

Cómo traducir un script de JavaScript con IA

Gustavo Sánchez Muñoz

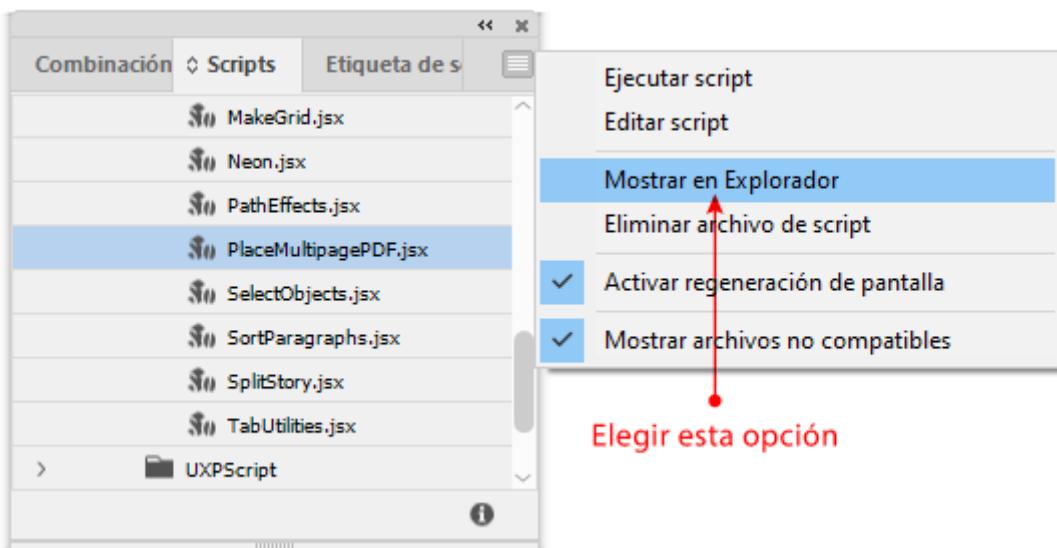
(Diciembre de 2025)

Adobe permite usar en sus programas scripts de JavaScript que son muy útiles y fáciles de modificar. Estos scripts se incluyen con el programa o se encuentran en la red (gratis o de pago), pero uno de sus pequeños inconvenientes suele ser que están en un idioma que nos es incómodo usar.

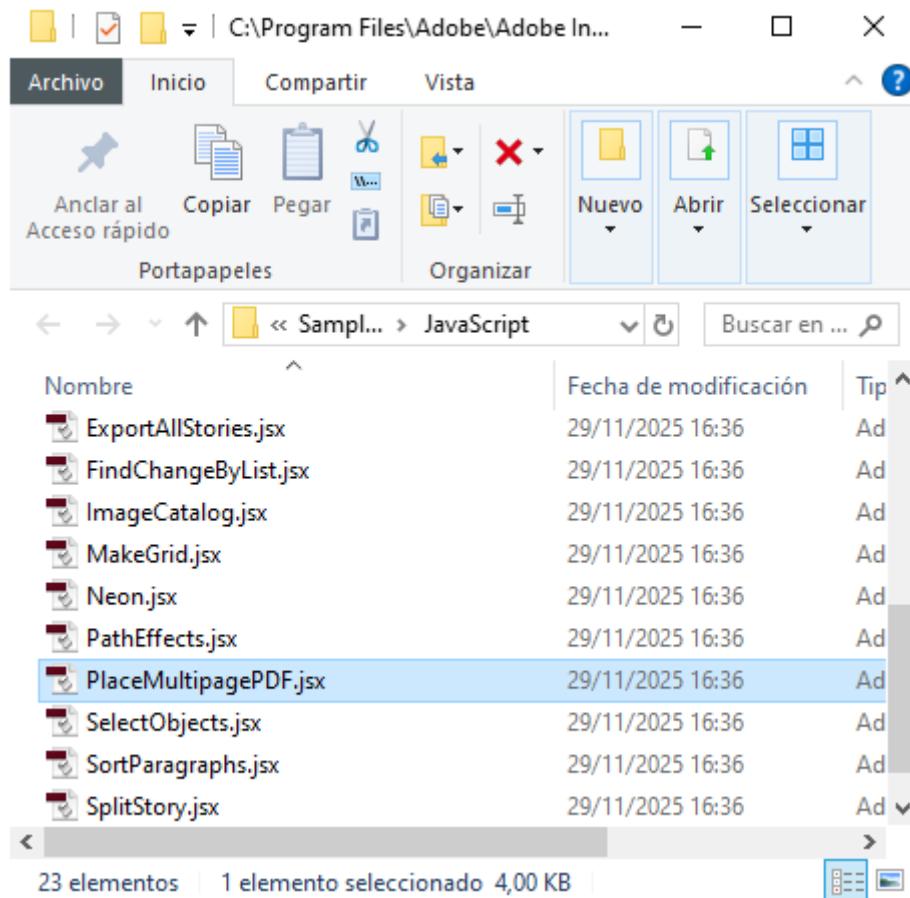
la solución hoy día es muy sencilla: Usar Inteligencia Artificial (IA) para traducir los mensajes y opciones de usuario. No hace falta saber nada de programación, la IA se encarga de localizar los mensajes y pasarlo al idioma pedido. En esta página veremos cómo hacerlo con un ejemplo de un script de Adobe InDesign que permite colocar PDF multi página. Los pantallazos se refieren a castellano o catalán, pero podemos elegir el idioma que nos apetezca siempre que la IA en cuestión sea capaz de usarlo para traducir.

El procedimiento

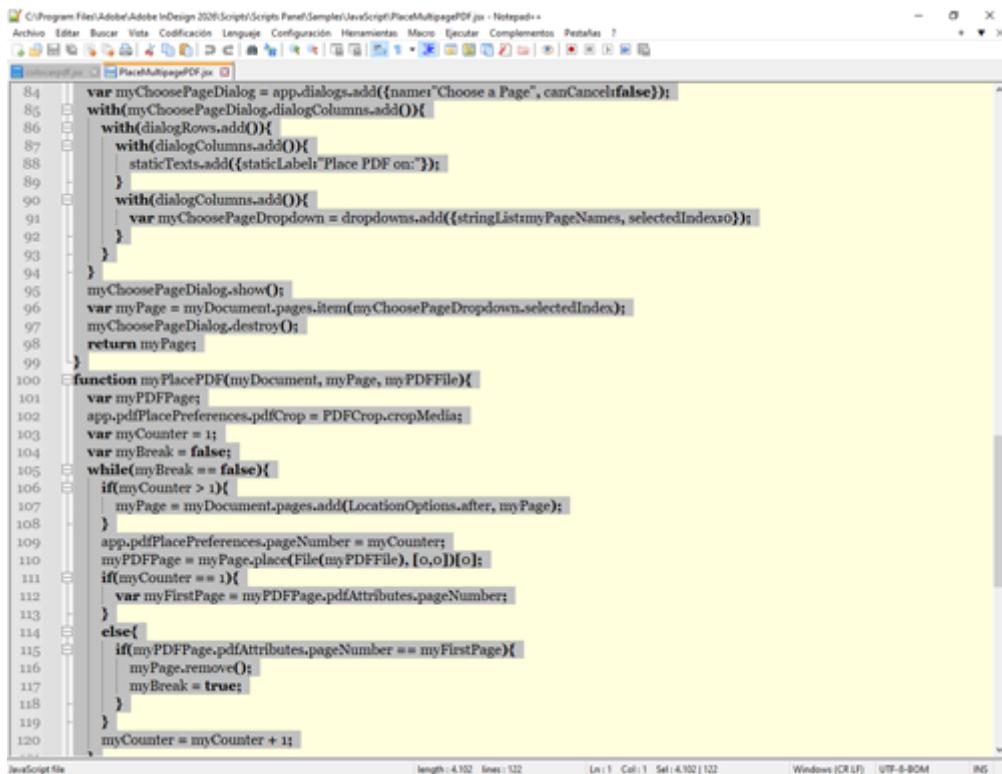
Lo primero es localizar el archivo con el *script*. Si está ya instalado, en Adobe InDesign, por ejemplo, se hace con la opción "Mostrar en Explorador" disponible en la esquina superior derecha de la paleta "Scripts".



Esto abrirá en una ventana la carpeta en la que se encuentran esos *scripts* de InDesign. Si tuviéramos el script sin instalar, por ejemplo, en el escritorio, este paso obviamente no haría falta.



Abrimos el *script* con cualquier editor de texto sin formato (el Bloc de notas, Notepad++, etc.) y copiamos su texto íntegro (es texto sencillo sin formato, no tiene misterio).



```

84 var myChoosePageDialog = app.dialogs.add({name:"Choose a Page", canCancel:false});
85 with(mChoosePageDialog.dialogColumns.addO){
86   with(dialogRows.addO){
87     with(dialogColumns.addO){
88       staticTexts.add({staticLabel:"Place PDF on:"});
89     }
90   }
91   with(dialogColumns.addO){
92     var myChoosePageDropdown = dropdowns.add({stringList:myPageNames, selectedIndex:0});
93   }
94 }
95 myChoosePageDialog.showO;
96 var myPage = myDocument.pages.item(myChoosePageDropdown.selectedIndex);
97 myChoosePageDialog.destroyO;
98 return myPage;
99
100 function myPlacePDF(myDocument, myPage, myPDFFile){
101   var myPDFPage;
102   app.pdfPlacePreferences.pdfCrop = PDFCrop.cropMedia;
103   var myCounter = 1;
104   var myBreak = false;
105   while(myBreak == false){
106     if(myCounter > 1){
107       myPage = myDocument.pages.add(LocationOptions.after, myPage);
108     }
109     app.pdfPlacePreferences.pageNumber = myCounter;
110     myPDFPage = myPage.place(File(myPDFFile), [0,0][0]);
111     if(myCounter == 1){
112       var myFirstPage = myPDFPage.pdfAttributes.pageNumber;
113     }
114     else{
115       if(myPDFPage.pdfAttributes.pageNumber == myFirstPage){
116         myPage.remove();
117         myBreak = true;
118       }
119     }
120     myCounter = myCounter + 1;
121 }

```

Abrimos en un navegador [Gemini](#) o [ChatGPT](#), escribimos "Traduce al catalán este javascript cambiando los mensajes para el usuario y cuadros de diálogo que haya en él:" y pegamos el texto íntegro del *script*.



Aquí tens el codi JavaScript traduït al català, amb els missatges d'usuari i els textos dels diàlegs modificats:

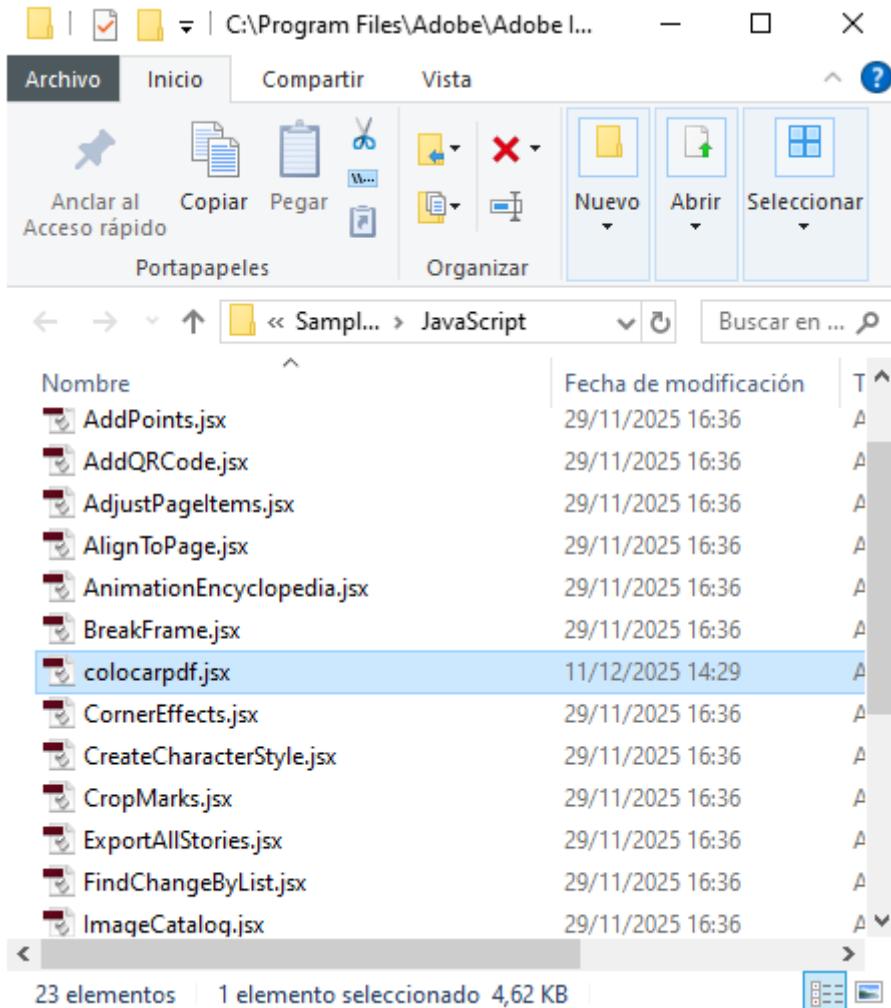
JavaScript

```

//PlaceMultipagePDF.jsx
//Un JavaScript d'InDesign
/*
@@@BUILDINFO@@@ "PlaceMultipagePDF.jsx" 3.0.0 15 December 2009
*/
//Colloca totes les pàgines d'un PDF de diverses pàgines.
//
//Per a més informació sobre scripting d'InDesign/InCopy, consulta la documentació
//disponible a http://www.adobe.com/devnet/indesign/sdk.html
//o visita el fòrum d'usuari a usuari de Scripting d'InDesign a http://www.adobe.com/forums
//
main();
function main(){
  //Assegura't que la interacció de l'usuari (visualització de diàlegs, etc)
  app.scriptPreferences.userInteractionLevel = UserInteractionLevels.interact;
  //Mostra un quadre de diàleg estàndard per obrir fitxers.
  var myPDFFile = File.openDialog("Tria un fitxer PDF");
  if((myPDFFile != "")&&(myPDFFile != null)){
    var myDocument = myPage =

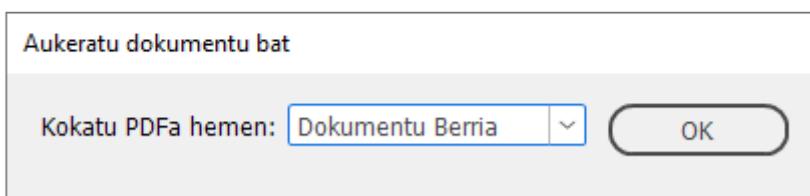
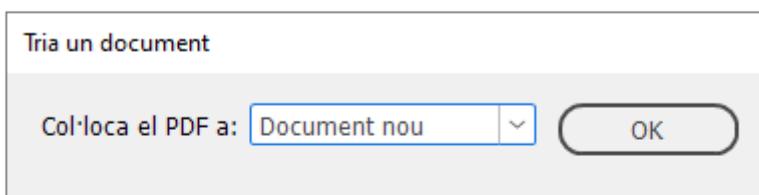
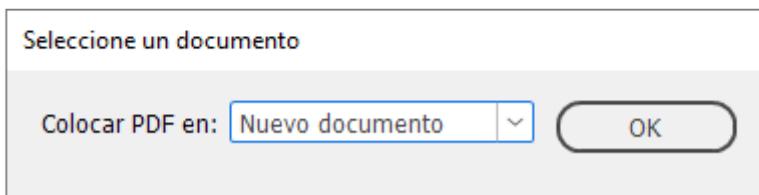
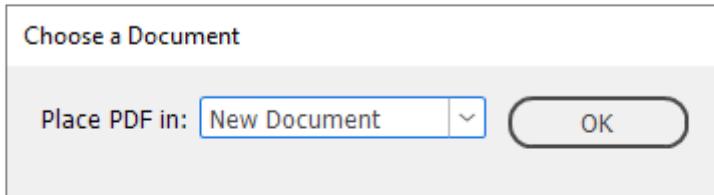
```

Una vez tengamos el resultado (como vemos en la imagen superior si le pedimos a Gemini que traduzca al catalán el mencionado script de InDesign), lo pegamos en un nuevo documento de texto que guardamos con un nombre adecuado sin espacios en blanco ni caracteres especiales y la extensión ".jsx" (por ejemplo: "colocarpdf.jsx").



Ahora, guardamos ese archivo en la misma carpeta en la que están los demás *scripts*.

No hace falta cerrar y abrir el programa. Con abrir y cerrar la paleta de "Scripts" allí estará el nuevo *script* españolizado (o catalanizado o...).



Estas son capturas de pantalla de los resultados; de arriba abajo: El original en inglés, el castellano, catalán y vasco (y no, no hablo los dos últimos).

Itzulpenen Laburpena

Testu Originala	Euskarazko Itzulpena	Kokapena
"Choose a PDF File"	"Aukeratu PDF fitxategi bat"	File.openDialog elkarritzeta-koadroa
"New Document"	"Dokumentu Berria"	Dokumentu zerrendako aukera
"Choose a Document"	"Aukeratu dokumentu bat"	myChooseDocument elkarritzeta-koadroaren izenburua
"Place PDF in:"	"Kokatu PDFa hemen:"	myChooseDocument elkarritzeta-koadroaren etiketa
"Choose a Page"	"Aukeratu orrialde bat"	myChoosePage elkarritzeta-koadroaren izenburua
"Place PDF on:"	"Kokatu PDFa hemen:"	myChoosePage elkarritzeta-koadroaren etiketa

Un buen detalle es que en los resultados (al menos Gemini) te ofrece una tabla con los textos originales y las traducciones que ha hecho. Si vemos algo que no nos gusta podemos cambiar esa traducción concreta buscando el mensale y haciendo los cambios en el nuevo *script*.

Este procedimiento vale para cualquier otro JavaScript, por cierto.