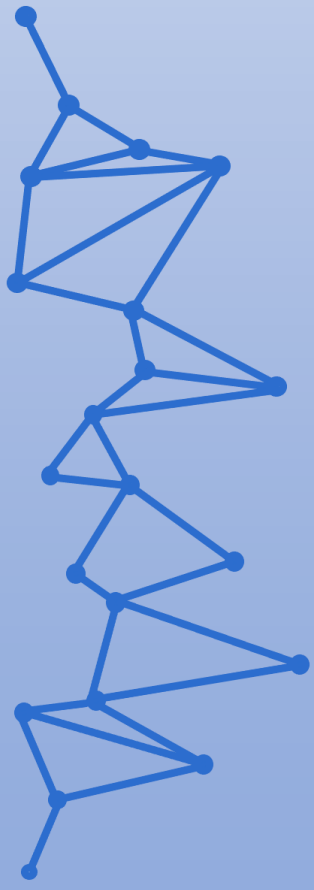
Curso de Especialización de
Inteligencia Artificial y Big Data (IABD)



Programación de Inteligencia Artificial

UD02. Aplicaciones de IA en la nube y servicios API.
Resumen.

JUAN ANTONIO GARCIA MUELAS

En los últimos años han aparecido modelos entrenados para tareas concretas o que se pueden ajustar sus parámetros para el problema a aplicar. En general, estos servicios se utilizan a través de API (Application Programming Interfaces), que viene a ser la forma en que se comunican servidores y aplicaciones cliente o entre aplicaciones dentro del mismo entorno de proyecto.

Destacan estas formas de funcionamiento para las API:

SOAP: se trata de la **más clásica y menos flexible**. La información intercambiada va en **XML**.

RPC: la llamada consiste en que el cliente **ejecuta una función** o procedimiento en el **servidor** y éste **devuelve el resultado.**

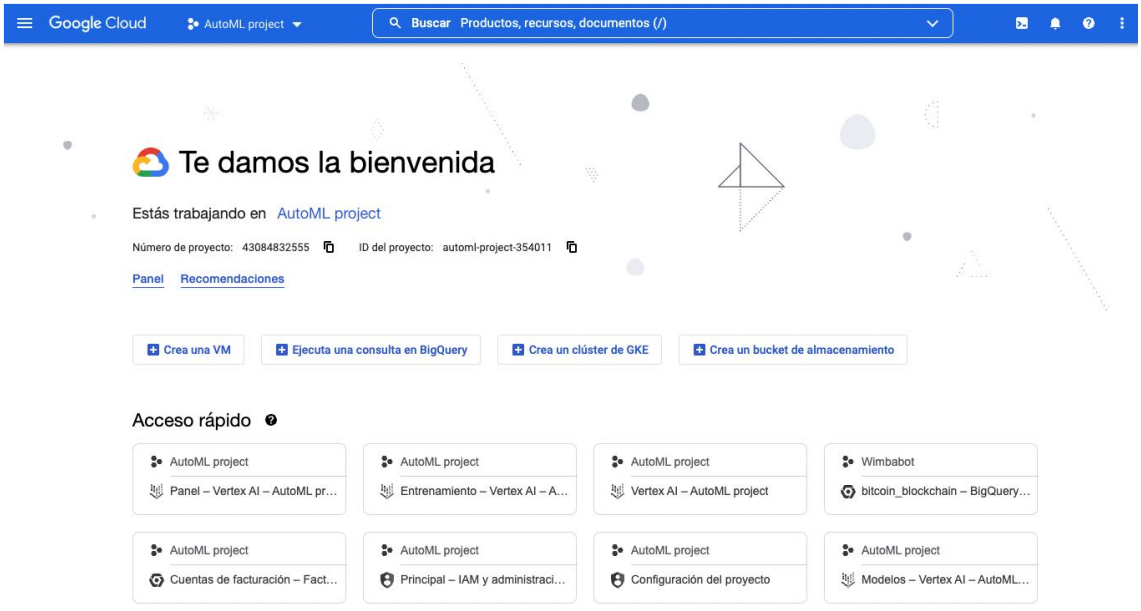
WebSocket: este **protocolo** ha sido creado para la **API web,** y utiliza **objetos JSON** para pasar datos. Es **bastante eficiente** porque se establece una **comunicación bidireccional** entre servidor y cliente.

REST: son **las más utilizadas actualmente, debido a su flexibilidad.** El cliente envía las solicitudes al servidor como datos. Esta entrada hace que el servidor ejecute funciones internas y devuelva los datos de salida a éste.

1. GOOGLE CLOUD PLATFORM

Cuenta con un catálogo de servicios muy extenso, con servicios "llave en mano" o para procesos muy concretos y habituales. Como **Document AI,** que permite el **análisis y extracción** de **información** de un tipo concreto de **documentos**, como facturas o impresos de solicitud de hipotecas, y el usuario apenas tiene que preocuparse de la parte de inteligencia artificial.

Es necesario **registro y crear cuenta de facturación**, aunque la mayoría de los servicios son **gratuitos hasta cierto volumen de datos.** A su **interfaz** se le denomina **consola**.



Consola de Google Cloud Plattform

VertexAI y AutoML.

Vertex AI reúne los servicios de Google Cloud que permiten crear modelos de aprendizaje automático en una interfaz con APIs únicas y unificadas. Se pueden utilizar modelos ya entrenados y listos para usar, o se **pueden entrenar modelos adaptados** a la lógica de negocio concreta **usando AutoML**.

En esta opción, se despliega su propio menú, que representa de forma casi ordenada, las fases que se siguen en cualquier proceso de preparación y despliegue de un modelo de aprendizaje automático. En cada apartado del proceso, será necesario fijar los parámetros propios del proyecto.

Una de sus **ventajas** es que el **proceso**, desde la carga de datos hasta la puesta en producción es **fluido y sencillo**.

Las partes de su "**Panel**", son:
Conjunto de datos: selecciona las fuentes de datos (Google Storage, BigQuery, subida de CSV...) para analizar y generar estadísticas.
Entrenamiento: se configuran parámetros, técnicas a aplicar (supervisado, clasificación, regresión)… Una buena parte, la realiza de forma automática. Una vez concluido el entrenamiento, vamos a la sección…
Modelo: figuran las métricas de desempeño. Para probar el modelo, es necesario crear un "**endpoin**t" o **acceso al modelo**, que se resuelve con un click de botón.

El **aprendizaje supervisado**, es la **clase de aprendizaje automático** que se encarga de los **problemas de regresión y clasificación.**

El aprendizaje **no supervisado y por refuerzo, son técnicas** del aprendizaje supervisado.

VisionAI.

Producto de **reconocimiento de imagen entrenado** genérico que puede ser útil cuando no se cuenta con conjuntos de datos ni experiencia en creación y entrenamiento de modelos. La API de Vision de GCP ofrece:

- ✓ **Reconocimiento facial (faces)**
- ✓ Reconocimiento **de objetos** en una **imagen (Objets)**
- ✓ Identificación de **etiquetas** para una **imagen**
- ✓ Extracción de **texto** de una **imagen**
- ✓ Detección de **elementos no seguros en imagen** (violencia, sexo, racismo, etc)

En el caso de reconocimiento facial, el **modelo detecta los elementos del rostro** y según sus **posiciones relativas**, ha sido entrenado para etiquetar respecto a las **principales emociones**. También detecta **orientación** del **rostro** en la **imagen y objetos** a través de la detección de bordes y formas.

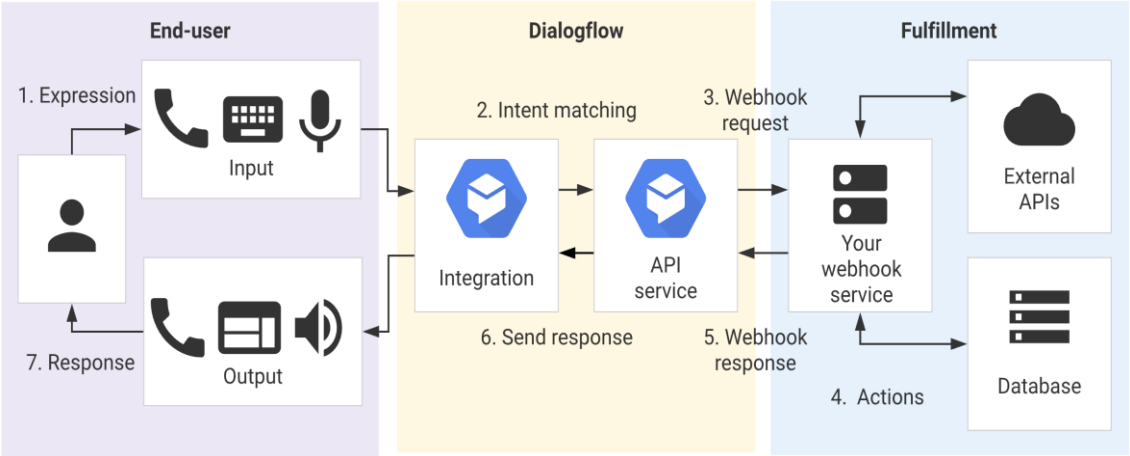
Una utilidad muy extendida para esta API es la de **detección de texto en imagen**. El modelo **reconoce los trazos típicos** de caracteres escritos y lo entrega por bloques.

Si estas funcionalidades no son suficientes hay que recurrir a **AutoML Vision**.

Dialogflow.

Para aplicaciones de comprensión del **lenguaje natural**. Está orientada y optimizada para crear una interfaz de usuario de conversación, con la facilidad extra de poder integrarla en todo tipo de aplicaciones, sistemas automatizados, robots, etc.

Permite la **implementación de Agent Assist**, un complemento para trabajar con agentes humanos **en Atención al Cliente** (ej: reservas de mesa, citas).



GCP DialogFlow. (CC-BY-SA)

Acepta audio y texto. Puede responder con texto y voz sintética. Es parte de **Conversational AI**.

Crear un asistente conversacional requiere, básicamente, dos cosas:

- ✓ **Identificar las intenciones** del usuario cuando hace una pregunta o petición.
- ✓ Una vez interpretada la pregunta e identificada la intención, **recurrir a las respuestas disponibles** para entregar la más adecuada al usuario.

Lo primero, es **crear el agente**, luego **dar de alta los "intents" o intenciones** (saludos, bienvenida, preguntas) y luego los **"utterances" o cuestiones sugeridas** (ejemplos de expresiones que podrían utilizar los usuarios). Se recomienda **entre 10 o 15 por "intent".** Finalmente, definimos las **respuestas**.

2. Amazon Web Services.

Lanzada en **2006**, es una de las más conocidas y utilizadas a nivel mundial. Se accede a través de HTTP, utilizando protocolos REST y SOAP.

Uno de los servicios más valorados para proyectos emergentes y que necesitan escalabilidad, es la **arquitectura sin servidor**, utilizando las denominadas **funciones "Lambda",** lanzado en **2014.**

En el ámbito de la inteligencia artificial, cuenta con un **servicio administrado** para la creación, entrenamiento y despliegue de modelos de aprendizaje automático **denominado SageMaker.**

SageMaker.

Es el entorno integrado para la **creación, entrenamiento y despliegue de modelos** de machine learning **en AWS**. Se gestiona a través de su **panel de control**, llamado **SageMaker Studio y** su **interfaz Canva**. Cuenta con herramientas extra como **Autopilot, que analiza el dataset** y la morfología de los datos objetivo, y automatiza la selección del tipo de modelo y parámetros asociados al entrenamiento. Pero este servicio tiene un coste mayor que un proyecto normal.

AWS provee de un entorno para pruebas y aprendizaje con las **funcionalidades más básicas y recursos de computación de forma gratuita**. Se trata de **SageMaker Studio Lab**. Es necesario registrarse y hay ciertos límites de uso, pero permite utilizar proyectos de prueba de aprendizaje automático dentro de otros proyectos que se tengan desplegados en AWS.

Rekognition.

Es el servicio de **reconocimiento de imagen de AWS ya pre-entrenado y desplegado,** que se puede probar a través de una interfaz demostrativa, y que se puede integrar en cualquier proyecto mediante llamadas a su API.

Tiene un primer modo, de reconocimiento de "etiquetas", que muestra los objetos que coincidan con ellas. El modo de moderación (contados sensibles). El de reconocimiento facial, y el modo Rekognition para identificar a determinadas personas o determinado texto (ejemplo: matrículas). Se puede revisar el resultado mediante el objeto JSON devuelto con los datos.

Comprehend.

Módulo para procesamiento de lenguaje natural.

Recuerda que el procesamiento de lenguaje natural (NLP) es el **conjunto de técnicas computacionales en** el ámbito del aprendizaje automático, que tienen como objetivo **identificar la intención del interlocutor**, contenida en una expresión **escrita,** o **hablada y transformada a texto escrito**, y **clasificarla o asociarla** a opciones o valores de variables de salida.

Divide el texto en unidades con sentido y las analiza.

Una función muy utilizada en este tipo de modelos es el **análisis de sentimiento**, que nos puede dar un valor aproximado del **grado de positividad o negatividad** que hay en un texto.

3. IBM Cloud.

Watson tiene varias herramientas disponibles, entre las que encontramos **módulos pre-entrenado y listos para integrar**, vía API en una aplicación o proyecto, así como entornos en los que trabajar con modelos de aprendizaje automático a bajo nivel.

La **herramienta de trabajo básica** de IBM para aprendizaje automático es **Watson Studio.** Facilita el desarrollo con un **entorno gráfico** al estilo **Node.js**, pero **también permite** utilizar un entorno de **tipo Notebooks.**

IBM Watson natural Language Understanding, en su extensión de análisis de texto, logra un resultado muy completo, tanto en la fase de extracción y clasificación de entidades y expresiones clave, como en la fase de interpretación de emociones para cada expresión clave.

4. Microsoft Azure.

Cognitive services.

Es la suite de modelos listos para usar de **Microsoft.** Implementa servicios de reconocimiento de **voz, texto**, comprensión de **lenguaje natural,** reconocimiento de **imagen** y analítica avanzada de datos para toma de decisiones.

AzureML.

Es el servicio para **crear y entrenar modelos de cero**, con la facilidad de conectarlos al resto de servicios necesarios en un proyecto de forma eficiente e integrada. Cuenta con un escritorio específico de trabajo, denominado Studio, en el que se puede iniciar y gestionar los modelos, sus implementaciones, métricas e historiales de ejecución.

Permite trabajar con cuadernos **Jupyter Notebook**, utilizar librerías para aprendizaje automático y la flexibilidad de **configurar los hiperparámetros** en el código. Cuenta con una

herramienta de depurado y **permite la integración con** el entorno de programación **Visual Studio Code.**

5. Open AI.

Aunque reciente, su **API basada en el algoritmo GPT-3 ha escalado muy rápidamente** y es muy valorada para proyecto de AI.

El modelo está desplegado como una herramienta de auto-completado, básicamente.

Se trata de un modelo generador de textos, que puede completar o crear el fragmento de texto que se le solicite, con una petición en lenguaje natural.

Los **parámetros de entrada** en la API son:

- ✓ **Prompt**: es la más importante. Es la **instrucción** a partir de la cual se configura el completado o respuesta
- ✓ **Temperatura**: es el **margen de "riesgo" o libertad que se le da al modelo** para ser más creativo o riguroso. Es un **valor entre 0 y 1** en el que el valor de 0 representa ningún margen de libertad. Con la temperatura a 0, el modelo responde de forma muy determinista y nos dará la misma respuesta para la misma entrada.

En la documentación se puede consultar cómo hacer las llamadas a la API desde una app o backend, pasando como parámetros principales:

- ✓ **model**: el modelo que se quiere usar. **Davinci 2 es el más potente**, pero también el más caro. Lo recomendable es empezar con él y tras ver de lo que es capaz, ir probando con los otros modelos más especializados si logramos más o menos un buen desempeño para nuestra aplicación. Los **otros modelos para NLG son Curie, Babbage y Ada**. Sus características se pueden consultar aquí.
- ✓ **prompt**: la instrucción de entrada.
- ✓ **temperature:** como hemos comentado antes, el valor entre 0 y 1 de la flexibilidad creativa que le permitimos al modelo.
- ✓ **max_tokens**: el número máximo de tokens que queremos que se generen. Los tokens son **conjuntos de caracteres consecutivos** que constituyen la unidad de trabajo del modelo. **La relación entre tokens y palabras es de 3 a 4 (100 tokens equivalen a unas 75 palabras).**