiento y gestión de las máquinas virtuales, servicios de almacenamiento y redes.

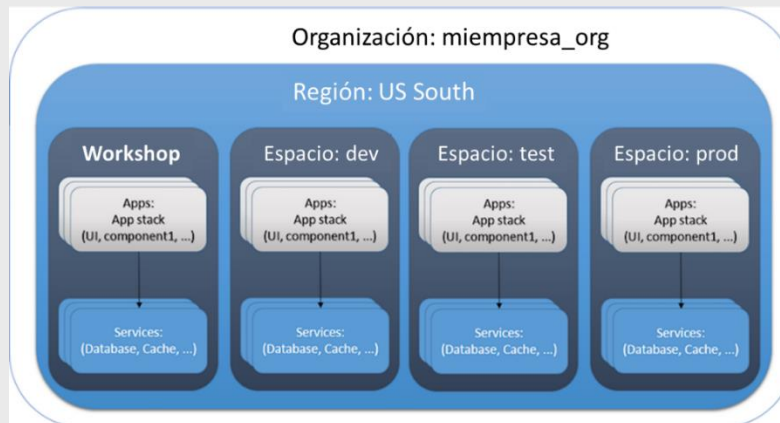
## OpenWhisk

Tecnología de ejecución de código bajo demanda (orientado a eventos) en un ambiente altamente escalable sin requerimientos de infraestructura para los clientes. Acelera el desarrollo de acciones pequeñas, distintas e independientes.

Debido a sus características, OpenWhisk está impulsando el internet de las cosas.

## 1.2 Organizaciones, regiones y espacios en Bluemix

Bluemix se encuentra organizado en organizaciones, regiones y espacios:



### Organizaciones

Las organizaciones se extienden a través de diversas regiones y se encuentran delimitadas por algunas componentes:

- Miembros: personas con acceso a la cuenta bajo un rol determinado (Manager, Developer, Auditor).
- Dominios: Ruta en Internet utilizadas para las aplicaciones.
- Cuota: límites para la organización en términos de número de servicios y cantidad de memoria

### Regiones

Una región es un territorio geográfico definido en el cual es posible desplegar aplicaciones y servicios. Bluemix permite crear aplicaciones e instancias de servicios en diferentes regiones.

Nombre Región	Ubicación Geográfica	Prefijo	cf API endpoint	UI console
US South region	Dallas, US	ng	api.ng.bluemix.net	<a href="https://console.ng.bluemix.net">console.ng.bluemix.net</a>
United Kingdom region	London, England	eu-gb	api.eu-gb.bluemix.net	<a href="https://console.eu-gb.bluemix.net">console.eu-gb.bluemix.net</a>
Sydney region	Sydney, Australia	au-syd	api.au-syd.bluemix.net	<a href="https://console.au-syd.bluemix.net">console.au-syd.bluemix.net</a>
Germany region	Frankfurt, Germany	eu-de	eu-de.	<a href="https://console.eu-de.bluemix.net">console.eu-de.bluemix.net</a>

### Espacios

Los espacios permiten agrupar conjuntos de aplicaciones, servicios y miembros de la cuenta. Los espacios se encuentran asociados a una región específica de Bluemix.

Típicamente utilice espacios para agrupar recursos de desarrollo (dev), pruebas (test) y producción (prod).

# Módulo 2: Un mundo conectado por APIs

Duración: 90 min

Inicie este módulo creando una instancia del servicio de **The Weather Company Data** y **API Connect**

## 2.1 API para consulta del clima

- Crear una nueva API desde cero (from scratch)
- Nombrar la API **Weather**, verificar que la ruta base sea **/weather** y utilizar la ruta **/today** agregando el parámetro **city**
- En **Assemble**, agregar dos elementos **Invoke**, un **Gatewayscript** y un **Map** en el orden que se muestra:



- Colocar el siguiente URL en el primer **Invoke**, configurando la respuesta en el objeto **geocode\_response**

```
https://maps.googleapis.com/maps/api/geocode/json?address={city}
```

- Configurar el Gatewayscript con el siguiente código

```
var map = apim.getvariable('geocode_response.body');
var location = map.results[0].geometry.location;

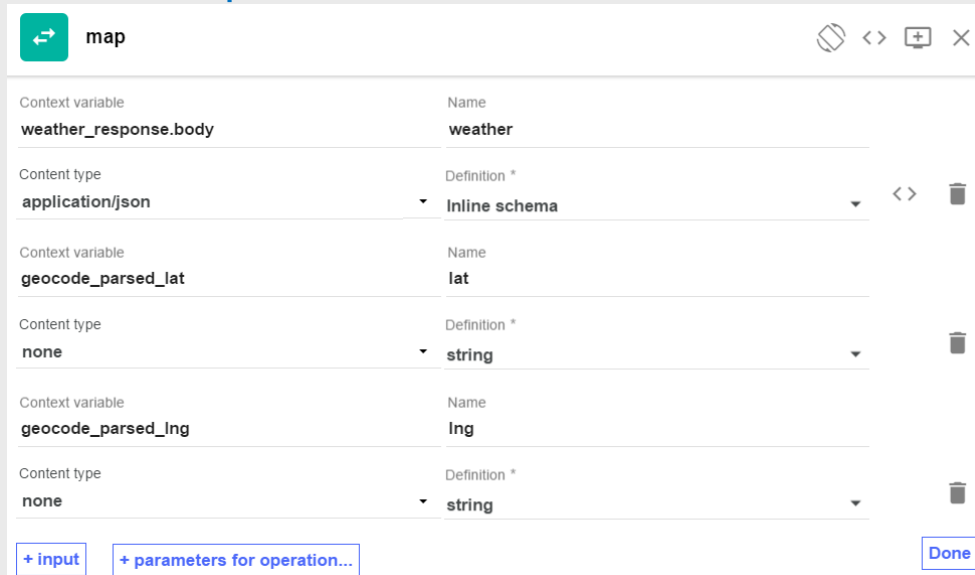
var respObj = {
  "google_maps_link": "https://www.google.com/maps?q=" + location.lat + "," + location.lng
};

apim.setvariable('geocode_parsed_lat', location.lat);
apim.setvariable('geocode_parsed_lng', location.lng);
```

- Colocar el siguiente URL en el segundo **Invoke**, configurando la respuesta en el objeto **weather\_response**. Asegúrese de configurar el **usuario** y **password** de su servicio de clima.

```
https://twcservice.mybluemix.net/api/weather/v1/geocode/${geocode_parsed_lat}/${geocode_parsed_lng}/observations.json?language=es-MX&units=m
```

- Configure las entradas del elemento **Map** cree 3 variables de entrada dando clic en [+ input](#) :
  - `weather_response.body`: tipo **inline schema** (copie el contenido del archivo **schema-weather-input.yml** proporcionado) y tipo de contenido **application/json**
  - `geocode_parsed_lat`: tipo **string**
  - `geocode_parsed_lng`: tipo **string**
- El input del elemento **Map** debería lucir tal como se muestra:

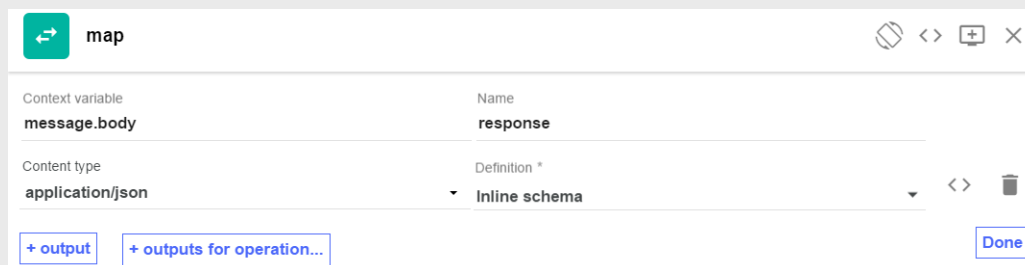


The screenshot shows the configuration for a 'map' node. It has three input variables defined:

Context variable	Name	Content type	Definition *
<code>weather_response.body</code>	<code>weather</code>	<b>application/json</b>	<b>Inline schema</b>
<code>geocode_parsed_lat</code>	<code>lat</code>	<b>none</b>	<b>string</b>
<code>geocode_parsed_lng</code>	<code>lng</code>	<b>none</b>	<b>string</b>

Buttons at the bottom: [+ input](#), [+ parameters for operation...](#), and [Done](#).

- Ahora configure la salida, para ello cree una variable dando clic en [+ outputs for operation...](#) :
  - `message.body`: tipo **inline schema** (copie el contenido del archivo **schema-weather-response.yml** proporcionado) y tipo de contenido **application/json**
- El output del elemento **Map** debería lucir tal como se muestra:



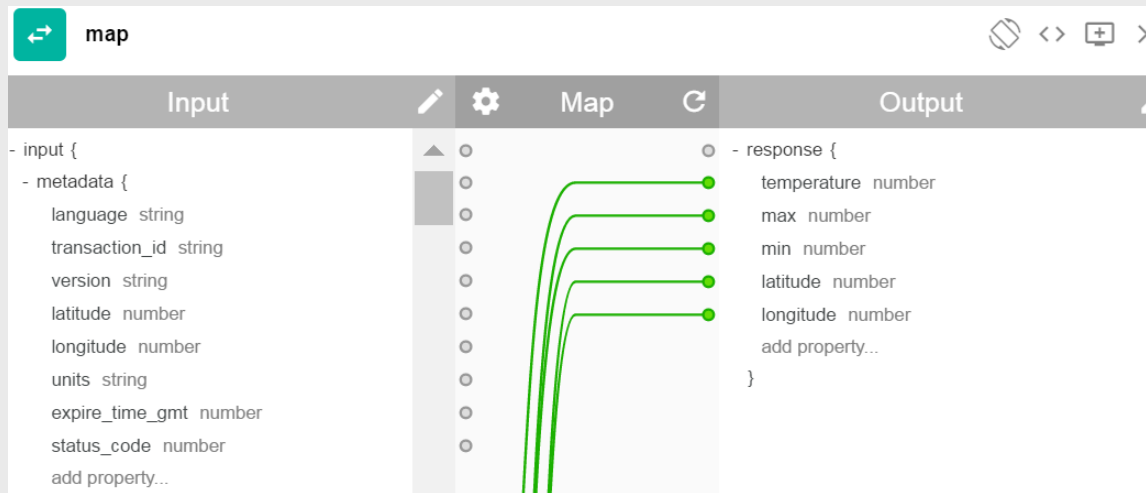
The screenshot shows the configuration for a 'map' node with one output variable defined:

Context variable	Name	Content type	Definition *
<code>message.body</code>	<code>response</code>	<b>application/json</b>	<b>Inline schema</b>

Buttons at the bottom: [+ output](#), [+ outputs for operation...](#), and [Done](#).

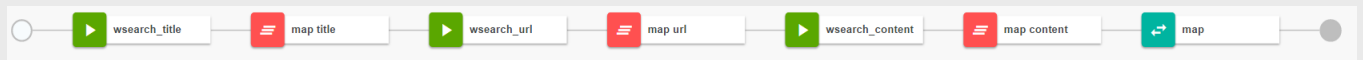
- Configure el mapeo vinculando las propiedades **temp**, **max\_temp** y **min\_temp** de la variable de entrada **weather** a sus propiedades correspondientes de la variable de salida **response**. Asimismo, vincule las variables de entrada **lat** y **lng** con las propiedades de salida **latitude** y **longitude**





## 2.2 Resumen de Wikipedia

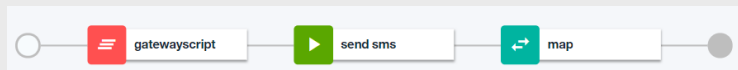
- Cree otra API, ahora seleccionando la opción de **Import an existing OpenAPI** y utilice el archivo **api-wikipedia\_1.0.0.yaml** (definición de API Swagger)
- El ensamblado de la API debería lucir tal como se muestra:



- Lanzar una prueba para verificar que la API recupera la información de Wikipedia correctamente.

## 2.3 Envío de Mensajes de Texto

- Cree otra API, seleccionando nuevamente la opción de **Import an existing OpenAPI** y utilice el archivo **api-sms\_1.0.0.yaml** (definición de API Swagger)
- El ensamblado de la API debería lucir tal como se muestra:



- Probar la API enviando un mensaje al número 5535542253
- Si desea realizar una prueba con un número diferente, comunique al facilitador para instrucciones.

Asegúrese de agregar las APIs a un producto y catálogo, así como de publicar cualquier cambio realizado.

# Módulo 3: Un asistente cognitivo para cada industria

---

Inicie este módulo creando una instancia del servicio de [Watson Conversation](#)

## 2.1 Asistente de viaje

Utilice el siguiente texto para las intenciones:

- #numero-reservacion
  - ✓ cuál es mi número de reservación?
  - ✓ número de itinerario
  - ✓ número de reservación de mi hotel
  - ✓ número de reservación de vuelo
  - ✓ numero vuelo
- #info-vuelo
  - ✓ A qué hora sale mi vuelo?
  - ✓ A qué hora tengo que estar en el aeropuerto?
  - ✓ Cuando despegue mi vuelo?
  - ✓ Cuando parte el avión
  - ✓ Cuando sale el avión
  - ✓ Cuando sale mi vuelo?
  - ✓ Dame información acerca de mi vuelo
  - ✓ En qué aerolínea vuelo?
- #direccion-hotel
  - ✓ cuál es la dirección del hotel
  - ✓ dónde encuentro mi hotel?
  - ✓ dónde está mi hotel
  - ✓ en que calle encuentro mi hotel
  - ✓ quiero ir al hotel
- #info-hospedaje
  - ✓ a qué hora es el checkin
  - ✓ a qué hora es el checkout
  - ✓ cuál es el nombre de mi hotel
  - ✓ cuál es mi hotel
  - ✓ dónde me estoy hospedando
  - ✓ dónde me voy a hospedar
  - ✓ en qué habitación me estoy quedando

Utilice el siguiente texto para las entidades:

- @vuelo
  - Ida: salida, primero, primer vuelo
  - vuelta
- @respuesta
  - Si: de acuerdo, ok
  - No: en otro momento

Utilice el siguiente texto para los nodos:

- Bienvenido a tu viaje a \$destino\_viaje, estoy seguro que va a ser le-gen-da-rio!
- Tu número de reservación es \$numero\_reservacion
- Aquí está tu información del vuelo de ida: \$vuelo\_ida
  - Que disfrutes tu vuelo de ida \$vuelo\_ida
- Aquí está tu información del vuelo de ida: \$vuelo\_vuelta
  - Que disfrutes tu vuelo de vuelta \$vuelo\_vuelta
- ¿A qué vuelo te refieres ida o vuelta?
- El hotel se encuentra ubicado en \$direccion\_hotel. ¿Quieres que te de indicaciones para llegar?

```
{
  "output": {
    "context": {
      "flag_mapa": "true"
    },
    "text": {
      "values": [
        "Sigue las indicaciones en el mapa:"
      ],
      "selection_policy": "sequential"
    }
  }
}
```

Para probar el asistente, configure las siguientes variables de contexto:

Variable	Valor	Variable	Valor
\$numero_reservacion	VPF-57623-41	\$direccion_hotel	Vasconcelos #300 Ote, Col. Residencial San Agustín
\$vuelo_ida	Aeroméxico AM0935 MEX - MTY. Salida 18:24 Terminal 2	\$vuelo_vuelta	Aeroméxico AM0935 MTY- Ex. Salida 20:35 Terminal B

## 2.2 Telecomunicaciones

Utilice el siguiente texto para las intenciones:

- #info\_plan
  - ✓ que incluye mi plan?
  - ✓ qué tipo de plan tengo
  - ✓ que incluye mi línea
  - ✓ que tipo de plan tengo contratado
  - ✓ que tengo incluido en mi plan
  - ✓ como se llama mi plan
  - ✓ cuánto cuesta mi plan al mes?
- #saldo\_corte
  - ✓ cuál es el saldo de mi cuenta
  - ✓ quiero consultar mi saldo
  - ✓ cuál fue el total de mi corte anterior
  - ✓ cuánto tengo que pagar
  - ✓ de cuánto es mi factura del mes pasado?
- #pagar\_plan
  - ✓ quiero pagar mi último estado de cuenta
  - ✓ quiero hacer el pago de mi factura
  - ✓ puedo pagar mi factura?
  - ✓ quiero pagar mi línea
  - ✓ quiero pagar mi factura

Utilice el siguiente texto para las entidades:

- @respuesta
  - no: en otro momento
  - si: correcto, así es, sí, ok

Utilice el siguiente texto para los nodos:

- Gracias por comunicarte a ATT, ¿Cómo puedo ayudarte hoy?
  - Bienvenido a ATT, cuál es tu consulta?
- Estimado \$cliente, actualmente tienes un plan \$plan\_nombre el cual incluye \$plan\_detalle
- El saldo al corte es de \$saldo. ¿Hay algo más que pueda hacer por tí?
- Vamos a procesar tu pago. Indícame tu tarjeta a 16 dígitos
  - `input.text.length()==16`
  - Ahora indícame los dos dígitos del mes de vigencia.
  - El número de tarjeta no es válido
- Indícame los dígitos correspondientes al año de vigencia
- Excelente, para terminar indícame los dígitos de seguridad.

```
{
  "context": {
    "procesar_pago": "true"
  },
  "output": {
    "text": {
      "values": [
        "Permíteme validar y procesar tus datos de pago"
      ],
      "selection_policy": "sequential"
    }
  }
}
```

- \$pago\_valido=="true"
  - Tu pago fue procesado correctamente. ¿Puedo ayudarte con algo más?
- anything\_else
  - Hubo un problema al procesar tu pago.

Para probar el asistente, configure las siguientes variables de contexto:

Variable	Valor	Variable	Valor
\$cliente	Juan Pérez Sánchez	\$plan_detalle	400 Min, 400 SMS 2000 Megabytes
\$plan_nombre	ATT con Todo 400	\$saldo	\$450
\$pago_valido	Pago_ok		

## 2.3 Educación e Inclusión Social

Para esta actividad, importa el workspace **3. Educacion.json**. A continuación, se describe el proceso para probar la invocación de la API de Conversation.

1. Ingresa en Postman la siguiente URL, reemplazando **{workspace-id}**. Asegúrate que el verbo utilizado sea POST
 

```
https://gateway.watsonplatform.net/conversation/api/v1/workspaces/{workspace-id}/message?version=2016-07-11
```
2. En autorización, selecciona **Basic Auth** y captura tu **usuario** y **contraseña** del servicio. Da clic en **Update request**
3. Como cuerpo de la petición, captura el siguiente texto:

```
{
  "input": { "text": "" }
}
```

- Analiza la respuesta, encontrarás el atributo **conversation\_id** que utilizarás en tu siguiente llamada:

```
{
  "input": {
    "text": "Hola, quiero saber qué es una tarjeta de crédito"
  },
  "context": {
    "conversation_id": "{conversation-id}",
    "system": {
      "dialog_stack": [
        "root"
      ],
      "dialog_turn_counter": 1
    }
  }
}
```

Vuelve a enviar la petición, y a partir de esta segunda llamada al servicio, utilizarás el mismo texto del punto número 4, modificando los valores de “**text**”, “**dialog\_stack**” y “**dialog\_turn\_counter**”. El texto que utilizarás en las siguientes llamadas es:

- Mi proveedor de telefonía no está cumpliendo con el contrato y me está cobrando más
- Si

## 2.4 Integración en una aplicación web

1. Abrir la aplicación **Educacion** el editor de NodeJS y localizar el archivo app.js
2. Reemplace en la línea 53 **{conversation-workspace-id}** por su Id de Workspace de **Watson Conversation**.
3. Reemplace en la línea 54 **{conversation-username}** y **{conversation-password}** por sus credenciales de **Watson Conversation**. En este fragmento de código se instancia el objeto del SDK de WDC.
4. El código localizado en la línea 80 realiza la **invocación del servicio**.
5. La validación localizada en la línea 88 permite identificar la **existencia de variables** de contexto en la respuesta de **Watson Conversation**.

# Siguientes Pasos

## Enlaces útiles

- Explore a fondo los detalles de la plataforma en la documentación <https://console.ng.bluemix.net/docs/>
- Watson API Explorer: Agilice la integración de los servicios cognitivos en su aplicación <https://watson-api-explorer.mybluemix.net/>
- Ejemplos ilustrativos que le ayudarán a generar ideas de como utilizar Bluemix <https://ibm-bluemix.github.io/>
- Manténgase al día con las nuevas características y servicios en Bluemix <https://console.ng.bluemix.net/docs/whatsnew/index.html>
- Casos de uso: empresas del mundo real que aprovechan Bluemix para hacer negocios <https://www.ibm.com/cloud-computing/bluemix/case-studies>