

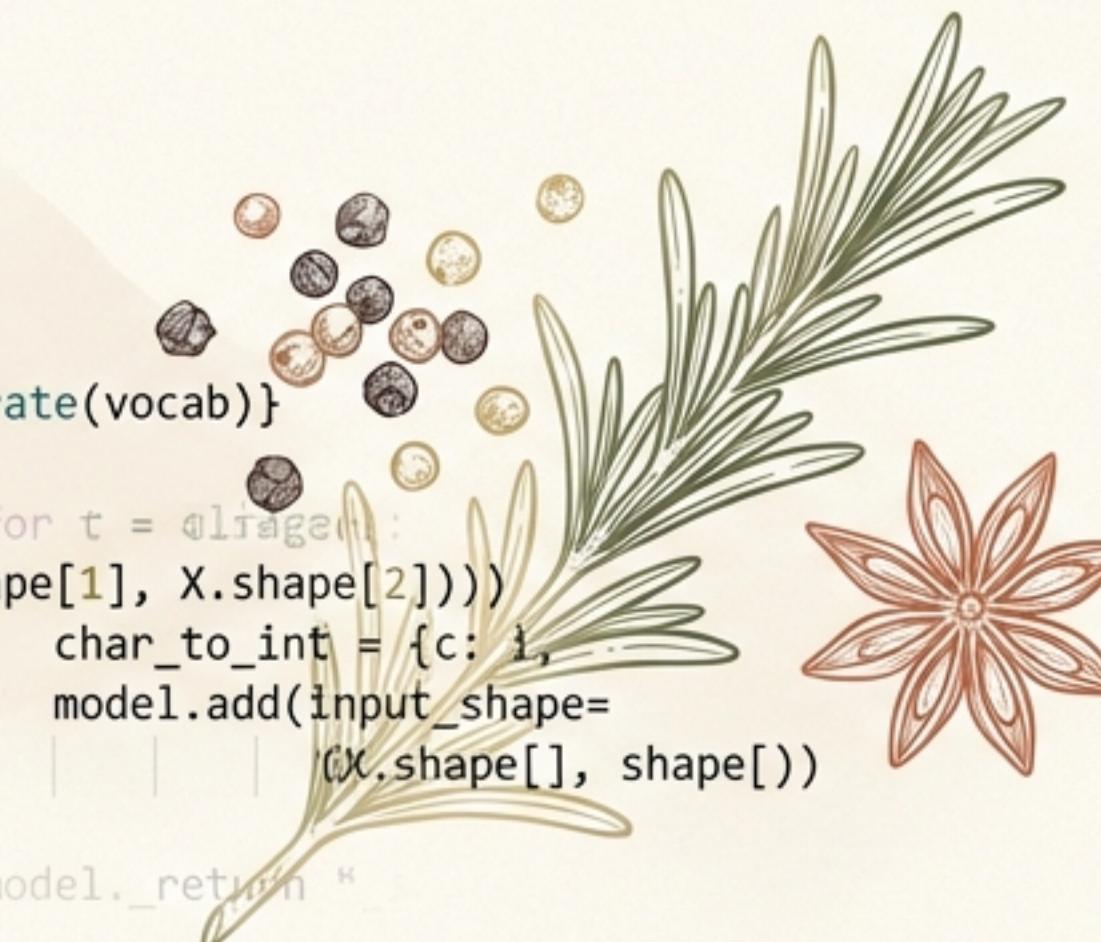
Cocinando una IA: La Receta del Texto Predictivo

Un recorrido paso a paso, desde los datos crudos hasta la primera predicción.

```
import numpy as np

data = open("corpus.txt", "r").read()
vocab = sorted(list(set(data)))
char_to_int = {c: i for i, c in enumerate(vocab)}

seq_length = 100
model.add(LSTM(256, input_shape=(X.shape[1], X.shape[2])))
for t = 0 to len(data) - seq_length:
    char_to_int = {c: i,
    model.add(input_shape=
              (X.shape[], shape[]))
    model._return *
```



El objetivo: una IA que completa tus frases.



Construiremos un sistema desde cero que **aprende patrones** del lenguaje para predecir la siguiente **palabra** más probable en una secuencia. Este es el principio fundamental detrás del **autocompletado** en tu teclado.

Paso 1: Reunir los ingredientes (Las librerías).



El ingrediente principal. La biblioteca de Google para construir el “cerebro” de nuestra IA. Nos da las piezas (capas) y las instrucciones para ensamblarlas.



Nuestras herramientas de cocina. `Pandas` nos ayuda a manejar los datos como si fueran hojas de cálculo, y `NumPy` es esencial para las operaciones matemáticas complejas que la IA necesita.



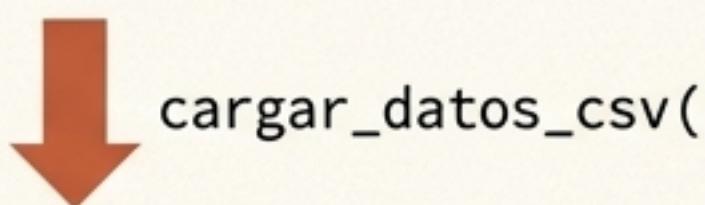
El traductor y el envase. `Tokenizer` convierte palabras en números (el único idioma que entiende la IA) y `Pickle` nos permite guardar este “diccionario” para usarlo más tarde.



Paso 2: La preparación (Limpieza de los datos).

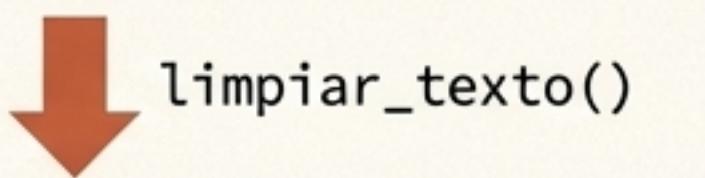


```
ARCHIVO_DATOS = 'csv/texto_predictivo.csv'
```



```
cargar_datos_csv()
```

Leer y eliminar nulos/duplicados. No queremos confundir a la IA con datos sucios o repetidos.



Convertir a minúsculas (`lower()`) y eliminar puntuación (`re.sub`). Esto estandariza nuestro ingrediente principal: el texto.



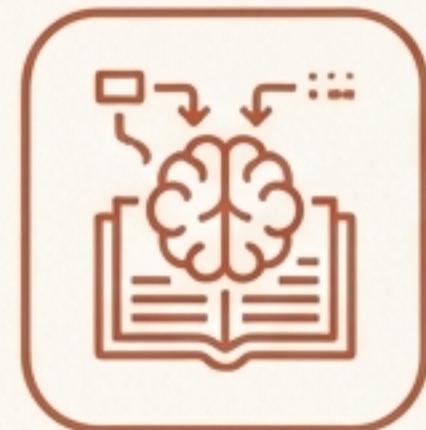
```
['hola como estas', 'estoy bien gracias', ...]
```

```
def limpiar_texto(text):
    text = text.lower()
    text = re.sub(r'^\w\s+', '', text)
    return text
```

Paso 3: Traducir la receta al idioma de la IA.

El `Tokenizer`: Nuestro diccionario palabra <-> número.

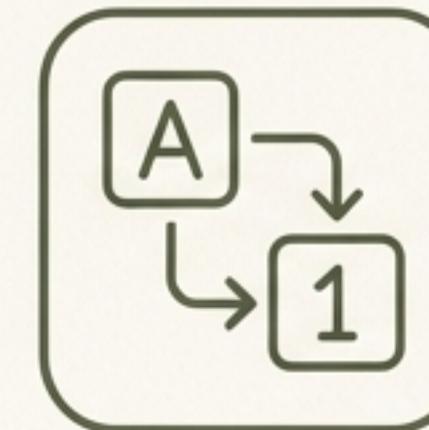
Parte 1: El Aprendizaje (`fit_on_texts`)



Primero, el `Tokenizer` lee todas nuestras frases para crear un vocabulario único. A cada palabra se le asigna un número.

```
{'hola': 1, 'como': 2, 'estas': 3, 'gracias': 4, ...}
```

Parte 2: La Traducción (`texts_to_sequences`)



Luego, usa este diccionario para convertir cada frase en una secuencia de números.

"hola como estas" —————→

```
[1, 2, 3]
```

La IA solo entiende de números. Este paso es fundamental para que pueda procesar el lenguaje.

Creando las lecciones: de frases a ejemplos de entrenamiento.

****Frase Original**:**
`nos vemos luego`



****Forma Tokenizada**:**
`[5, 6, 7]`

Lección 1

A partir de `[5]` (nos), la IA debe aprender a predecir `6` (vemos).



Lección 2

A partir de `[5, 6]` (nos vemos), la IA debe aprender a predecir `7` (luego).



Texto explicativo: No le enseñamos a la IA frases enteras. La descomponemos en secuencias progresivas. Cada secuencia se convierte en un par de '**contexto**' (X) y '**respuesta correcta**' (Y). Así, la IA aprende a predecir la siguiente palabra en cualquier punto de una frase.

La uniformidad es clave para el ‘horno’ matemático.

Izquierda: Relleno (pad_sequences)

El Qué

Añadimos ceros a la izquierda de las secuencias más cortas.

El Porqué

El modelo es una matriz matemática. Exige que todas las entradas (filas) tengan exactamente la misma longitud. El padding='pre' asegura esta consistencia.

$$\begin{array}{ccc} [5, 6] & \longrightarrow & [0, 5, 6] \\ [8, 9, 10] & \longrightarrow & [8, 9, 10] \end{array}$$

Derecha: Categorización (to_categorical)

El Qué

Convertimos la palabra de salida (ej: '3') en un vector de ceros con un '1' en la posición correcta.

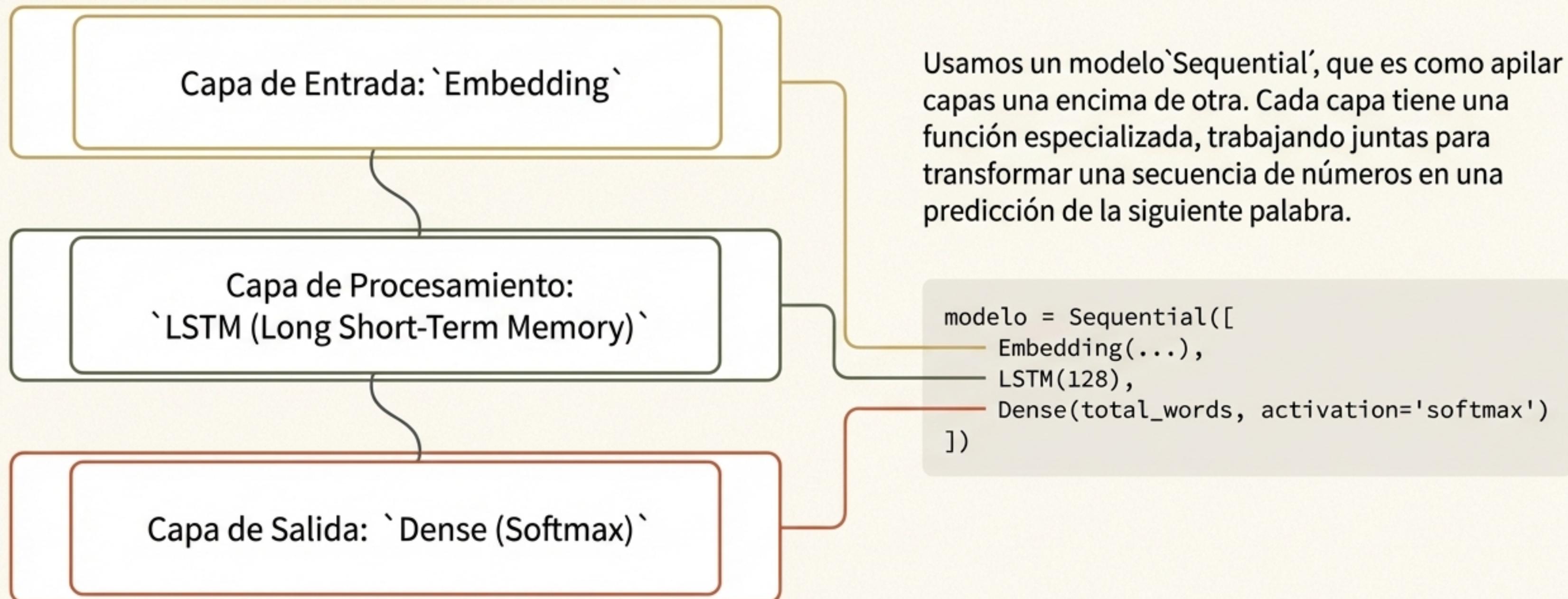
El Porqué

En lugar de predecir un número, el modelo predice la *probabilidad* para cada palabra del vocabulario. Este formato es ideal para medir el error y aprender.

$$3 \longrightarrow [0, 0, 0, 1, 0, \dots]$$

Paso 4: Construyendo el cerebro de la IA.

La arquitectura de nuestra Red Neuronal LSTM.



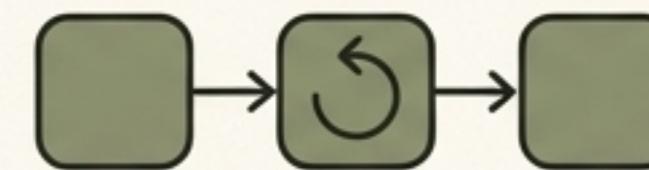
Anatomía de nuestro cerebro artificial.

Capa 1: Embedding - El traductor de matrices



No solo convierte números, sino que aprende el *significado* contextual. Aprende que palabras como 'rey' y 'reina' están relacionadas matemáticamente.

Capa 2: LSTM - La neurona con memoria



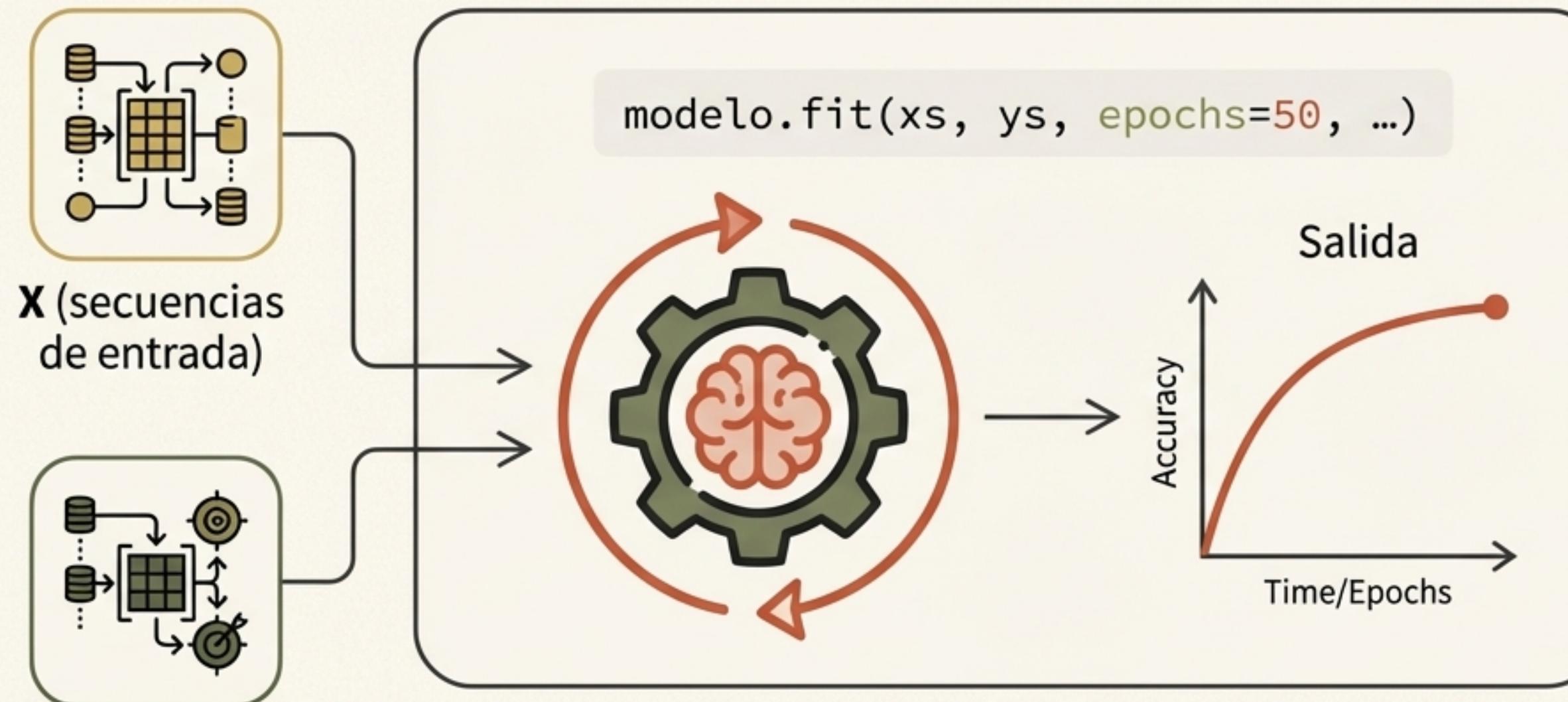
El verdadero núcleo. La capa LSTM (Long Short-Term Memory) tiene la capacidad de *recordar* información de palabras anteriores en la frase para entender el contexto y hacer una predicción más inteligente.

Capa 3: Dense (Softmax) - El tomador de decisiones



La capa final. Calcula una puntuación de probabilidad para CADA palabra de nuestro vocabulario. La activación `softmax` asegura que todas las probabilidades sumen 100%, y nos permite elegir la palabra más probable.

Paso 5: ¡A cocinar! El proceso de entrenamiento.



Epochs

epochs=50: Repasará todo el libro de recetas (los datos) 50 veces, mejorando en cada pasada.



Optimizer

optimizer='adam': El "profesor" inteligente que corrige los errores del modelo de la manera más eficiente.

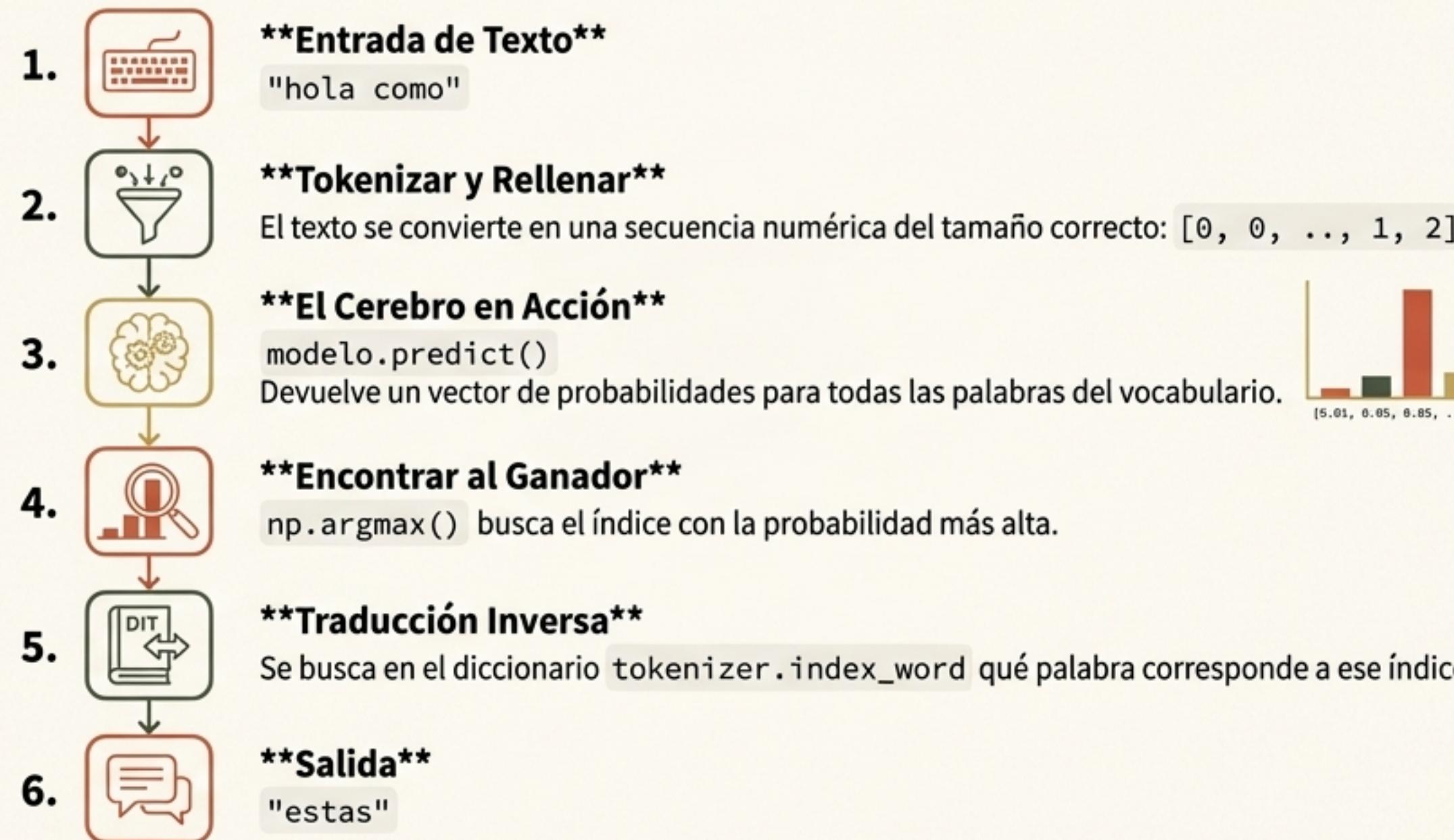


Callback

EarlyStopping: El "vigilante" que detiene el entrenamiento si la IA deja de mejorar. Esto ahorra tiempo y evita que el modelo "memorice" en lugar de "aprender".

Paso 6: Probando el plato (La predicción).

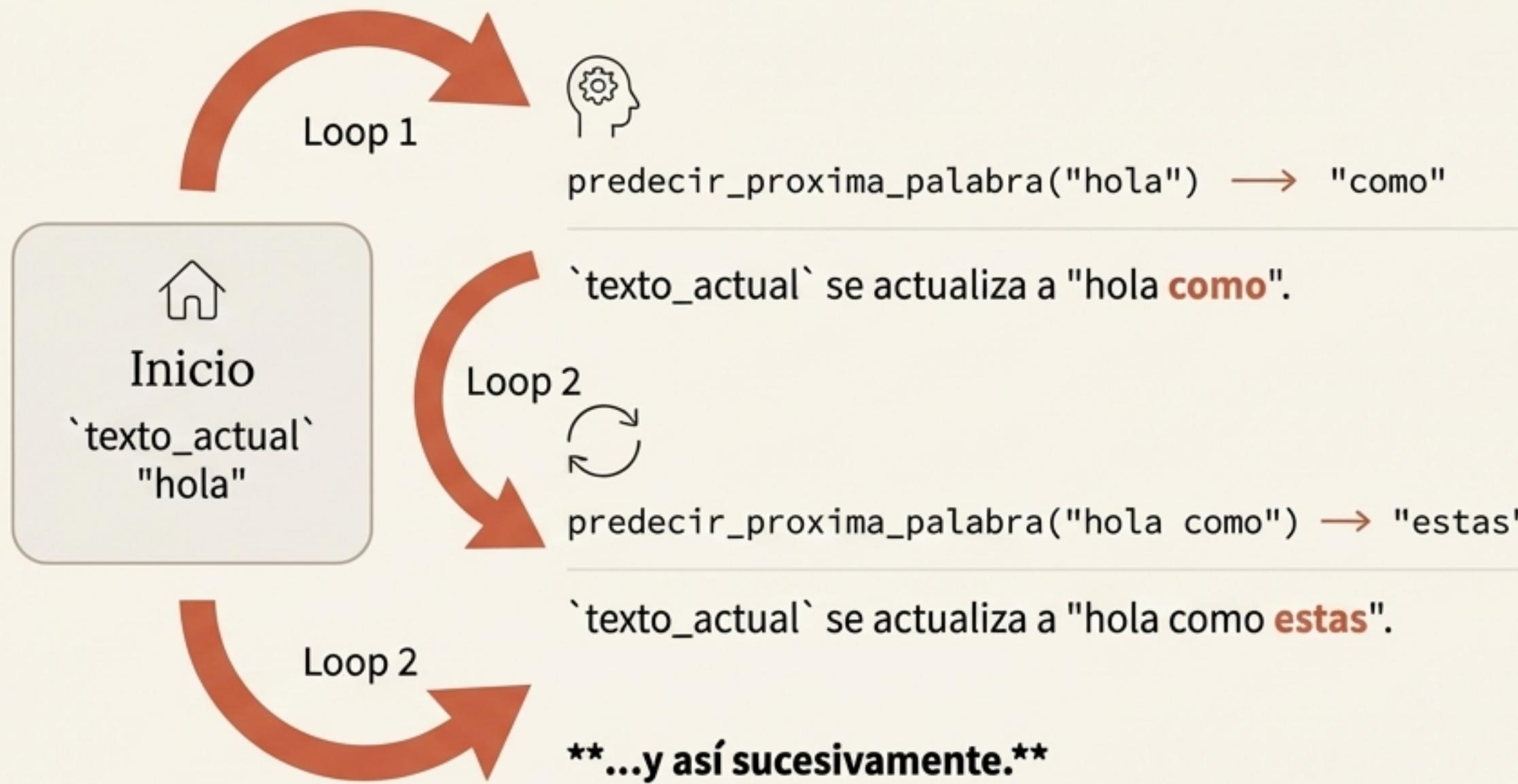
Así es como la IA adivina la siguiente palabra.



Nota importante

Incluimos un umbral de confianza. Si la probabilidad más alta es muy baja (ej: <10%), no devolvemos nada para evitar decir 'tonterías'.

Sirviendo el menú completo: de una palabra a una frase.

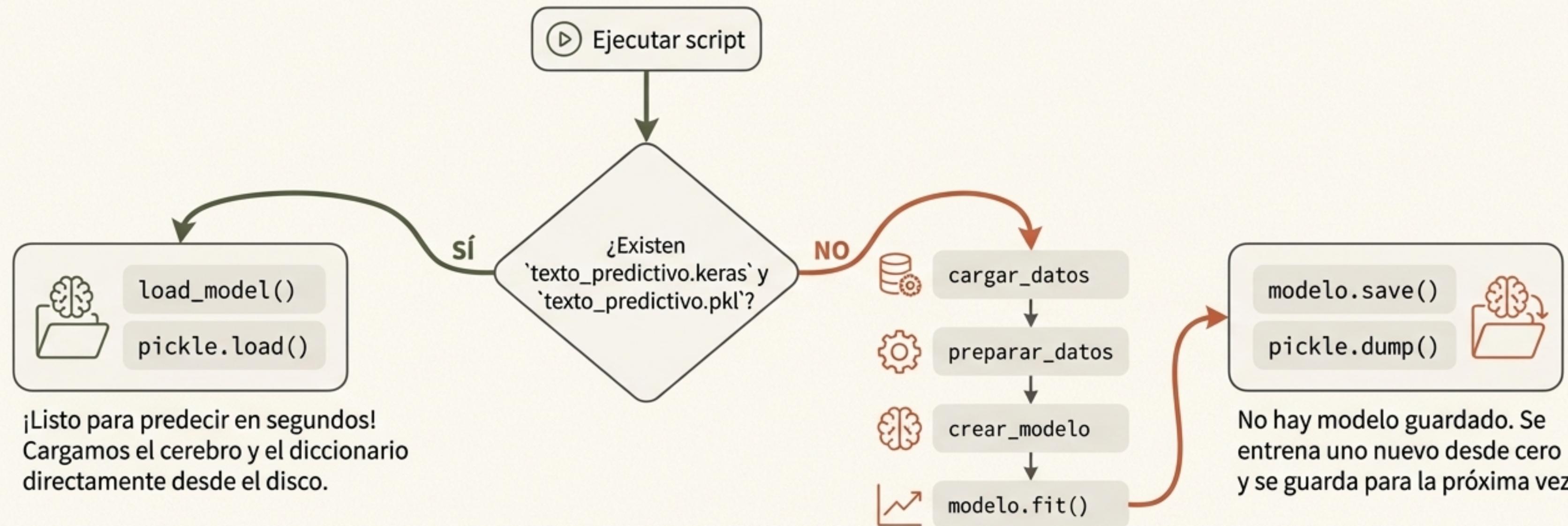


Reglas de Lógica Adicional

- Se implementa una salvaguarda para evitar repetir la misma palabra dos veces seguidas (ej: 'gracias gracias').
- El bucle se detiene si la IA no está segura de la siguiente palabra o si se alcanza el límite de palabras.

Una cocina inteligente: guardar la receta para el futuro.

La lógica de `inicializar()` para cargar o entrenar.



****Punto Clave**:** Este es un paso crítico para la eficiencia. El entrenamiento puede llevar tiempo; la carga es casi instantánea.

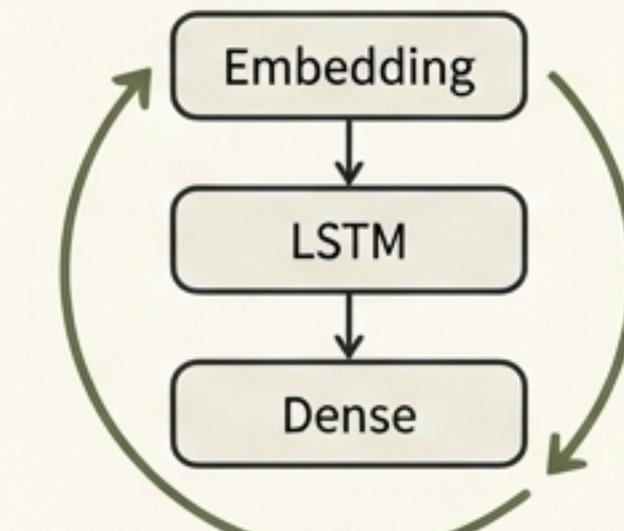
La Receta Completa: Del Dato Crudo a la Predicción Inteligente.

1. Preparación (Mise en Place)

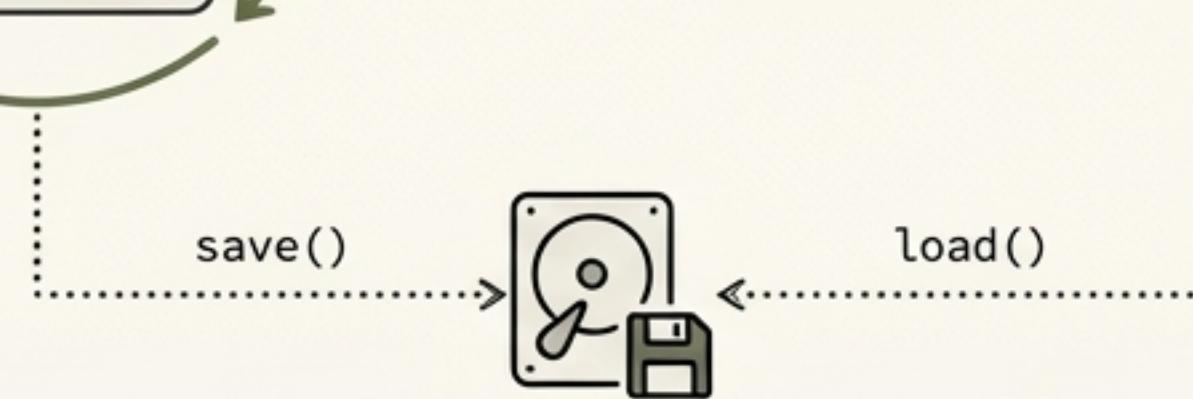
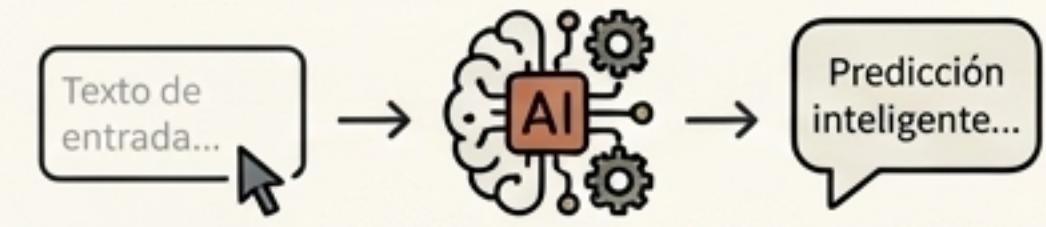


2. Cocción (El Cerebro)

entrenamiento (`.fit()`)



3. Servicio (La Inferencia)



Hemos transformado texto caótico en una herramienta estructurada y predictiva. Cada paso, desde la limpieza inicial hasta la arquitectura del modelo, es un ingrediente esencial en la receta de esta inteligencia artificial.