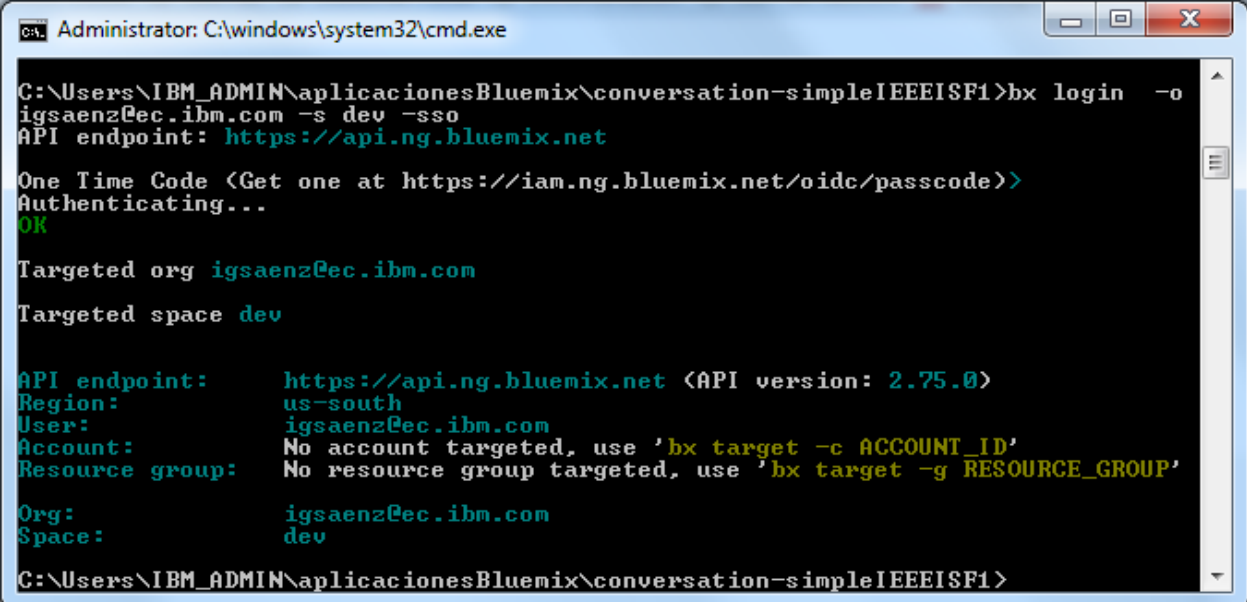


GUIA DE LABORATORIO CREACIÓN DE SERVICIOS WATSON CONVERSATION, CLOUDANT DB Y VISUAL RECOGNITION

1. Iniciamos sesión en bluemix con esta serie de comandos:

- bx api <https://api.ng.bluemix.net> (para apuntar a la region donde está creada la cuenta)
- bx login -u igsaenz@ec.ibm.com -o igsaenz@ec.ibm.com -s dev (para definir la organización y el espacio sobre el que se va a trabajar), si el procedimiento no funciona por tener una cuenta federada se debe utilizar el comando:

bx login -o igsaenz@ec.ibm.com -s dev -sso



```
C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simpleIEEEISF1>bx login -o
igsaenz@ec.ibm.com -s dev -sso
API endpoint: https://api.ng.bluemix.net

One Time Code <Get one at https://iam.ng.bluemix.net/oidc/passcode>>
Authenticating...
OK

Targeted org igsaenz@ec.ibm.com
Targeted space dev

API endpoint: https://api.ng.bluemix.net <API version: 2.75.0>
Region: us-south
User: igsaenz@ec.ibm.com
Account: No account targeted, use 'bx target -c ACCOUNT_ID'
Resource group: No resource group targeted, use 'bx target -g RESOURCE_GROUP'

Org: igsaenz@ec.ibm.com
Space: dev

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simpleIEEEISF1>
```

2. Creación servicio de Base de datos:

- Ejecutar el comando: bx service create cloudantNoSQLDB Lite servicioBDD

```
Administrator: C:\windows\system32\cmd.exe

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx service create cloudantNoSQLDB Lite servicioBDD
Invoking 'cf create-service cloudantNoSQLDB Lite servicioBDD '...

Creating service instance servicioBDD in org igsaenz@ec.ibm.com / space dev as igsaenz@ec.ibm.com...
OK

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>
```

- b. Creamos las credenciales del servicio con el comando: `bx cf create-service-key servicioBDD credsservicioBDD`

```
Administrator: C:\windows\system32\cmd.exe

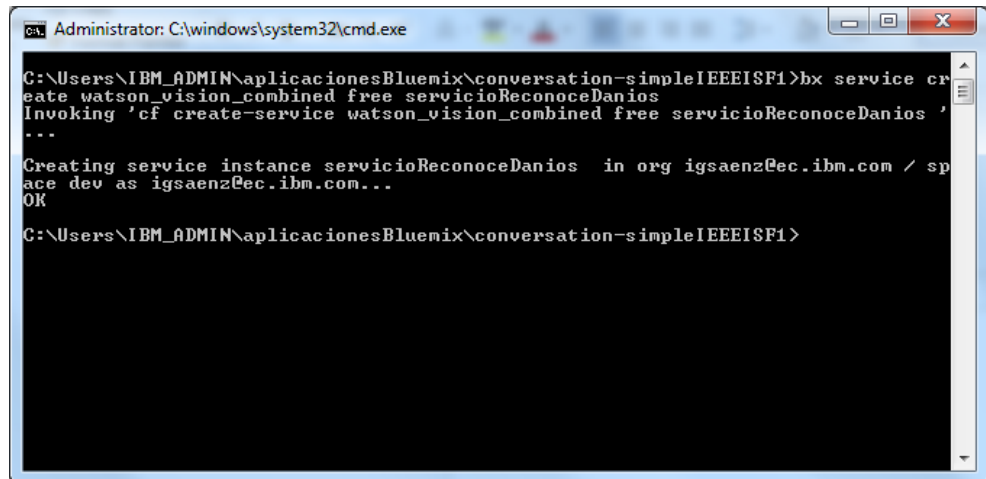
C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx cf create-service-key servicioBDD credsservicioBDD
Invoking 'cf create-service-key servicioBDD credsservicioBDD '...

Creating service key credsservicioBDD for service instance servicioBDD as igsaenz@ec.ibm.com...
OK

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>
```

3. Creación del servicio de Watson Visual Recognition.

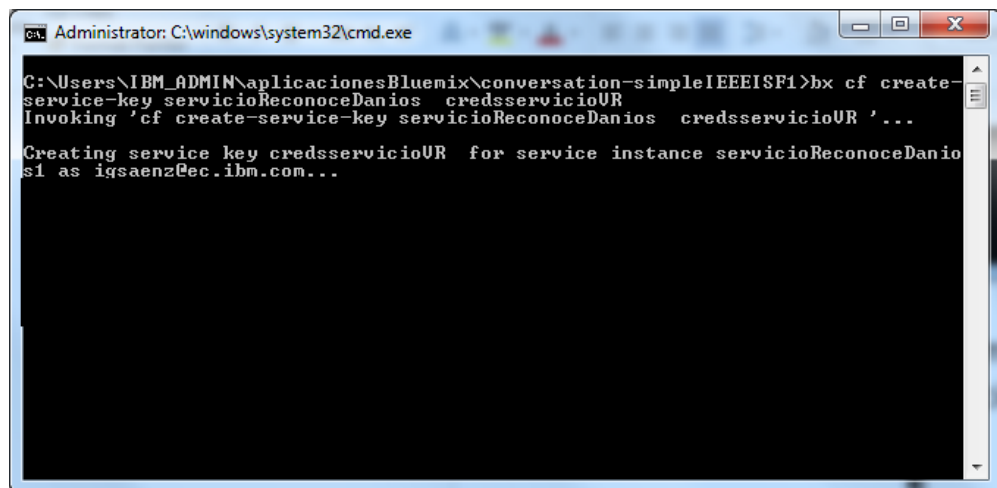
- c. Ejecutar el comando: `bx service create watson_vision_combined servicioReconoceDanios`



```
Administrator: C:\windows\system32\cmd.exe

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx service create watson_vision_combined free servicioReconoceDanios
Invoking 'cf create-service watson_vision_combined free servicioReconoceDanios '...
Creating service instance servicioReconoceDanios in org igsaenz@ec.ibm.com / space dev as igsaenz@ec.ibm.com...
OK
C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>
```

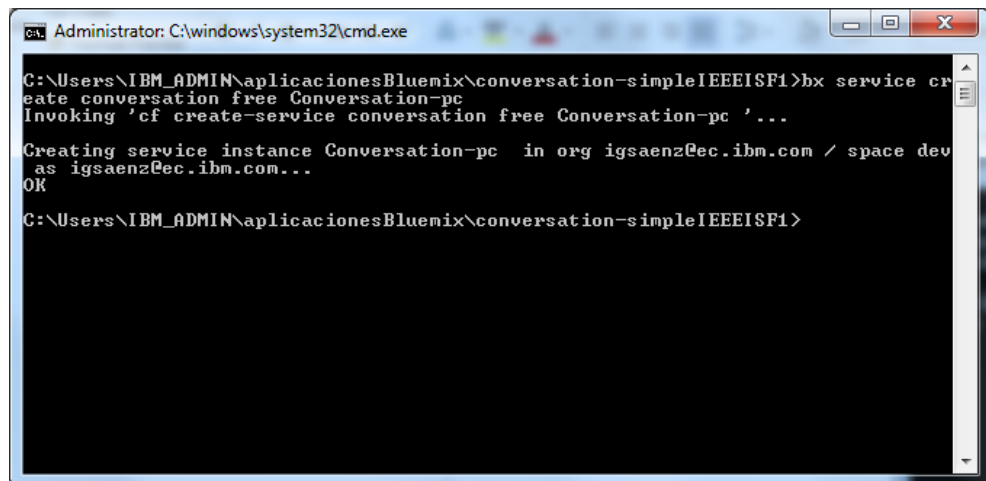
- d. Creamos las credenciales del servicio con el comando: `bx cf create-service-key servicioReconoceDanios credsservicioVR`



```
Administrator: C:\windows\system32\cmd.exe

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx cf create-service-key servicioReconoceDanios credsservicioVR
Invoking 'cf create-service-key servicioReconoceDanios credsservicioVR '...
Creating service key credsservicioVR for service instance servicioReconoceDanios1 as igsaenz@ec.ibm.com...
```

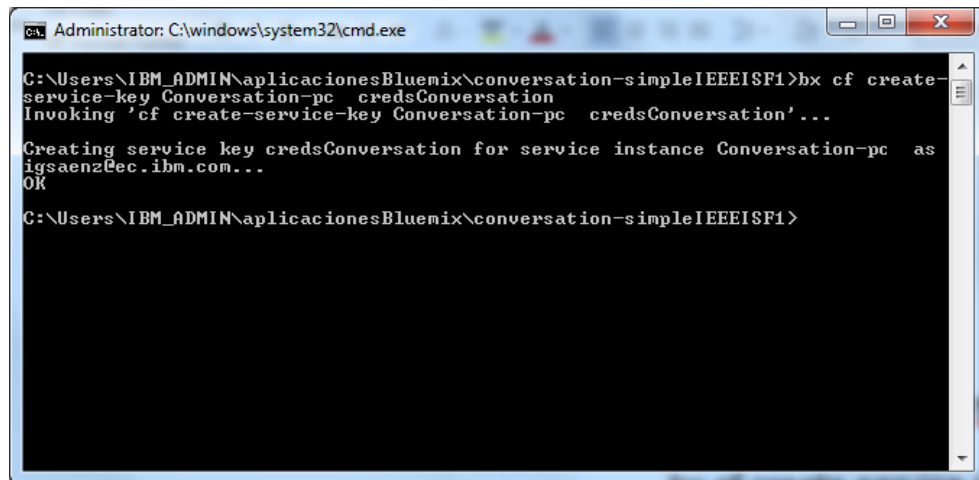
4. Creación del servicio Watson Conversation.
- e. Ejecutamos el comando: `bx service create conversation free Conversation-pc`



```
Administrator: C:\windows\system32\cmd.exe

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx service create conversation free Conversation-pc
Invoking 'cf create-service conversation free Conversation-pc '...
Creating service instance Conversation-pc in org igsaenz@ec.ibm.com / space dev as igsaenz@ec.ibm.com...
OK
C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>
```

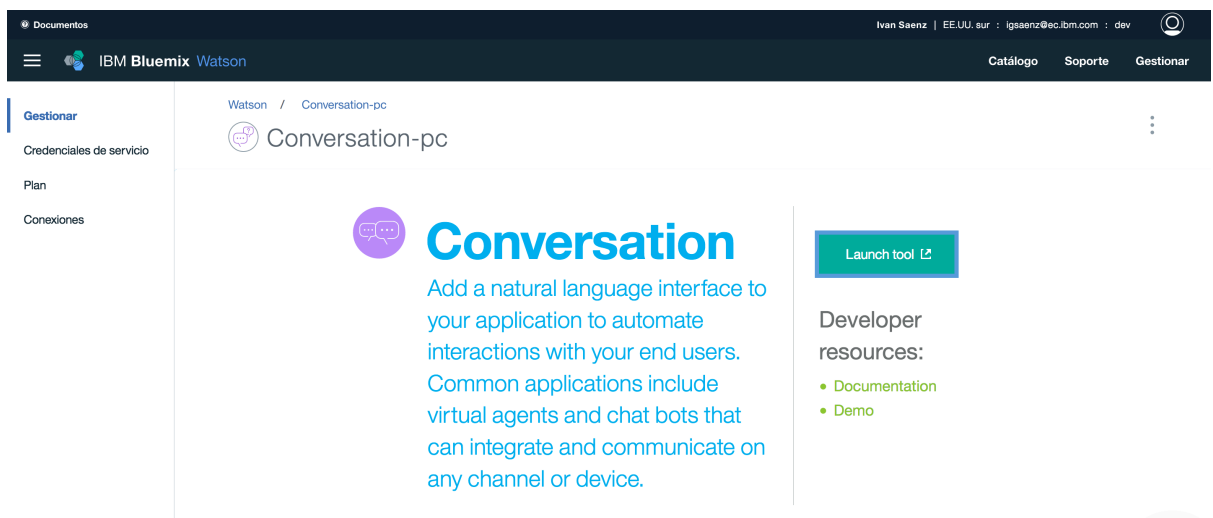
- f. Creamos las credenciales del servicio con el comando: `bx cf create-service-key Conversation-pc credsConversation`



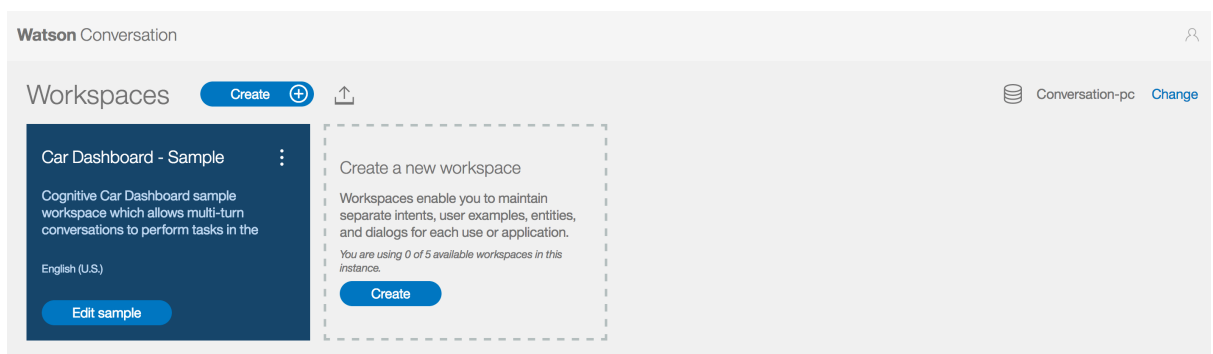
```
Administrator: C:\windows\system32\cmd.exe

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx cf create-
service-key Conversation-pc credsConversation
Invoking 'cf create-service-key Conversation-pc credsConversation'...
Creating service key credsConversation for service instance Conversation-pc as
igsaenz@ec.ibm.com...
OK
C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>
```

5. Ingresamos al panel de control (dashboard) y en el listado de servicios buscamos el servicio Conversation-pc que creamos.
6. El servicio se nos muestra instrucciones para configurar la primera aplicación que lo utilice.



7. Hacemos click en Launch tool.



8. Creamos un Workspace, un workspace es un contenedor de los artefactos que definen el flujo de conversación para una aplicación. Hacemos click en Create. Y damos un nombre al workspace, una descripción, dejamos el lenguaje como Español y hacemos clic en "Create".

Create a workspace

Workspaces enable you to maintain separate intents, user examples, entities, and dialogs for each use or application.

Name

Chat Seguro Vehicular

Description

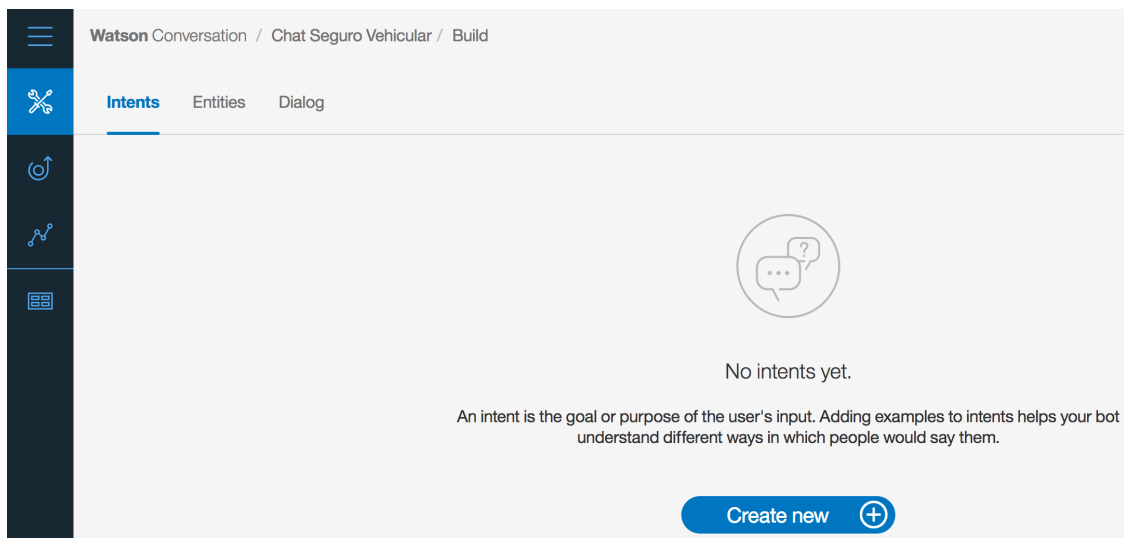
Guarda las definiciones de las interacciones del Chat Seguro Vehicular.

Language

Spanish

Create

9. Se abre la sección de Intents, un intent es la intención del input del usuario. Añadir ejemplos de intenciones ayuda al bot a entender diferentes formas en que las personas los podrían decir.



10. Creamos las intenciones necesarias para el saludo, el reporte de un incidente y la despedida. Por cada intención debemos ingresar ejemplos de frases y/o palabras que un usuario escribiría en cada intención.

Intent name

#saludo

User example

ayudame



hola



buenos dias



buenas tardes



buenas noches



buenas



que tal



11. Al finalizar se muestra el listado de intents.

Intents			Entities		Dialog	
Create new (+)			Import	Export	Delete	
2 intents			Sort by: Newest		▼	
>	#despedida	chao			5	
>	#saludo	buenos dias			6	

12. El siguiente paso es construir un diálogo. Un diálogo define el flujo de la conversación en la forma de un árbol lógico. Cada nodo tiene una condición que lo dispara basada en lo ingresado por el usuario. Para hacerlo debemos ingresar a la sección Dialogo:

Watson Conversation / Chat Seguro Vehicular / Build

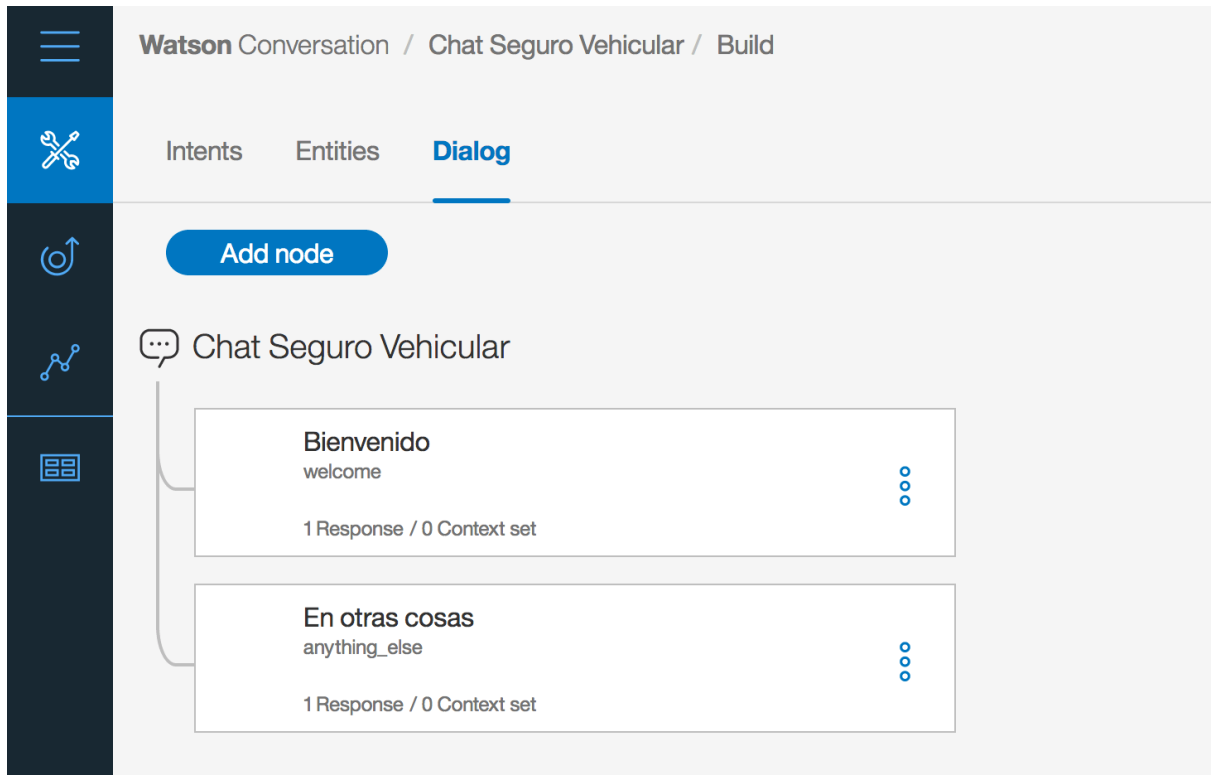
Intents Entities **Dialog**

No dialog yet

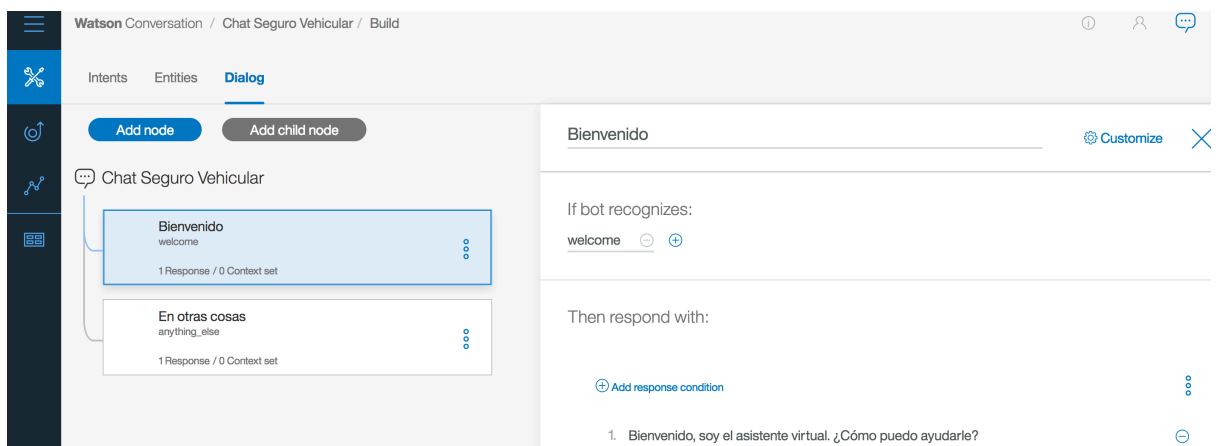
A dialog uses intents, entities, and context from your application to define a response to each user's input. Creating a dialog defines how your bot will respond to what your users are saying.


Create (+)

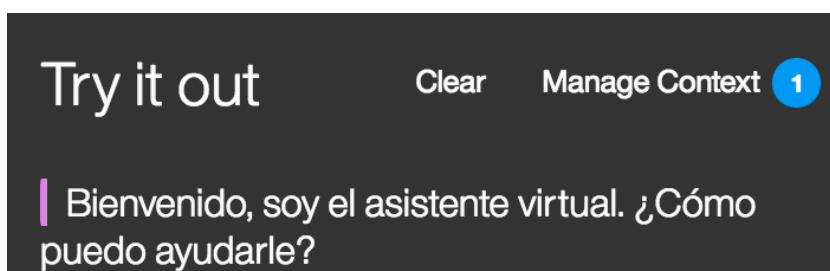
13. Clic en Crear. Se muestran dos nodos, el llamado “Bienvenido” contiene un saludo que será desplegado a los usuarios cuando ingresan a la herramienta; el llamado “En otras cosas” contiene frases que son usadas para responder a los usuarios cuando el texto introducido no se reconozca...




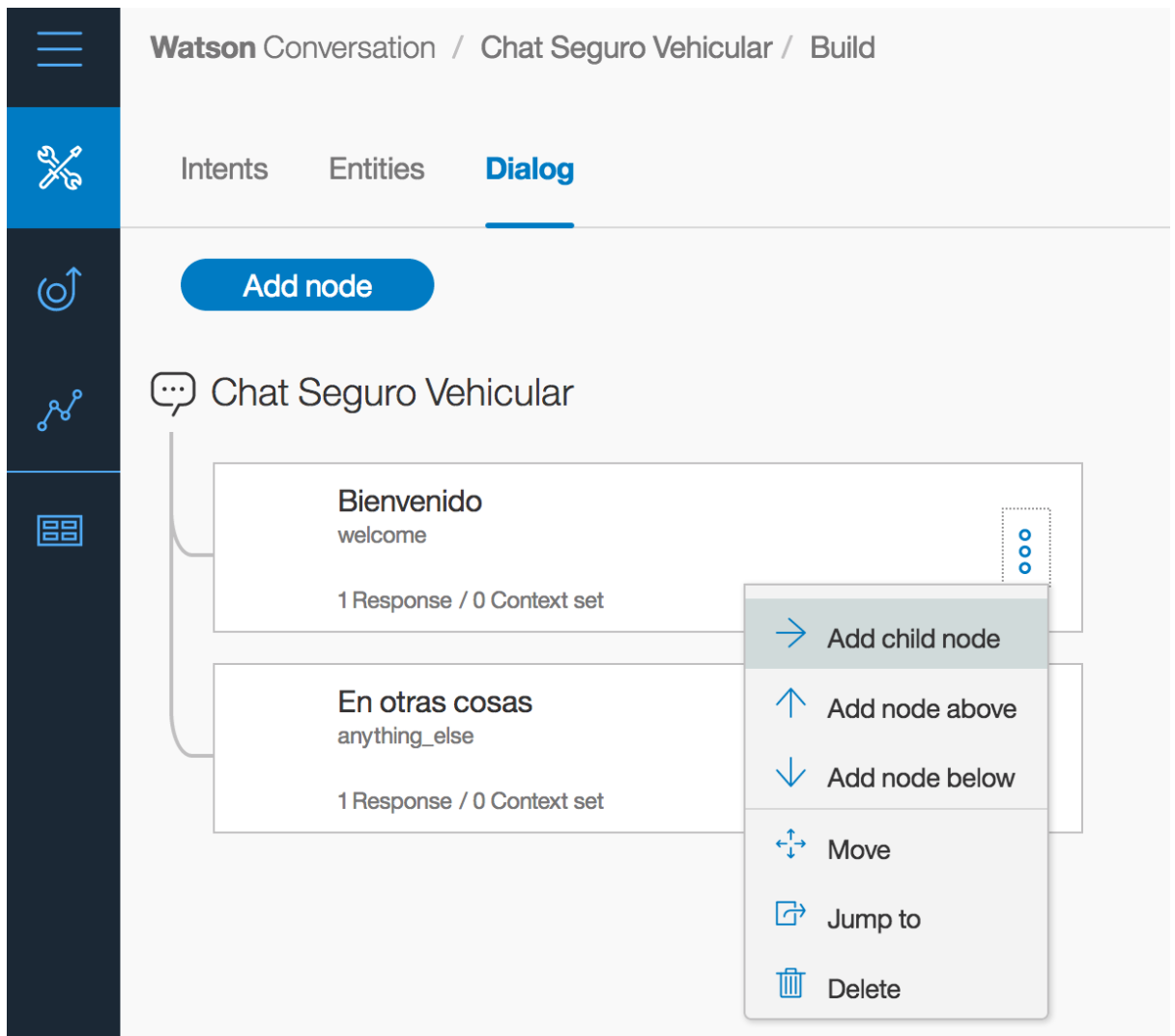
14. Hacemos clic sobre el nodo Bienvenido y reemplazamos la respuesta por defecto con el texto de bienvenida deseado.



15. Hacemos clic en el ícono  para probar el mensaje inicial. Se desplegará nuestro texto modificado.



16. Cerramos el diálogo y añadimos otro nodo para manejar las intenciones entre el nodo “Bienvenido” y el nodo “En otras cosas”. Para hacerlo clic sobre  y seleccionamos “Add node below”



Watson Conversation / Chat Seguro Vehicular / Build

Intents Entities **Dialog**

Add node

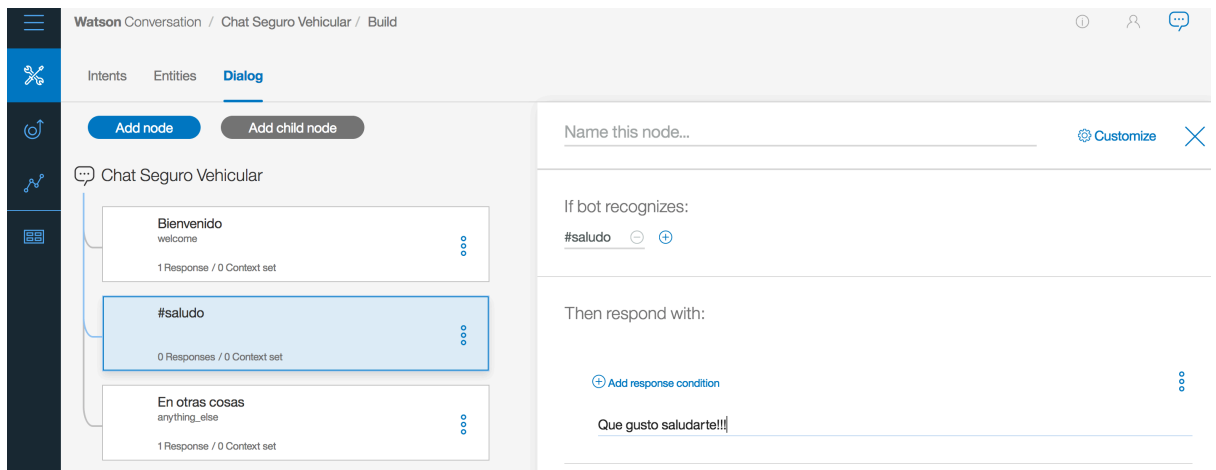
Chat Seguro Vehicular

Bienvenido
welcome
1 Response / 0 Context set

En otras cosas
anything_else
1 Response / 0 Context set

- Add child node
- ↑ Add node above
- ↓ Add node below
- ↔ Move
- 📄 Jump to
- 🗑 Delete

17. Utilizamos la intención saludo y proveemos una respuesta.



Watson Conversation / Chat Seguro Vehicular / Build

Intents Entities **Dialog**

Add node **Add child node**

Chat Seguro Vehicular

Bienvenido
welcome
1 Response / 0 Context set

#saludo
0 Responses / 0 Context set

En otras cosas
anything_else
1 Response / 0 Context set

Name this node...

If bot recognizes:
#saludo

Then respond with:
Que gusto saludarte!!!

18. Creamos un nodo debajo de este para la intención de despedida y lo poblamos con una respuesta de despedida.

The screenshot shows the 'Dialog' editor in the Watson Conversation interface. The breadcrumb path at the top is 'Watson Conversation / Chat Seguro Vehicular / Build'. Below this, there are tabs for 'Intents', 'Entities', and 'Dialog', with 'Dialog' being the active tab. On the left sidebar, there are icons for a menu, a wrench (representing the Dialog editor), a refresh, a flow diagram, and a table. The main area has two buttons: 'Add node' (blue) and 'Add child node' (grey). Below these buttons is the title 'Chat Seguro Vehicular' with a speech bubble icon. A vertical line on the left indicates the flow of the dialog. Three nodes are visible in the flow: 1. A white node titled 'Bienvenido' with the response 'welcome' and '1 Response / 0 Context set'. 2. A blue node titled '#saludo' with the response '1 Response / 0 Context set'. 3. A white node titled 'En otras cosas' with the response 'anything_else' and '1 Response / 0 Context set'. A context menu is open over the '#saludo' node, showing four options: 'Add child node' (with a right arrow icon), 'Add node above' (with an up arrow icon), 'Add node below' (with a down arrow icon, which is highlighted), and 'Move' (with a four-way arrow icon).

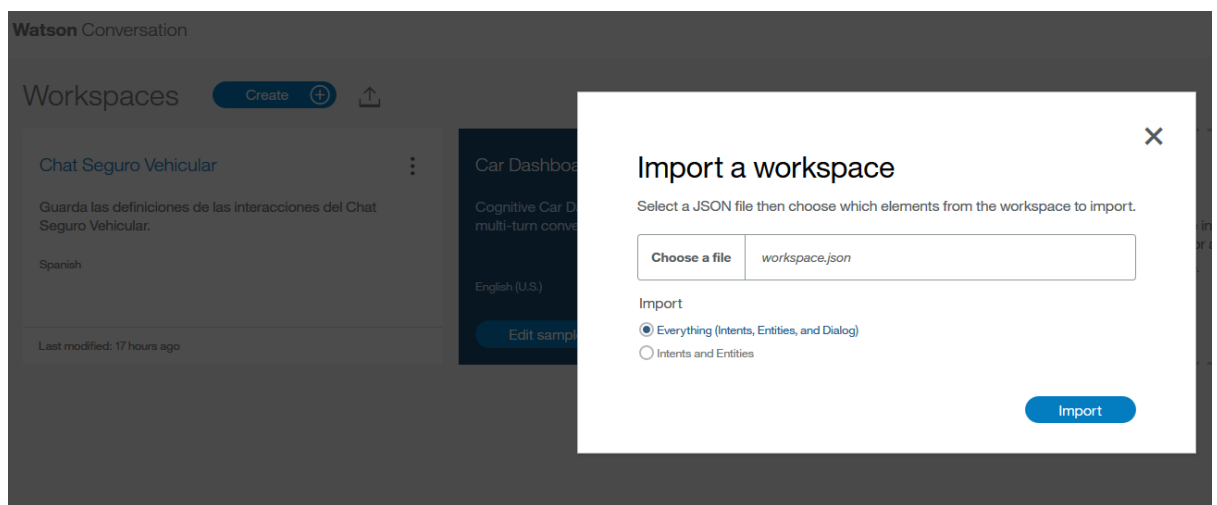
Name this node...
Customize
X

If bot recognizes:
#despedida


Then respond with:
Add response condition

Hasta la próxima!!! Ha sido un gusto atenderte,

19. El repositorio GitHub incluye un archivo con el export del workspace final en la ruta /entrenamientoConversacion/workspace.json. Para actualizarlo ingresamos a la consola de configuración del servicio Conversation e importamos el workspace:



20. Entrenamiento Visual Recognition. En la consola de administración del Servicio Visual Recognition lanzamos la herramienta de configuración. Para hacerlo ingresamos el API KEY que se obtiene de las credenciales de la consola. Creamos un clasificador llamado ClasificadorSinistro y dentro de él creamos tres clases: AutosConVidriosRotos, AutosChocados y Negative. Dentro de cada uno de estos hacemos upload de los archivos .zip llamados VidriosRotos, AutosChocados y SinChoques respectivamente. Los archivos de entrenamiento se encuentran en la ruta /entrenamientoVR de los archivos importados del repositorio. Luego de arrastrar cada comprimido y aceptar debemos esperar mientras el servicio entrena.


Visual Recognition Tool
2*****bb1
Update key

Create a new classifier

Creating a custom classifier will allow you to identify highly specialized subject matter. A classifier is created from a group of images trained by the Visual Recognition service to identify the subject matter of interest.

Classifier name

Classes
Upload at least 2 classes, each in a zipped file with at least 10 photos.

AutosConVidriosRotos

VidriosRotos.zip


Class name

Drag .zip here to train class
Or [choose your file](#)

Negative (Optional)

Drag .zip here to train class
Or [choose your file](#)

Add class
Cancel
Create


Visual Recognition Tool
2*****bb1
Update key

Create classifier

ClasificadorSiniestro
ClasificadorSiniestro_432578065
● training

Drag images here to classify them
Or [choose your files](#)

General
default
● ready


Drag images here to classify them
Or [choose your files](#)

Food
food
● ready

Drag images here to classify them
Or [choose your files](#)

21. Configuración servicio Base de Datos Cloudant. Ingresamos a la página del servicio y presionamos Launch:

Datos y análisis / servicioBDD1

 servicioBDD1

Cloudant NoSQL DB

LAUNCH

Cloudant NoSQL DB is a fully managed data layer designed for modern web and mobile applications that leverages a flexible JSON schema. Cloudant is built upon and compatible with Apache CouchDB and accessible through a secure HTTPS API, which scales as your application grows. Cloudant is ISO27001 and SOC2 Type 1 certified, and all data is stored in triplicate across separate physical nodes in a cluster for HA/DR within a data center.

Fully managed DBaaS

Work with self-describing JSON documents through a RESTful API that makes every document in your Cloudant database accessible as JSON via a URL. Documents can be retrieved, stored, or deleted individually or in bulk and can also have files attached. IBM takes care of the provisioning, management, and scalability of the data store, freeing up your time to focus on your application.

Powerful query, analytics, replication, and sync

Cloudant indexing is flexible and powerful, and includes real-time MapReduce, Apache Lucene-based full-text search, advanced Geospatial, and declarative Cloudant Query. Cloudant makes it easy to conduct advanced analytics on JSON data with dashDB Warehousing and Apache Spark integrations. Replication enables cross-geo deployments and Cloudant Sync provides data access for mobile devices to run connected or off-line.

22. Ingresamos a la sección Base de Datos a la izquierda:



23. Creamos una base de datos llamada aseguradoracognitiva



24. Luego de creada la base seleccionamos Documento, Añadir Documento. En el recuadro de texto pegamos el contenido del archivo datos.json que se encuentra en la carpeta /baseDatos de la carpeta exportada del repositorio git.



25. El documento se crea y ya es posible consultar datos desde la aplicación:

