

Manual de Instalación del código

Concept Curve Embeddings Indexation

Inteligencia Artificial con memoria ilimitada

Introducción:

Este código permite crear aplicaciones de IA para analizar documentos completos, como manuales operativos, documentos médicos y legales, sin necesidad de entrenamiento de la IA, entregando resultados perfectos sin alucinaciones y a un costo mínimo.

No se utiliza R.A.G. ni embeddings, no requiere espacios de almacenamiento y es compatible con cualquier modelo LLM, sea grande o pequeño, en modo online u offline, sea tecnología presente o futura.

Estas capacidades se logran mediante el paradigma de **Curva de Conceptos** desarrollada por **Daniel Bistman**.

Para obtener toda la información relacionada al paradigma de la Curva de Conceptos y su aplicación en software real, consulte al Agente CC, una de las personalidades de ChatGPT diseñada para explicar este método: <https://tinyurl.com/agente-cc>

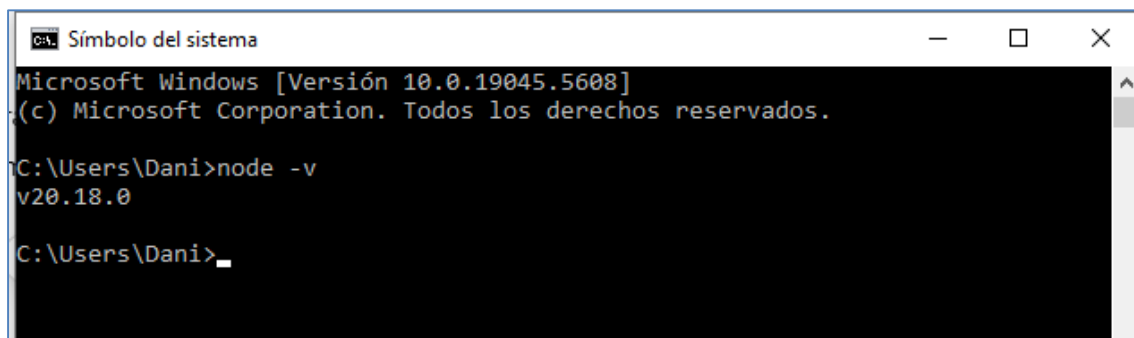
Sección 1 – Instalación (Windows, Linux y Mac)

1. Requisitos del sistema

Para ejecutar esta aplicación, solo necesitas instalar algunos paquetes en tu sistema. Asegúrate de cumplir con los siguientes requisitos:

Paso 1.1: Node.js (versión 16 o superior)

Para verificar si ya está instalado, ejecuta en la terminal de Windows el comando `node -v`




```
Microsoft Windows [Versión 10.0.19045.5608]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dani>node -v
v20.18.0

C:\Users\Dani>
```

Si no está instalado, puedes descargarlo ingresando a <https://nodejs.org/> y haciendo click en el botón:

Descargar Node.js (LTS) 

En Debian/Ubuntu:

```
sudo apt update && sudo apt install nodejs npm
```

En Mac (usando Homebrew):

```
brew install node
```

Paso 1.2: Paquetes de Node.js utilizados

La aplicación requiere los siguientes paquetes de Node.js, los cuales se instalarán automáticamente más adelante en el proceso de instalación:

- axios: Para hacer solicitudes HTTP.
- body-parser: Para procesar datos en las solicitudes.
- cors: Para manejar permisos de acceso entre dominios.
- dotenv: Para gestionar variables de entorno.
- express: Framework para crear el servidor backend.
- marked: Para convertir texto Markdown a HTML.
- pdf-parse: Para extraer texto de archivos PDF.
- tesseract.js: Para reconocimiento óptico de caracteres (OCR).

No es necesario instalarlos manualmente, ya que se instalarán automáticamente en el siguiente paso.

2. Instalación

2.1 Descargar el código fuente

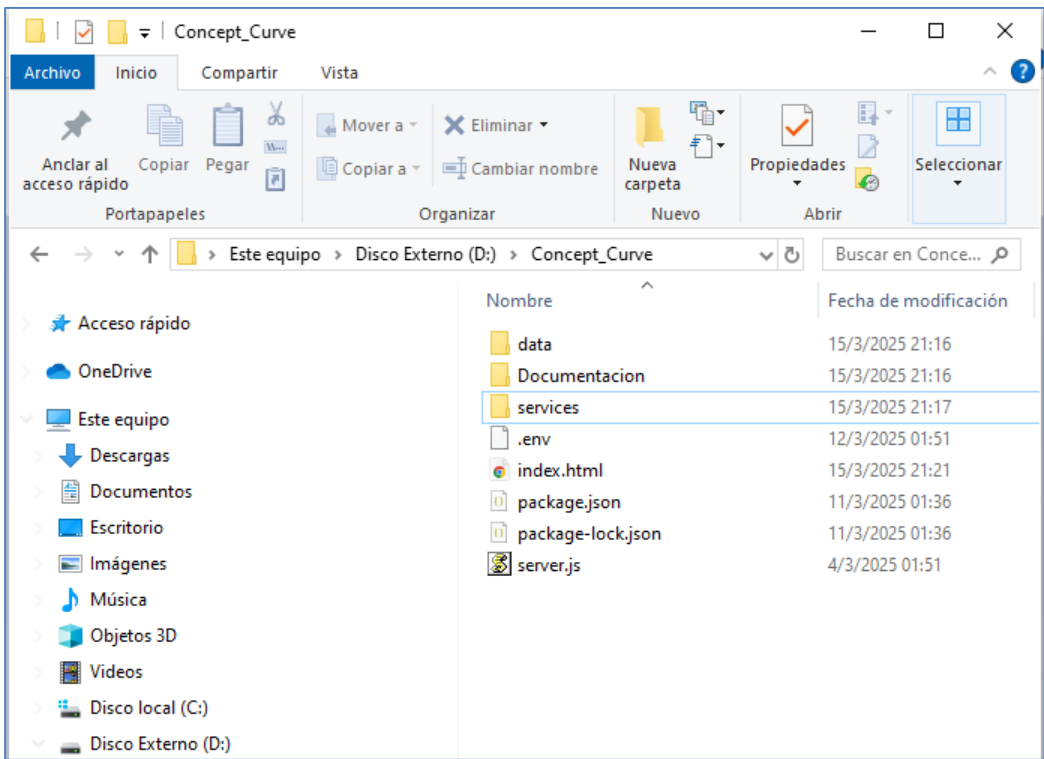
Extrae el archivo ZIP en la carpeta de tu elección. Para esta versión de demostración en español se han entregado dos versiones del software precargado:

Versión 1 - `conceptCurve_spanish_Biblia.zip`: Los primeros seis libros de la Biblia (BTX Biblia Textual), un documento con alrededor de 250 mil tokens divididos en 211 chunks.

Versión 2 - `conceptCurve_spanish_CCCN.zip`: La totalidad del Código Civil y Comercial de la Nación Argentina, dividido en 37 chunks con unos 220 mil tokens, el cual ChatGPT no puede leer ni sobre cual está entrenado.

(Más información al respecto se encuentra en la Sección 2 – Configuraciones)

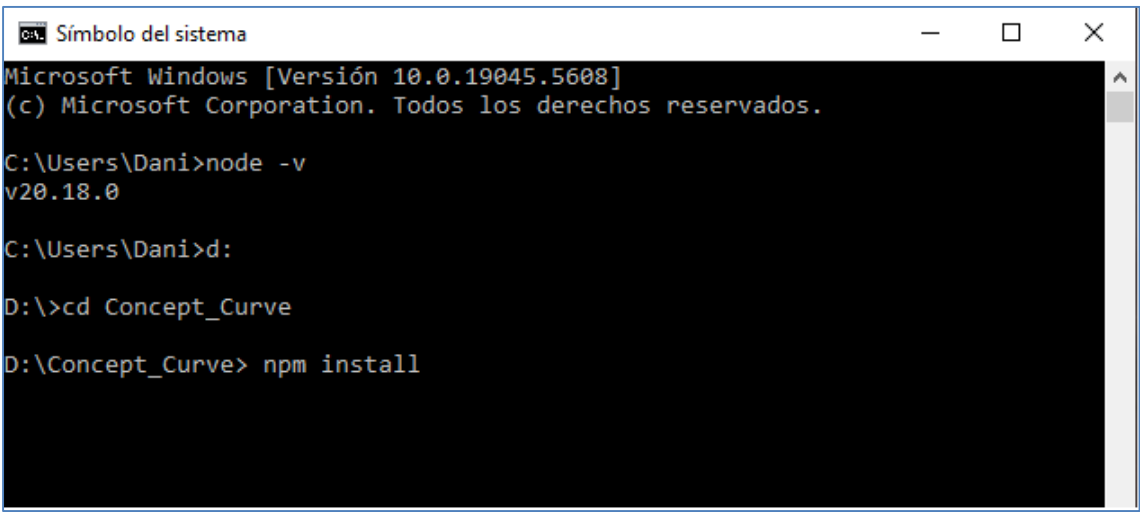
Luego de extraer los archivos en el directorio de tu elección, la estructura de directorios quedará como se ve en la figura.



De esta instalación, aún quedan por agregar las dependencias de Node.js las cuales luego de ejecutar el siguiente paso, aparecerán dentro de un directorio llamado “node_modules”.

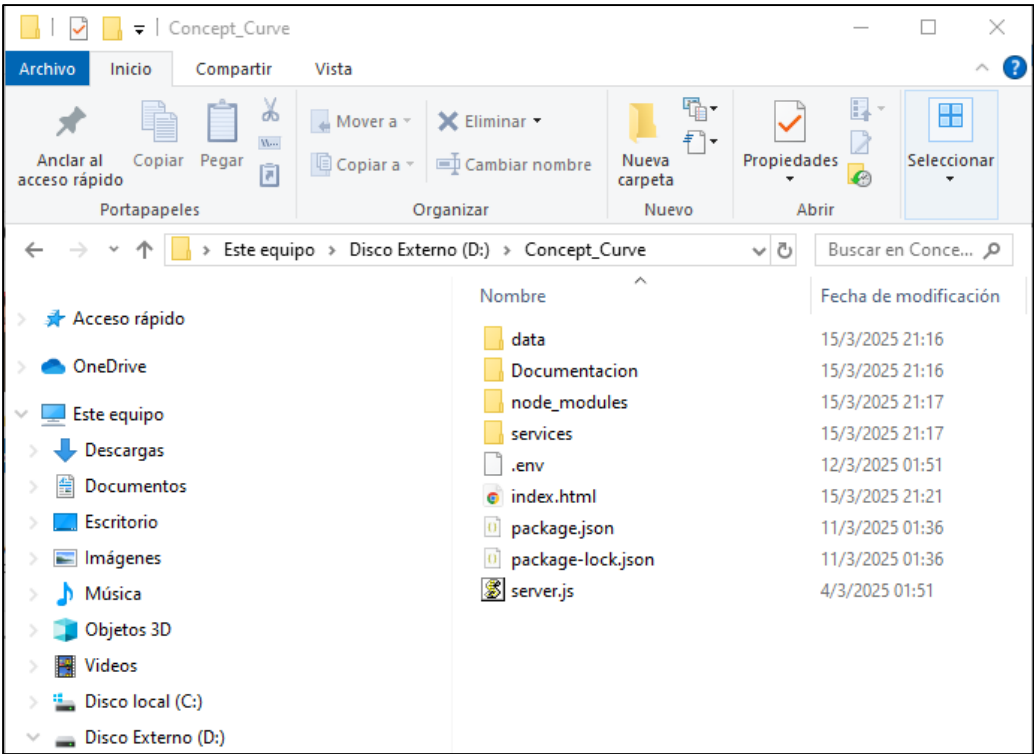
2.2 Instalar dependencias

Navega hasta la carpeta del proyecto, el mismo directorio que contiene el archivo package.json y ejecuta el comando `npm install` para instalar las dependencias necesarias.



2.3. Estructura de archivos final

Luego de instaladas las dependencias de Node.js, la estructura final de archivos quedará como se ve en la figura.



El directorio **“data”** contiene los chunks del documento a consultar, el archivo **“_Document_Index.txt”** y el archivo **“_Context_Index.txt”**

El directorio **“Documentacion”** contiene la documentación del código y los manuales de operación para instalar y configurar el software.

El directorio **“node_modules”** contiene todas las dependencias y bibliotecas necesarias para la ejecución del software.

El directorio “services” contiene la implementación del algoritmo principal (consulta.js), junto con funciones auxiliares (functions.js y smartFunctions.js). Se encarga del procesamiento de datos, análisis de consultas y optimización de respuestas en el backend.

El archivo “.env” es el archivo de configuración que almacena las variables de entorno necesarias para el funcionamiento de la aplicación, como la API Key de OpenAI. Es importante completar este archivo antes de ejecutar el software.

El archivo “index.html” es la interfaz principal de la aplicación (frontend). Puede abrirse en un navegador para acceder a la funcionalidad del software.

El archivo “package.json” contiene la información del proyecto, incluyendo sus dependencias, scripts y metadatos. Es un archivo clave para gestionar el entorno de desarrollo y producción.

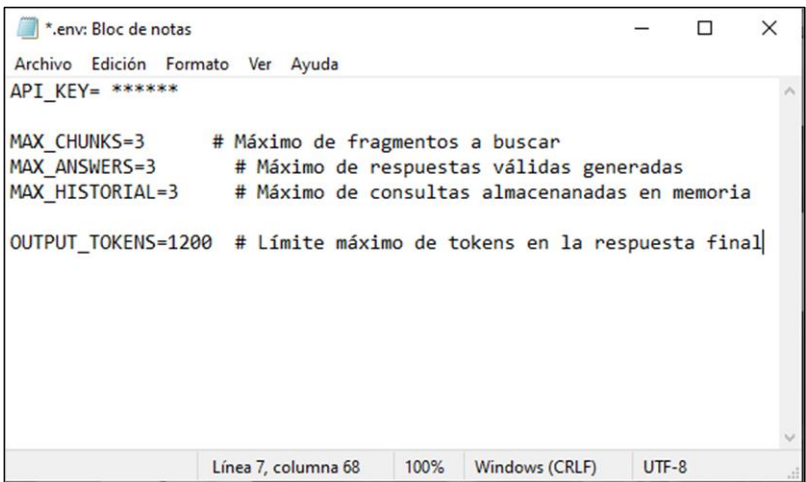
El archivo “package-lock.json” asegura que las versiones de las dependencias sean consistentes entre diferentes instalaciones, evitando incompatibilidades.

El archivo “server.js” es el punto de entrada del backend. Gestiona las solicitudes y la comunicación con el frontend.

2.3. Configuración

Antes de ejecutar la aplicación debes tener una API Key contratada del proveedor OpenAI.

Verifica el archivo “.env” de configuración, el cual se encuentra en el directorio raíz. En el caso de este instructivo D:\Concept_Curve> y coloca en ese archivo la API Key de OpenAI



Para aprender cómo contratar una API Key, se pueden consultar videos tutoriales o directamente al asistente Agente CC ---> <https://tinyurl.com/agente-cc>

2.4. Problemas de Instalación – Troubleshooting

Si encuentra problemas durante la instalación, asegúrese de haber seguido correctamente los pasos anteriores teniendo actualizadas las dependencias Node.js, la API_KEY contratada, y actualizada en el archivo “.env”.

Si los problemas persisten, puede buscar soluciones en la documentación oficial de Node.js o consultar a un asistente de IA como ChatGPT para resolver errores específicos.

3. Ejecutar la aplicación

3.1 Ejecutar la aplicación Backend

Para iniciar la aplicación, usa el siguiente comando en la terminal: node server.js

```
Símbolo del sistema - node server.js
Microsoft Windows [Versión 10.0.19045.5608]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dani>node -v
v20.18.0

C:\Users\Dani>d:

D:\>cd Concept_Curve

D:\Concept_Curve>node server.js
Servidor escuchando en http://localhost:3000
```

En este punto, Windows Firewall solicitará permiso para permitir el acceso a Internet del archivo `server.js`. Esto se debe a que dicho archivo funciona como servidor backend y necesita conectarse a la API de OpenAI para realizar consultas.

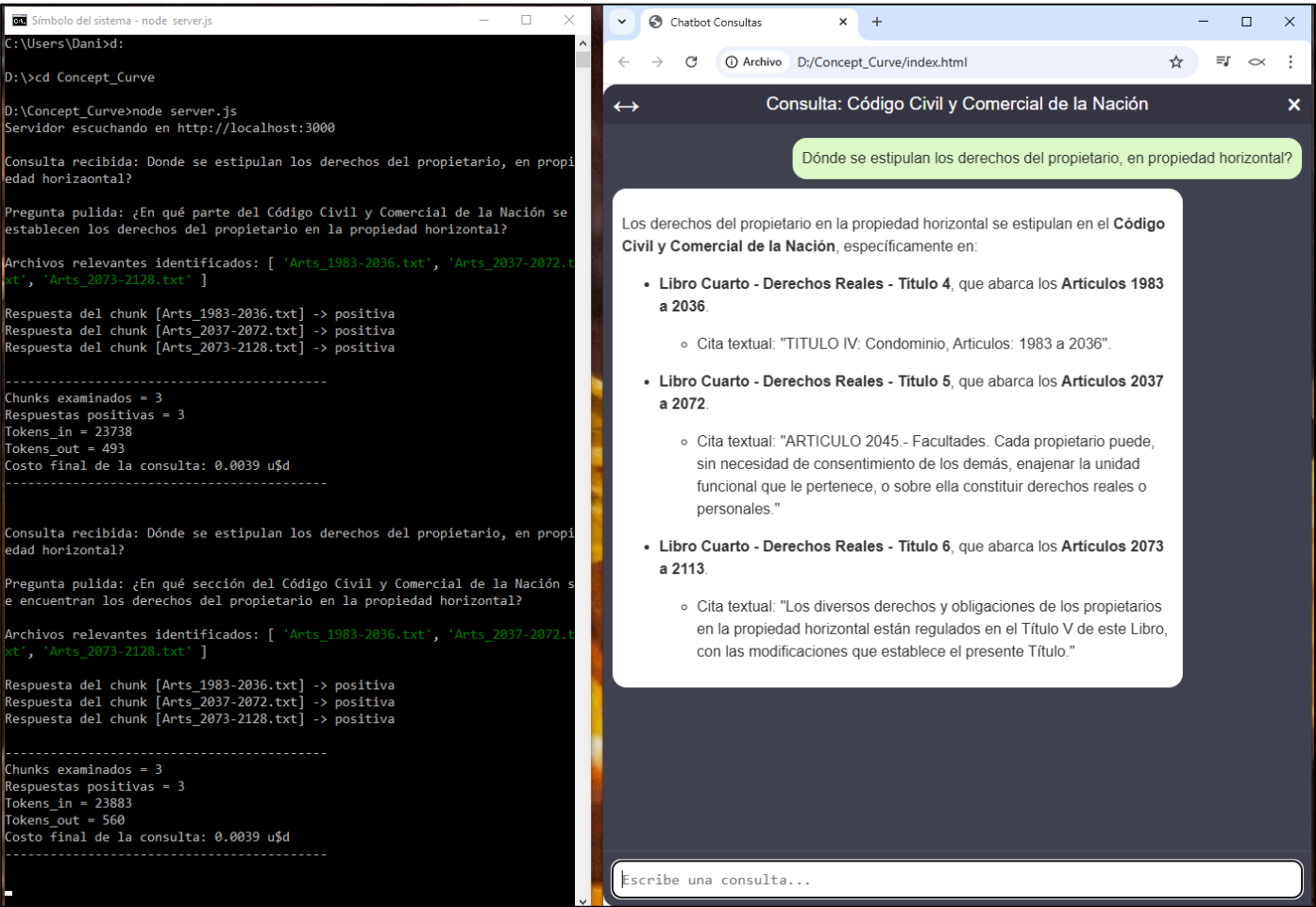
3.2. Ejecutar la aplicación Frontend

Ve al explorador de Windows y ejecuta el archivo `index.html` haciendo doble click en el archivo. Esto iniciará la interfaz de la aplicación, desarrollada en HTML.

En Linux y Mac

```
xdg-open index.html # En Linux
```

```
open index.html # En Mac
```



Listo! La aplicación ya está funcionando.