

Abstract geometric lines in black on a white background, forming various overlapping polygons and shapes.

PROMETEO
by thePower

TALLER DE N8N
DÍA 1

AGENDA DÍA 1

Introducción a n8n

Instalación y configuración

Empezando con n8n

Automatización de procesos con n8n

Agentes de Inteligencia Artificial

--- DESCANSO ---

Bases de datos y Agentes IA

Memoria de los Agentes de IA

OBJETIVOS

Objetivos del Curso

Crear agentes de IA con n8n

Aprovechas todo el potencial del nocode, la automatización y la IA

Detectar oportunidades para desarrollar agentes de IA en tu entorno de trabajo actual.

¿QUÉ ES N8N?

N8n es una plataforma para crear agentes de IA sin necesidad de programar. Interactuar con usuarios, procesar la información, etc.

Tendencias:

NoCode

Automatización

Inteligencia Artificial

Triger primer nodo que activa como

Acción nodo que recibe una tarea y da una respuesta, por ejemplo, con un modelo OpenIA

Workflow Por ejemplo recibir una consulta, se procesa y analiza por un agente y realiza una acción como respuesta.

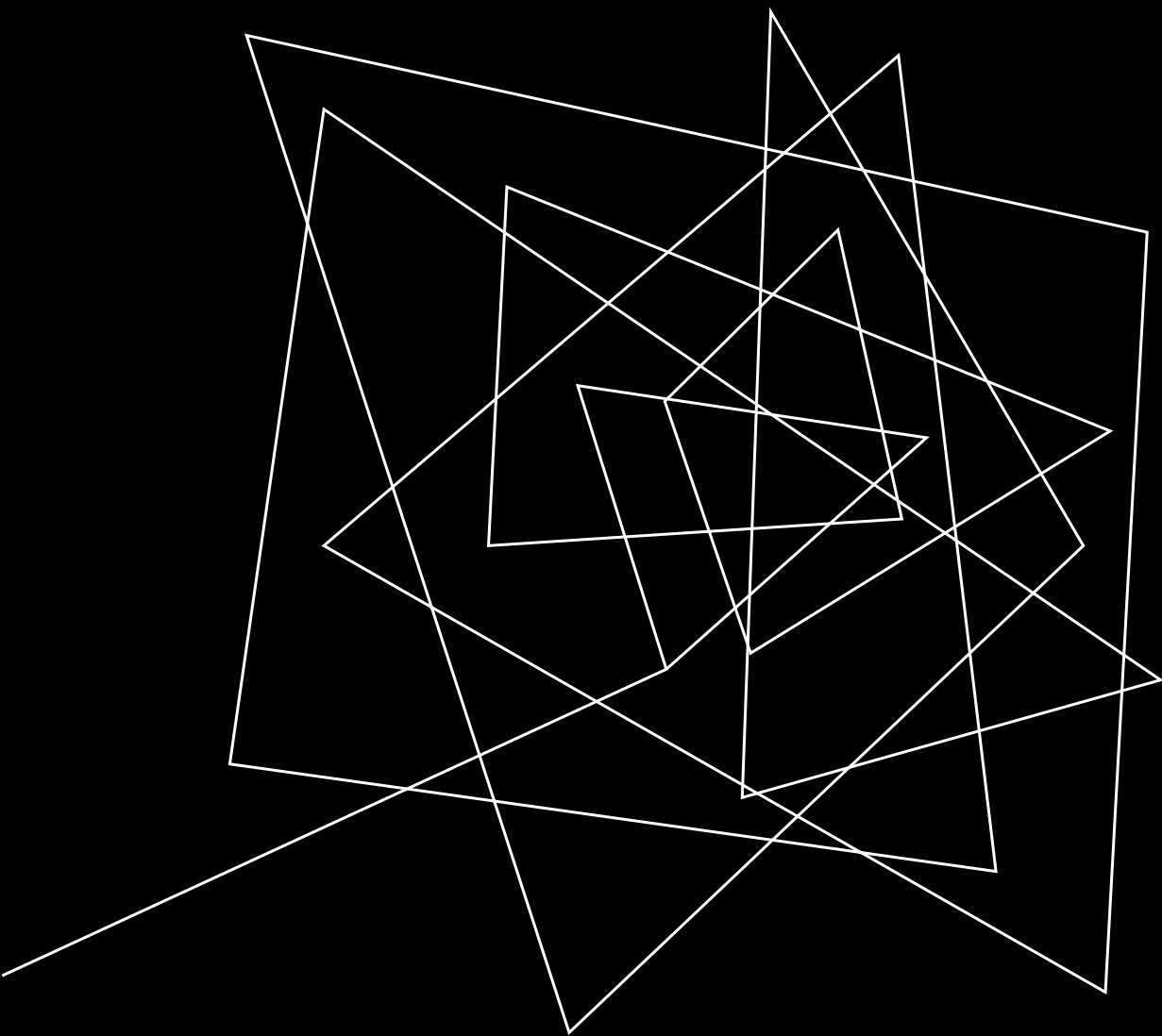
¿QUÉ ES N8N?

N8n es una plataforma para crear agentes de IA sin necesidad de programar. Interactuar con usuarios, procesar la información, etc.

Tendencias:

- NoCode
- Automatización
- Inteligencia Artificial

- Trigger primer nodo que activa
- Acción nodo que recibe una tarea y da una respuesta, por ejemplo, con un modelo OpenIA
- Workflow Por ejemplo recibir una consulta, se procesa y analiza por un agente y realiza una acción como respuesta.



SISTEMAS RAG



TÉCNICAS DE ENTREGA EFECTIVAS

Definición

RAG significa **Retrieval-Augmented Generation**.

Es una técnica que **combina recuperación de información con generación de texto** mediante modelos de lenguaje como GPT.

Retrieval (Recuperación):

Se hace una búsqueda (por ejemplo, en Pinecone, base de datos, etc.) usando embeddings para encontrar fragmentos de texto relevantes.

Augmented (Contextualizado):

Los textos recuperados se insertan como contexto en el prompt del modelo generativo (por ejemplo, GPT-4).

Generation (Generación):

El modelo genera una respuesta más precisa porque ahora tiene datos externos relevantes, incluso si no estaban en su entrenamiento.



EMBEDDINGS

Definición

¿Qué es un embedding?

Un embedding es una representación matemática (un vector, es decir, una lista de números) que transforma un texto, palabra o incluso imagen en algo que una máquina puede entender midiendo significado, no solo forma.

En otras palabras: convierte un texto en números que conservan su significado semántico.

¿Para qué sirven los embeddings?

Búsqueda semántica: encontrar cosas aunque no uses las mismas palabras.

Comparar similitud de textos, usuarios, productos...

Clasificación de contenido.

Chatbots con memoria y contexto.

Indexar en Pinecone, Weaviate, RedisVector, etc.

Recomendaciones personalizadas (tipo "Netflix", "Spotify").



EMBEDDINGS

Definición

Supón que tienes tres frases:

"Hola, ¿cómo estás?"

"¿Qué tal te encuentras?"

"Necesito un taxi"

Cuando generas sus embeddings, podrías obtener vectores así (simplificado):

Frase 1 → [0.12, -0.89, 0.45, ...]

Frase 2 → [0.13, -0.91, 0.44, ...]

Frase 3 → [0.87, 0.22, -0.65, ...]

TOKENS

¿Qué es un token?

Un token es una unidad mínima de texto que un modelo de lenguaje usa para procesar el lenguaje.

⚠ No es igual a una palabra ni a un carácter.

Puede ser una palabra, parte de una palabra o incluso solo un signo de puntuación.

¿Cuántos tokens ocupa un texto?

Como regla general aproximada:

1 token \approx 4 caracteres en inglés

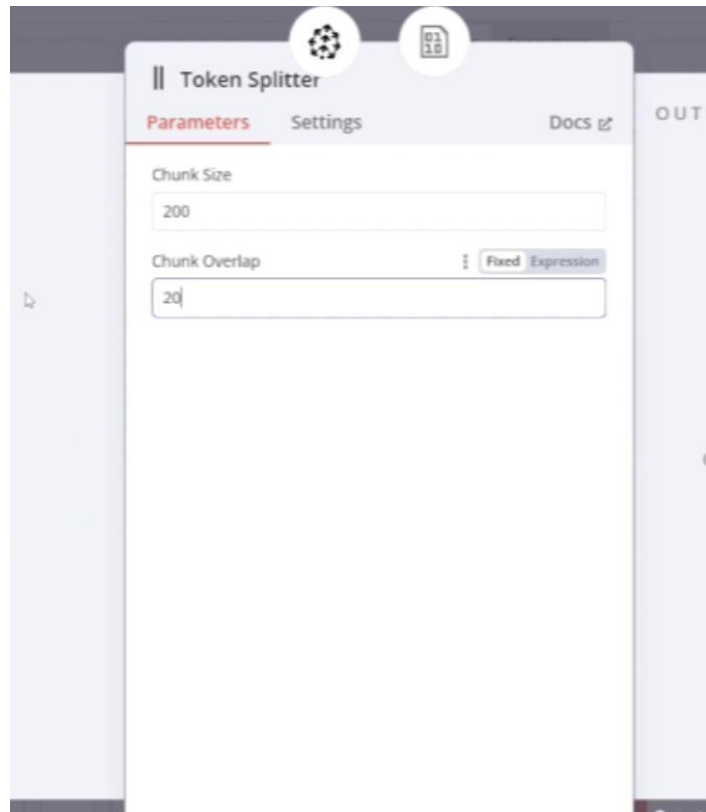
100 tokens \approx 75 palabras en español (aprox.)

Ejemplo:

"Este es un ejemplo sencillo para explicar los tokens."

→ Son unas 10 palabras → ~15 tokens

TOKENS



Token Splitter

Nodo Token Splitter de n8n, usado típicamente para dividir texto largo en fragmentos más pequeños antes de procesarlos con modelos de lenguaje (por ejemplo, embeddings, GPT, etc.).

Parámetros clave:

Chunk Size: 200

Esto significa que el texto se dividirá en bloques de 200 tokens.

No son caracteres ni palabras, sino tokens, que pueden ser palabras, partes de palabras o incluso signos de puntuación (según el modelo que uses, como GPT-3.5 o GPT-4).

Chunk Overlap: 20

Esto define que cada fragmento tendrá una superposición de 20 tokens con el anterior.

Esto se hace para que no se “pierda contexto” entre fragmentos (por ejemplo, que una frase no se corte bruscamente y se pierda sentido).

Ejemplo:

Supón que tienes un texto de 600 tokens. Con Chunk Size = 200 y Overlap = 20, se dividiría así:

Chunk 1: tokens 0–199

Chunk 2: tokens 180–379

Chunk 3: tokens 360–559

Como ves, el inicio de cada nuevo bloque se solapa 20 tokens con el anterior.