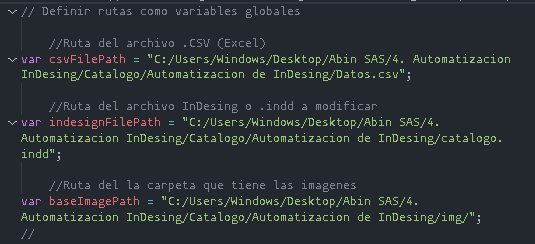
**MANUAL DEL PROGRAMADOR**

**DESCRIPCIÓN DETALLADA DEL PROGRAMA**

El programa que se presenta es un script en JavaScript para Adobe InDesign. Su objetivo es automatizar el proceso de actualización de un documento de InDesign con datos provenientes de un archivo CSV (Excel). A continuación, se detalla cada parte del script:

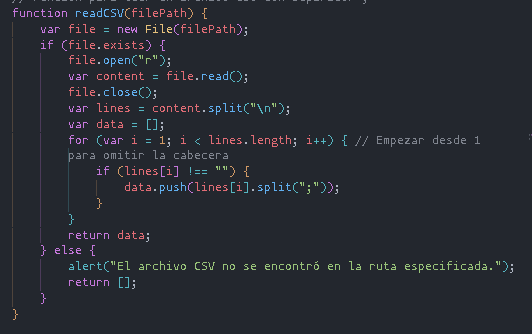
**1. Definición de variables globales**

Se definen varias rutas como variables globales para facilitar la gestión de archivos en el script. Estas variables incluyen la ruta del archivo CSV, el documento de InDesign a modificar, y la carpeta de imágenes.



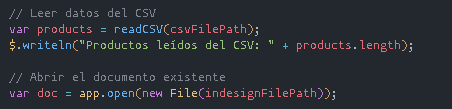
**2. Función para leer el archivo CSV**

La función `readCSV` abre y lee el archivo CSV. Se espera que el archivo use el punto y coma (`;`) como delimitador de campos. La función devuelve una matriz con los datos del CSV, omitiendo la cabecera.



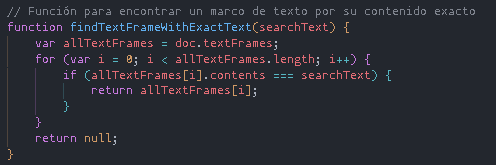
**3. Leer CSV y abrir el documento de InDesign**

El documento de InDesign se abre usando la ruta especificada. La variable `doc` almacena una referencia al documento abierto.



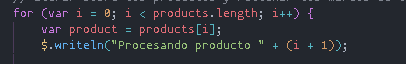
**4. Función para Encontrar un Marco de Texto por Contenido**

La función `findTextFrameWithExactText` busca en todos los marcos de texto del documento el marco que contiene un texto específico. Esta función es crucial para actualizar los marcos de texto con los datos del CSV.

****

**5. Procesar Cada Producto del CSV**

Se itera sobre cada línea de datos del CSV (cada producto) y se actualizan los marcos de texto correspondientes en el documento de InDesign. Se actualizan los marcos de texto para "Código", "REF", "PREMIUM", "PARTMO", "Descripción", y se coloca una imagen si está disponible.



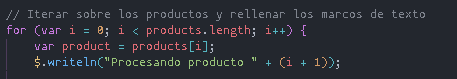
**6. Verificación de la Apertura del Documento**

El script primero verifica si el documento de InDesign se ha abierto correctamente. La variable `doc` contiene la referencia al documento abierto, y si `doc` es verdadero, el script continúa con el procesamiento.



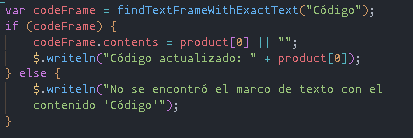
**7. Iteración sobre los Productos**

El script itera sobre cada producto en el array `products`, que contiene los datos leídos del archivo CSV.



**8. Actualización de Marcos de Texto**

Para cada producto, el script busca y actualiza varios marcos de texto en el documento de InDesign basándose en el contenido esperado. Cada marco de texto se identifica por su contenido específico.

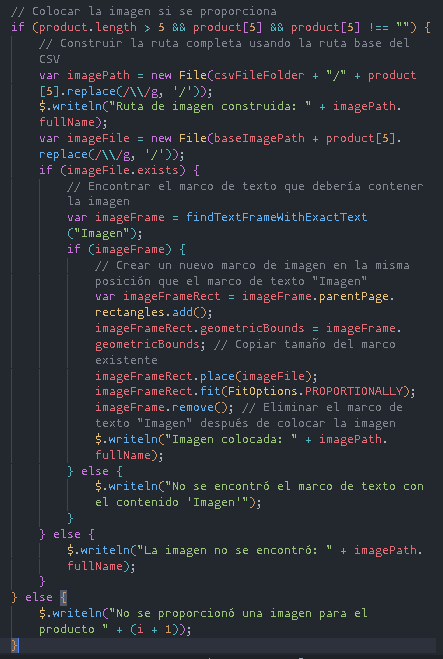


* `findTextFrameWithExactText("Código")`: Busca un marco de texto que contenga el texto "Código".
* `codeFrame.contents = product[0] || "";`: Actualiza el contenido del marco de texto con el valor correspondiente del CSV. Si no hay valor, se usa una cadena vacía.
* `$.writeln("Código actualizado: " + product[0]);`: Imprime el nuevo valor del código.

Este proceso se repite para otros marcos de texto como "REF", "PREMIUM", "PARTMO", y "Descripción".

**9. Manejo de Imágenes**

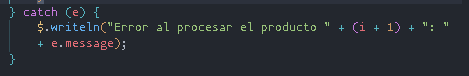
El script también gestiona la colocación de imágenes en el documento de InDesign si el producto tiene una imagen asociada. Se construye la ruta de la imagen y se verifica su existencia antes de colocarla en el documento.



* `if (product.length > 5 && product[5] && product[5] !== "")`: Verifica si hay una imagen asociada para el producto.
* `var imagePath = new File(csvFileFolder + "/" + product[5].replace(/\\/g, '/'));`: Construye la ruta completa para la imagen.
* `var imageFile = new File(baseImagePath + product[5].replace(/\\/g, '/'));`: Crea un objeto de archivo para la imagen.
* `if (imageFile.exists)`: Verifica si el archivo de imagen existe.
* `var imageFrame = findTextFrameWithExactText("Imagen");`: Busca el marco de texto que debería contener la imagen.
* `var imageFrameRect = imageFrame.parentPage.rectangles.add();`: Crea un nuevo marco de imagen en la página.
* `imageFrameRect.place(imageFile);`: Coloca la imagen en el marco.
* `imageFrameRect.fit(FitOptions.PROPORTIONALLY);`: Ajusta la imagen en el marco.
* `imageFrame.remove();`: Elimina el marco de texto original "Imagen" después de colocar la imagen.

**10. Manejo de Errores**

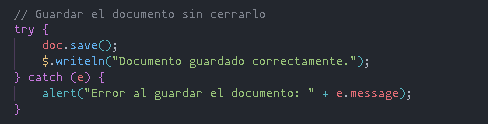
El script incluye manejo de errores para capturar y reportar cualquier problema durante el procesamiento de los productos.



* `catch (e)`: Captura excepciones que puedan ocurrir durante la actualización de los marcos de texto o el manejo de imágenes.
* `$.writeln("Error al procesar el producto " + (i + 1) + ": " + e.message);`: Imprime un mensaje de error que incluye el número del producto y el mensaje de la excepción.

**11. Guardar el Documento**

Finalmente, el documento de InDesign se guarda para asegurar que todos los cambios se apliquen.

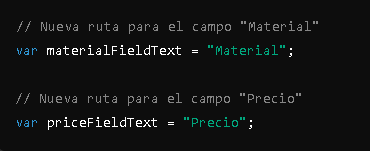


**CÓMO ACTUALIZAR LOS DATOS PARA INCLUIR NUEVOS TEXTOS**

Para añadir nuevos textos al script y actualizar los datos, sigue estos pasos:

**1. Definir Nuevas Variables para Nuevos Textos**

Añade nuevas variables globales para los nuevos textos que quieres incluir en los marcos de texto.



**2. Actualizar el CSV para Incluir Nuevas Columnas**

Modifica tu archivo CSV para incluir columnas para los nuevos datos. Por ejemplo, agrega columnas para "Material" y "Precio".

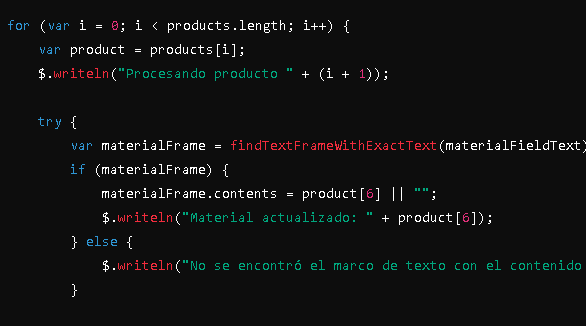


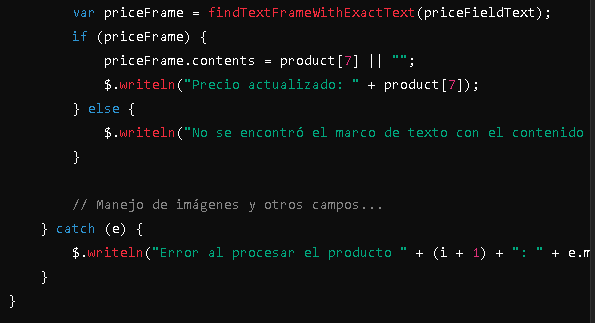
**3. Modificar la Función de Lectura CSV**

Asegúrate de que la función `readCSV` pueda manejar las nuevas columnas del CSV. No es necesario cambiar esta función a menos que quieras cambiar el delimitador o el formato de lectura.

**4. Actualizar el Bucle de Procesamiento para Nuevos Campos**

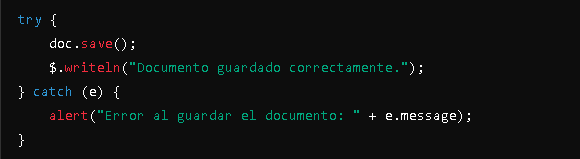
Añade lógica en el bucle de procesamiento para manejar los nuevos campos. Asegúrate de que los nuevos marcos de texto ("Material" y "Precio") se encuentren y actualicen correctamente.





**5. Guardar el Documento**

El proceso de guardar el documento sigue siendo el mismo. Asegúrate de que el documento se guarda al final de la ejecución del script.



Con estas modificaciones, tu script estará actualizado para manejar nuevos campos de texto en el documento de InDesign, permitiendo una mayor flexibilidad en la automatización de la creación del catálogo.