Google Cloud Dialogflow

1. INTRODUCCIÓN

Google define Dialogflow como "una plataforma con comprensión del lenguaje natural que te facilita el diseño de una interfaz de usuario de conversación y su integración a tu aplicación para dispositivos móviles, aplicación web, dispositivo, bot, sistema de respuesta de voz interactiva y más".

Dialogflow permite el análisis de varios tipos de entrada, como texto o audio, y puede responder con texto o con voz sintética.

2. CONCEPTOS

1. Agentes

Son módulos de comprensión del lenguaje natural que manejan los tipos de conversaciones requeridas con los usuarios finales.

Dialogflow tiene dos tipos de agentes:

- **CX** – es un tipo de agente avanzado, adecuado para crear agentes grandes o complejos; los componentes básicos del diseño de conversaciones son los flujos y las páginas, y se usan controladores de estado para controlar las rutas de conversación.
- **ES** – es el tipo de agente estándar, adecuado para crear agentes pequeños, medianos o moderadamente complejos; los componentes básicos del diseño de las conversaciones son los intents, y se usan los contextos para controlar las rutas de conversación.

En este documento se habla sobre el **Dialogflow ES**.

2. Intents

Un intent clasifica la intención del usuario final para un turno de conversación. Para manejar una conversación completa, se definen muchos intentos para un agente y cuando un usuario escribe o dice algo Dialogflow hace coincidir la expresión con el mejor intent.

Un intent contiene los siguientes elementos:

- *Frases de entrenamiento* – son frases de ejemplo que podrían decir los usuarios finales; son las frases que usa Dialogflow para hacer coincidir las expresiones de los usuarios finales con los intents definidos en los agentes; no es necesario definir todos los ejemplos posibles.
- *Acción* – se define por cada intent y se usa para activar ciertas acciones en el sistema.

- *Parámetros* – cuando un intent coincide en el entorno de ejecución, Dialogflow proporciona los valores extraídos de la expresión del usuario final como parámetros; a diferencia de la entrada sin procesar del usuario final, los parámetros son datos estructurados que se pueden usar con facilidad para realizar alguna lógica o generar respuestas.
- Respuestas – proporcionan respuesta al usuario final, pueden ser de texto, de voz o visuales.

3. Entidades

Cada parámetro de un intent, llamado tipo de entidad, que dicta como se extraen los datos de una expresión.

Dialogflow proporciona entidades predefinidas para muchos tipos comunes de datos, como pueden ser: fechas, horas, colores, direcciones de correo electrónico. También se pueden definir entidades personalizadas.

4. Contextos

Sirven para controlar el flujo de una conversación y son de dos tipos:

- Contextos de entrada – controlan la detección de coincidencias con intents;
- Contextos de salida – controlan los contextos activos; cuando se detecta una coincidencia con un intent, se activan los contextos de salida configurados para ese intent.

5. Intents de seguimiento

Se usan para establecer contextos para grupos de intents de forma automática. Un intent de seguimiento es elemento secundario del intent superior asociado y solo coincide cuando el intent superior coincide en el turno anterior de conversación.

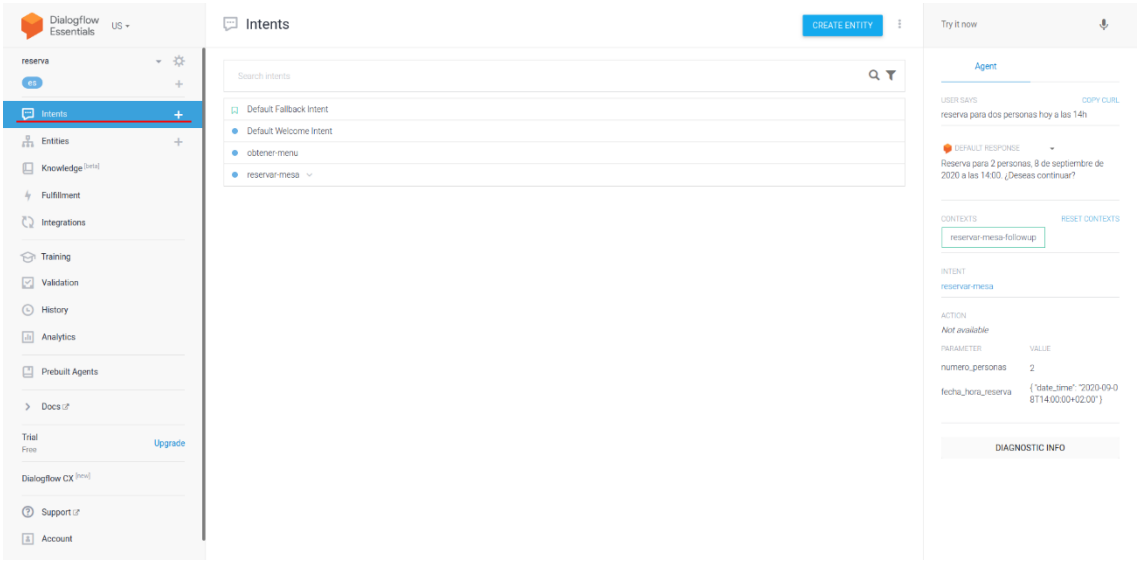
6. Fulfillment

De forma predeterminada, un agente responde al intent coincidente con una respuesta estática. Con *fulfillment* (entrega) se pueden proporcionar respuestas más dinámicas, llamando al servicio definido. Cada intent tiene una configuración que permite la entrega.

3. INTERFAZ DIALOGFLOW

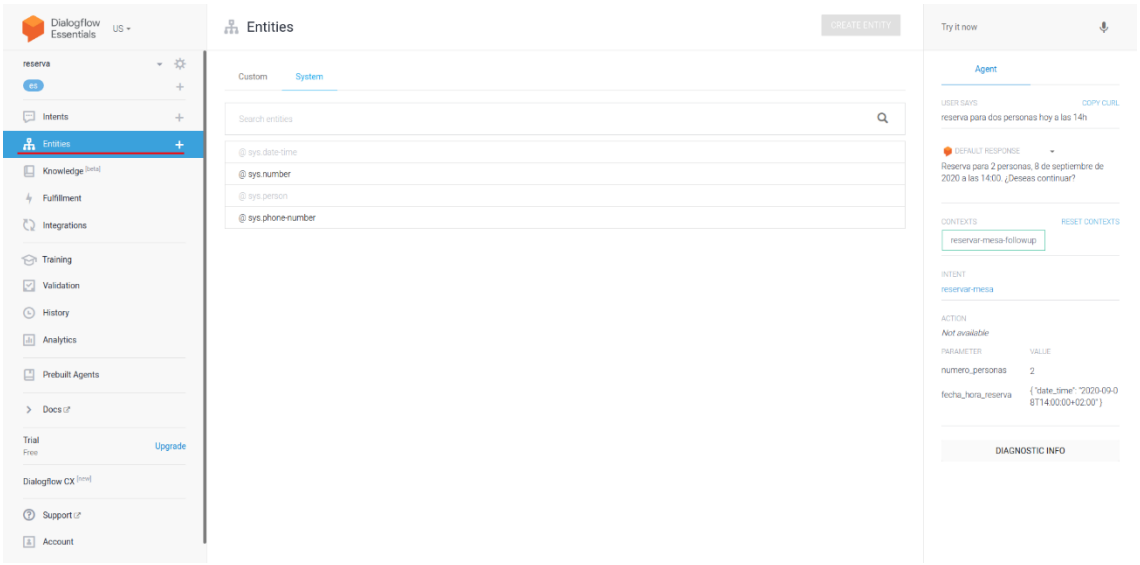
1. Intents

Permite la creación y gestión de intents con toda la configuración necesaria.



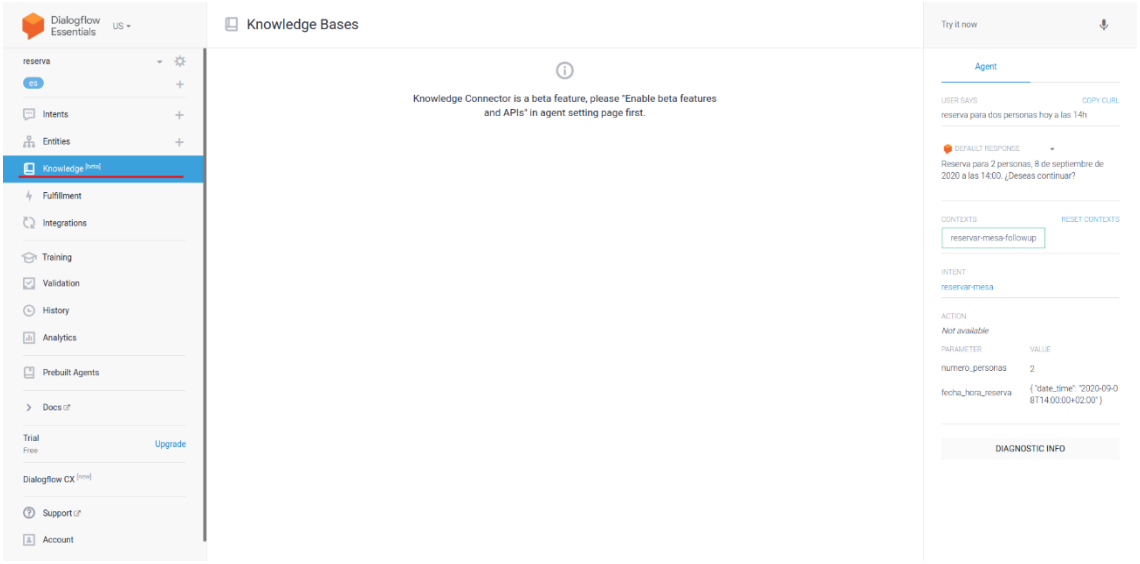
2. Entities

Permite la creación y gestión de las entidades necesarias para el agente.



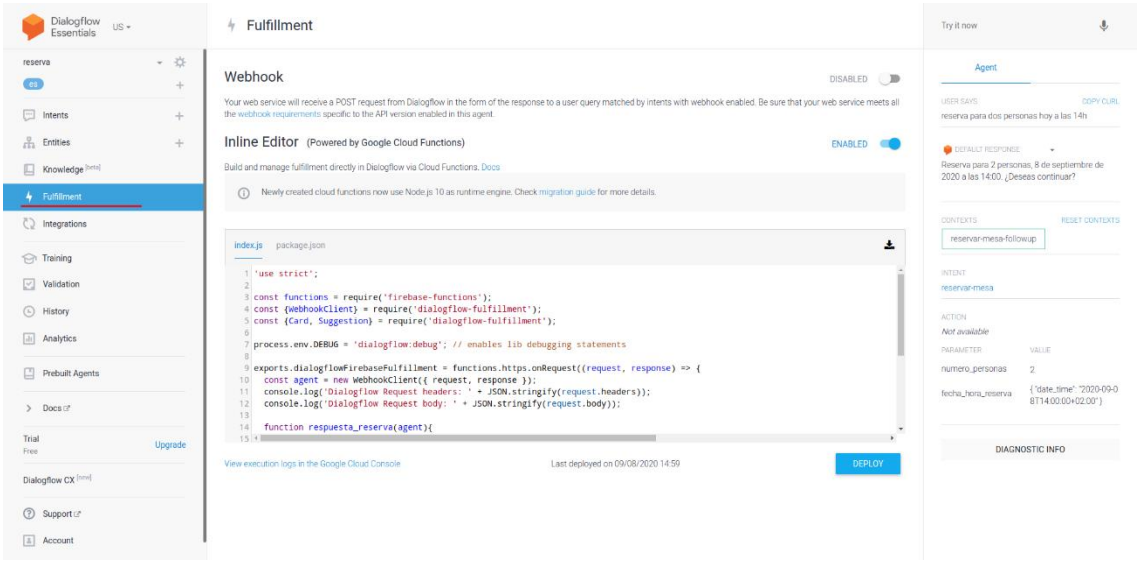
3. Knowledge

Permite definir y gestionar conectores de conocimiento que complementan los intents definidos. Estos analizan documentos de conocimiento (FAQs, artículos...) para encontrar respuestas automáticas. Es una funcion en Beta por lo que hay que activarla en la configuración del agente.



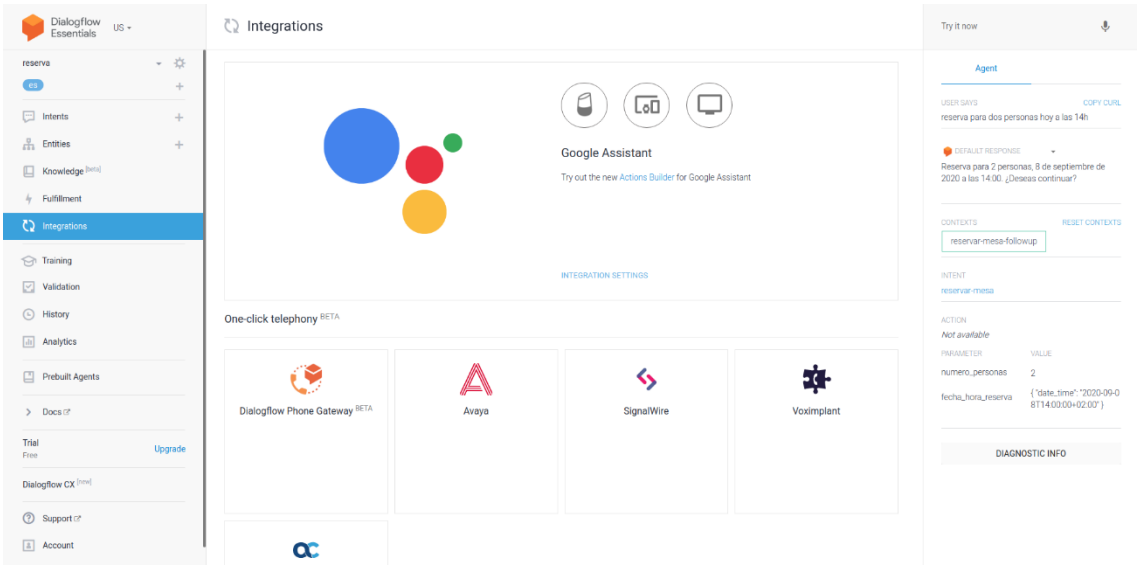
4. Fulfillment

Permite definir respuestas más dinámicas usando Google Cloud Functions.



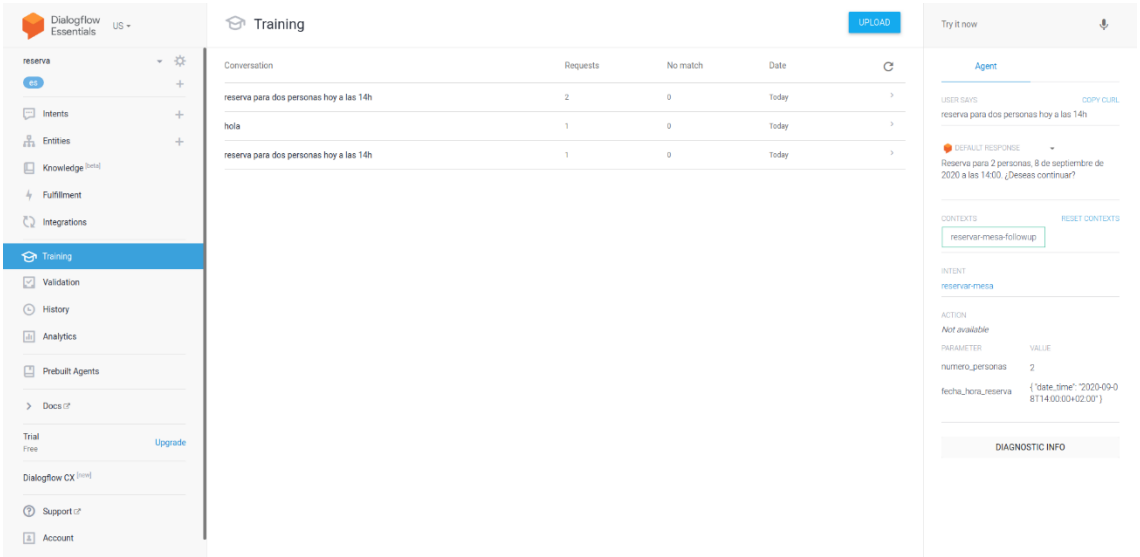
5. Integrations

Permite ver y configurar las integraciones con otros servicios (Slack, Messenger ...).



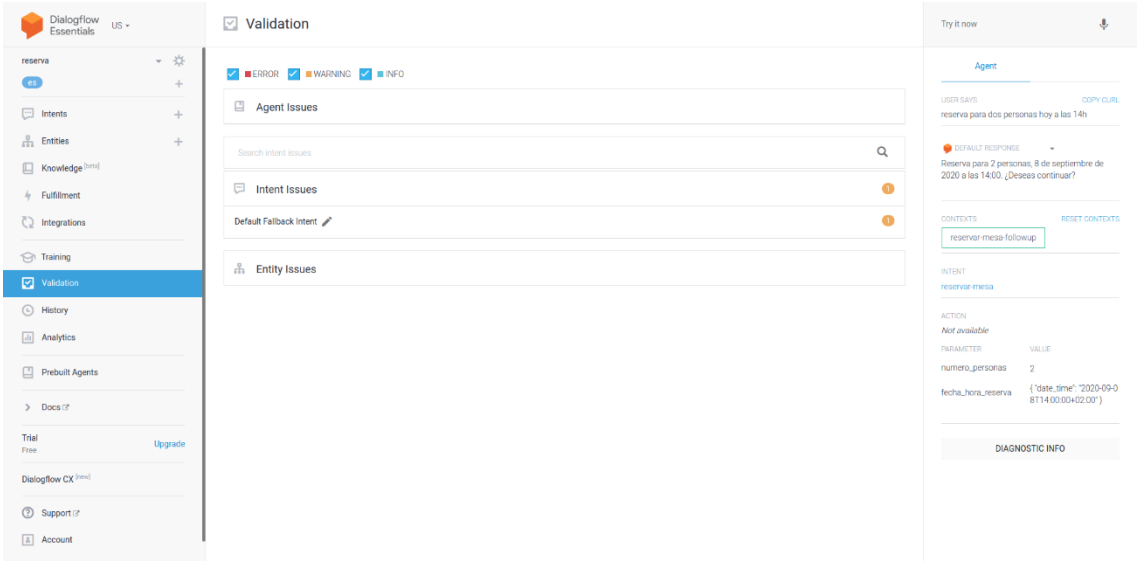
6. Training

Permite ver y añadir a los intentos las frases que se han usado para probar el agente. También permite la subida de conjuntos de datos de entrenamiento.



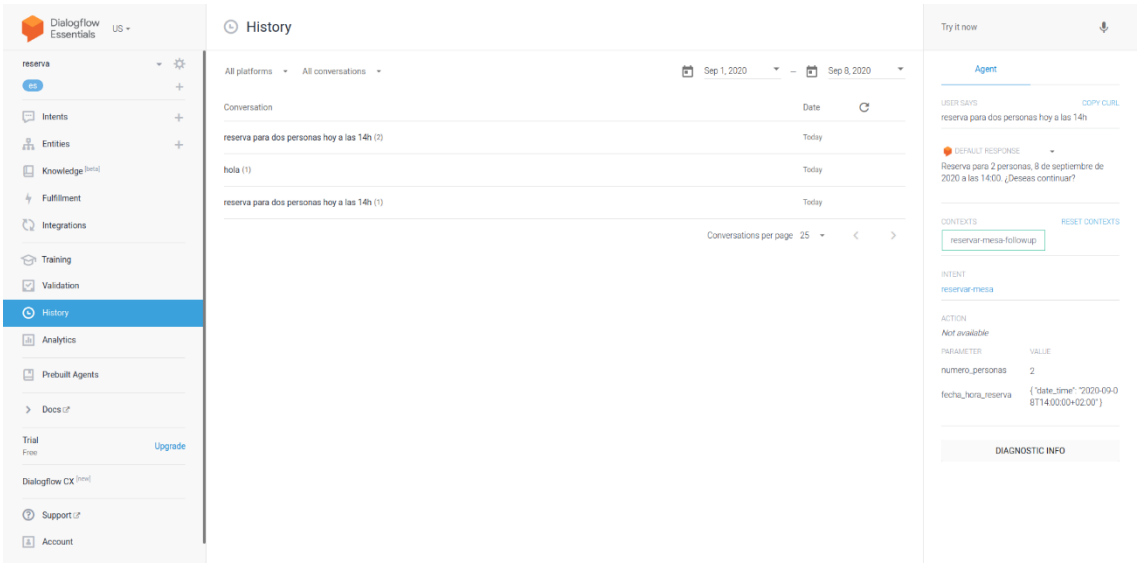
7. Validation

Permite ver información, advertencias y errores del agente.



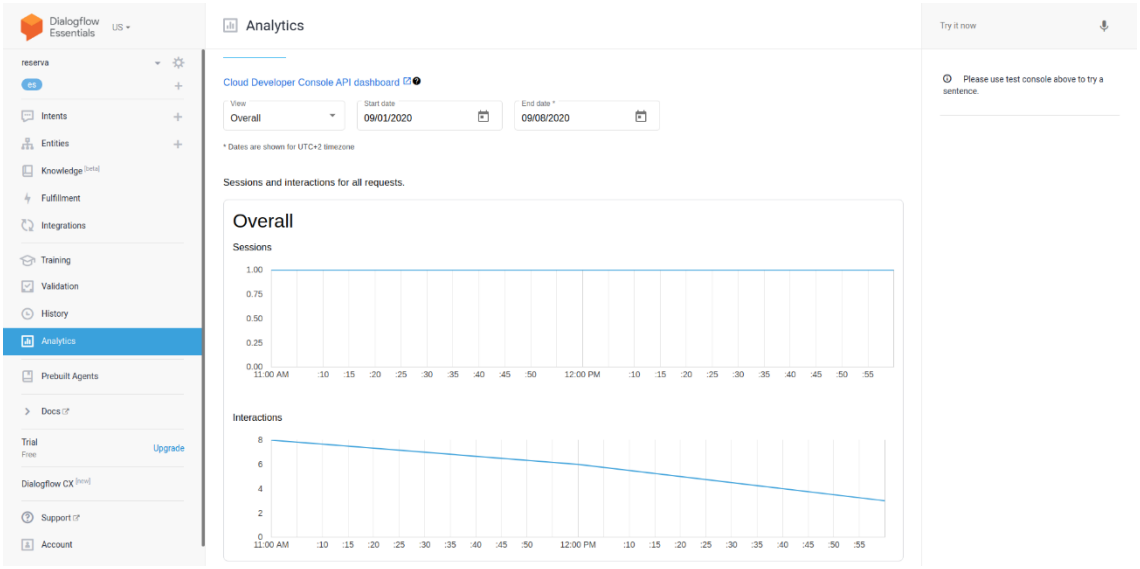
8. History

Permite ver todos los mensajes de todos los usuarios



9. Analytics

Permite ver las métricas (interacciones, numero de sesiones) del agente.



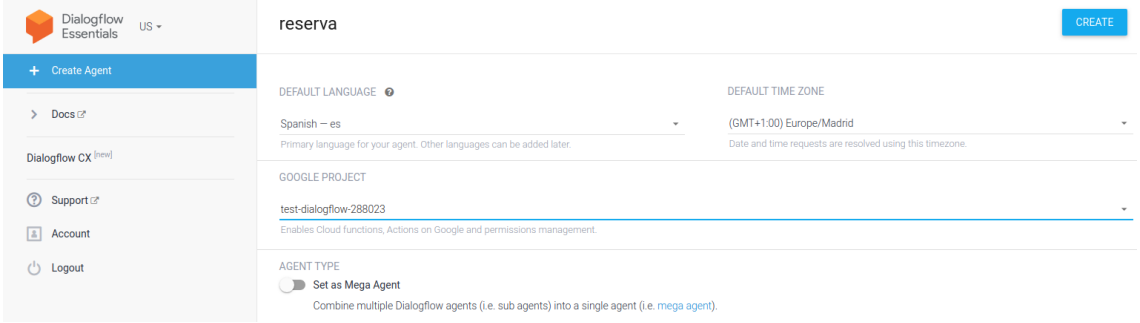
4. EJEMPLO CREACIÓN DE AGENTES

En este ejemplo crearemos un agente que se encarga de las reservas de un restaurante. Definiremos dos intents, uno para obtener el menú del día y otro para reservar una mesa. El intent de reserva tendrá intents de seguimiento para confirmar que el usuario desea continuar con la reserva y para obtener los datos de contacto.

Para crear agentes hay que navegar a la Consola de Dialogflow ES y pinchar en *Create Agent*.

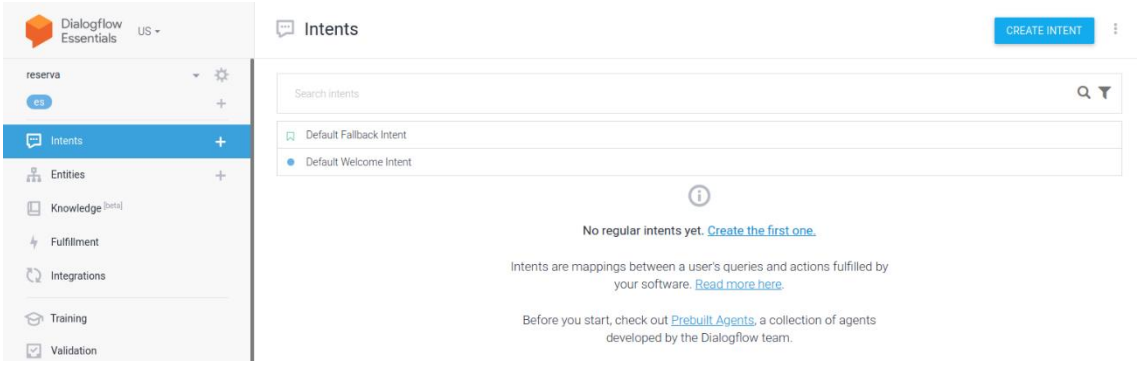
1. Creación del agente

Escribimos el nombre del agente, elegimos el idioma y la zona horaria, y creamos o elegimos un proyecto *de Google Cloud Platform*.

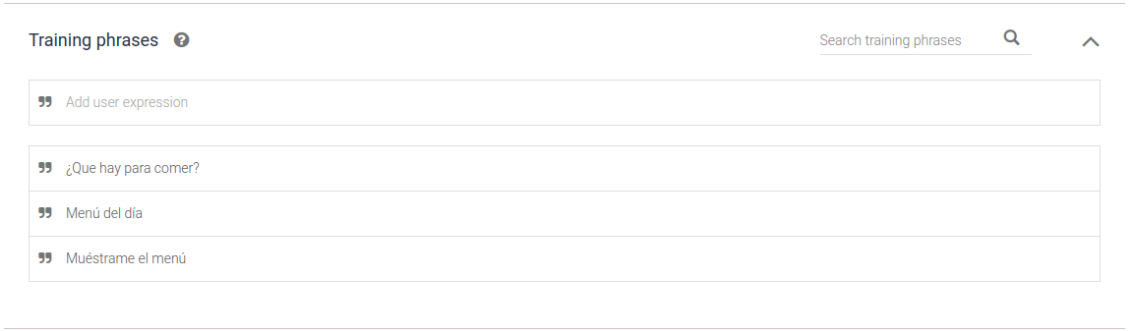


7. Intent obtener-menu

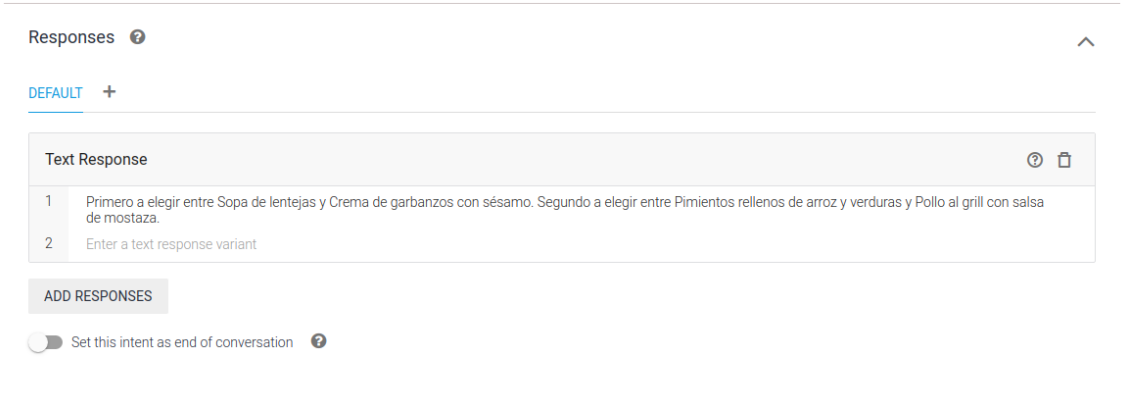
Creamos un intent para obtener el menú del día.



Añadimos las frases de entrenamiento.

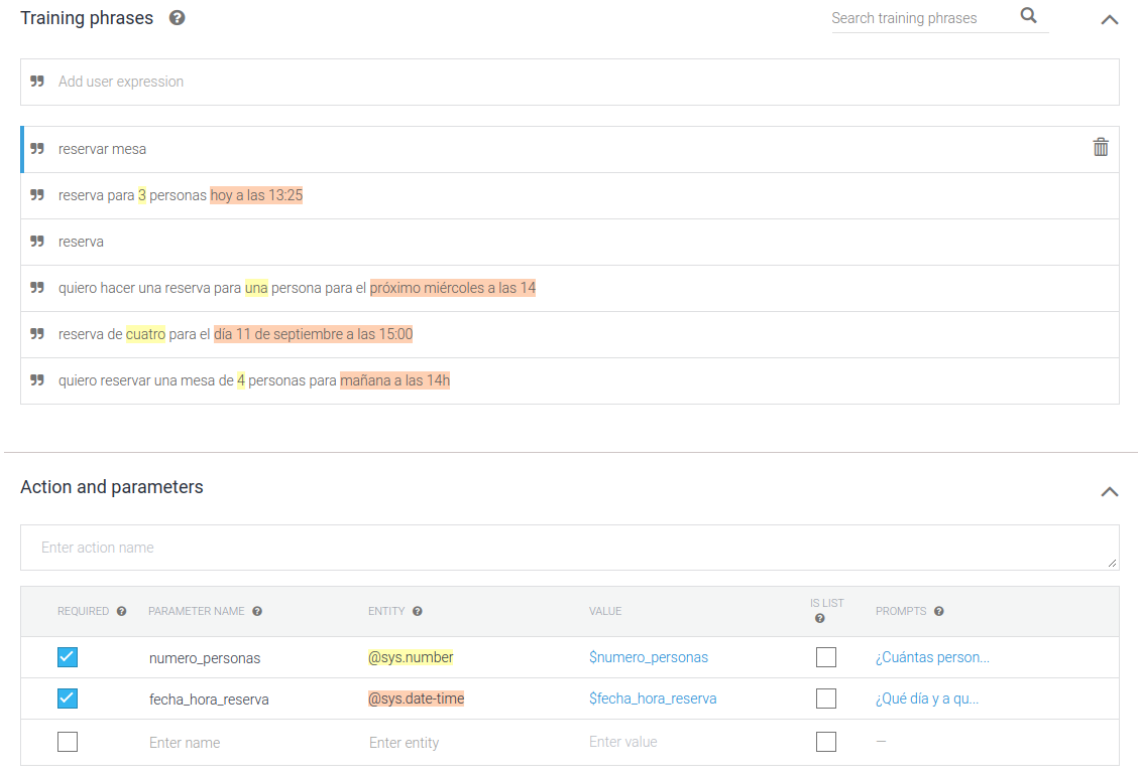


Creamos la respuesta.

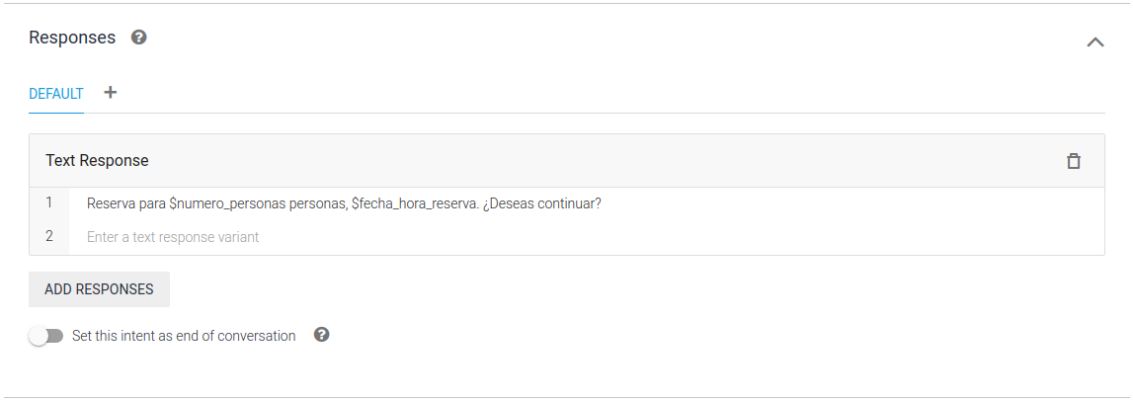


8. Intent reservar-mesa

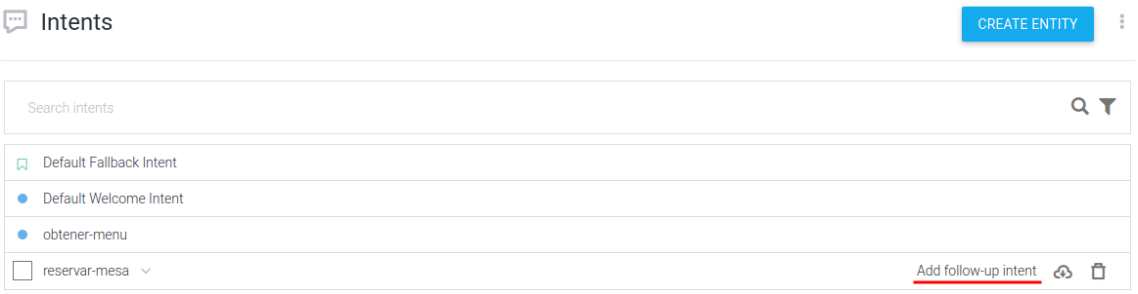
Añadimos las frases de entrenamiento y definimos los parámetros *numero_personas* y *fecha_hora_reserva*. Marcamos las casillas *REQUIRED* para especificar que los parámetros son necesarios. En *PROMPTS* definimos las preguntas que el agente hará a los usuarios si la solicitud no contiene los parámetros.



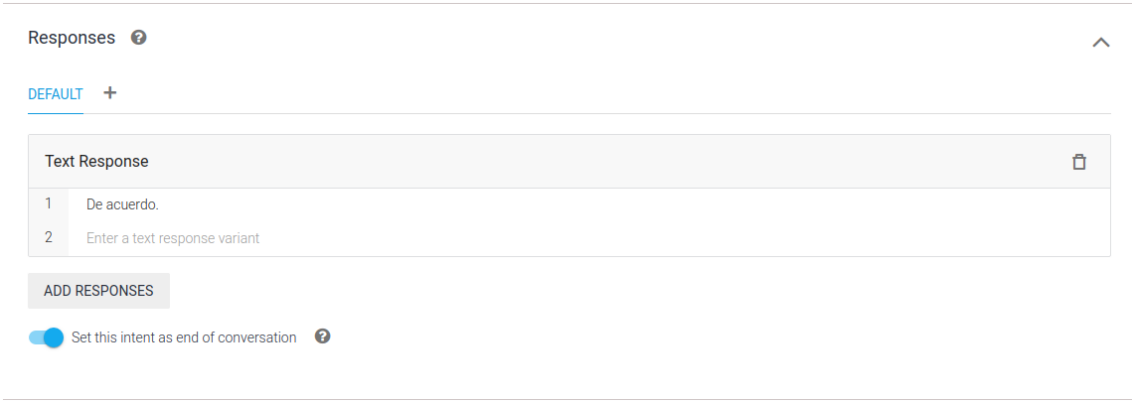
Definimos la respuesta usando los parámetros obtenidos.



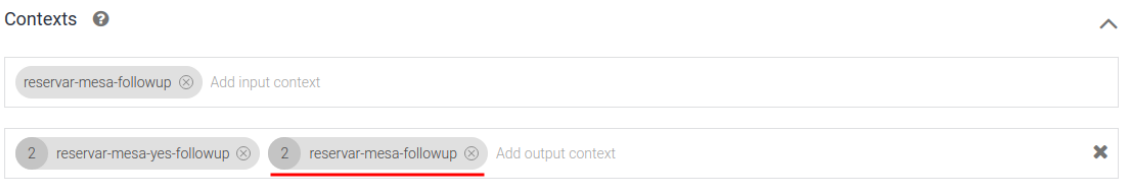
Creamos dos intents de seguimiento, afirmación y negación, para el intent *reservar-mesa* pinchando en *Add follow-up intent*. Se despliega un selector con varios tipos de intents. Anadimos *yes* y *no*.



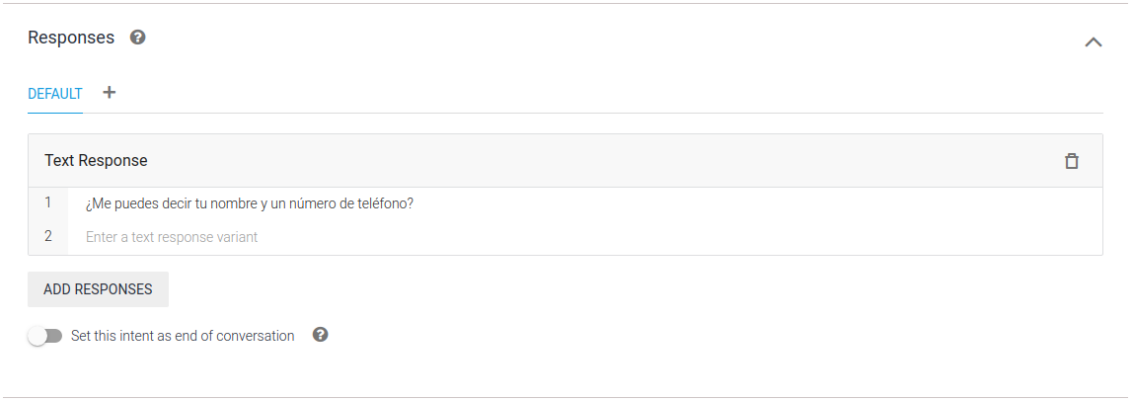
Navegamos al intent *reservar-mesa - no*. Observamos que Dialogflow ha insertado algunas frases de entrenamiento. Definimos las respuestas y marcamos *Set this intent as end of conversation*.



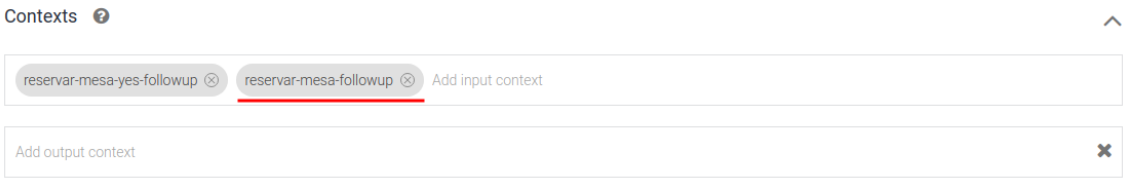
Navegamos al intent *reservar-mesa - yes*. Observamos que Dialogflow ha insertado algunas frases de entrenamiento. Pasamos el contexto reservar-mesa-followup en el output para que el intent de seguimiento de este intent (*reservar-mesa - yes*) tenga los datos obtenidos en el intent *reservar-mesa*.



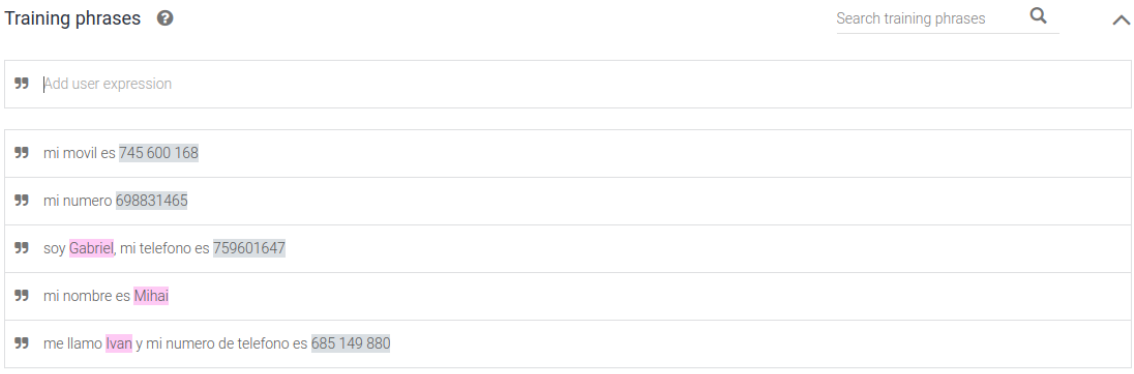
Definimos las respuestas.

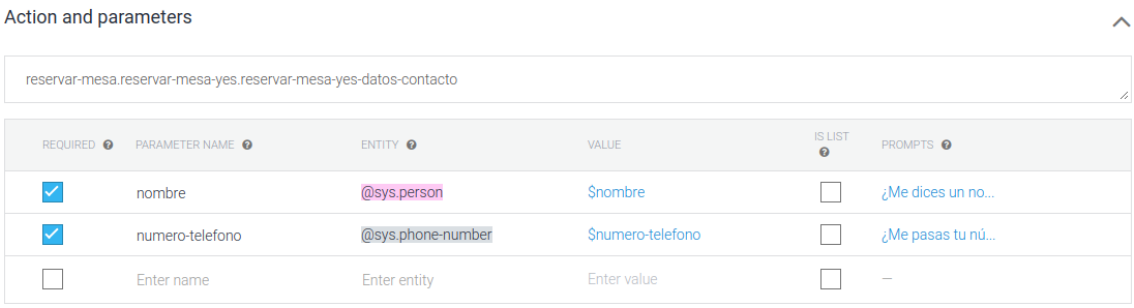


Creamos un intent de seguimiento para el intent *reservar-mesa - yes*. Añadimos el contexto *reservar-mesa-followup* en el input.

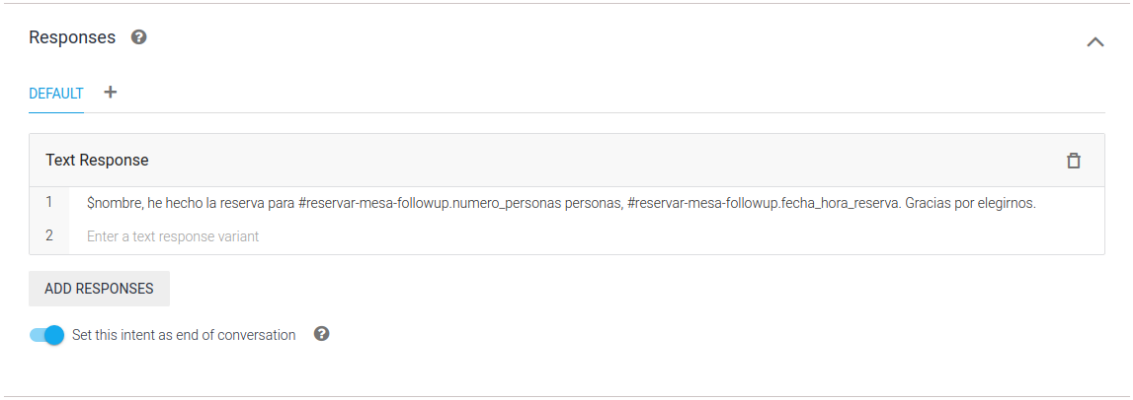


Añadimos las frases de entrenamiento y definimos los parámetros *nombre* y *numero-telefono*. Igual que en el intent *reservar-mesa*, marcamos las casillas *REQUIRED* para especificar que los parámetros son necesarios y definimos los *PROMPTS*.





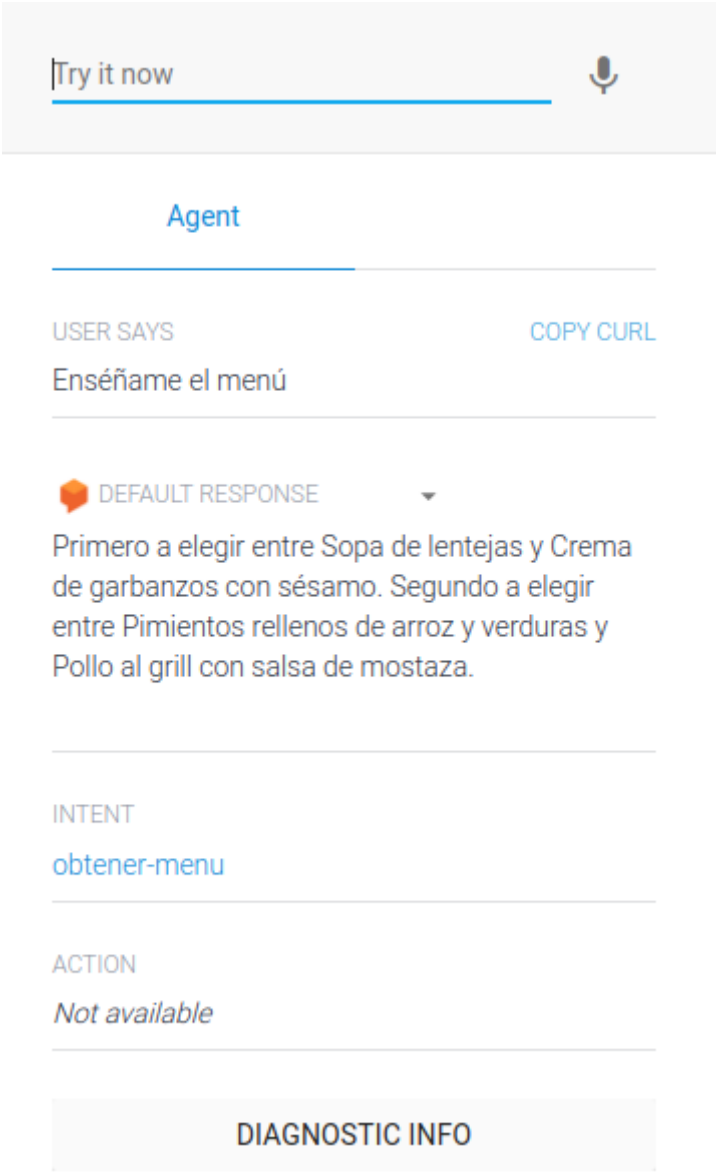
Definimos la respuesta, usando los parámetros obtenidos y los del contexto *reservar-mesa-followup*. Marcamos *Set this intent as end of conversation*.



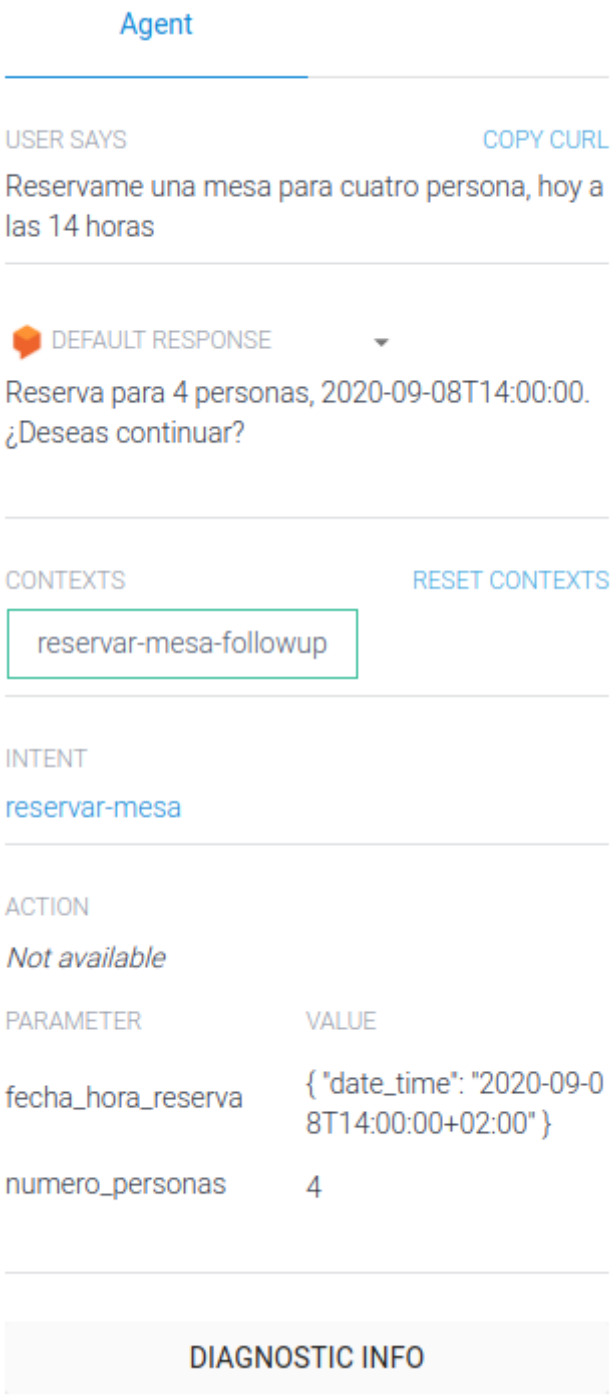
9. Pruebas

Desde la interfaz de Dialogflow se puede probar el agente.

Probamos con la frase "Enséñame el menú" y observamos que la intención del usuario coincide con el intent *obtener-menu*.

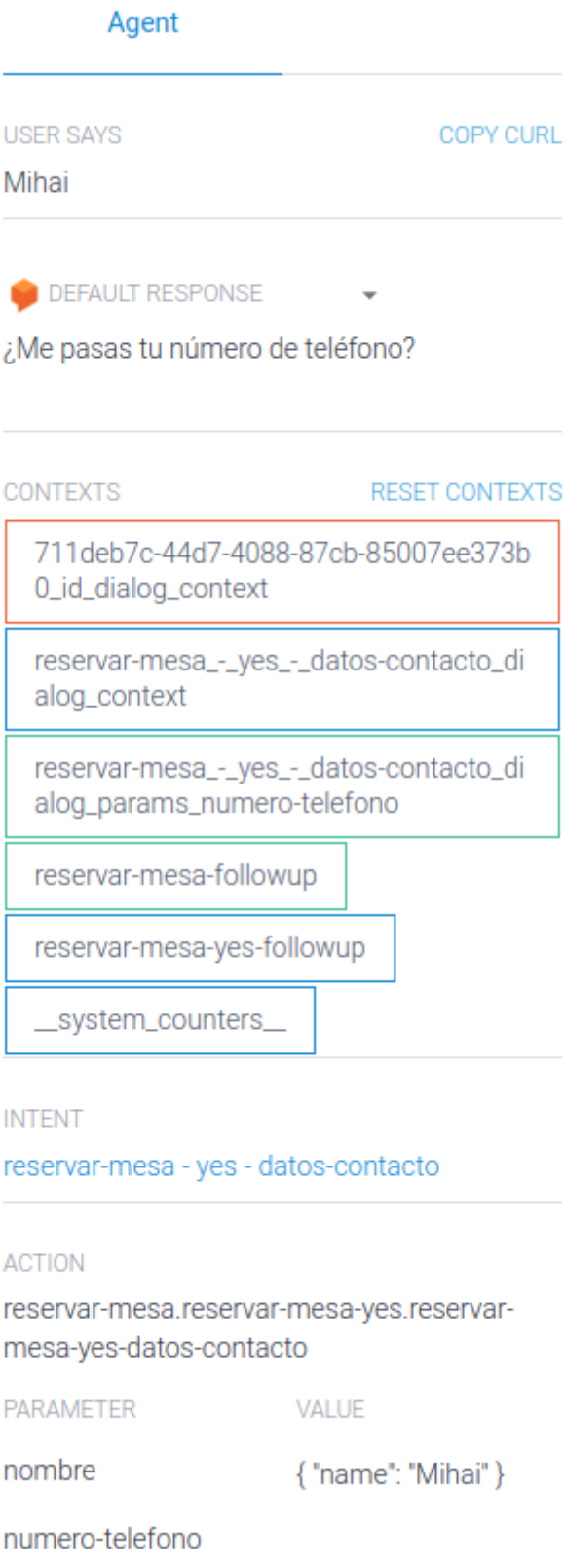


Probamos con la frase "Reservame una mesa para cuatro personas, hoy a las 14 horas" y observamos que la intención del usuario coincide con el intent reservar-mesa y que se han obtenido también los parámetros.

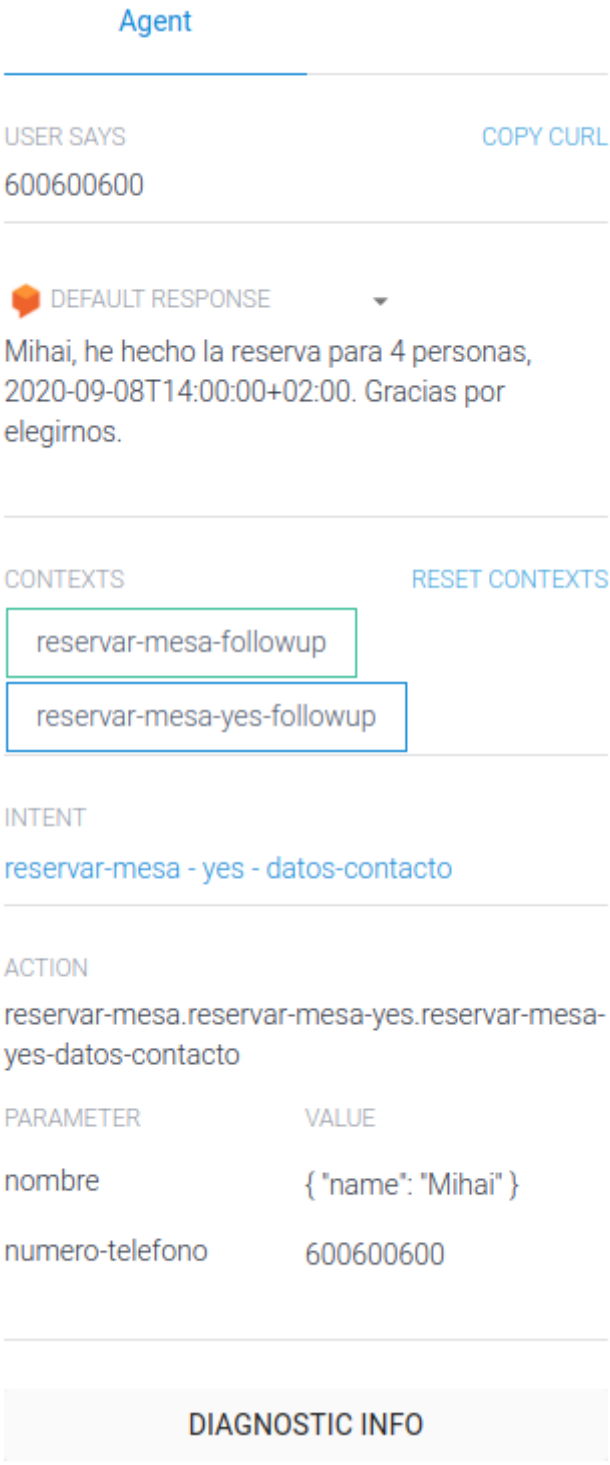


Continuamos la conversación con la frase "si". Observamos que el agente pregunta por el nombre y el número de teléfono.

Le pasamos solo el nombre. Observamos que se ha obtenido el parámetro *nombre* y que el agente pregunta por el teléfono.



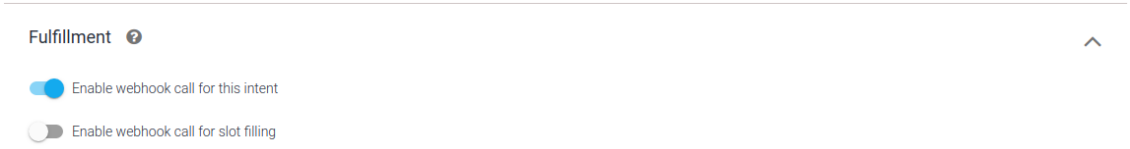
Le pasamos el número de teléfono. Observamos la respuesta del agente y los parámetros.


Agent
USER SAYS
COPY CURL
600600600
DEFAULT RESPONSE
Mihai, he hecho la reserva para 4 personas, 2020-09-08T14:00:00+02:00. Gracias por elegirnos.
CONTEXTS
RESET CONTEXTS
reservar-mesa-followup
reservar-mesa-yes-followup
INTENT
reservar-mesa - yes - datos-contacto
ACTION
reservar-mesa.reservar-mesa-yes.reservar-mesa-yes-datos-contacto
PARAMETER
VALUE
nombre
{ "name": "Mihai" }
numero-telefono
600600600
DIAGNOSTIC INFO

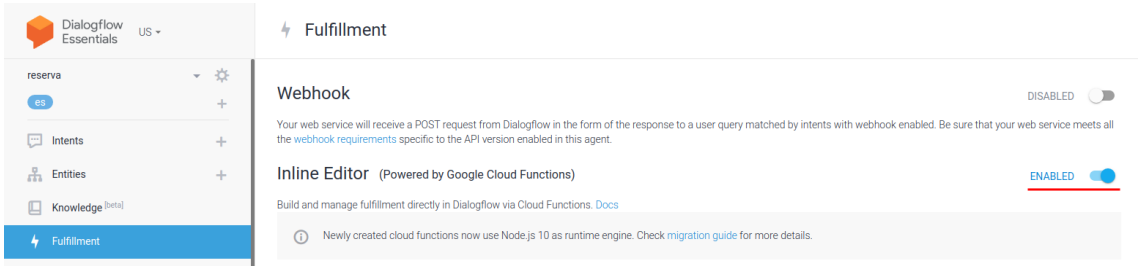
10. Fulfillment

Como se puede observar, el formato de la fecha (ISO 8601) no es amigable. Lo podemos cambiar usando *Fulfillment*.

Navegamos a ***Intents*** y habilitamos ***Enable webhook call for this intent*** para los intents *reservar-mesa* y *reservar-mesa - yes - datos-contacto*.



Luego, navegamos a ***Fulfillment*** y habilitamos ***Inline Editor***. Esto nos permite crear funciones directamente en Dialogflow.



Hay dos ficheros: index.js y package.json. Los completamos con la siguiente información:

- index.js - definimos y registramos la función *respuesta_reserva* para el intent *reserva-mesa*; esta función parsea la fecha y cambia el formato a uno más amigable.

```
'use strict';

const functions = require('firebase-functions');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Card, Suggestion} = require('dialogflow-fulfillment');

process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements

exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request,
response) => {
  const agent = new WebhookClient({ request, response });
  console.log('Dialogflow Request headers: ' +
JSON.stringify(request.headers));
  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));

  function respuesta_reserva(agent){
    const fecha = new Date(agent.parameters.fecha_hora_reserva.date_time);
    const opcionesFecha = {year: 'numeric', month: 'long', day: 'numeric'};
    const opcionesHora = {hour: '2-digit', minute: '2-digit', timeZone:
'Europe/Madrid'};
```

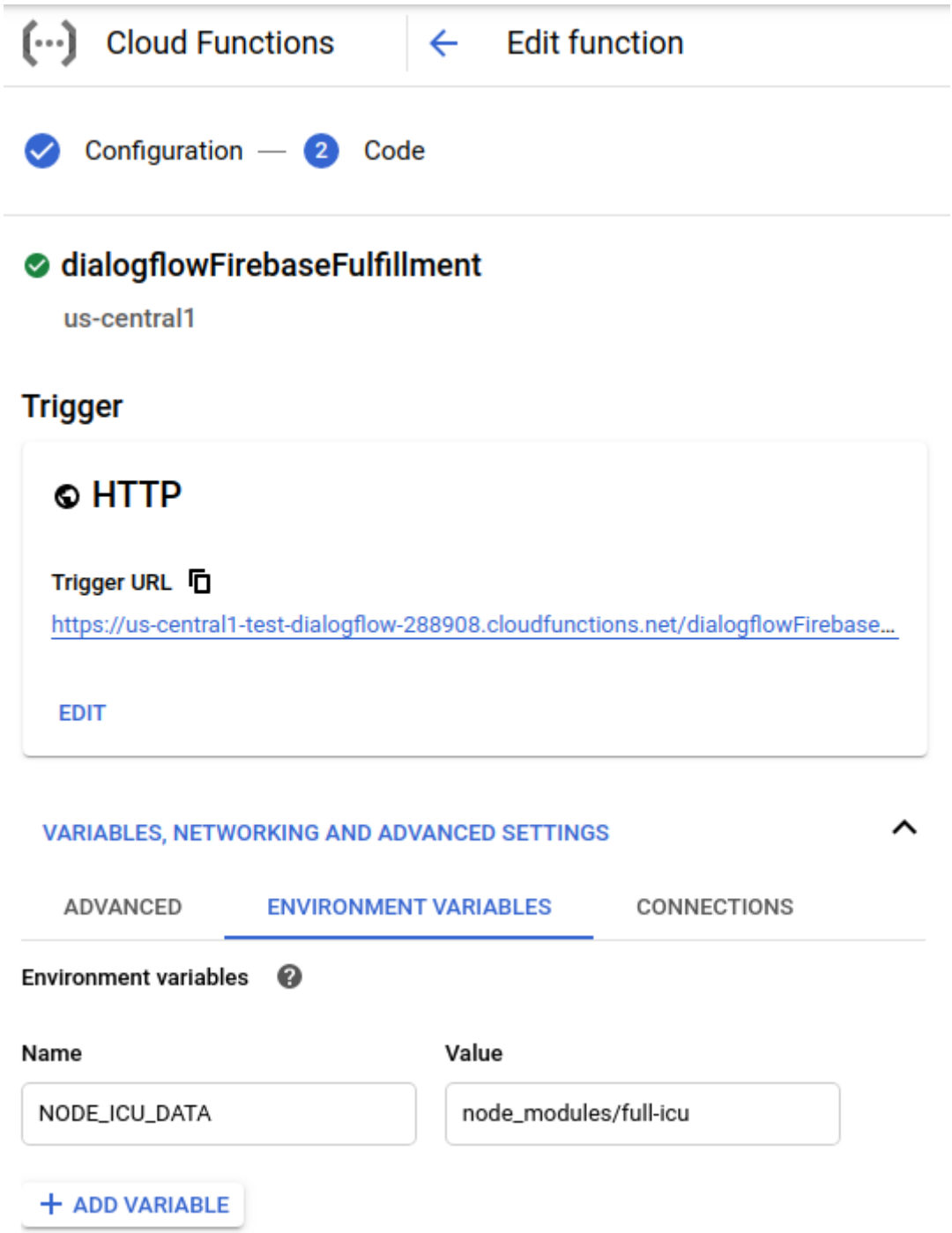
```
    agent.add(`Reserva para ${agent.parameters.numero_personas} personas,
${fecha.toLocaleDateString('es-ES', opcionesFecha)} a las
${fecha.toLocaleTimeString('es-ES', opcionesHora)}.`);
  }

  // Run the proper function handler based on the matched Dialogflow intent
name
  let intentMap = new Map();
  intentMap.set('reservar-mesa', respuesta_reserva);
  agent.handleRequest(intentMap);
});
```

- package.json - añadimos la dependencia *full-icu* para poder formatear la fecha en español.

```
{
  "name": "dialogflowFirebaseFulfillment",
  "description": "This is the default fulfillment for a Dialogflow agents
using Cloud Functions for Firebase",
  "version": "0.0.1",
  "private": true,
  "license": "Apache Version 2.0",
  "author": "Google Inc.",
  "engines": {
    "node": "10"
  },
  "scripts": {
    "start": "firebase serve --only functions:dialogflowFirebaseFulfillment",
    "deploy": "firebase deploy --only
functions:dialogflowFirebaseFulfillment"
  },
  "dependencies": {
    "actions-on-google": "^2.2.0",
    "firebase-admin": "^5.13.1",
    "firebase-functions": "^2.0.2",
    "dialogflow": "^0.6.0",
    "dialogflow-fulfillment": "^0.5.0",
    "full-icu": "^1.3.0"
  }
}
```

Hacemos clic en ***Deploy*** para desplegar el código.

Por último, navegamos a la lista de funciones en Google Cloud, hacemos clic en *dialogflowFirebaseFulfillment* y luego en ***Edit***. Añadimos la variable *NODE_ICU_DATA* con valor *node_modules/full_icu* para especificar que queremos usar el conjunto de datos entero (*full-icu*) para la internacionalización. Pinchamos ***Next*** y luego ***Deploy*** para desplegarlo.



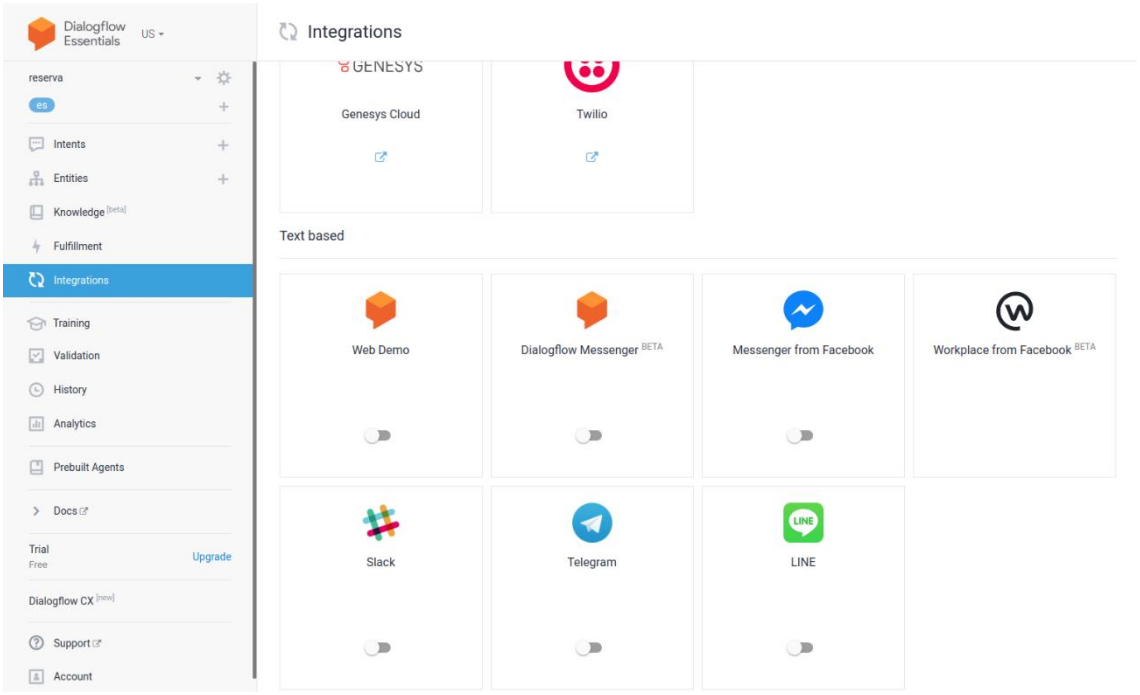
Probamos el agente con la frase "reserva para dos personas hoy a las 14h" y observamos como ha cambiado el formato de la fecha.

11. Interacciones

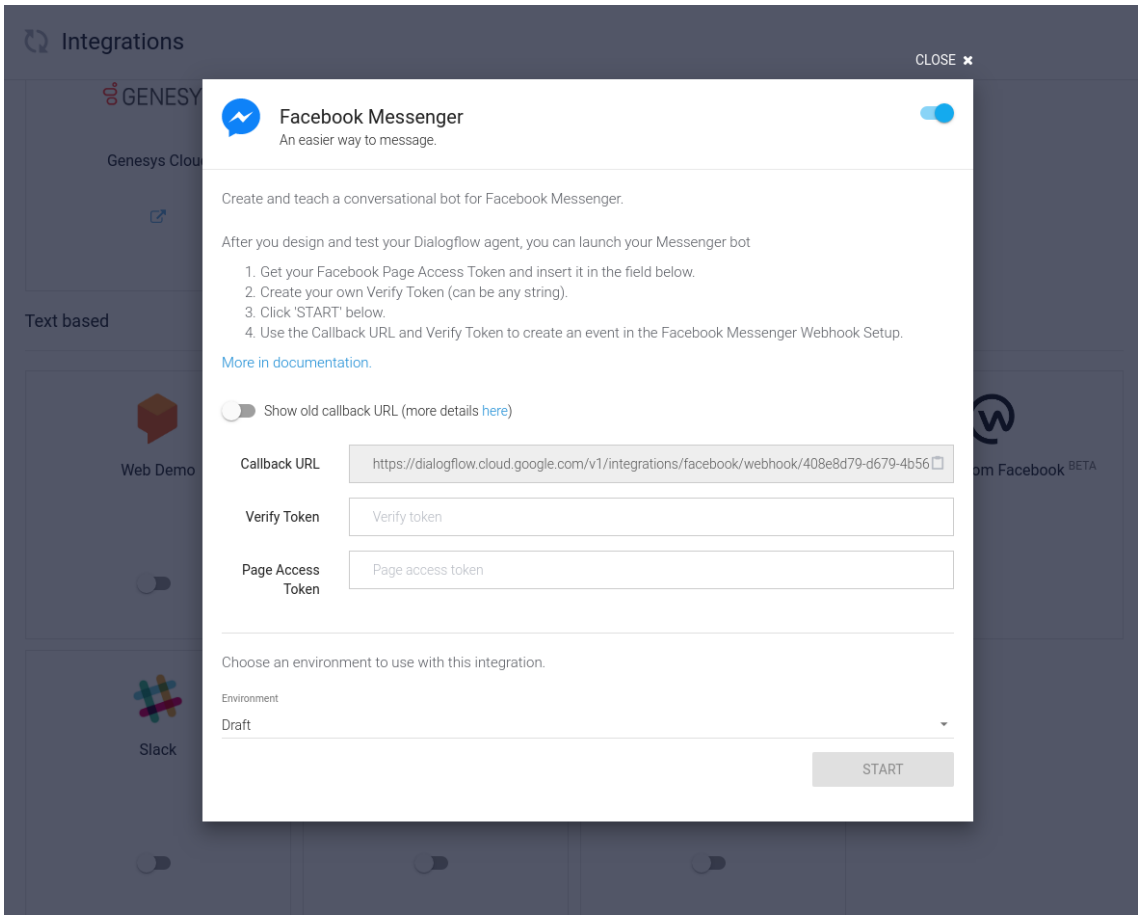
Se puede interaccionar con el agente usando las integraciones que proporciona Google Dialogflow (Messenger, Telegram, Slack ...), pero también a través de la API.

a. Integraciones

Las integraciones se encuentran en ***Integrations***.

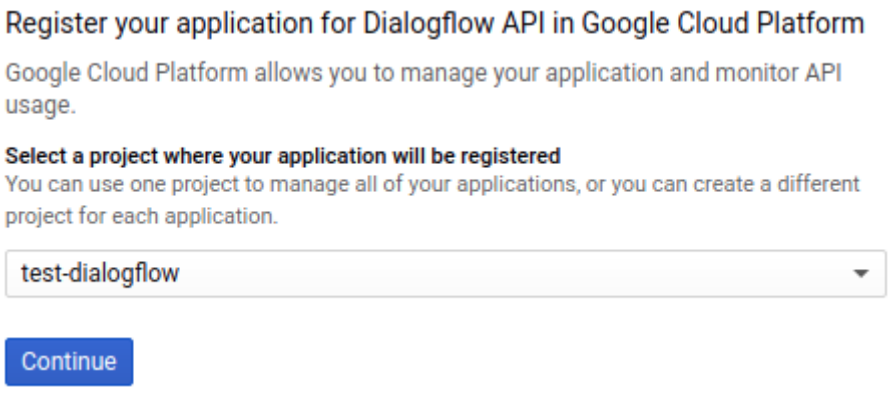


Cada integración tiene su propia configuración y se tiene que rellenar en la interfaz.

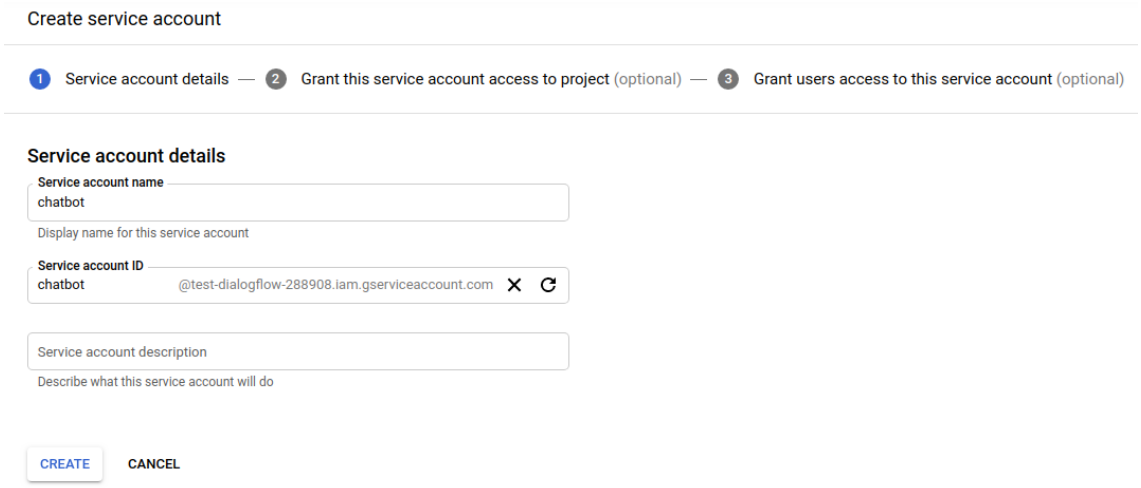


b. API

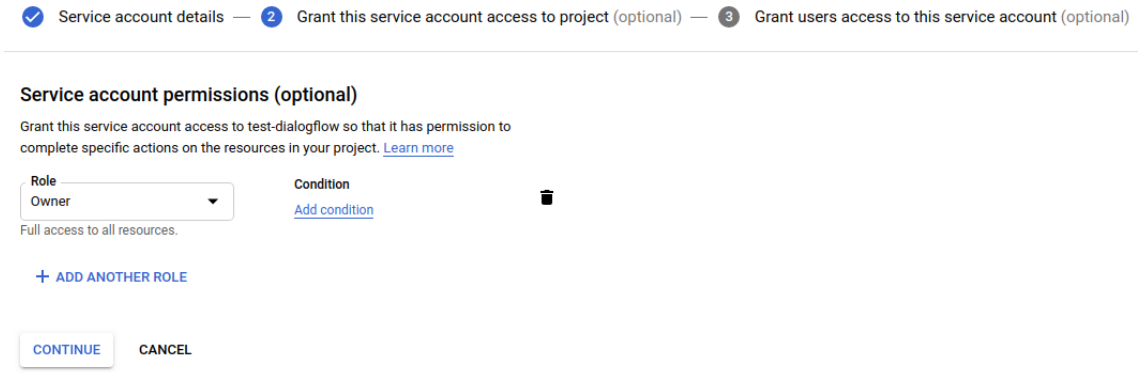
Para poder usar la API tenemos que activar el API de Dialogflow en el proyecto.



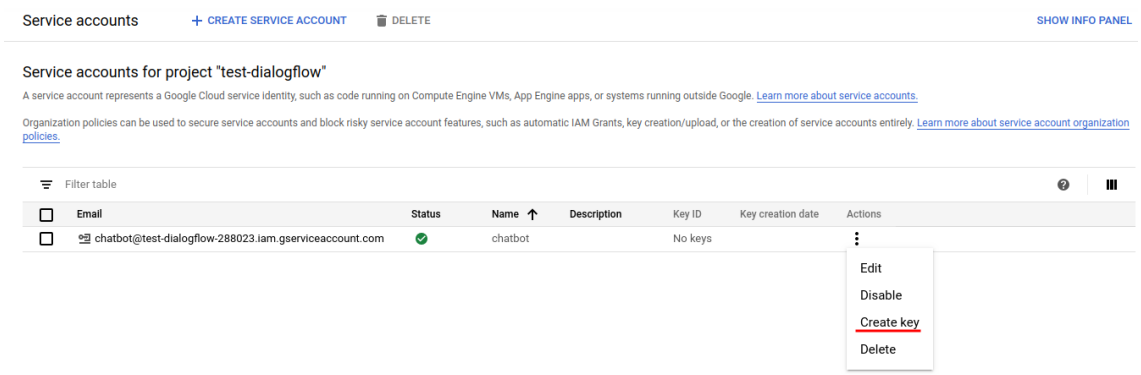
También tenemos que crear una cuenta de servicio especificando el nombre y el ID de la cuenta.



Para la prueba le concedemos el Rol "Owner" aunque no es recomendable.



Una vez creada la cuenta de servicio generamos la clave privada en formato JSON.



Esto activa la descarga automática de la clave privada.

Creamos un proyecto en Python e instalamos el paquete *dialogflow*.

```
pip install dialogflow
```

Creamos un fichero llamado cliente.py con el código de un pequeño cliente.

```
import os
import json
import uuid
import dialogflow_v2 as dialogflow


def establecer_conexion():
    sesion_cliente = dialogflow.SessionsClient()
    with open(os.environ['GOOGLE_APPLICATION_CREDENTIALS']) as credenciales:
        id_proyecto = json.load(credenciales)['project_id']

    sesion = sesion_cliente.session_path(id_proyecto, str(uuid.uuid4()))
    return sesion_cliente, sesion


if __name__ == '__main__':
    sesion_cliente, sesion = establecer_conexion()

    while True:
        text = input('Yo: ')
        if text == 'q':
            break

        text_input = dialogflow.types.TextInput(text=text, language_code='es-ES')
        query_input = dialogflow.types.QueryInput(text=text_input)
        respuesta = sesion_cliente.detect_intent(session=sesion,
query_input=query_input)
        print(f'Chatbot: {respuesta.query_result.fulfillment_text}')
```

En el código se define la función *establecer_conexion* que devuelve la sesión del cliente y la del usuario. En el *main* obtenemos la conexión, y en el bucle leemos desde la entrada estándar y enviamos el texto al agente. El agente devuelve la respuesta y la imprimimos.

Para la correcta función del cliente se tiene que pasar la ruta de la clave publica descargada.

```
GOOGLE_APPLICATION_CREDENTIALS=credenciales.json python cliente.py
```

Resultado:

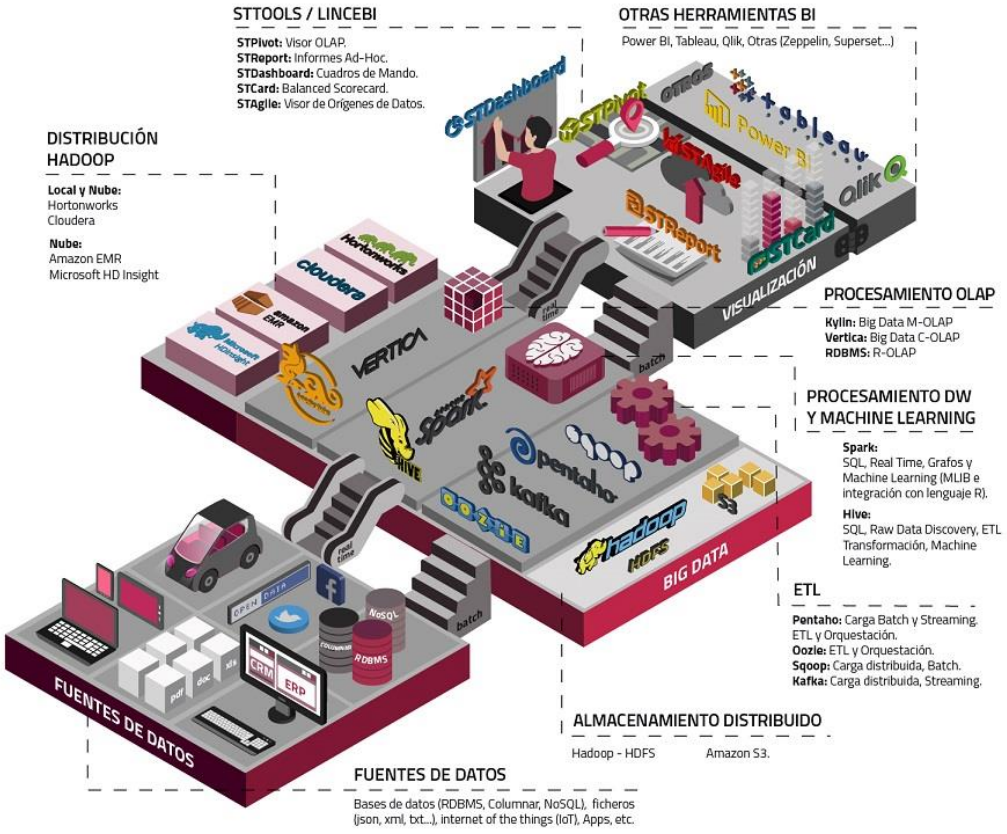
```
Yo: menu del dia
Chatbot: Primero a elegir entre Sopa de lentejas y Crema de garbanzos con
sésamo. Segundo a elegir entre Pimientos rellenos de arroz y verduras y Pollo
al grill con salsa de mostaza.
```

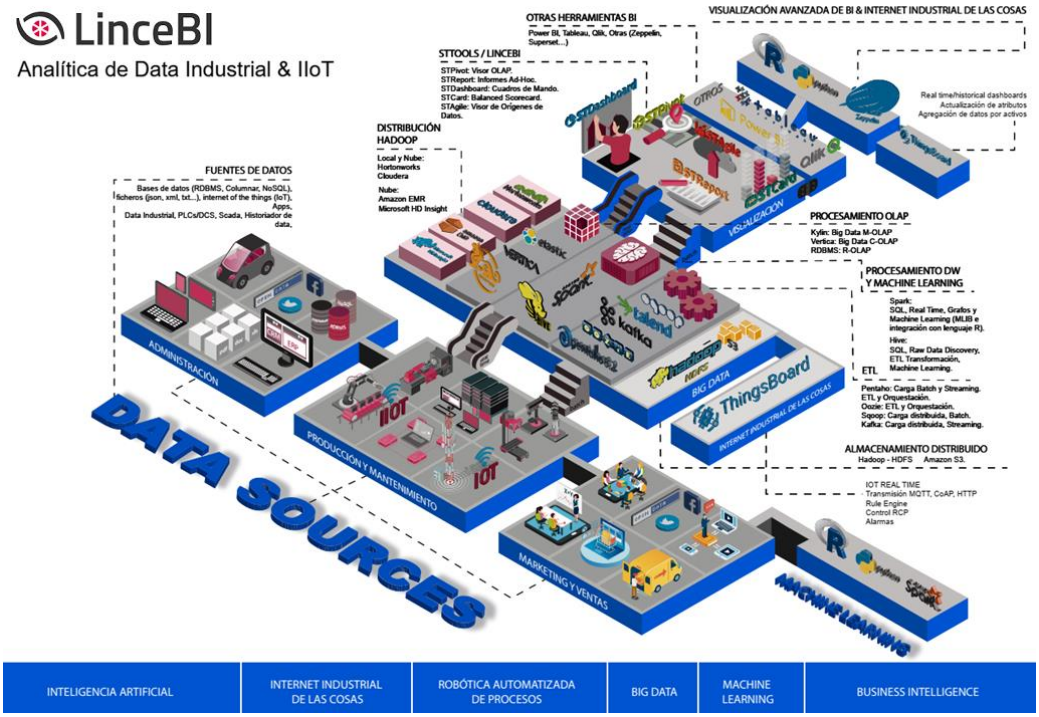
5. CONCLUSIONES

Google Dialogflow permite la creación de chatbots con comprensión del lenguaje natural. Puede analizar texto o audio y generar respuesta en formato texto o con voz sintética. Existen dos versiones, ES que sirve para crear agentes pequeños, medianos o modernamente complejos, y CX que sirve para crear agentes grandes o muy complejos. Son diferentes en cuanto al diseño, ES funciona con intents y contextos, CX con páginas y flujos. En la siguiente versión de este documento veremos Google Dialogflow CX.

6. TECNOLOGÍAS

Recientemente, hemos sido nombrados Partners Certificados de Vertica, Talend, Microsoft, Snowflake, Kylligence, Pentaho, etc.


STTOOLS / LINCEBI
STPivot: Visor OLAP.
STReport: Informes Ad-Hoc.
STDashboard: Cuadros de Mando.
STCard: Balanced Scorecard.
STAgile: Visor de Orígenes de Datos.
OTRAS HERRAMIENTAS BI
Power BI, Tableau, Qlik, Otras (Zeppelin, Superset...)
DISTRIBUCIÓN HADOOP
Local y Nube:
Hortonworks
Cloudera
Nube:
Amazon EMR
Microsoft HD Insight
PROCESAMIENTO OLAP
Kylin: Big Data M-OLAP
Vertica: Big Data C-OLAP
RDBMS: R-OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
Spark:
SQL, Real Time, Grafos y Machine Learning (MLIB e integración con lenguaje R).
Hive:
SQL, Raw Data Discovery, ETL Transformación, Machine Learning.
ETL
Pentaho: Carga Batch y Streaming. ETL y Orquestación.
Oozie: ETL y Orquestación.
Sqoop: Carga distribuida, Batch.
Kafka: Carga distribuida, Streaming.
ALMACENAMIENTO DISTRIBUIDO
Hadoop - HDFS
Amazon S3.
FUENTES DE DATOS
Bases de datos (RDBMS, Columnar, NoSQL), ficheros (json, xml, txt...), internet of the things (IoT), Apps, etc.
VISUALIZACIÓN
BIG DATA
FUENTES DE DATOS


LinceBI
Analítica de Data Industrial & IIoT
OTRAS HERRAMIENTAS BI
VISUALIZACIÓN AVANZADA DE BI & INTERNET INDUSTRIAL DE LAS COSAS
STTOOLS / LINCEBI
DISTRIBUCIÓN HADOOP
FUENTES DE DATOS
PROCESAMIENTO OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
ETL
ALMACENAMIENTO DISTRIBUIDO
ThingsBoard
DATA SOURCES
IIOT
MACHINE LEARNING
INTELIGENCIA ARTIFICIAL
INTERNET INDUSTRIAL DE LAS COSAS
ROBÓTICA AUTOMATIZADA DE PROCESOS
BIG DATA
MACHINE LEARNING
BUSINESS INTELLIGENCE

7. INFORMACIÓN SOBRE STRATEBI

Stratebi es una empresa española, con sede en Madrid y oficinas en Barcelona, Alicante y Sevilla, creada por un grupo de profesionales con amplia experiencia en sistemas de información, soluciones tecnológicas y procesos relacionados con soluciones de Open Source y de inteligencia de Negocio.

Esta experiencia, adquirida durante la participación en proyectos estratégicos en compañías de reconocido prestigio a nivel internacional, se ha puesto a disposición de nuestros clientes.

Somos **Partners Certificados en Microsoft PowerBI** con una dilatada experiencia

Stratebi es la única empresa española que ha estado presente todos los Pentaho Developers celebrados en Europa habiendo organizado el de España.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi **son** profesores y responsables de proyectos **del Master en Business Intelligence de la Universidad UOC, UCAM, EOI…**

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source. Todobi.com

Stratebi es partner de las principales soluciones Analytics: Microsoft Power BI, Talend, Pentaho, Vertica, Snowflake, Kyligence, Cloudera…

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.

8. OTROS

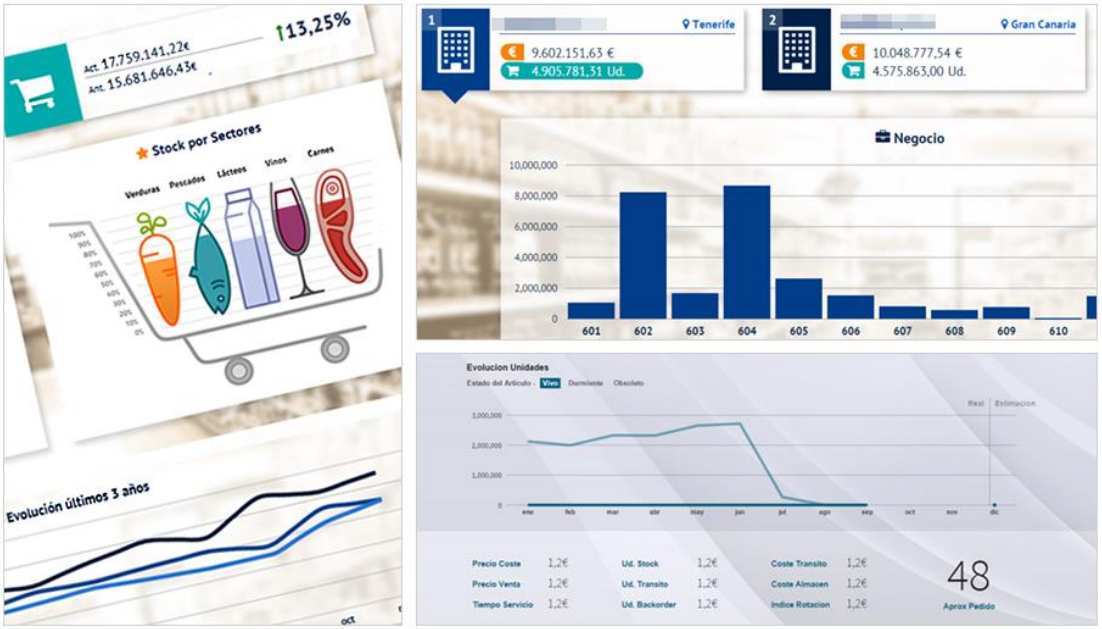
Trabajamos en los principales sectores y con algunas de las compañías y organizaciones más importantes de España.

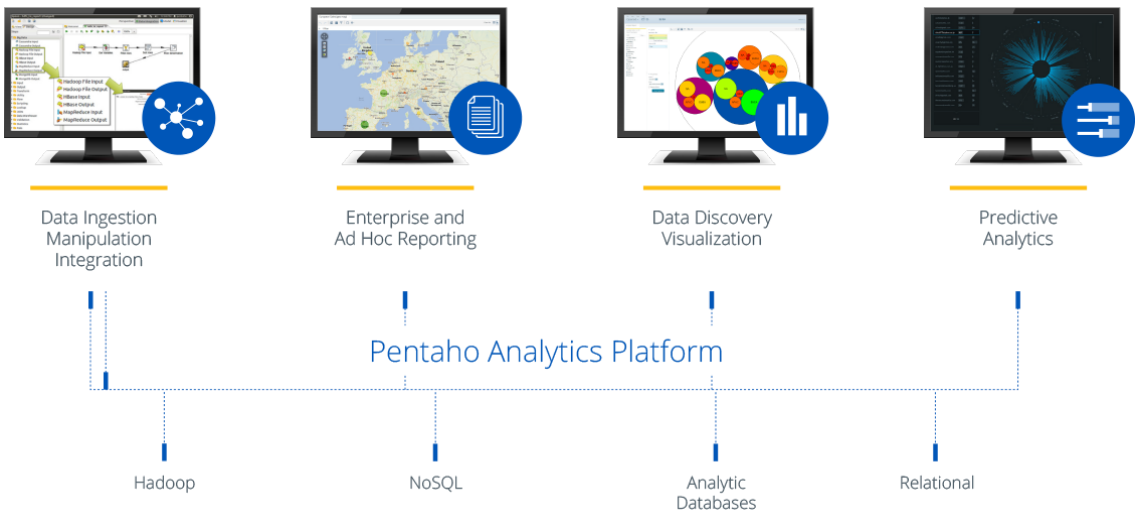
SECTOR PRIVADO

SECTOR PÚBLICO

9. EJEMPLOS DE DESARROLLOS ANALYTICS

A continuación, se presentan **ejemplos de algunos screenshots** de cuadros de mando diseñados por Stratebi, con el fin de dar a conocer lo que se puede llegar a obtener, así como Demos Online en la web de Stratebi:

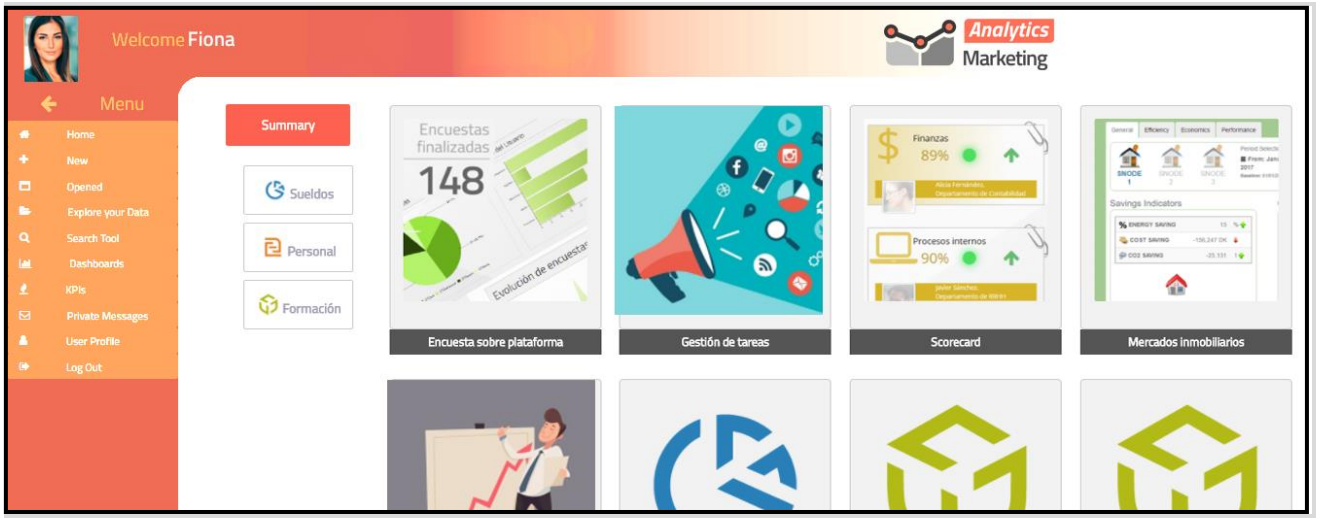





facebook DASHBOARD COMENTARIOS
Sexo
Sentimiento
Localización
65%
35%
28%
60%
12%
Tipo de recurso
Fotos
Videos
Status
Preguntas
Evolución Comentarios por días


smart MADRID
Última Actualización
18:24
TOP 5 PUNTOS DE MEDIDA
PROXIMIDAD DE PUNTOS DE MEDIDA
EVOLUCIÓN INTENSIDAD DE TRÁFICO


Créditos Matriculados
400
Procedencia
FUENTE-ÁLAMO
Alumnos Matriculados: 50
Sexo
50%
50%
Rango de Edad
18-20
20-22
22-24
24-26
26-28
28-30
30-35
35-40
GRADO EN BIOLOGÍA
Créditos Matriculados: 6.500
Buscar Plan de Estudio
INTERÉS PLATAFORMA
181 Inscritos en Cursos
8 Desinscritos en Cursos
181 Inicios de Cursos
110 Inscritos en Eventos Virtuales
63 Inscritos en Debates
181 Inscritos en Talleres
USUARIOS ACTIVOS POR PAÍS
INSCRITOS EN CURSOS POR COMUNIDAD EDU
JORNADA COMPLETA
IDIOMAS
TIC
TIPO DE CENTRO
TIPO DE AUSENCIA
Nº de profesores por sexo
36%
64%
Nº total de profesores
1.587
Media de edad
43 años
Búsqueda
TERRITORIAL › Madrid - Capital


Welcome Fiona
Analytics
Marketing
Menu
Home
New
Opened
Explore your Data
Search Tool
Dashboards
KPIs
Private Messages
User Profile
Log Out
Summary
Sueldos
Personal
Formación
Encuestas finalizadas
148
Evolución de encuestas
Encuesta sobre plataforma
Gestión de tareas
Finanzas
89%
Procesos internos
90%
Scorecard
Savings Indicators
Mercados inmobiliarios