

Trabajo de final de grado

## Ingeniería en tecnologías Industriales

### Ampliación y mejoras en una herramienta de corrección de la memoria de un TFE mediante Microsoft Excel

MEMORIA

**Autor:** Carlos Pascual Escalante  
**Director:** Manuel Moreno Eguilaz  
**Convocatoria:** Mayo 2023



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





## Resumen

El objetivo de este trabajo es continuar y mejorar el trabajo de otros compañeros. Principalmente de dos trabajos llamados **Herramienta de verificación de TFE basado en Excel VBA** y **Herramienta de corrección de la memoria de un TFE mediante macros de Microsoft Excel**. Esta es la razón principal del nombre de este proyecto de final de grado: **Ampliación y mejoras en una herramienta de corrección de la memoria de un TFE mediante Microsoft Excel**.

Se explicarán los motivos detrás de la elaboración de este trabajo. Los procedimientos y herramientas utilizadas para la mejora de las funciones ya existentes, mejoras en la presentación de la interfaz, la organización del código y la implementación de funciones nuevas. El lenguaje de programación utilizado será Visual Basic for Applications (VBA) para Excel. Se probarán los límites de este lenguaje en la búsqueda de nuevas mejoras para esta herramienta de corrección.

## Resum

L'objectiu d'aquest treball és continuar i millorar el treball d'altres companys. Principalment de dos treballs anomenats **Eina de verificació de TFE basat en Excel VBA** i **Eina de correcció de la memòria d'un TFE mitjançant macros de Microsoft Excel**. Aquesta és la raó principal del nom d' aquest projecte de final de grau: **Ampliació i millores en una eina de correcció de la memòria d' un TFE mitjançant Microsoft Excel**.

S'explicaran els motius darrere de l'elaboració d'aquest treball. Els procediments i eines utilitzades per a la millora de les funcions ja existents, millores en la presentació de la interfície, l'organització del codi i la implementació de funcions noves. El llenguatge de programació utilitzat serà Visual Basic for Applications (VBA) per a Excel. Es provaran els límits d'aquest llenguatge en la recerca de noves millores per a aquesta eina de correcció.

## Summary

The goal of this work is to continue and improve the work of other colleagues. Mainly two jobs called **Excel VBA-based TFE Verification Tool** and **TFE Memory Correction Tool using Microsoft Excel macros**. This is the main reason for the name of this final degree project: **Extension and improvements in a tool for correcting the memory of a TFE using Microsoft Excel**.

The reasons behind the elaboration of this work will be explained. The procedures and tools used for the improvement of existing functions, improvements in the presentation of the interface, the organization of the code and the implementation of new functions. The programming language used will be Visual Basic for Applications (VBA) for Excel. The limits of this language will be tested in the search for new improvements for this correction tool.



## Sumario

<b>1. Glosario</b> .....	6
<b>2. Prefacio</b> .....	7
<b>3. Introducción</b> .....	8
<b>4. Consideraciones previas</b> .....	9
<b>5. Desarrollo del programa</b> .....	12
<b>6. Comprobación en un caso real</b> .....	38
<b>7. Planificación temporal</b> .....	59
<b>8. Estudio Económico</b> .....	61
<b>9. Estudio Ambiental</b> .....	63
<b>10. Impacto social</b> .....	64
<b>11. Perspectiva de género</b> .....	65
<b>Conclusiones</b> .....	66
<b>Agradecimientos</b> .....	67
<b>Bibliografía</b> .....	68

# 1. Glosario

**ETSEIB:** Escola Tècnica Superior de Ingenieria Industrial de Barcelona

**TFG:** Trabajo de Fin de Grado

**VBA:** Visual Basic for Applications

**TFE:** Trabajo de Fin de Estudios

## 2. Prefacio

### 2.1. Origen del proyecto

El motivo principal de este proyecto es aprender más en profundidad sobre la herramienta de Microsoft Excel ya que en mi experiencia como estudiante y trabajador he visto que es una herramienta que se usa bastante hoy en día. Y que por lo tanto si aprendo habilidades más avanzadas sobre este programa podría tener más posibilidades de acceder a trabajos mejores. Además, que aprender macros ayuda a automatizar muchas tareas en cualquier área que incluya Microsoft Excel y optimizar el tiempo de trabajo.

Las asignaturas que más me han gustado del grado han sido las de programación y como no tenía muy claro sobre qué tema hacer el TFG pues al ver esta propuesta en la bolsa de TFG's me decanté por ella por las razones que he mencionado anteriormente y la buena experiencia que he tenido en las asignaturas de programación del grado.

### 2.2. Motivación

Llegar a un nivel de dominio avanzado de Excel y con ello mejorar mi perfil profesional. Poner a prueba aquí los conocimientos adquiridos en las asignaturas de programación para aprender un lenguaje nuevo de programación. Ver hasta qué punto puedo llegar a automatizar y mejorar esta herramienta de verificación y corrección.

### 2.3. Requisitos previos

Uno de los requisitos previos que veo más relevantes son el conocimiento del lenguaje de Visual Basic, como se activa en el Excel, como hacer que los archivos internos y los que se descargan habiliten el uso de las macros, donde se programa y las diferentes ventanas que hay en el editor de Visual Basic etc. En definitiva, saber usar este lenguaje de programación. Concretamente aplicado a Word ya que manipularemos un archivo Word.

En mi caso yo no sabía nada así que tuve que aprenderlo desde 0. Me ha ayudado bastante un curso online sobre macros [1] que hice antes de ponerme con el proyecto, también me ayudó la lectura de los trabajos que preceden a este para entrar más en materia y por último la lógica en programación adquiridas en los cursos del grado me han ayudado a aprender y adaptarme más rápido al lenguaje VBA.

## 3. Introducción

### 3.1. Objetivos del proyecto

El objetivo del proyecto es encontrar mejoras y poner en práctica algunas propuestas de mejoras de los trabajos que preceden a este. Más específicamente los trabajos de una Herramienta de verificación de TFE basado en Excel VBA del compañero Ignasi Alés Roca [2] y Herramienta de corrección de la memoria de un TFE mediante macros de Microsoft Excel del compañero Pau Palau [3]. De tal forma que se puedan reducir al máximo los tiempos de corrección de una memoria de TFG hecha en la ETSEIB en cuanto al formato de esta.

### 3.2. Alcance del proyecto

Lo que se quiere conseguir es la búsqueda de mejoras y solución de errores en cuanto a la verificación y corrección de los siguientes aspectos en cuanto al formato de una memoria de TFG de la ETSEIB:

- Portada
- Índice
- Ortografía
- Parágrafos
- Pies de página
- Encabezados
- Bibliografía

Además de la búsqueda e implementación de nuevas funciones que se le puedan atribuir a esta herramienta como se verá en detalle más adelante, la puesta en práctica de algunas propuestas de mejora que aparecían en trabajos anteriores con esta temática y por último cambios en la interfaz y la organización del código que bajo mi punto de vista le daban un toque más vistoso por un lado y hace más fácil la lectura del código por el otro.



## 4. Consideraciones previas

Para entender este proyecto en su totalidad hace falta saber las directrices en las que se basa, el grado de familiarización que se tenga con el lenguaje VBA de Excel y Word y por último el contexto de este proyecto.

### 4.1. Directrices (plantilla proporcionada por la ETSEIB)

El documento que se usa como referencia para decir si el formato de un documento de TFE de la ETSEIB es correcto o no es una plantilla que proporciona la misma ETSEIB y que por lo tanto la verificación y/o corrección de las distintas partes del formato de un TFE se harán en base a este documento. Su nombre de documento Word es: *tfg\_tfm\_plantilla.doc* [4].

En cuanto a los detalles del documento, el trabajo se inicia con una portada con una marca de agua de la universidad, el título del grado / máster, el nombre del autor, director y la convocatoria en la cual se realizó el trabajo. Seguidamente, hay una página en blanco y, a continuación, se muestra un resumen definiendo el objetivo, métodos, resultados y conclusiones presentados en el trabajo. En la siguiente sección aparece el índice. Se indica que está creado de forma automática y que únicamente hay que actualizarlo. No se detalla el tamaño de la letra en el índice, pero todo el documento se realiza en Arial y el texto (no títulos) es en 11. A continuación, aparece el primer capítulo opcional del trabajo. Se trata del glosario y es interesante para definir ciertos términos que puedan ser no conocidos por cualquier lector del trabajo. Otro capítulo opcional que sigue es el prefacio. Este capítulo permite al autor poner en contexto el origen del proyecto, así como darle una idea al lector de cómo es el propio autor, ya que permite explicar sus motivaciones y requerimientos previos que se debían tener a la hora de empezar a redactar el TFE.

Posteriormente se presenta una introducción del trabajo detallando los objetivos y alcance del proyecto. El siguiente apartado ya detalla cómo construir los siguientes puntos del documento. También se muestra como añadir figuras y ecuaciones al documento y cómo numerarlas.

Finalmente, se llega a las conclusiones, los agradecimientos y la bibliografía. En las conclusiones, se describen los resultados a los que se ha llegado en la realización del trabajo y alguna mejora propuesta para poder obtener algún resultado diferente. En los agradecimientos, se puede dar reconocimiento a aquella gente que haya podido

contribuir de forma excepcional en la realización del proyecto. Se deben citar a los otros trabajos y autores en caso de usar fuentes provenientes de proyectos anteriores. La bibliografía debe contener todas las referencias usadas a lo largo del trabajo de forma ordenada según aparecen en el documento. La bibliografía se divide en referencias bibliográficas y bibliografía complementaria. Se diferencian en que las referencias bibliográficas se citan en el texto y las complementarias no. Todas las referencias citadas deben ir enumeradas por orden de aparición, con su fecha de publicación.

Según se observa en la plantilla de TFE, los encabezados de las páginas pares e impares son diferentes. Esto también se cumple en los pies de página y hay que tener en cuenta. Este tipo de plantillas ofrece una estandarización en los documentos y permite al autor redactar de forma sencilla preocupándose tan sólo del contenido.

## 4.2. Grado de familiarización con el lenguaje VBA

Para llevar a cabo este proyecto es vital entender el lenguaje VBA ya que es el que se utilizará para crear las macros que conformarán toda la herramienta de corrección. Yo en mi caso sabía muy poco de las macros al empezar, pero gracias a un curso que hice [1], la consulta de la documentación VBA [5] por internet y la lectura y entendimiento de los códigos VBA de dos TFE referentes a este tema ([2] y [3]) he reunido los recursos suficientes para incorporar nuevas funcionalidades a esta herramienta y a mejorar funciones ya existentes tanto como de verificación como de corrección.

En cuanto al lenguaje VBA, se basa en la manipulación de objetos. Para ello, estos objetos poseen propiedades y métodos. Una propiedad de un objeto es una característica de este objeto mientras que un método es una función que puede llevar a cabo el objeto. Cada objeto posee un número diferente de propiedades y métodos. Por ejemplo, una sección de un documento Word puede ser un objeto (objeto *Sections* [6]) mientras que los encabezados de las páginas pares (*headers(wdHeaderFooterEvenPages)* [7]) de esa sección son una propiedad de la sección. Otro ejemplo, un rango de celdas en concreto puede ser un objeto (objeto *range* [8]) y la eliminación de los valores de este rango de celdas es un método (método *ClearContents* [9]) de este objeto.

Los objetos se pueden agrupar entre ellos y formar colecciones de objetos. A esta colección de objetos también se le puede tratar como a un objeto. Por ejemplo, todas las imágenes de un documento Word se agrupan en una colección (Colección

*InlineShapes* [10]). Esta colección al ser también un objeto tiene métodos, por lo tanto, uno de los métodos más comunes de las colecciones es que te cuente cuantos objetos hay en la colección (método *InlineShapes.Count* [11]). Por lo tanto, con uno de los métodos de la colección de imágenes de Word podrías saber cuántas imágenes hay en el documento.

## 5. Desarrollo del programa

Para la realización de este trabajo ha sido de gran ayuda el código VBA de Ignasi [2] y Pau [3] no solo para comprender de primera mano la estructura del programa y detectar errores o más oportunidades de mejora sino también para aprender nuevas estructuras de control el flujo que me han servido para implementar las nuevas funciones que explicaré en detalle más adelante. Cabe mencionar también que este programa trae integradas las funciones de verificación, corrección y de redacción de informe, pero el núcleo del trabajo está en la mejora de la corrección y la verificación del TFE. Por no mencionar un ligero cambio en la interfaz gráfica y la organización del código VBA.

Para hacer el trabajo lo más compacto y útil solo explicaré los cambios añadidos a las funciones ya existentes y las nuevas funciones implementadas.

### 5.1. Interfaz gráfica

Decidí poner a prueba los conocimientos que adquirí en el curso de macros que hice [1] cambiando el modelo de interfaz de tal forma que mantenga las funciones ya implementadas pero que de un aspecto más de aplicación manteniendo la profesionalidad.

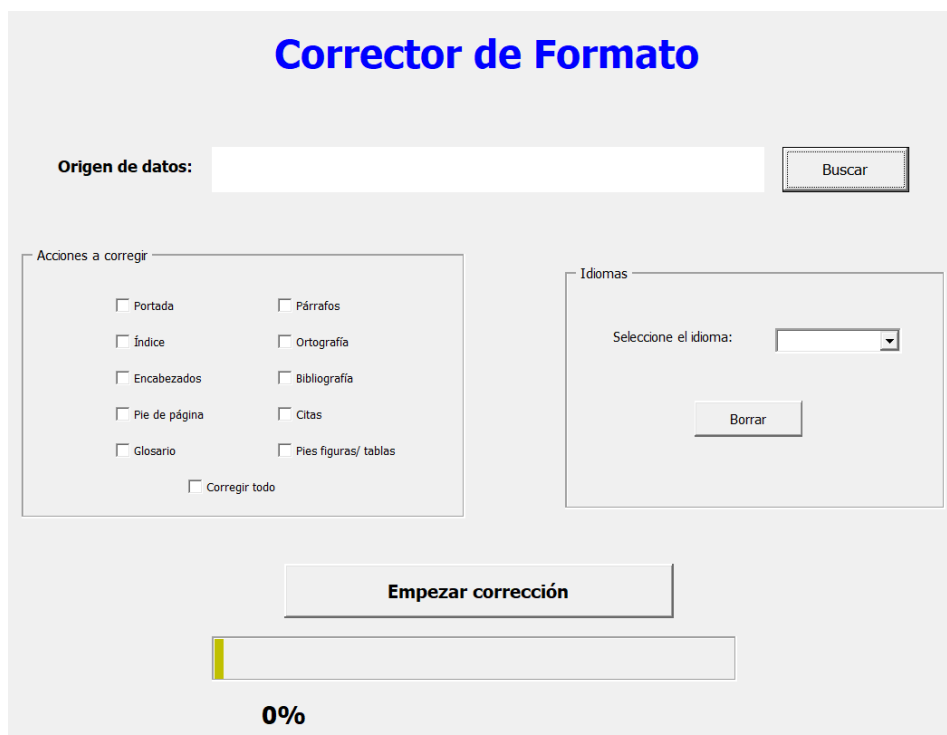


Fig. 5.1 Interfaz gráfica del programa. Fuente: Propia

En cuanto a la parte del diseño usé la herramienta llamada formulario (*Userform* [12]) junto con las siguientes herramientas del cuadro de herramientas asociadas al formulario como etiquetas (*Label*), botones (*Commandbutton*), marcos (*Frame*), casillas (*CheckBox*) y cuadros de texto combinado (*ComboBox*).

Para más información sobre cada una de estas herramientas mirar en la documentación del cuadro de herramientas [13]. Como se puede apreciar en la figura 5.1 también se ha incorporado la barra de progreso debajo del botón que provoca el comienzo del proceso de verificación y corrección de la memoria y la posterior redacción del informe.

En cuanto a la parte de la funcionalidad, la estructura es la misma que en los trabajos que preceden a este. Es decir, el programa tiene una función principal que llama a las otras en función de si estas han sido marcadas por el usuario mediante un *CheckBox*. La única particularidad aquí es la forma de unir el código de las funciones con algunos elementos del formulario que se hace a través de los eventos (*Events* [14]). Para definir un evento hace falta determinar qué es lo que hará en este evento y bajo qué condiciones se activará. En este caso, los eventos que más se han usado son los asociados a cuando presionas un botón del formulario o se abre o cierra el libro de Excel. Este último evento ha sido muy útil para que una vez se abra el libro de Excel se active el formulario con el programa y solo sea visible el formulario y dar un aspecto de aplicación profesional.

Otra cosa relevante a mencionar es el cambio en la organización de las funciones en el programa con un módulo dedicado a cada función para ordenar el código y saber dónde está cada cosa y no tener que buscar una función en una línea de código muy larga ya que puede llegar a ser mentalmente agotador.

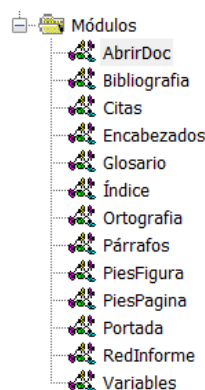


Fig. 5.2 Funciones principales del programa distribuidas en módulos. Fuente: Propia

Como se puede apreciar en la imagen hay un módulo llamado variables que es donde se guardan las variables que se usan en distintas funciones también llamadas variables públicas.

Por lo tanto, el programa se ejecutará siguiendo el siguiente orden:

1. Abrir el documento a modificar. Mediante el botón buscar y su evento asociado que abre una ventana de búsqueda de archivos. Una vez se seleccione el archivo deseado, este se abrirá en segundo plano y su dirección de archivo aparecerá en la etiqueta de nombre ruta (la etiqueta de fondo blanco).
2. Cada función integrada en el programa se activará en función de si su casillero asociado este seleccionado o no. Las funciones se ejecutan en orden de aparición en el formulario.
3. Cada vez que se ejecuta una función se ejecuta el proceso de verificación y una vez se ha detectado el número de errores se le notifica al usuario y se le pregunta si quiere corregir o no estos errores.
4. La barra de progreso avanza cada vez que se analiza un nuevo parámetro.
5. Una vez se han ejecutado todos los parámetros, se guarda el documento modificado con un nombre diferente en la misma ubicación de la memoria y se cierra para tener el documento original y el modificado.
6. Se procede a hacer el informe.

## 5.2. Función principal

Hay que recordar que hay que activar la referencia *Microsoft Word Object Library* [15] para que el VBA pueda trabajar con objetos con los que trabaja Microsoft Word.

La función principal se activa cuando el usuario presiona el botón que a la vista del usuario tiene el nombre de "Empezar corrección" mediante el evento *Click* [16] asociado al botón.

Primeramente, se borra la zona de la hoja del Excel donde se presentan los resultados por si ha quedado algún dato de algún análisis anterior. Después llama a las funciones en función de si su casillero ha sido marcado por el usuario y a la vez va actualizando la barra de progreso conforme se van llamando las funciones.

Hay una casilla llamada *Chck\_Todo* que incluye todas las funciones y por lo tanto ejecuta todas las funciones y posteriormente desactiva todas las casillas por si el usuario activa otra casilla en el formulario para evitar que se ejecute dos veces y se produzcan errores. Después se guarda el documento modificado con un nombre diferente en la misma ubicación de la memoria y se cierra para tener el documento original y el modificado. Posteriormente se redacta el informe.

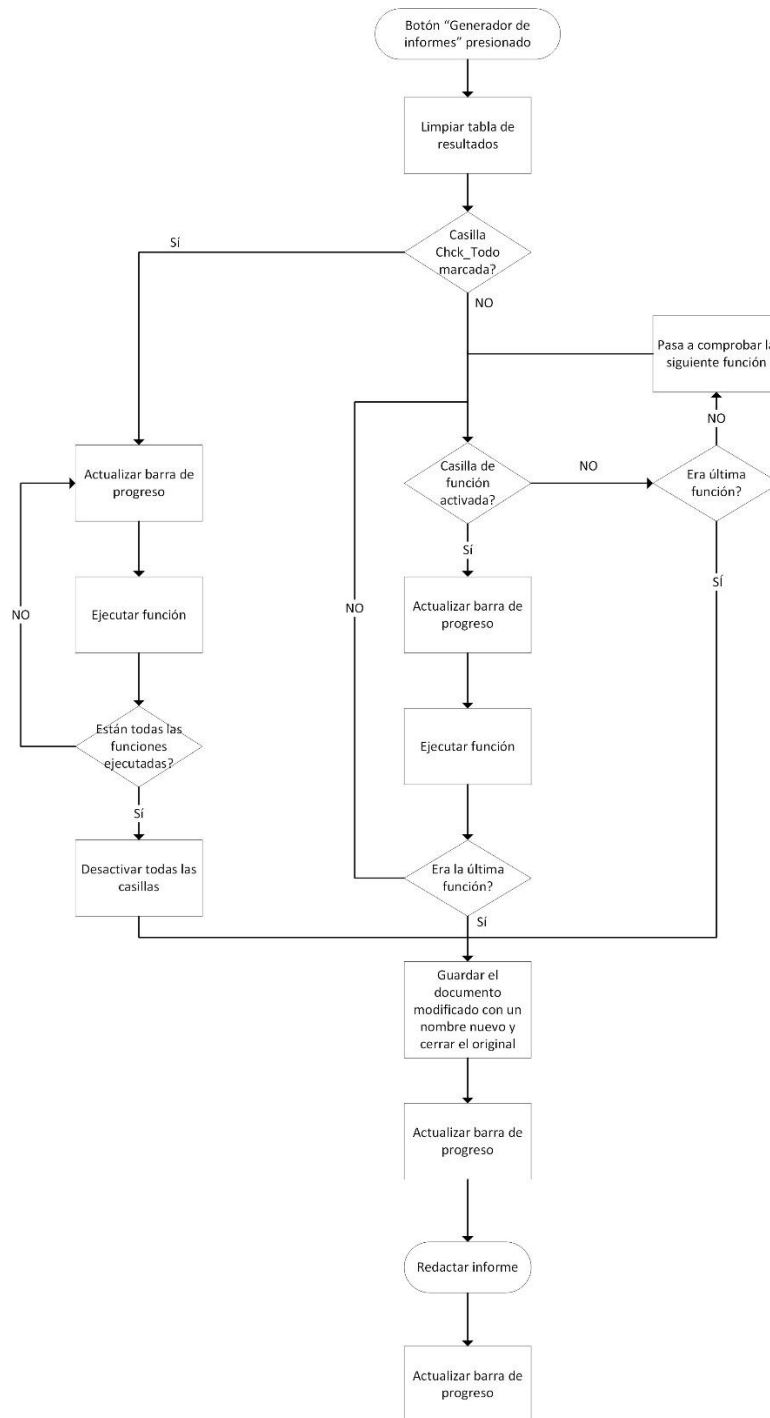


Fig. 5.3 Diagrama de flujo de la macro de la función principal. Fuente: Propia

### 5.3. Control de cambios

El control de cambios es una herramienta que se implementó en el trabajo de Pau [3] que servía para llevar un registro de los cambios que hacían las diferentes funciones de corrección en el documento y para consultar con el usuario si estaba de acuerdo con estos cambios o no.

En cuanto a las mejoras de esta función además de informar del número de errores que hay en un parámetro determinado gracias al proceso de verificación y preguntar si los quiere corregir, ahora los corrige desde el control de cambios y se adapta al tipo de parámetro que está corrigiendo. Es decir, diferencia entre la naturaleza de corrección de un parámetro con otro ya que por ejemplo el control de cambios te puede decir el número de faltas de ortografía que hay en la memoria que será un número en concreto, pero no podrá decir ningún en concreto si está corrigiendo la portada. En ese caso o estará bien o estará mal. El aviso será distinto. Entonces esta nueva mejora hace que el control de cambios se adapte al parámetro que está verificando.

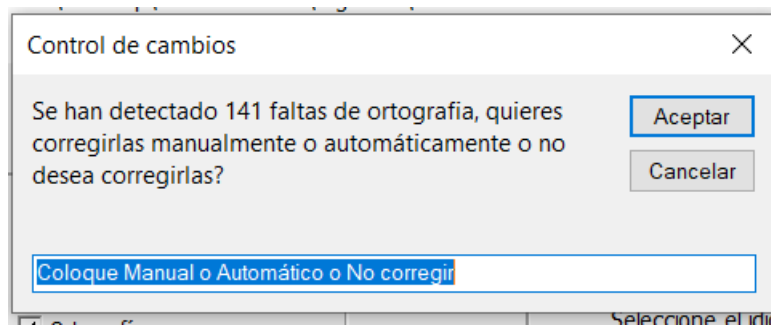


Fig. 5.4 Respuesta del control de cambios al verificar la ortografía. Fuente: propia.

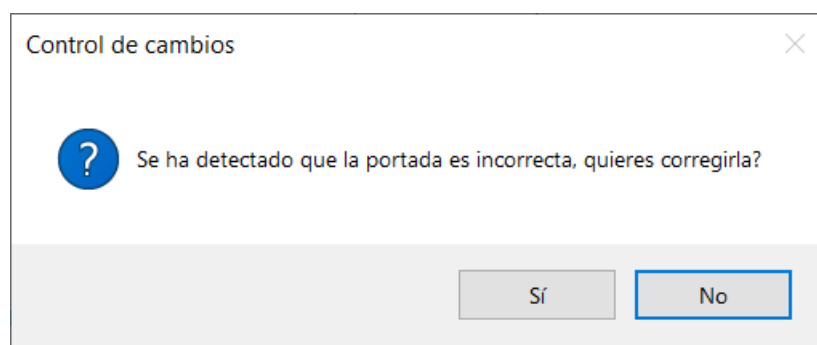


Fig. 5.5 Respuesta del control de cambios al verificar la portada. Fuente: propia.

Para conseguir esta adaptación ha sido necesario que el código del control de cambios se integre en el código de cada función adaptándose a sus necesidades. Aun así,



podemos distinguir un patrón que sigue el código macro del control de cambios en cada función:

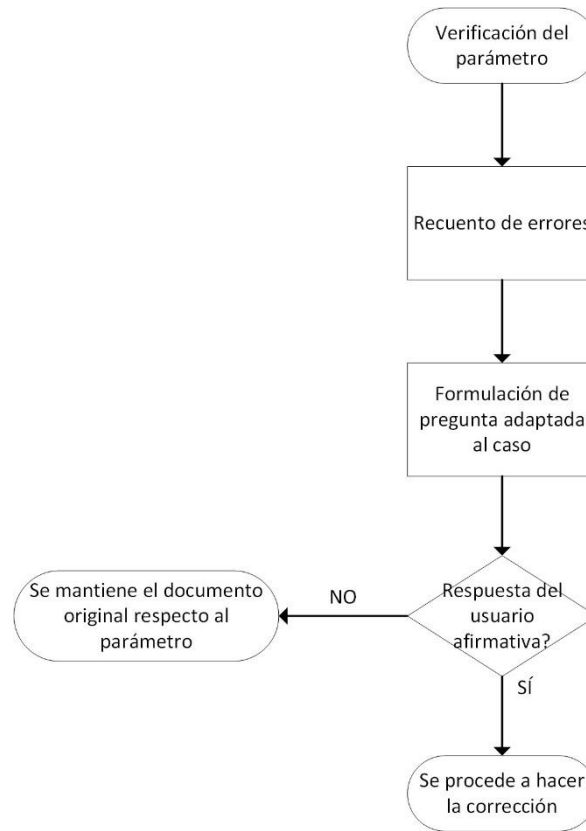


Fig. 5.6 Diagrama de flujo general de la macro de control de cambios. Fuente: propia

Como se puede apreciar en la imagen, el código integrado del control de cambios se sitúa entre el proceso de verificación y el proceso de corrección en general. Hay casos en que se solapan con el proceso de corrección como en el caso de la portada que empieza a hacer más preguntas, pero eso lo explicaré en el siguiente. El proceso de corrección se activará en función de la respuesta del usuario. Si la respuesta es afirmativa, empieza el proceso de corrección. Si es negativa, el proceso de corrección no se activa y se mantiene el documento original respecto al parámetro que se está verificando en ese momento. Y por último cabe mencionar que el método *Trackrevisions* [17] mencionado en el trabajo de Pau no me ha servido mucho en este enfoque, pero considero que es una herramienta muy potente que se le puede sacar partido con un planteamiento distinto.

## 5.4. Corrección de portada

En cuanto a la corrección de la portada, primero verifica si es correcta en base de la existencia de las palabras autor, director y convocatoria teniendo en cuenta el idioma que ha seleccionado el autor.

Una vez repasada la verificación pasamos a la corrección. En el caso de que no se encuentren estas palabras en el idioma deseado, el control de cambios nos avisará de que la portada es incorrecta y de si queremos corregirla. En caso afirmativo, primeramente, se eliminarán todos los elementos de la portada actual y se irán colocando los elementos genéricos de la portada siguiendo con el formato establecido por la plantilla de la ETSEIB [4]. Cuando llegue a un campo no genérico como el nombre del TFE o del autor, el control de cambios hará una pregunta al usuario referente a ese campo mediante la función *Inputbox* [18] y la respuesta será colocada en el rango correspondiente. De esta forma se conseguirá una portada con el formato y datos deseados. También tiene en cuenta el idioma seleccionado por el usuario ya que los campos genéricos serán del idioma que el usuario haya puesto en el formulario.

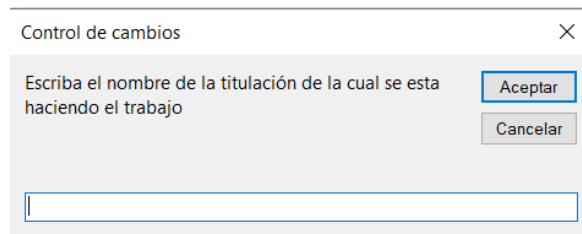


Fig. 5.7 Ejemplo de pregunta que hace el control de cambios

El efecto secundario al que no he podido encontrar solución es que cuando respondes una pregunta el programa te lleva al Word y te enseña como lo ha puesto y para volver a la aplicación tienes que hacerlo manualmente.



Fig. 5.8 Formato de portada definida por la plantilla de la ETSEIB. Fuente: ETSEIB

Entonces, la macro se resume en el siguiente diagrama:

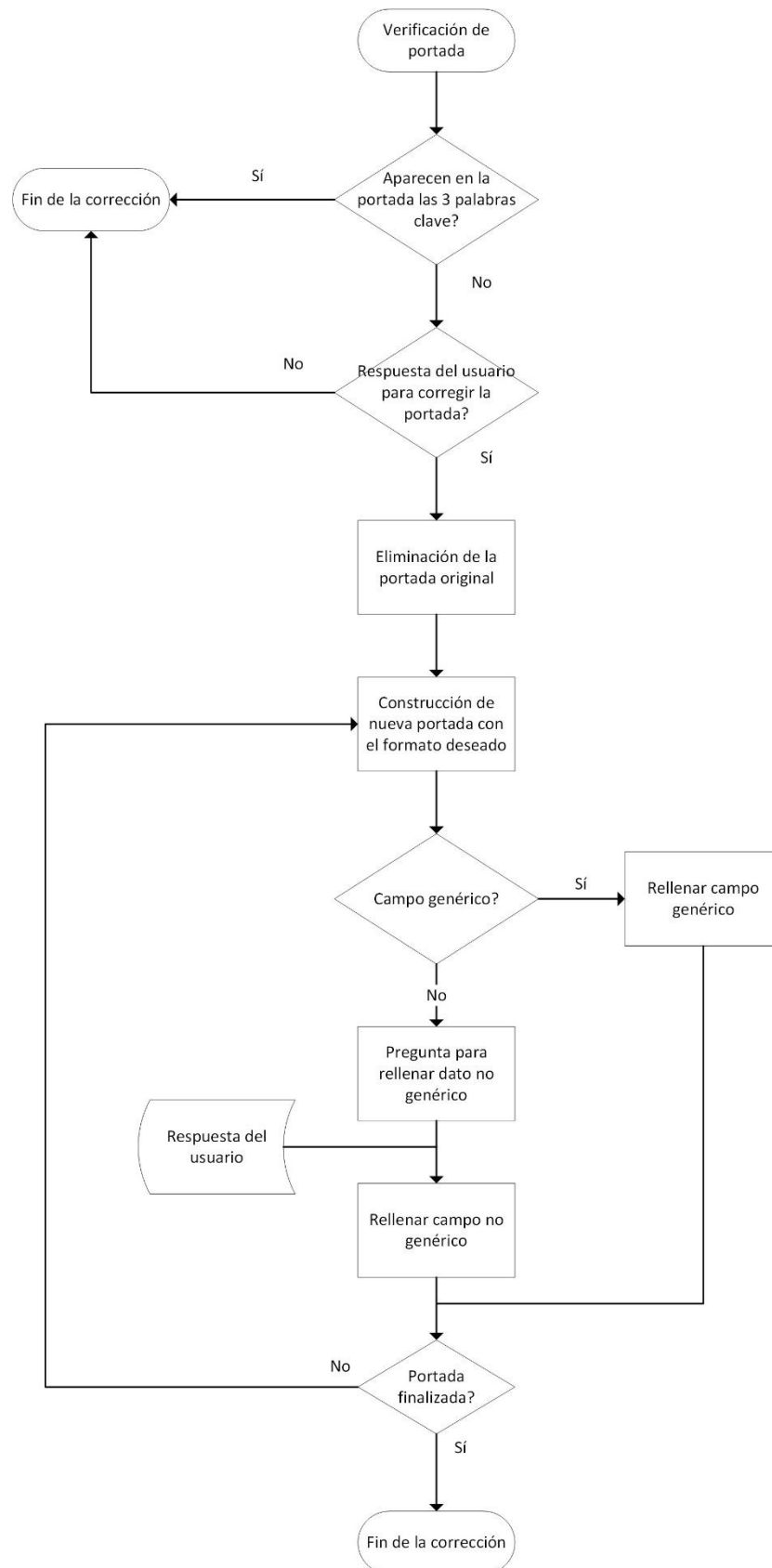


Fig. 5.9 Diagrama de flujo de la macro de corrección de la portada. Fuente: propia.

El diagrama de la imagen se repite para los 3 idiomas con sus respectivos cambios de idioma en los campos genéricos y en la formulación de las preguntas.

Una propuesta de mejora sería encontrar la forma de volver al programa automáticamente después de responder cada pregunta o que no se vaya al Word en primera instancia.

## 5.5. Corregir índice

Repasando brevemente lo que hacía esta función basándome en los trabajos de Ignasi [2] y Pau [3], esta función se basaba en el objeto *TablesOfContents* [19] para comprobar si hay un índice. En el caso de que hubiera índice, se comprueba el formato del índice siguiendo los estándares de la plantilla de la ETSEIB [4].

Bien, por mi parte lo que hice fue añadir que en el caso de que no encuentre un índice, el programa se vaya a la página 5 con el método *GoTo* [20] que es donde debe estar el índice según la plantilla y se inserta un salto de sección para insertar un índice mediante el método *Add* [21] del objeto *TablesOfContents*. Todo esto siempre y cuando el usuario dé el consentimiento mediante el control de cambios. A la hora de insertar el nuevo índice se tienen en cuenta todos los factores de formato deseados para el índice. El único cabo suelto para la creación de este índice depende de si los títulos de todo el TFE tienen el formato adecuado para ser detectados y añadidos al índice. Por eso, en el parámetro *AddedStyles* del método *Add* he seleccionado el estilo "Título 1" que es una colección de formatos genéricos que se suelen encontrar en todos los documentos Word para que de esta forma también detecte los subapartados. Posteriormente se actualiza el nuevo índice para que los capítulos se correspondan con las páginas y se añade la palabra "Sumario" sobre el índice.

Este código se ha hecho bajo la suposición de que solo hay uno o cero índices (*TableOfContents* [22]) ya que no figura en la plantilla oficial y es una posibilidad muy remota de que haya más de una tabla de contenidos en un TFE. En todo caso se usarían listas de números o viñetas o tablas que son más intuitivas y sencillas de usar y manipular.

Algunas propuestas de mejora serían detectar todos los títulos de la memoria y modificar su formato de tal forma que sean detectados y añadidos al nuevo índice o detectar el formato de los títulos e introducir este estilo de formato en la colección de formatos que detecta el índice. Otra propuesta sería tener en cuenta la posibilidad de que pueda haber más de una tabla de contenido y detectar la que se quiere comprobar de alguna forma como por ejemplo viendo si esa tabla de contenidos posee la palabra "Sumario".

La macro se resumiría en el siguiente diagrama:

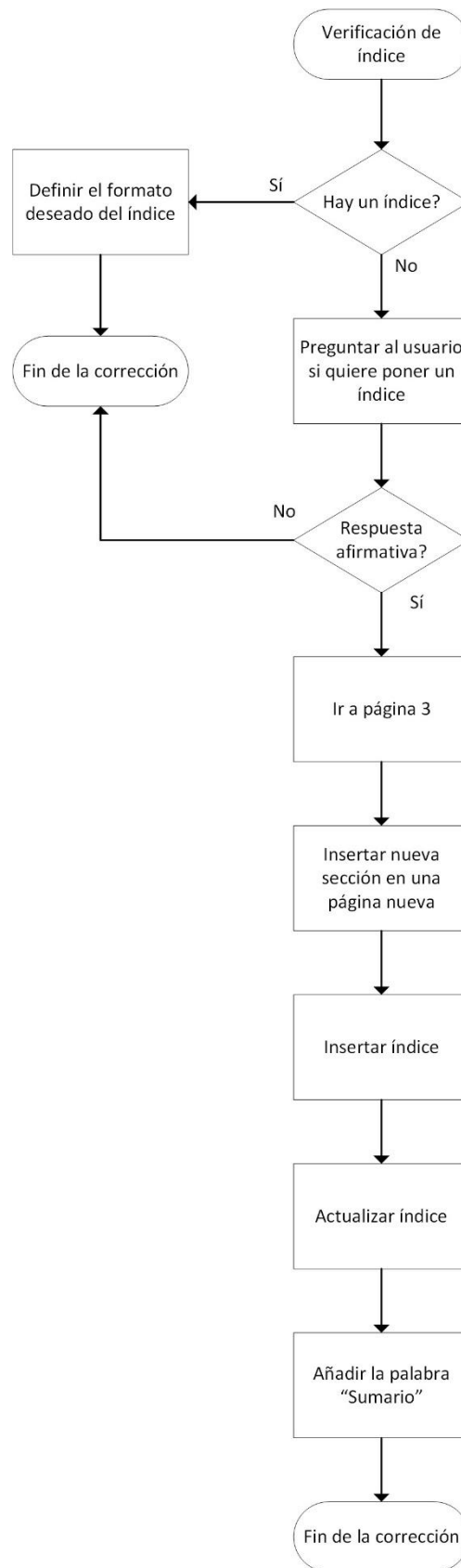


Fig. 5.10 Diagrama de flujo de la macro de corrección del índice. Fuente: propia.

## 5.6. Corregir encabezados

Este programa comprueba y corrige los encabezados según las directrices de la plantilla de la ETSEIB. Esta plantilla establece que la portada no debe llevar encabezado y que los encabezados de las páginas pares e impares deben ser diferentes.

- En las páginas impares se comprueba si existe la palabra "Pág." en los tres idiomas disponibles y si hay al menos tres palabras.
- En las páginas pares se comprueba si existen las palabras "Pág." y "Memoria" en los tres idiomas disponibles.

En el caso de que no se cumplan alguna de estas condiciones, se corrigen mediante el control de cambios.

En cuanto a este parámetro no he logrado encontrar una mejora relevante. Pero lo que sí he encontrado es un error en la verificación de los encabezados para las secciones de una página.

Para empezar hay que tener en cuenta que para acceder a un encabezado primero tienes que acceder a la sección que engloba el encabezado. Entonces, en cada sección hay 3 tipos de encabezado:

- El encabezado de la primera página de la sección (*Headers(wdHeaderFooterFirstPage)* [23]).
- El encabezado de las páginas pares de la sección (*Headers(wdHeaderFooterEvenPages)* [23]).
- El encabezado de las páginas impares de la sección (*Headers(wdHeaderFooterPrimary)* [23]).

Por otro lado, se analiza el documento mediante la iteración de las diferentes secciones del documento y se hacen comprobaciones separando las páginas pares de las impares de cada sección. Entonces, los errores están en el caso concreto de que la sección tenga una página. Da igual que esta sea par o impar porque dará un error extra incorrecto. Más detalladamente, en el caso de que la página de esta sección sea par, cuando se analicen las páginas pares de esta sección lo analizará correctamente y dará lo que tenga que dar. Pero cuando se analicen las páginas impares de esta sección, al no haber impares, el programa detecta una página virtual impar que no figura en el documento del TFE y al no tener definido el encabezado el programa lo interpreta como un error de página impar en esta sección de una página. Y lo mismo pasa en el análisis de las páginas pares en el caso

de que la sección solo tenga una página y esta página sea impar.

Para paliar el error lo que se ha hecho es calcular el número de páginas de cada sección mediante el método *Information(wdActiveEndAdjustedPageNumber)* [24] que devuelve la última página de la sección seleccionada y comprobar si la sección tiene una o más páginas. Si la sección tiene más de una página proceder con la verificación o corrección. En el caso de que solo tenga una página se comprueba si es par o impar en función del análisis que se esté realizando:

- Si se está realizando el análisis de las páginas pares, se comprueba si la página es par o no mediante el operador *Mod* [25]. En el caso de que sea par, se procede con la verificación o corrección de lo contrario no se procede.
- Si se está realizando el análisis de las páginas impares, se comprueba si la página es impar o no mediante el operador *Mod* [25]. En el caso de que sea impar, se procede con la verificación o corrección de lo contrario no se procede.

Para ver el código macro del filtro de errores de una forma más esquematizada:

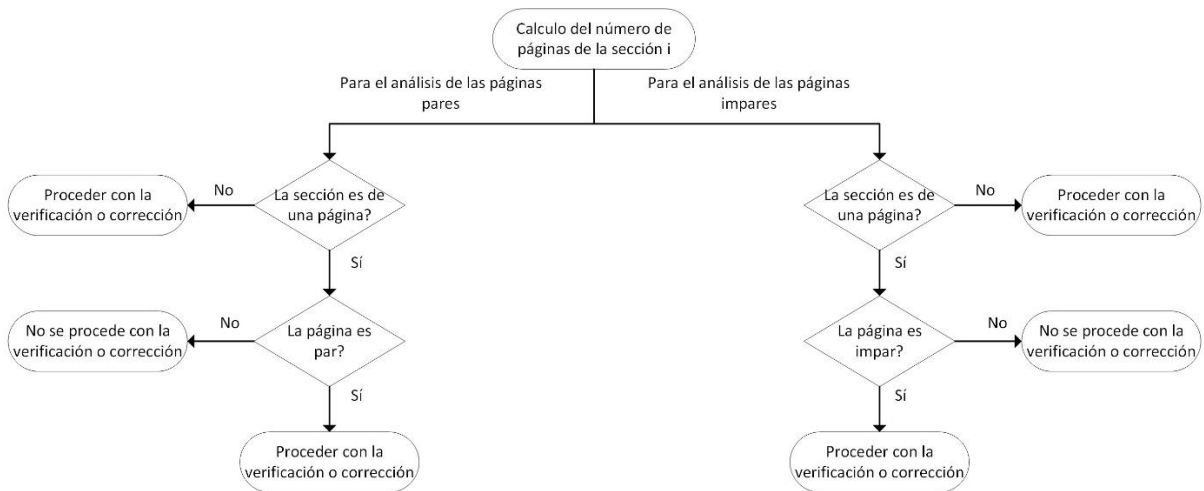


Fig. 5.11 Diagrama de flujo de la macro de filtrado de errores de los encabezados.

Fuente: propia.

Este filtro se aplicará en todas las pruebas lógicas referentes al verificado de los encabezados detallado en el trabajo de Ignasi [2]. Por último, otro cambio relevante a mencionar es la reorganización del código debido a la integración del control de cambios en el código de la función explicado detalladamente en el apartado 5.3.

## 5.7. Corregir pies de página

En cuanto a este parámetro, la función obtenida en los anteriores trabajos consiste en comprobar si los pies de página de todas las páginas menos la portada contiene una imagen del escudo de la facultad y añadir la imagen en el caso de que se detecte la ausencia de una en una sección en concreto. Siguiendo con la plantilla, estas imágenes deben estar posicionadas en función de si la página es par o impar:

- Si la página es par, el escudo debe estar situado a la izquierda del pie de página.
- Si la página es impar, el escudo debe estar situado a la derecha del pie de página.

Una vez más, para acceder a los pies de página es necesario el método de las secciones *Footers* [26]. Al igual que los encabezados, los pies de página tienen los mismos 3 tipos de pies. Uno para la primera página, otro para las páginas pares y otro para las impares.

Bien, aparte de incorporar el código correspondiente a este apartado de los dos trabajos en los que me baso principalmente ([2] y [3]) unidos y ordenados, una mejora que he hecho ha sido incorporar el filtro de errores adaptado a los pies de página explicado detalladamente en el apartado 5.6 de este trabajo. La estructura es la misma que en los encabezados. La diferencia es que se usa el método de las secciones que selecciona los pies de página (*Footers* [26]) en lugar de los encabezados.

La mejora más relevante en este parámetro de formato es la comprobación de si la imagen está bien colocada a la derecha o a la izquierda del pie de página en función de si la página es par o impar. Esta mejora no hubiera sido posible sin el descubrimiento de la propiedad *Left* [27] de un objeto *ShapeRange* [28] que determina si la imagen está situada a la derecha o a la izquierda del pie de página.

Por lo tanto, el código de los pies de página junto con las mejoras seguiría el siguiente orden:

1. Comprobar si el encabezado de la portada es distinto a los demás y también ver si los pies de página de las páginas pares son distintos a las pares en todo el documento.
2. Comprobar si existe para las páginas pares de cada sección la imagen del escudo de la facultad y ver si está bien colocada. En este punto se aplica el



filtro de errores.

3. Repetir el punto 2 para las páginas impares.
4. Recuento de errores y preguntar al usuario si quiere corregir mediante el control de cambios.
5. Si la respuesta es afirmativa, se procede a corregir. De lo contrario el código llega a su fin. En la corrección también se aplica el filtro de errores.

Resumiendo, en un diagrama de flujo:

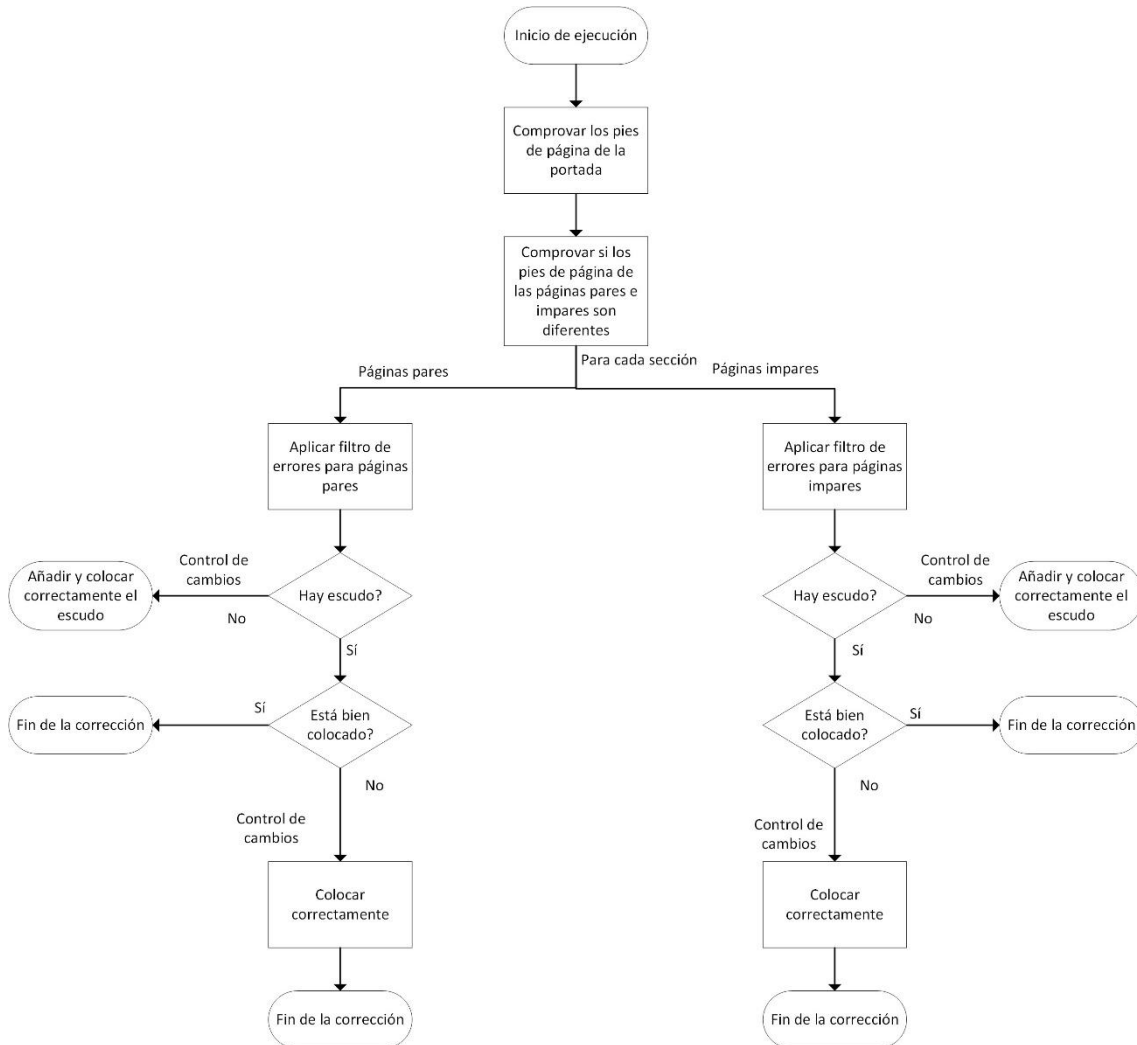


Fig. 5.12 Diagrama de flujo de la verificación y corrección de los pies de página enfocada en las novedades. Fuente: Propia

Como podemos ver en el diagrama de flujo de la imagen se resume todo el proceso de verificación y corrección. Aun así faltan detallar cosas como el hecho de corregir los pies de página de la portada si hace falta e imponer que los pies de página de las páginas pares e impares sean diferentes en el caso de que no lo sean. Estas funciones las tengo en

cuenta en el programa. Lo que quiero explicar más detalladamente son las novedades como esta comprobación de la posición adecuada de las imágenes en los pies de página. Por no mencionar que estos detalles que obvio ya están detallados en el trabajo de Pau [3].

## 5.8. Glosario

Esta es una función nueva que he implementado gracias a la idea de mi tutor. El objetivo principal es comprobar que existe un glosario y en el caso de que exista uno, comprobar que todas las palabras dentro del glosario aparezcan como mínimo una vez más en el documento.

En primer lugar, para comprobar que haya un glosario hemos usado buscado la palabra glosario en todo el documento mediante el método *find* [29] combinado con la estructura de control de flujo *Do – Until* [30]. Si encontramos esta palabra en el documento damos por hecho que hay glosario. De lo contrario no hay glosario y se terminaría la función.

Vamos al caso de que sí hay glosario. Pasamos a comprobar que las palabras del glosario aparezcan una o más veces en el texto. Primero tenemos que identificar estas palabras del glosario diferenciándolas de las demás de alguna manera. Como en este caso la plantilla de la ETSEIB no especifica el formato de este capítulo ya que es opcional, podemos imponer algunas condiciones que nos puedan ayudar a diferenciar y seleccionar las palabras del glosario. Las 3 condiciones que son:

- Que la palabra esté antes de “:” ya que al ser una palabra del glosario una vez aparezca se procederá a su definición y comúnmente se usan los dos puntos para definir alguna palabra.
- Que esté en negrita. Son palabras especiales así que es lógico destacarlas.
- Que se fuente sea de 11 puntos para diferenciar la palabra de algunos títulos que suelen estar en negrita y a veces tienen dos puntos delante.

Como se puede apreciar, se intenta que las condiciones sean lo más genéricas y comunes para que esta función se pueda aplicar al máximo número de casos. Gracias a estas condiciones ya podemos distinguir y seleccionar las palabras del glosario. Entonces lo que se hará será guardar cada palabra en la variable “Plbra” y hacer una búsqueda de esta palabra por todo el texto y usar un contador (variable “p”) para contar las veces que aparezca en el texto. De esta forma se logrará identificar si alguna palabra del glosario no aparece como mínimo por segunda vez en el texto y se anotará en la tabla de resultados.

Una vez acabado el análisis, el control de cambios informa al usuario cuantas palabras no han aparecido en el texto.

Para esquematizarlo todo:

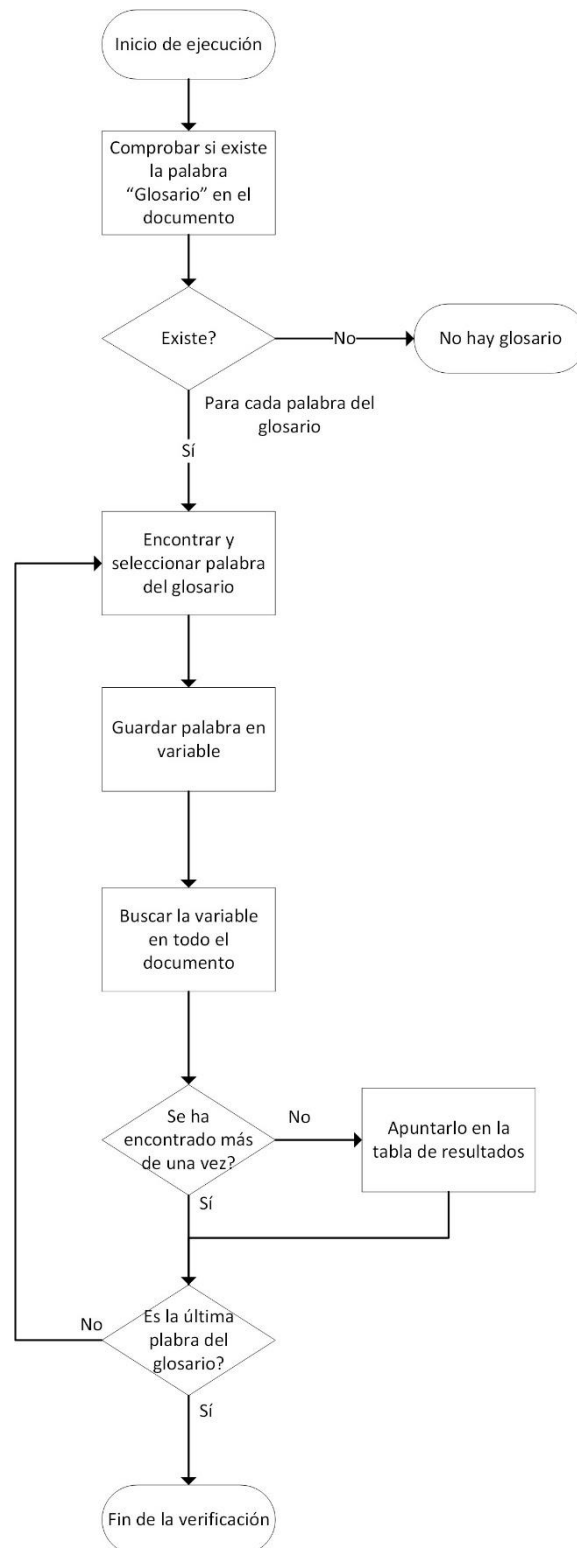


Fig. 5.13 Diagrama de flujo de la nueva función del Glosario. Fuente: Propia.

## 5.9. Corregir párrafos

Esta función lo que hace es comprobar si los párrafos de todo el documento siguen el formato impuesto por la plantilla. Para repasarlo una vez más:

- Tipo de letra: Arial
- Tamaño de fuente: 11
- Texto justificado

Se verifican estas 3 condiciones para cada párrafo y los párrafos que no lo cumplan se corrigen.

En cuanto a los párrafos, había poco que mejorar ya que con el código que había se cumplía con los objetivos de verificar y corregir los párrafos.

Una mejora discreta que hice fue que cuando se eliminan las anotaciones duplicadas en la tabla de resultados mediante el método *RemoveDuplicates* [31] se borran los bordes de la tabla. Por lo tanto, mediante el método *End(xlDown)* [32] y la grabadora de macros conseguí seleccionar esta zona que tiene los bordes borrados y volver a poner sus respectivos bordes. Por lo tanto, el diagrama de flujo de esta pequeña mejora quedaría:

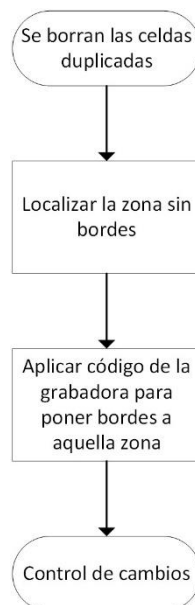


Fig. 5.14 Diagrama de flujo de mejora en código de párrafos. Fuente: Propia

Por último, cabe mencionar que después de esta mejora se activa el control de cambios informando el número de errores que ha habido y si autoriza el usuario corregirlos.

## 5.10. Ortografía

Esta función se encarga de corregir las faltas de ortografía del documento. Como siempre un pequeño resumen de lo que hace hasta ahora:

Primero de todo se asigna el lenguaje con el cual el documento está escrito en función de lo que haya marcado el usuario en el apartado de idioma. Una vez definido el idioma del documento, se detectan los errores mediante el método *SpellingErrors* [33] y se van añadiendo a la tabla de resultados para después ser corregidas automáticamente mediante el método *GetSpellingSuggestions* [34] escogiendo la primera sugerencia que se presente. La pega de este tipo de corrección es que no tiene en cuenta el contexto del texto y por lo tanto se suele equivocar varias veces. Otra limitación son las expresiones en otros idiomas que pueden aparecer en el documento.

Lo mejora que he añadido consiste en que mediante el control de cambios el usuario pueda elegir entre dos métodos de corrección: el automático que es el que ya estaba y otro más manual, pero se asegura de corregir correctamente todos los errores ortográficos detectados en el documento. Este segundo método se basa en activar el método *CheckSpelling* [35] que lo que hace es activar una ventana en el Word que lo que hace es ir error por error y nos da diferentes sugerencias de corrección o nos deja que lo corriamos de acuerdo al contexto del texto. Usando este método se eliminarían las limitaciones con las expresiones en otros idiomas. De esta forma el programa solo tiene que avisar al usuario que debe ir al Word que está en segundo plano una vez esta ventana este activada.

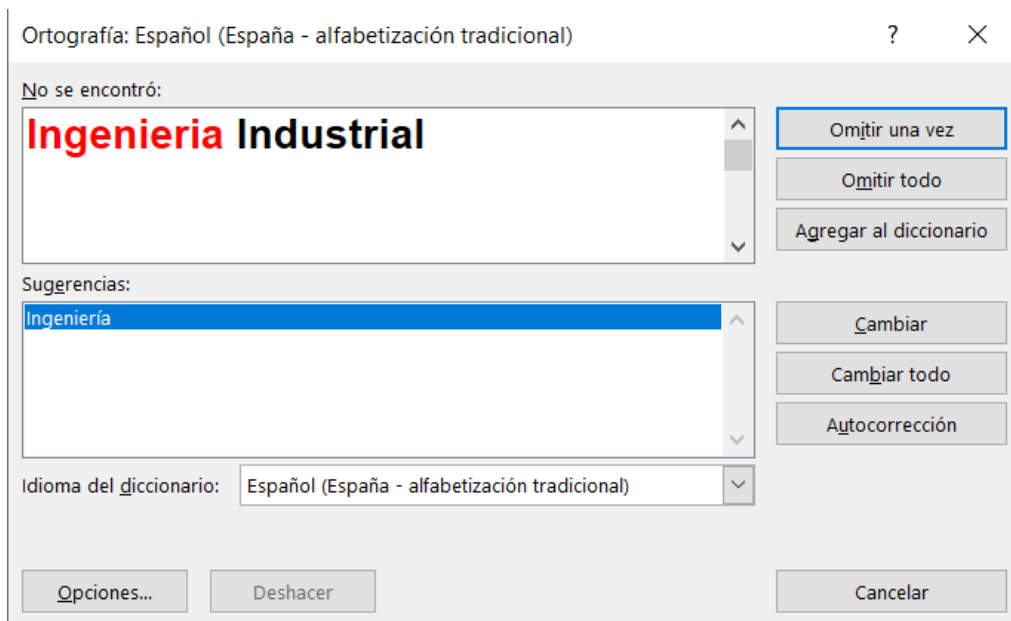


Fig. 5.15 Ventana de corrección ortográfica de Word. Fuente: propia.

Una vez hemos acabado de corregir los errores simplemente cerramos la ventana de corrección ortográfica y volvemos al programa que nos estará esperando con una pregunta del control de cambios referente a una nueva función que he decidido incorporarla al parámetro de la ortografía.

Esta nueva función consiste en la búsqueda de espacios dobles en el documento, apuntarlos en la tabla de resultados y sustituirlos por un espacio normal. Para llevar esto a cabo primero se buscarán todos los espacios dobles mediante el método Find [28] combinado con la estructura de control de flujo *Do – Until* [29]. De esta forma se encontrarán todos los espacios dobles y se anotarán en la tabla de resultados. Posteriormente se hará el recuento de espacios dobles detectados y se preguntará al usuario si quiere corregirlos mediante el control de cambios. En el caso de que se decante por corregirlos, se seleccionarán todos los espacios dobles uno por uno y se irán sustituyendo por un espacio normal mediante la propiedad *Text* [36].

Por lo tanto, el diagrama de flujo de la corrección de la ortografía quedaría:

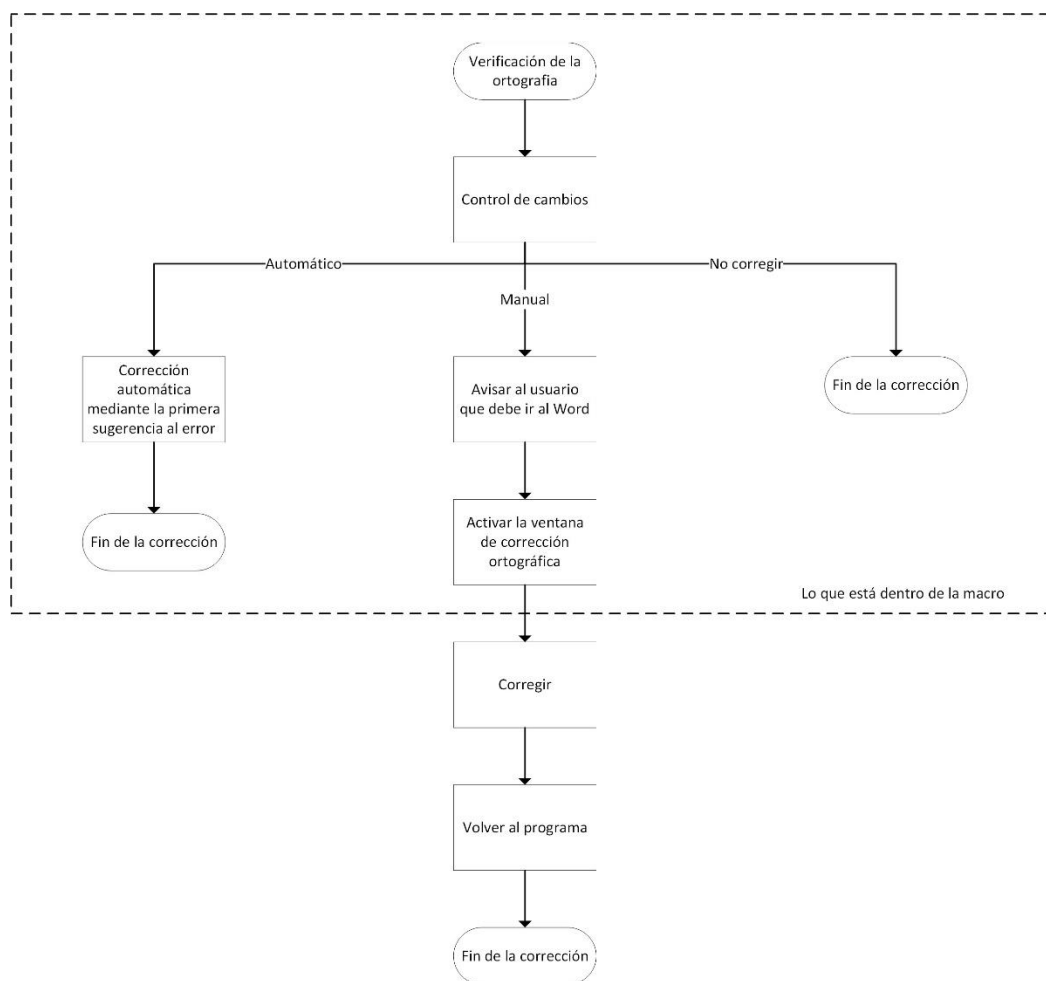


Fig. 5.16 Diagrama de flujo de la corrección de la ortografía. Fuente: Propia.

Cabe mencionar que lo que está fuera del recuadro de líneas discontinuas lo tiene que hacer el usuario manualmente ya sea el usar la ventana de corrección ortográfica de Word o el volver al programa una vez se han corregido los errores.

Por otro lado, el diagrama de flujo de la nueva función sería:

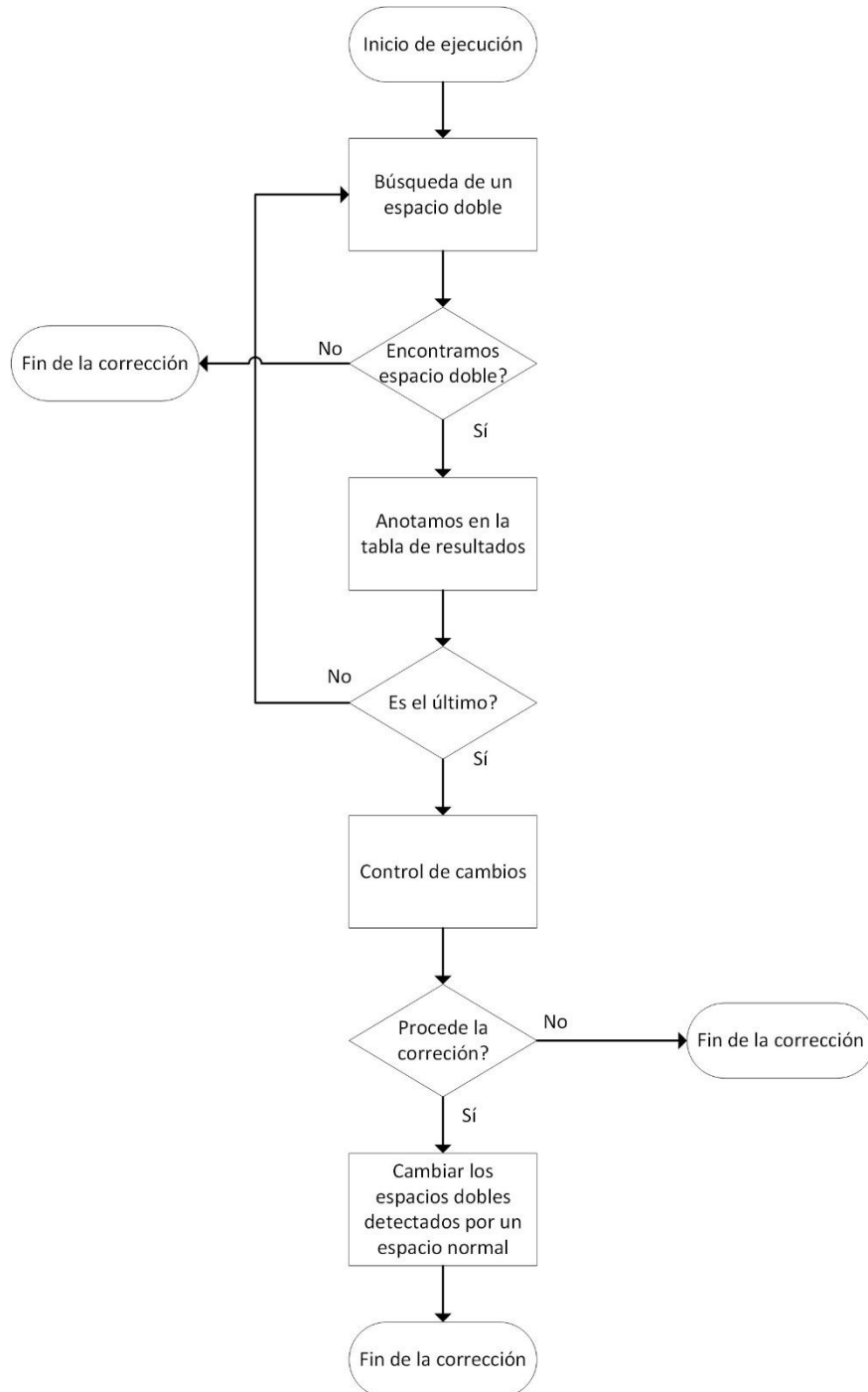


Fig. 3.17 Diagrama de flujo de la macro que detecta y corrige los espacios dobles.

Fuente: propia.

## 5.11. Verificar bibliografía

Esta función lo que hace hasta el momento es comprobar si existe la bibliografía mediante la búsqueda de la palabra "Bibliografía" por todo el documento y verifica que las URL de las referencias tengan las fechas actualizadas comprobando que las fechas sean superiores al año 2000.

La mejora que yo he implementado consiste en comprobar que todas las referencias bibliográficas se citan en el texto y anotar en la tabla de resultados las referencias que no se han citado. Para ello, he tenido en cuenta que cada referencia bibliográfica se presenta según el formato que establece la plantilla. Es decir, el siguiente formato: [1]. Este formato se puede poner en forma de lista definiéndolo como un nuevo formato de número en el Word y trabajar con este tipo de numeración. O se puede hacer manualmente línea tras línea manteniendo el formato. La gran diferencia entre estas dos formas de presentar la bibliografía es que con el formato de numeración los números de referencia no se detectan cuando los buscas con el método *Find* [28] y cuando se hace línea por línea sí que se detectan los números de las referencias en la bibliografía.

Por lo tanto, en primer lugar, se comprobará si existe una lista numerada en el documento con el formato que requiere la bibliografía mediante el método *ListString* [37]. Si se encuentra esta clase de lista, se activa un contador (variable *k*) que hará que no se active el código adaptado a la bibliografía hecha línea por línea. Una vez detectada esta lista numerada se hace una iteración por todas las referencias dentro de la lista y por cada iteración se busca el número de la referencia en el texto. Si no se encuentra en todo el texto del documento se apunta en la tabla de resultados.

Por otra parte, si no se detecta ninguna lista con el formato que requiere la bibliografía, se procede a hacer la búsqueda de los números de referencia de la bibliografía como si se hubiera hecho manualmente. En este caso el método *Find* [28] sí que detecta los números de las referencias bibliográficas. El procedimiento es muy similar al anterior pero aquí en lugar de recorrer listas tenemos que distinguir las referencias que aparecen en la bibliografía de las citas que aparecen en el texto. Para ello comprobamos que la referencia esté en negrita con el método *Bold* [38]. Por lo tanto, se usará esta condición para identificar las referencias bibliográficas. Después se procede a buscar si aparece citada en el texto y anotar en la tabla de resultados en el caso de que no esté.

Las dos funciones están hechas de tal forma que si se escoge una no se puede usar la otra. Por último, al final de cada función se llama al control de cambios para informar al



usuario sobre el número de referencias bibliográficas no citadas en el texto. De esta manera, el diagrama de flujo quedaría de la siguiente manera:

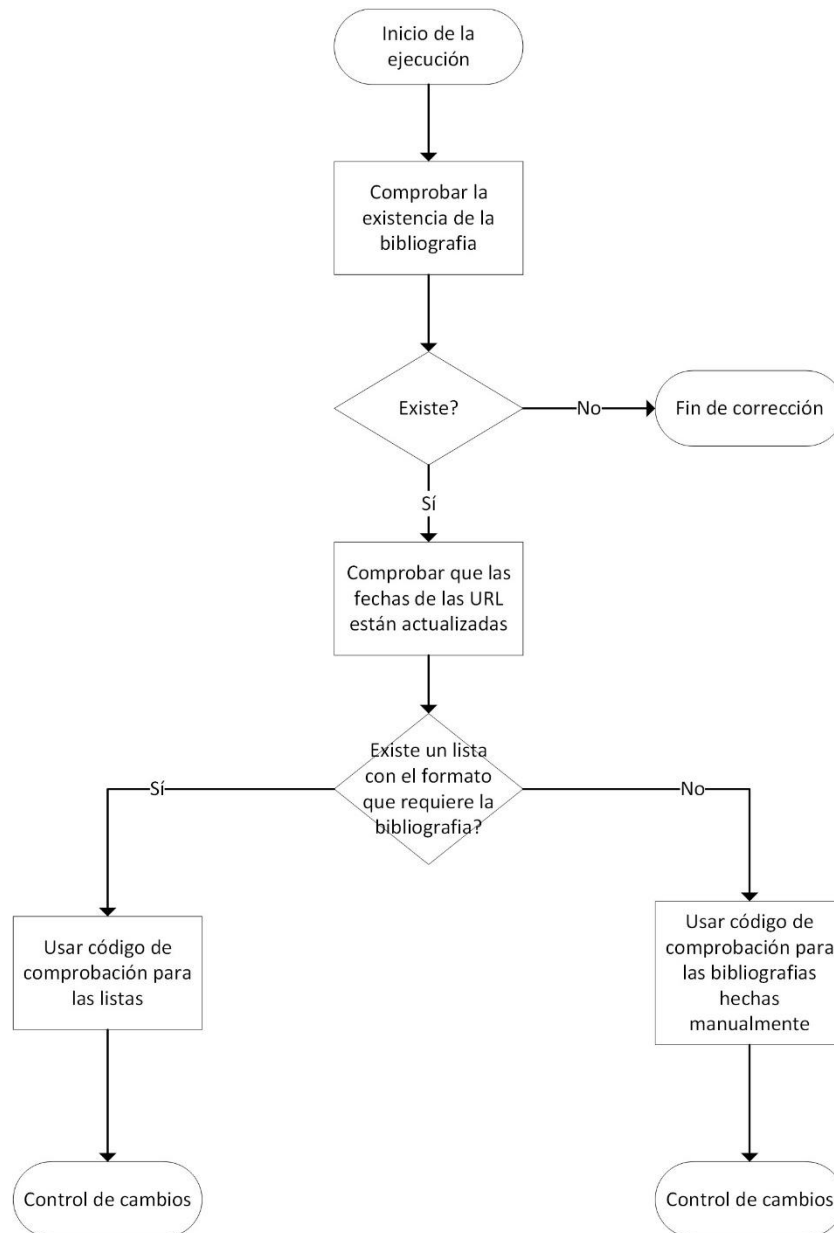


Fig. 3.18 Diagrama de flujo de la verificación de la bibliografía. Fuente: propia.

## 5.12. Corregir citas

Esta función apareció por primera vez en el trabajo de Ignasi [2] y se encarga principalmente de comprobar que las citas que van apareciendo en el texto estén ordenadas.

Pensaba no añadir esta función, pero consideré que me podría servir el algoritmo para comprobar el orden hecho por Ignasi [2]. Lo que hice fue aplicar esta comprobación de

orden a las referencias bibliográficas ya que si lo aplicamos a las citas del texto con el código que se tiene darían errores en bastantes casos reales ya que una referencia se puede citar más de una vez en el texto. Las referencias bibliográficas tienen un formato distinto (van entre claves y están en negrita) y no se repiten en la bibliografía.

Por lo tanto, el proceso de verificación es el mismo que se tenía en los anteriores trabajos, pero comprobando que las referencias bibliográficas estén en negrita y que después de la primera clave (I) de la referencia bibliográfica que se está buscando haya un número. Esto último se puede hacer con la función *IsNumeric*. De esta forma solo contarán las referencias que estén en la bibliografía. Una vez comprobadas todas las referencias, en el caso de que haya alguna desordenada el control de cambios informará al usuario y preguntará si quiere ordenarla. Esta es otra mejora incorporada.

Si el usuario desea ordenar las referencias, se procede a ordenar las citas. Para llevar esto a cabo, la idea clave ha sido ir iterando por las diferentes referencias bibliográficas hasta llegar a la referencia que no está ordenada y a partir de aquí se vuelven a enumerar todas las referencias. La comparación de la variable *contador* con la variable *orden* es clave para esto. El diagrama de flujo quedaría de la siguiente manera:

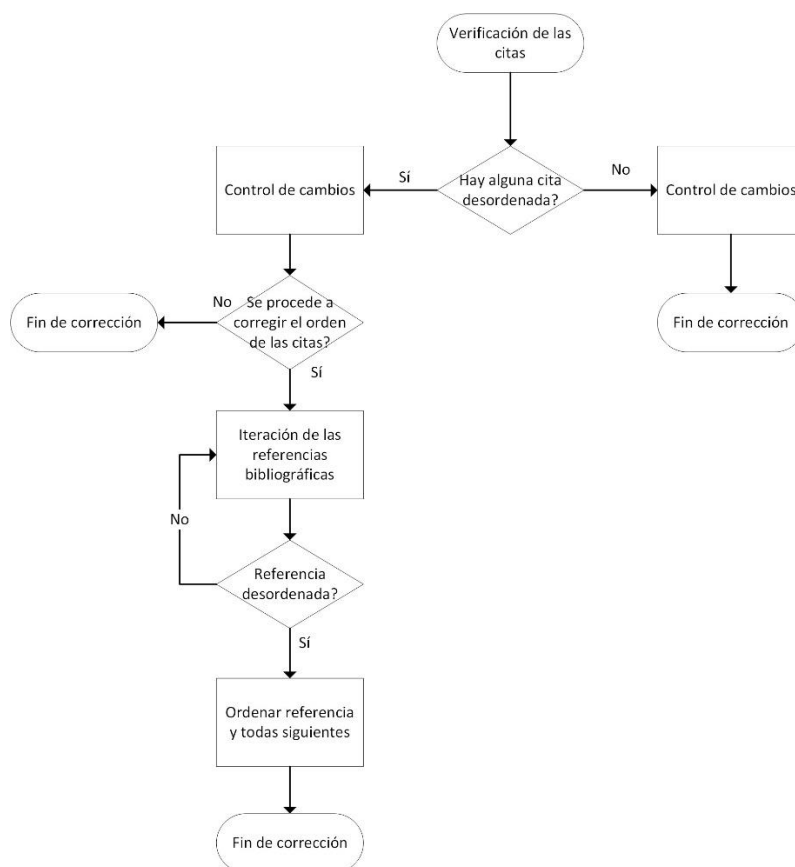


Fig. 3.19 Diagrama de la corrección del orden de las citas. Fuente: propia.

Una propuesta de mejora a esta función sería que se logre ordenar las citas del texto teniendo en cuenta que las citas se pueden repetir. Es decir, que conforme el número de la cita vaya subiendo, considere la posibilidad de que puedan aparecer los números de cita anteriores y no lo interprete como un error de orden.

### 5.13. Verificación y corrección de los pies de figura / tablas

Esta es otra de las nuevas funciones implementadas. Consiste en comprobar si los pies de figura y las tablas están correctamente citados y enumerados con el formato correcto que se impone en la plantilla.

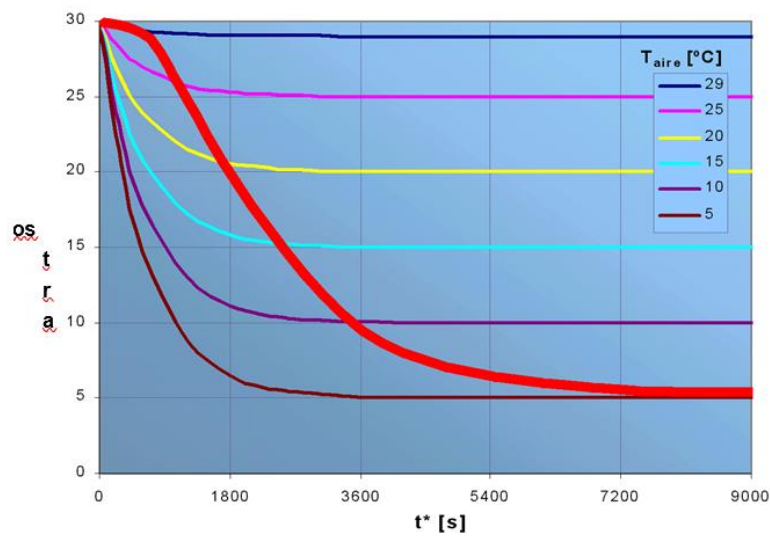


Fig. 4.1. Estimació de la corba real de temperatures del conjunt

Fig. 5.20 Ejemplo de cita de figura según la plantilla

Nos centraremos en el cumplimiento de este formato y derivados como *Figura a.b.* También se tiene en cuenta la elección del idioma para la verificación y corrección. En cuanto a los números que aparecerán en cada figura/ tabla, el primero hace referencia al capítulo de la memoria en que se encuentra la figura y el segundo al orden de aparición de la figura en el capítulo, es decir, si es la primera o segunda o tercera imagen del capítulo etc.

Para encontrar el primer número, encontraremos el número de cada capítulo y el número de página correspondiente donde empieza cada capítulo. Para ello, será de gran ayuda el texto de los hipervínculos del índice de la memoria (*Hyperlinks* [39]) siempre y cuando tenga un formato correcto (de eso se encarga la función del punto 5.5.). Estos dos números se guardarán en un diccionario (*Scripting.Dictionary* [40]). Una vez tenemos el diccionario

hecho, iteramos imagen por imagen gracias a la colección *InlineShapes* [10] y a la estructura de control de flujo *For - Next* [41]. Mediante un algoritmo que involucra el diccionario anteriormente creado y la página donde se encuentra la imagen, se consigue deducir a que capítulo pertenece la imagen seleccionada y el número de imagen dentro del capítulo. Una vez tenemos estos dos números, se comprueba si la cita de la imagen tiene un formato correcto y los números coinciden con los encontrados por el algoritmo. Se anotarán los errores en la tabla de resultados y el control de cambios preguntará si al usuario si quiere corregirlos. En caso de respuesta afirmativa se procede a corregir los pies de figura. La estructura del código de corrección es prácticamente la misma que la de verificación solo que cuando detecta errores en lugar de anotarlos procede a corregir el formato o los números o ambos según el caso. Ha resultado muy útil para esta función el método *Next* [42] del objeto *Selection* [43].

En cuanto a las tablas, el código sigue la misma estructura que el de los pies de figura las dos principales diferencias es que el formato de las citas de las tablas debe ser de la forma: Tabla a.b. La palabra "Tabla" variará en función del idioma escogido por el usuario. Hay que tener esto en cuenta a la hora de buscar la cita de cada tabla. La otra diferencia a tener en cuenta es que iteraremos de tabla en tabla mediante la colección *Tables* [44] y a la estructura de control de flujo *For - Next* [39].

Por último, pero no menos importante, esta función se adapta al idioma seleccionado.

La estructura del diagrama de flujo tanto para figuras como para tablas es la misma como se observa en las dos figuras siguientes:

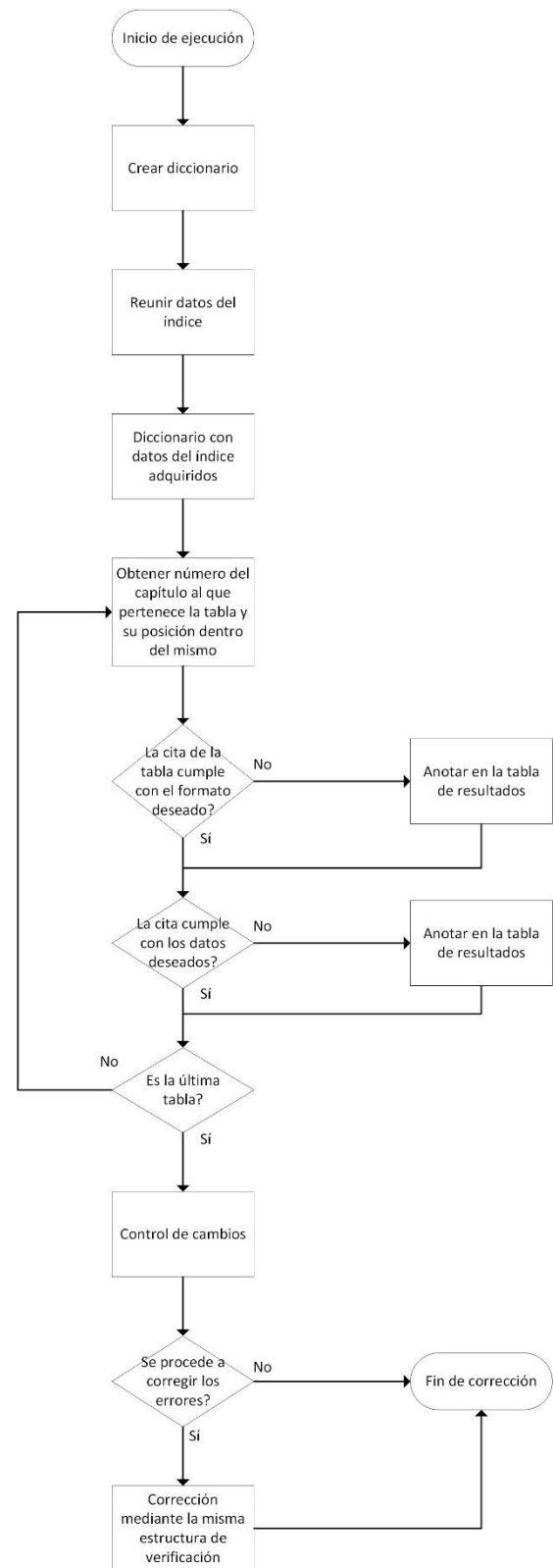
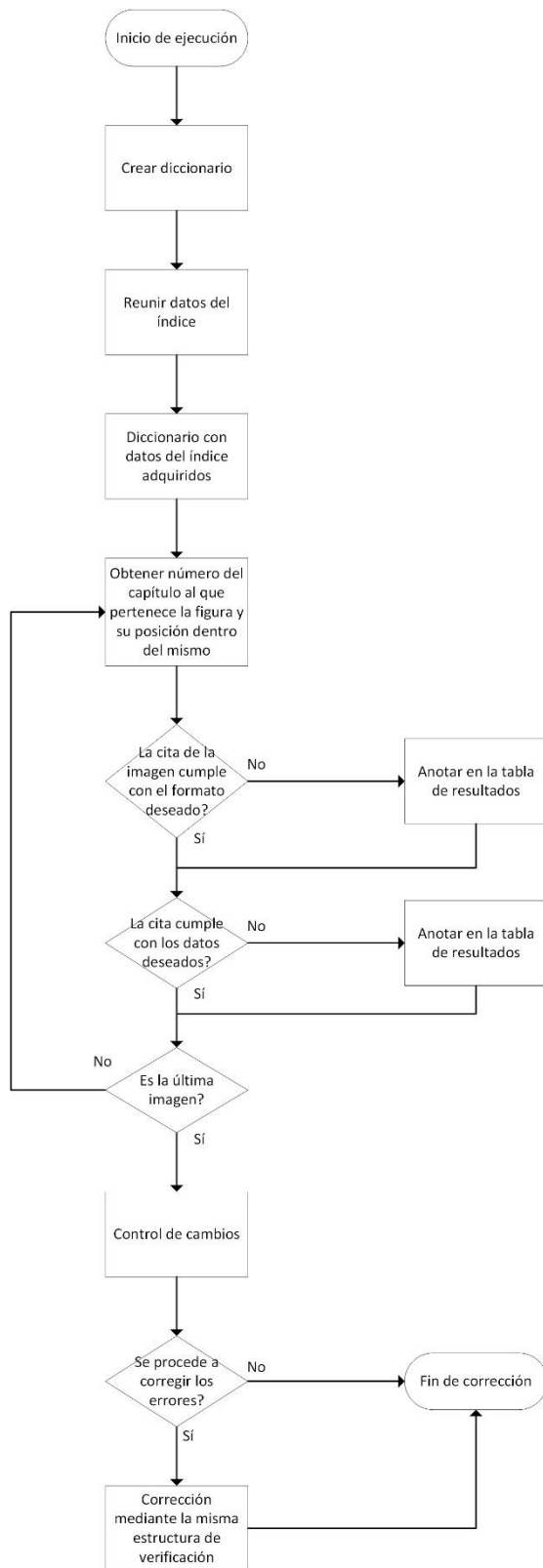


Fig 5.21 Diagrama de flujo de la macro de verificación y corrección de los pies de figura y tablas. Fuente: Propia.

## 6. Comprobación en un caso real

Para comprobar las nuevas mejoras, autoevaluarnos y detectar posibles limitaciones en la aplicación procedemos a hacer la corrección de una memoria de TFE real.

Corregiremos el TFE que precede a este, el TFM del compañero Pau Palau titulado: “Eina de correcció de la memòria d’un TFE mitjançant macros de Microsoft Excel” [3].

The image shows the 'Corrector de Formato' application interface. At the top, the title 'Corrector de Formato' is displayed in blue. Below the title, there is a section for 'Origen de datos:' with a text input field containing the file path 'C:\Users\Carlo\Desktop\Muestras TFG\ELpalasio\MEMORIA\_PAU\_PALAU (1).docx' and a 'Buscar' button to the right. Below this, there are two main panels. The left panel, titled 'Acciones a corregir', contains a list of checkboxes: 'Portada', 'Índice', 'Encabezados', 'Pie de página', 'Glosario', 'Párrafos', 'Ortografía', 'Bibliografía', 'Citas', and 'Pies figuras/ tablas'. The 'Corregir todo' checkbox is checked. The right panel, titled 'Idiomas', has a label 'Seleccione el idioma:' followed by a dropdown menu showing 'Català' and a 'Borrar' button below it. At the bottom center, there is a large 'Empezar corrección' button. Below the button is a progress bar that is currently empty, with '0%' displayed underneath it.

Fig. 6.1 Condiciones iniciales. Fuente: propia.

Como se puede apreciar en la imagen, una vez seleccionado el archivo deseado se guarda su dirección en la etiqueta de fondo blanco. Después, solo se selecciona la casilla de “Corregir todo” ya que se van a corregir todos los parámetros. Como esta memoria está en catalán, pues se selecciona como idioma de corrección el catalán. Y, por último, se puede apreciar que la barra de progreso aparece vacía ya que aún no se ha empezado el proceso

de corrección. Es decir, que no le hemos dado al botón de “Empezar corrección”.

Una vez pulsemos el botón de “Empezar corrección” se empezarán a llamar en orden a todas las funciones integradas.

## 6.1. Comprobación Portada

Empezamos con la portada, en este caso la portada presenta las 3 palabras clave de acuerdo con el idioma seleccionado: “Autor”, “director” y “Convocatòria”. Por lo tanto, la portada es correcta a pesar de que también le falta la imagen del escudo de fondo, pero al no estar dentro de los criterios de verificación no lo tendremos en cuenta.

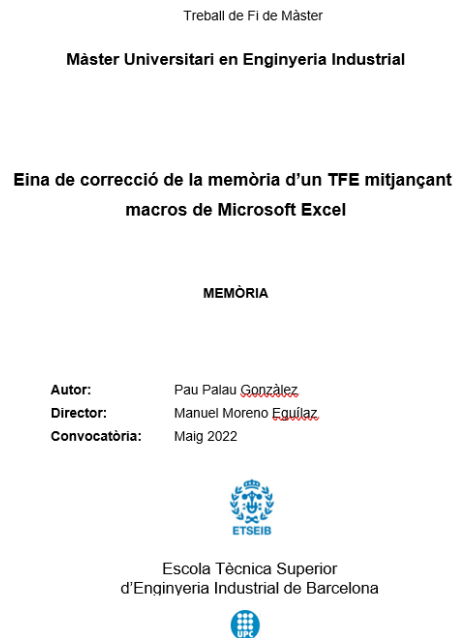


Fig. 6.2 Portada original del TFE de prueba. Fuente: [3]

Para poner a prueba las mejoras en la portada quitaremos las 3 palabras clave para que se considere incorrecta.

Entonces, como anteriormente he explicado, la función volverá a hacer una nueva portada y hará preguntas al usuario para los campos que no sean genéricos.

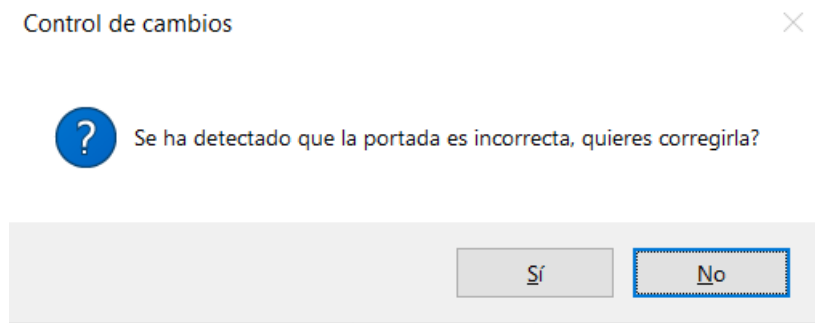


Fig 6.3 Primera pregunta del control de cambios. Fuente: propia.

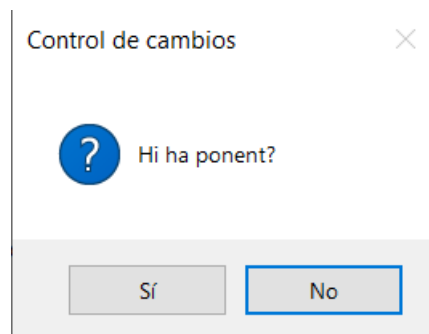


Fig 6.4 Una de las preguntas que hace el control de cambios respecto a un campo no genérico. Fuente: propia.

La pregunta de la figura 6.4 hace referencia a la existencia de un ponente para el trabajo. En caso afirmativo, se pasa a preguntar el nombre del ponente. En caso negativo, se pasa directamente a la convocatoria. En este caso no hay ponente por lo tanto no aparecerá en la portada corregida.



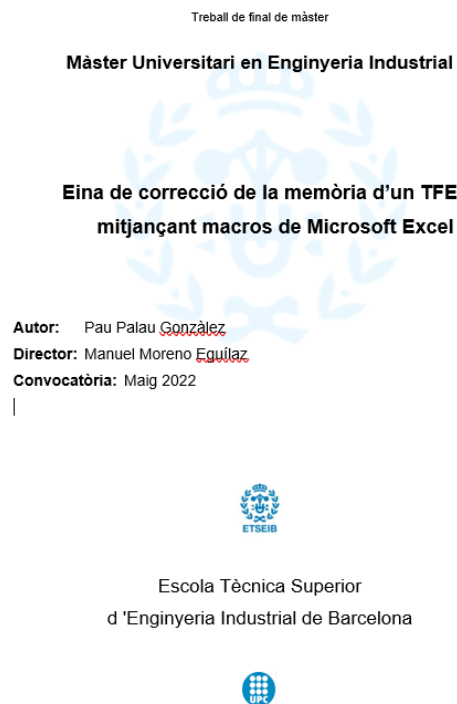


Fig. 6.5 Portada corregida. Fuente: propia.

Como se puede apreciar en la imagen, así quedaría la portada corregida. Posee el formato indicado por la plantilla y los datos no genéricos deseados gracias a las preguntas del control de cambios.

## 6.2. Comprobación índice

Aplicando la función podemos ver que detecta un índice. Por lo tanto, el control de cambios no dirá nada ya que si se encuentra un índice. La función aplicará los cambios para que el índice cumpla con el formato deseado. En este caso también cumple con el formato. Una limitación detectada consiste en que a la hora de ajustar el tamaño de la letra del índice, el programa lo aplica a todo el índice y en la versión de la plantilla se distingue entre el nombre de los capítulos y los subapartados de cada capítulo poniendo un tamaño de 13 puntos para los capítulos y uno de 12 puntos para los subapartados. En este caso ya se hacía esta distinción en el documento original, pero al pasarle el programa asigno a toda la tabla un tamaño de 13 sin distinción.

Entonces, una propuesta de mejora sería encontrar la forma de realizar esta distinción dentro del índice y asignar correctamente el tamaño de fuente en función de si es capítulo

o subapartado.

**SUMARI:**

<b>1. GLOSSARI</b>	<b>8</b>
<b>2. PREFACI</b>	<b>9</b>
2.1. Origen del projecte	9
2.2. Motivació	9
2.3. Requeriments previs	9
<b>3. INTRODUCCIÓ</b>	<b>10</b>
3.1. Objectius del projecte	10
3.2. Abast del projecte	10
3.3. Relació amb altres treballs	10
<b>4. ANTECEDENTS</b>	<b>11</b>
<b>5. CONSIDERACIONS PRÈVIES</b>	<b>13</b>
5.1. Plantilla TFE proporcionada per l'ETSEIB	13
5.2. Repàs de conceptes bàsics de Visual Basic	15
5.3. Eina de revisió de memòries	16
<b>6. DESENVOLUPAMENT DEL PROGRAMA</b>	<b>20</b>
6.1. Interfície gràfica	20
6.2. Funció principal	21
6.3. Control de canvis	23
6.4. Corregir_index	25
6.5. Corregir_capçaleres	27
6.6. Corregir_portada	29
6.7. Corregir_paràgrafs	31
6.8. Corregir_ortografia	33
6.9. Corregir peus de pàgina	35
6.10. Corregir_bibliografia	37
<b>7. COMPROVACIÓ DE L'EINA EN UN CAS REAL</b>	<b>38</b>
7.1. Comprovació de la portada	39

Fig. 6.6 Índex antes del programa

**SUMARI:**

<b>1. GLOSSARI</b>	<b>8</b>
<b>2. PREFACI</b>	<b>9</b>
2.1. Origen del projecte .....	9
2.2. Motivació .....	9
2.3. Requeriments previs .....	9
<b>3. INTRODUCCIÓ</b>	<b>10</b>
3.1. Objectius del projecte .....	10
3.2. Abast del projecte .....	10
3.3. Relació amb altres treballs .....	10
<b>4. ANTECEDENTS</b>	<b>11</b>
<b>5. CONSIDERACIONS PRÈVIES</b>	<b>13</b>
5.1. Plantilla TFE proporcionada per l'ETSEIB .....	13
5.2. Repàs de conceptes bàsics de Visual Basic .....	15
5.3. Eina de revisió de memòries .....	16
<b>6. DESENVOLUPAMENT DEL PROGRAMA</b>	<b>20</b>
6.1. Interfície gràfica .....	20
6.2. Funció principal .....	21
6.3. Control de canvis .....	23
6.4. Corregir_índex .....	25
6.5. Corregir_capçaleres .....	27
6.6. Corregir_portada .....	29
6.7. Corregir_paràgrafs .....	31
6.8. Corregir_ortografia .....	33
6.9. Corregir peus de pàgina .....	35
6.10. Corregir_bibliografia .....	37
<b>7. COMPROVACIÓ DE L'EINA EN UN CAS REAL</b>	<b>38</b>
7.1. Comprovació de la portada .....	39

Fig. 6.7 Índice después del programa. Fuente: propia

Ahora, pongámonos en el caso de que no hay índice. El programa detecta que no hay un índice y nos pregunta si queremos colocar uno.

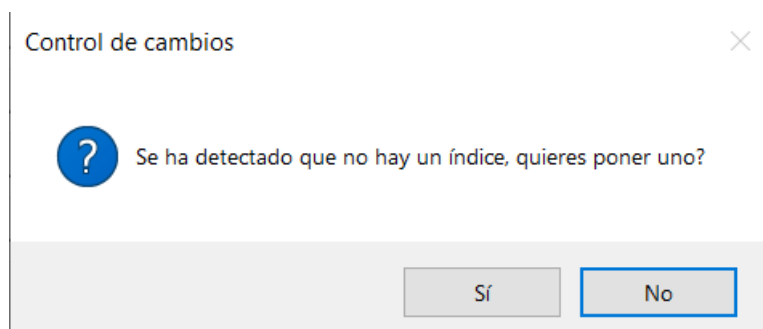


Fig. 6.8 Pregunta del control de cambios referente al índice

Contestamos de forma afirmativa y procede a hacer un índice que quedará de la siguiente manera:

Eina de correcció de memòries de TFE mitjançant macros de Microsoft Excel		Pàg. 5	
<b>Sumari</b>			
<b>1. GLOSSARI</b>			<b>7</b>
<b>2. PREFACI</b>			<b>8</b>
2.1. Origen del projecte			8
2.2. Motivació			8
2.3. Requeriments previs			8
<b>3. INTRODUCCIÓ</b>			<b>9</b>
3.1. Objectius del projecte			9
3.2. Abast del projecte			9
3.3. Relació amb altres treballs			9
<b>4. ANTECEDENTS</b>			<b>10</b>
<b>5. CONSIDERACIONS PRÈVIES</b>			<b>12</b>
5.1. Plantilla TFE proporcionada per l'ETSEIB			12
5.2. Repàs de conceptes bàsics de Visual Basic			14
5.3. Eina de revisió de memòries			15
<b>6. DESENVOLUPAMENT DEL PROGRAMA</b>			<b>19</b>
6.1. Interfície gràfica			19
6.2. Funció principal			20
6.3. Control de canvis			22
6.4. Corregir_index			24
6.5. Corregir_capçaleres			26
6.6. Corregir_portada			28
6.7. Corregir_paràgrafs			30
6.8. Corregir_ortografia			32
6.9. Corregir peus de pàgina			34
6.10. Corregir_bibliografia			36
<b>7. COMPROVACIÓ DE L'EINA EN UN CAS REAL</b>			<b>37</b>
7.1. Comprovació de la portada			38
7.2. Comprovació de l'index			40
7.3. Comprovació de les capçaleres			41
7.4. Comprovació dels peus de pàgina			42

Pàg. 6		Memòria	
7.5. Comprovació de l'ortografia			43
7.6. Comprovació dels paràgrafs			46
7.7. Comprovació de la bibliografia			46
<b>8. MILLORES APORTADES RESPECTE A L'ÚLTIMA VERSIÓ</b>			<b>47</b>
<b>9. IMPLEMENTACIONS FUTURES</b>			<b>48</b>
9.1. Enfoc de les funcions ja realitzades			48
9.2. Verificació de la gramàtica			49
<b>10. PLANIFICACIÓ TEMPORAL</b>			<b>50</b>
<b>11. ESTUDI ECONÒMIC</b>			<b>51</b>
<b>12. ESTUDI AMBIENTAL</b>			<b>52</b>
<b>CONCLUSIONS</b>			<b>53</b>
<b>AGRAÏMENTS</b>			<b>54</b>
<b>BIBLIOGRAFIA</b>			<b>55</b>
<b>LLISTAT DE FIGURES I TAULES</b>			<b>58</b>
<b>ANNEXOS</b>			<b>60</b>

Fig. 6.9 Índice introducido por el programa. Fuente: propia.

Como se puede apreciar en la imagen, se ha hecho una captura de pantalla de las dos páginas para mostrar que se han introducido todos los puntos. Se puede apreciar que el programa tiene en cuenta el idioma seleccionado a la hora de añadir el título al índice con la palabra "Sumari".

### 6.3. Comprobación encabezados

En cuanto a los encabezados nos enfocaremos en la demostrar la mejora ya que el programa ya corrige con éxito los encabezados en versiones anteriores de este trabajo. En este caso la mejora es la implementación de este filtro para las secciones de una página.

Haciendo la comprobación, se ha detectado que todos los encabezados del trabajo son correctos.

Ahora pongámonos en una situación donde se pueda demostrar este filtro. Para ello, nos iremos al documento de prueba junto con el cual he desarrollado el programa. Nos centraremos en la comparación de la verificación de la sección 3 que tiene solo una página par sin el filtro y luego con el filtro.

Por lo tanto, la situación sin el filtro quedaría de la siguiente forma:

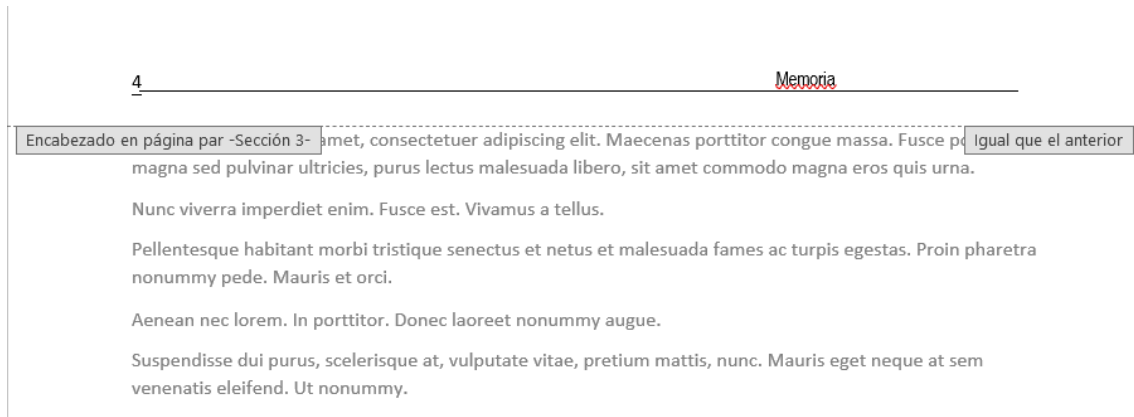


Fig. 6.10 Sección 3 del documento de pruebas personal. Fuente: propia.

Y daría los siguientes resultados:

C	
ENCABEZADOS	
	Encabezado incorrecto en las páginas impares en la sección 2. Falta la palabra Pág.
	Encabezado incorrecto en las páginas pares en la sección 2. Falta la palabra Pág.
	Encabezado incompleto en las páginas impares de la sección 2
	Encabezado incorrecto en las páginas pares en la sección 3. Falta la palabra Pág.
	Encabezado incompleto en las páginas impares de la sección 3
	Encabezado incorrecto en las páginas pares en la sección 4. Falta la palabra Pág.
	Encabezado incompleto en las páginas impares de la sección 4
	Encabezado incorrecto en las páginas impares en la sección 5. Falta la palabra Pág.
	Encabezado incorrecto en las páginas pares en la sección 5. Falta la palabra 'Memoria'
	Encabezado incompleto en las páginas impares de la sección 5
	Encabezado incorrecto en las páginas pares en la sección 6. Falta la palabra 'Memoria'
	Encabezado incorrecto en las páginas pares en la sección 7. Falta la palabra 'Memoria'
	Encabezado incorrecto en las páginas impares en la sección 8. Falta la palabra Pág.
	Encabezado incorrecto en las páginas pares en la sección 8. Falta la palabra 'Memoria'
	Encabezado incompleto en las páginas impares de la sección 8
	Encabezados del trabajo incorrectos

Fig. 6.11 Resultados de la verificación del documento personal sin el filtro aplicado.

Fuente: propia.

Podemos apreciar en la imagen que el programa interpreta una página que no existe que es la página impar de la sección 3 y esta sección solo tiene una página y esta página es par.

Ahora, aplicando el filtro tenemos los siguientes resultados:

ENCABEZADOS
Encabezado incorrecto en las páginas impares en la sección 2. Falta la palabra Pág.
Encabezado incorrecto en las páginas pares en la sección 2. Falta la palabra Pág.
Encabezado incompleto en las páginas impares de la sección 2
Encabezado incorrecto en las páginas pares en la sección 3. Falta la palabra Pág.
Encabezado incompleto en las páginas impares de la sección 4
Encabezado incorrecto en las páginas impares en la sección 5. Falta la palabra Pág.
Encabezado incorrecto en las páginas pares en la sección 5. Falta la palabra 'Memoria'
Encabezado incompleto en las páginas impares de la sección 5
Encabezado incorrecto en las páginas pares en la sección 6. Falta la palabra 'Memoria'
Encabezado incorrecto en las páginas pares en la sección 7. Falta la palabra 'Memoria'
Encabezado incorrecto en las páginas pares en la sección 8. Falta la palabra 'Memoria'
Encabezados del trabajo incorrectos

Fig. 6.12 Resultados de la verificación del documento personal con el filtro aplicado.

Fuente: propia.

Podemos ver que una vez aplicado el filtro el número de errores se reduce pero sobre todo la página impar inexistente de la sección 3 desapareció de la tabla de resultados lo cual cuadra más con lo que hay realmente en el documento.

## 6.4. Comprobación pies de página

Pasamos a los pies de página, aquí también se aplicó el filtró demostrado en el apartado anterior pero no nos centraremos en él aquí ya que tiene la misma estructura y funcionamiento. Nos centraremos en la mejora de que ahora es capaz de comprobar y colocar si se da el caso la imagen correspondiente al pie de página.

En primer lugar, se pasará la herramienta por el documento de Pau [3].

PIES DE PÁGINA
Logo de la universidad mal colocado en las páginas impares de la sección 2
Pies de página incorrectos

Fig. 6.13 Resultados análisis de los pies de página

Al tener solo 3 secciones y la última sección tiene solo páginas pares y el programa hace el bucle de análisis para todas las secciones menos la primera ya que, bajo las directrices de la plantilla, esta sección debería permanecer diferente a las demás ya que pertenecería solo a la portada y solo tendría una página y el programa solo analiza el encabezado de la primera página de la primera sección. Al no ser este el caso y la primera sección abarca

gran parte del documento, pues el programa no ha detectado los errores de la primera sección. Una de las propuestas de mejora de este apartado sería expandir la suposición inicial y abarcar casos como estos.

Por otro lado, viendo el documento se puede decir a simple vista que los escudos están bien colocados, pero a los ojos del programa, para estar bien colocados tienen que indicarse claramente la posición al colocar la el escudo por primera vez. Se indica con una opción de formato de la imagen dentro del pie de página. Adjunto un ejemplo:

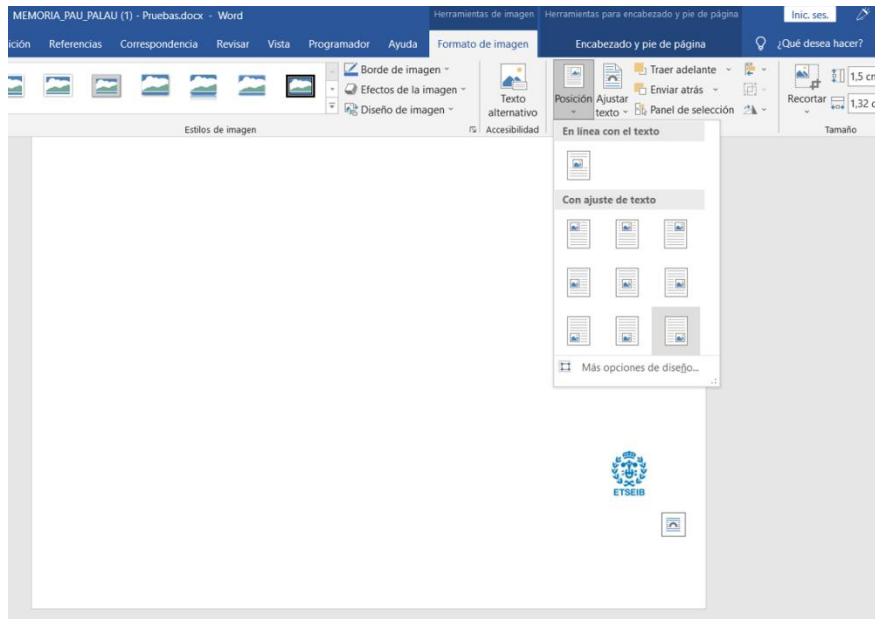


Fig. 6.14 Opción para indicar a Word la posición de la imagen en el pie de página.

Fuente: propia

La imagen de ejemplo es la opción que se debería escoger para indicar a Word que la imagen irá a la derecha del pie de página. Podemos hacer la prueba con el error que dio en el primer análisis. Si indicamos a Word que el escudo del pie de página de las páginas impares de la sección 2 va a la derecha de la forma en que se enseña en la imagen podemos demostrar la certeza del funcionamiento del programa. Los resultados después del cambio:

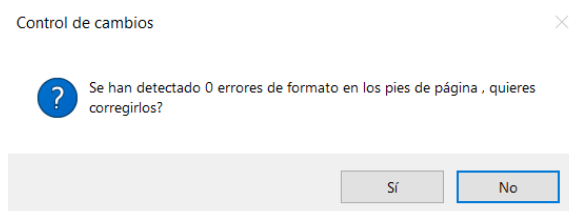


Fig. 6.15 Mensaje en la verificación de los pies de página. Fuente: propia.

Efectivamente, el control de cambios (para no saturar de imágenes de la tabla de resultados) nos informa que no se han detectado errores después de indicar a Word la ubicación con la opción anteriormente mostrada. Por lo tanto, para que el programa pueda evaluar si la posición de la figura es correcta es necesario que se indique la posición claramente mediante esta opción del formato de la imagen. De lo contrario el programa lo interpretará como mal colocado en el caso de la existencia de la imagen en el pie de página. Es importante tenerlo en cuenta en este caso ya que una vez corregido el error detectado no se puede apreciar un cambio de posición muy relevante y puede llevar a confusiones.

## 6.5. Comprobación glosario

Pasamos a probar una de las nuevas funciones implementadas. Los resultados:

GLOSARIO
Sí existe Glosario
Todas las palabras del glosario se encuentran en el texto.

Fig. 6.15 Resultados de un primer análisis

El primer resultado es correcto pero el segundo no ya que el programa a ha detectado otras palabras que se repiten puesto que las palabras del glosario en ese caso no están en negrita y por lo tanto no las detecta. Habrá detectado otras palabras que cumplan con las condiciones para ser palabras del glosario explicadas en el apartado 5.8.

## 1. Glossari

ETSEIB: Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

GETI: Grau en Enginyeria en Tecnologies Industrials

TFG: Treball de Fi de Grau

TFE: Treball de Fi d'Estudis

TFM: Treball de Fi de Màster

VBA: Visual Basic for Applications

Fig. 6.16 Glosario del trabajo de Pau. Fuente: [3]



- a) **Grammarly Business:** És un corrector d'ortografia, gramàtica i plagis. Presenta una interfície molt intuïtiva i el programa és senzill d'utilitzar. Grammarly Business pretén ajudar a l'usuari a produir escrits polits i d'alta qualitat que comptin amb una bona organització. Grammarly Business inclou: correccions sofisticades per a errors complexos d'ortografia, gramàtica i puntuació, suggeriments avançats per fer que l'escriptura sigui més concisa, atractiva, educada i inclusiva i una guia d'estil personalitzable. Té una versió gratuïta amb menys funcionalitats, també ofereix una versió de pagament amb més de 400 normes de gramàtica i el corrector de plagis. Té un cost de 29,95 US\$ mensuals o 139 US\$, si es paga anualment [2].
- b) **ProWritingAid:** Es tracta d'una plataforma per aprendre a redactar millor els informes i documents, però es pot fer servir com a corrector ortogràfic i de gramàtica. Com a punt a destacar, permet pujar memòries de treballs, ja que no té límit quant a mida del document o longitud. Permet als usuaris optimitzar els seus articles escrits, esborrar errors i rectificar els errors gramaticals i de puntuació de manera automatitzada. També permet als usuaris millorar la llegibilitat dels seus articles. Actualment, el

Fig. 6.17 Palabras que también cumplen las condiciones para ser palabras del glosario.

Fuente: [3]

Las dos anteriores imágenes evidencian lo anterior dicho respecto al segundo resultado. Esta claro que a este nivel de restricción de las palabras del glosario aun presenta limitaciones por lo tanto otra propuesta de mejora seria encontrar una cuarta restricción para reducir estos casos o encontrar otra forma de seleccionar solo las palabras del glosario.

Pongamos otro ejemplo de mi documento personal de pruebas para ver la funcionalidad de la verificación del glosario.

## 1. Glosario

**Posible:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

**Dúctil:** Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

**Dani:** Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

**MML:** Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

Fig. 6.18 Glosario de documento de pruebas personal. Fuente: propia.

Este sería el formato de un glosario el cual el programa puede verificar. La palabra “Dúctil” y las siglas “MML” no se encuentran en el texto. Veremos si el programa detecta lo mismo.

GLOSARIO	
Sí existe Glosario	
Dúctil --> palabra del glosario, no se encuentra en el texto.	
MML --> palabra del glosario, no se encuentra en el texto.	

Fig. 6.19 Resultados de la verificación del glosario del documento de pruebas personal.

Fuente: Propia.

Efectivamente, las dos palabras que se esperaban que iba a detectar las ha detectado.

## 6.6. Comprobación párrafos

Pasamos el programa al documento. Se detectan dos errores uno de falta de justificación y el otro que no está en Arial.

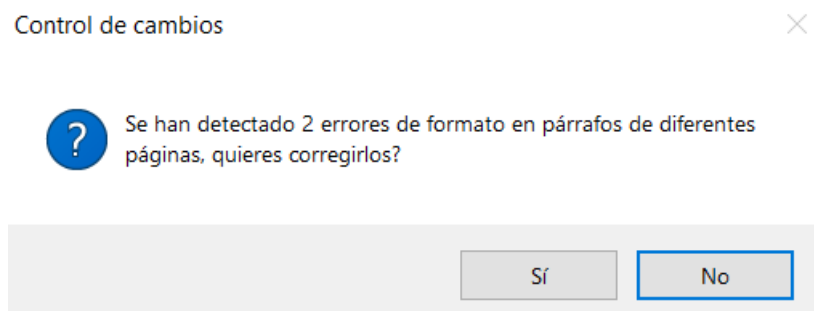


Fig. 6.20 Informe del control de cambios en la verificación de los párrafos. Fuente: Propia.

El primer error es una cita de una figura. Pero su número de palabras no supera las 20 palabras para que sea considerado párrafo, pero el programa lo consideró así igualmente. Se confirma esta primera suposición después de ver el resultado de la corrección.

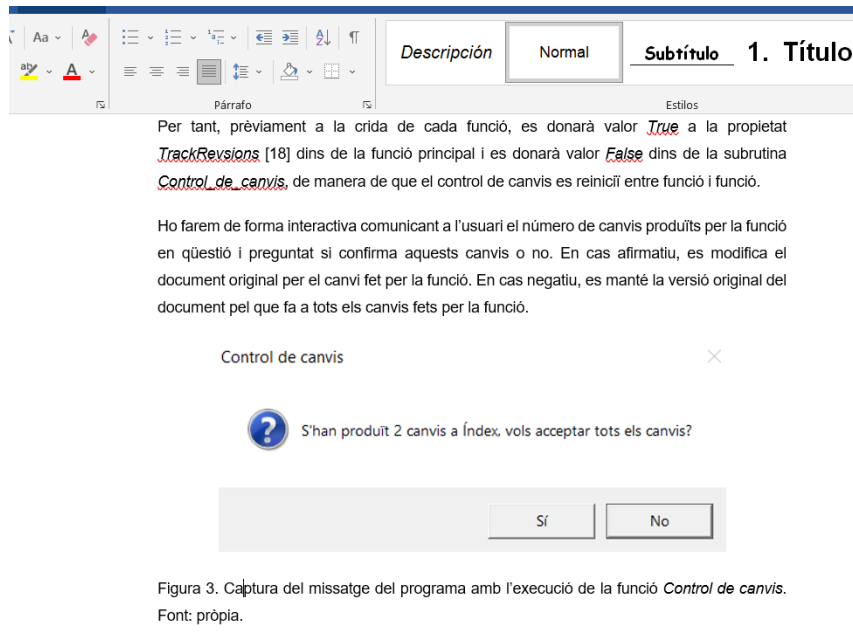


Fig. 6.21 Justificación de una cita de figura no esperada. Fuente: propia.

Como se puede ver se ha justificado la cita de la figura por lo tanto se le ha considerado párrafo sin ni siquiera superar las 20 palabras mínima para que el programa lo considere uno. Esto se puede deber a que el método *count* [47] usado para las palabras lo cuente de una forma distinta. Teorizo esto porque ya una vez me pasó concretamente con los encabezados.

El segundo error no lo comprendo ya que al analizar la página donde se produce este error no encuentro ningún párrafo que sea de un tipo de letra distinta la Arial. Y cuando se corrige, la página no cambia para nada.

La búsqueda y solución a estos errores las puedo dejar para futuras versiones.

## 6.7. Comprobación ortografía

Hacemos la comprobación de este parámetro.

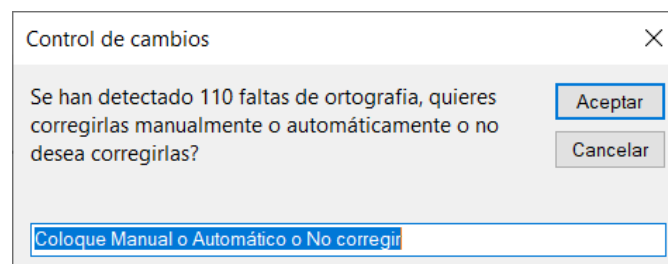


Fig. 6.22 Pregunta sobre el método de corrección que se usará. Fuente: propia.

Una vez encuentra el número de faltas de ortografía nos pregunta el método de corrección. Cogemos el manual porque es el que yo he implementado.

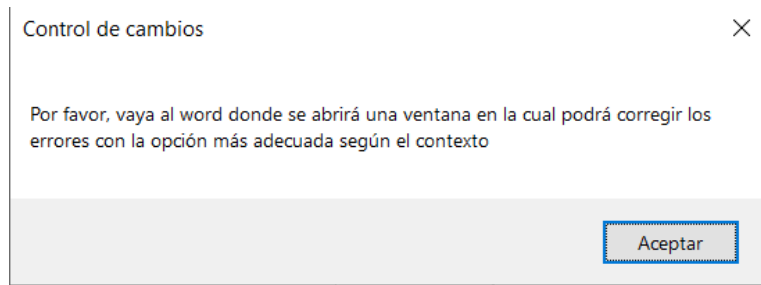
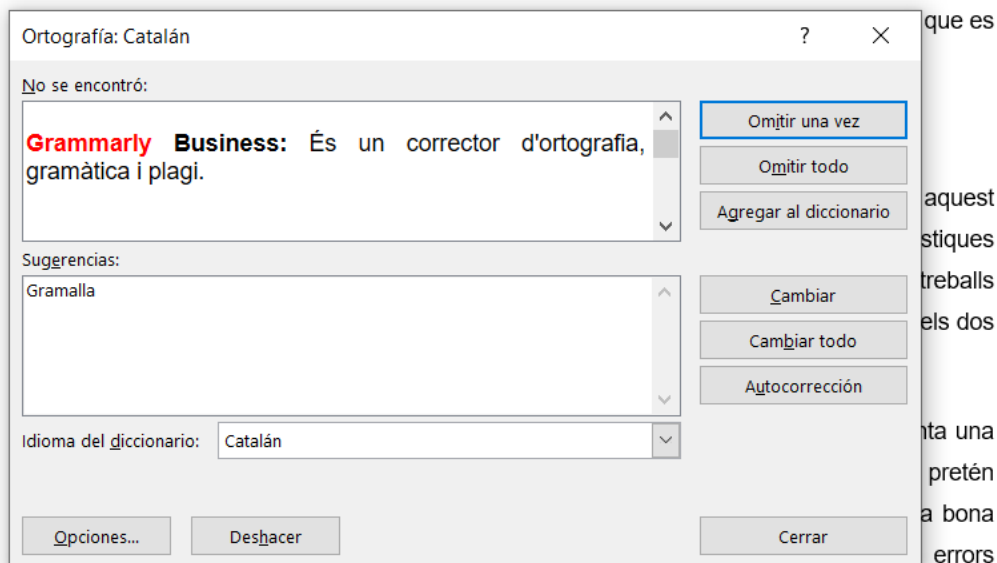


Fig. 6.23 Mensaje del control de cambios después de escoger la opción "Manual".

Fuente: propia.

Una vez en el Word, nos recibirá una ventana de corrección que nos facilitará la corrección ya que irá de error en error y en cada error podremos escoger cual es el más adecuado según la circunstancia. Es más lento que el automático, pero no nos equivocaremos ni obtendremos resultados no deseados.

## 4. Antecedents



complexos d'ortografia, gramàtica i puntuació, suggeriments avançats per fer que l'escriptura sigui més concisa, atractiva, educada i inclusiva i una guia d'estil personalitzable. Té una versió gratuïta amb menys funcionalitats, també ofereix una versió de pagament amb més de 400 normes de gramàtica i el corrector de plagis. Té un cost de 29,95 US\$ mensuals o 139 US\$, si es paga anualment [2].

Fig. 6.24 Ventana de corrección ortográfica. Fuente: propia

Además, que de esta forma ya no nos tendríamos que preocupar de las expresiones en otros idiomas.

Una vez hayamos cerrado la ventana de corrección volvemos al programa y el control de cambios nos avisa del número de espacios dobles en el texto y de si queremos corregirlos.

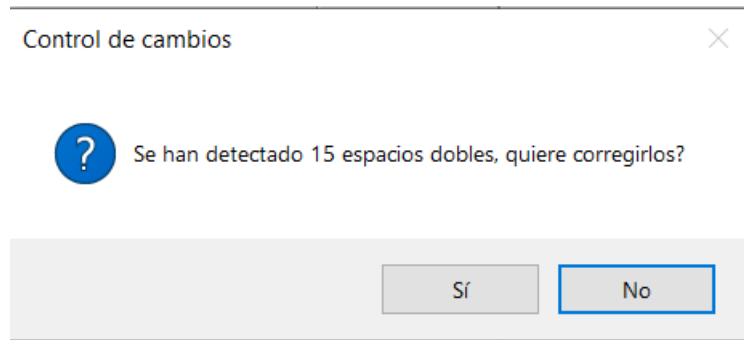


Fig. 6.25 Mensaje del control de cambios referente a los espacios dobles. Fuente: propia.

Respondemos de forma afirmativa y cambiará estos espacios dobles por espacios normales.

## 6.8. Comprobación bibliografía

Vemos los resultados en este parámetro:

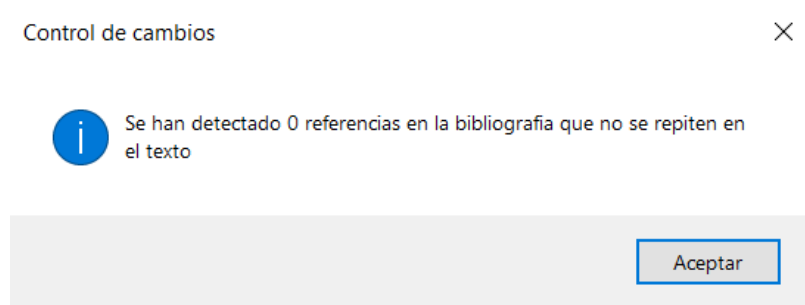


Fig. 6.26 Respuesta del control de cambios después de la verificación. Fuente: propia.

Vemos que todo está en su sitio ya que la bibliografía se ha enumerado con el formato esperado (el de la plantilla) y por lo tanto ha ido iterando referencia por referencia buscándolas en todo el documento y ha detectado que por lo menos se mencionan toda una vez en el texto.

Vayámonos a un caso no tan perfecto. Para ello, me remito una vez más a mi documento de pruebas personal que estará en el anexo. La bibliografía será la siguiente imagen y la referencia 2 y 4 no se repetirán. Veremos si coincide con los resultados del programa.

## 5. Bibliografía

- [1] <http://www.upc.es/slt/cat/publicacions/gl2/gl2.htm>, 21 de septiembre de 2000
- [2] <http://www.upc.es/slt/puro>, 24 de septiembre de 2015
- [3] <http://www.upc.es/slt/puro/vd>, 28 de junio de 1985
- [4] <http://www.plataformanaranja.com>, 21 de enero de 2023
- [5] <http://www.upc.es/slt/cat/publicacions/gl2/gl2.htm>, 21 de septiembre de 2000

Fig. 6.27 Bibliografía del documento de pruebas personal. Fuente: Propia.

El código de verificación de la bibliografía contempla dos posibilidades en cuanto a la forma de hacer la bibliografía explicadas en el apartado 5.11. El primer caso es el que se refleja en el TFM de Pau [3], una lista numerada con el formato adecuado y el caso que estamos tratando ahora es el de una bibliografía hecha línea a línea. Así demostramos que esta función funciona correctamente en estas dos condiciones ligeramente distintas.

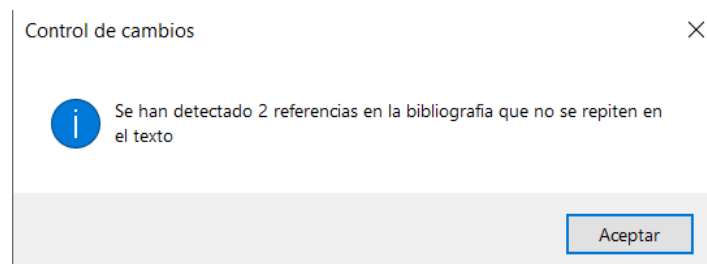


Fig. 6.28 Respuesta del control de cambios a un caso distinto. Fuente: propia.

Pinta bien ya que quedamos que eran dos las referencias que no se repiten.

G
<b>BIBLIOGRAFÍA</b>
Sí existe bibliografía
Hay 1 URLs que no tienen fecha actualizada
La referencia número 2 no se repite en el texto
La referencia número 4 no se repite en el texto

Fig. 6.29 Resultados del análisis de la bibliografía. Fuente: propia.

Efectivamente, el programa ha detectado que las referencias 2 y 4 no se repiten como se acordó en un principio.

## 6.9. Comprobación citas

Recordemos que esta función verificará y corregirá, si el usuario así lo quiere, el orden de enumeración de las referencias bibliográficas. Por lo tanto, si es una lista enumerada no lo detectará y tampoco es necesaria esta función en ese caso ya que las listas enumeradas se ordenan automáticamente. Esto es lo que pasará en este caso ya que la bibliografía en este caso es una lista enumerada.

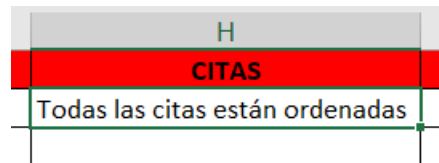


Fig. 6.30 Resultado del análisis de las citas. Fuente: propia.

Pongamos un caso donde no estén ordenadas:

### 5. Bibliografía

- [1] <http://www.upc.es/slt/cat/publicacions/gl2/gl2.htm>, 21 de septiembre de 2000
- [2] [http://www.upc.es/slt/puro\\_](http://www.upc.es/slt/puro_), 24 de septiembre de 2015
- [2] [http://www.upc.es/slt/puro/vd\\_](http://www.upc.es/slt/puro/vd_), 28 de junio de 1985
- [3] <http://www.plataformanaranja.com>, 21 de enero de 2023
- [4] <http://www.upc.es/slt/cat/publicacions/gl2/gl2.htm>, 21 de septiembre de 2000

Fig. 6.31 Bibliografía de prueba desordenada. Fuente: propia.

Aplicamos la función.

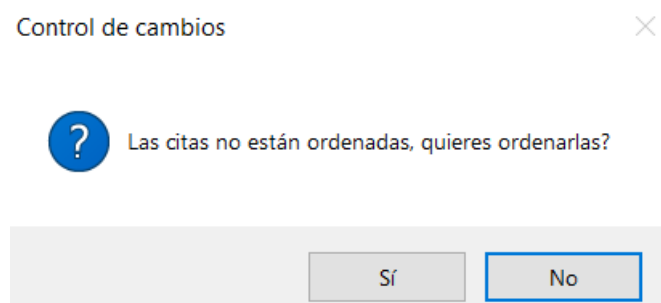


Fig. 6.32 Pregunta respecto a la corrección de las citas. Fuente: Propia.

El control de cambios detecta que las citas no están ordenadas y pregunta al usuario si

quiere ordenarlas. Respondemos afirmativamente.

## 5. Bibliografía

- [1] <http://www.upc.es/slt/cat/publicacions/gl2/gl2.htm>, 21 de septiembre de 2000
- [2] <http://www.upc.es/slt/puro>, 24 de septiembre de 2015
- [3] <http://www.upc.es/slt/puro/vd>, 28 de junio de 1985
- [4] <http://www.plataformanaranja.com>, 21 de enero de 2023
- [5] <http://www.upc.es/slt/cat/publicacions/gl2/gl2.htm>, 21 de septiembre de 2000

Fig. 6.33 Bibliografía de prueba ordenada. Fuente: propia.

Como se puede apreciar en la imagen, la corrección ha sido un éxito.

## 6.10. Comprobación pies de figura y tablas

Finalmente pasamos a la última función. Ejecutamos.

Nos da error. Esto es debido a algún error de a la hora de coger valores numéricos del índice para el diccionario porque el algoritmo se queda dentro de un bucle infinito debido a que no se encuentra el valor deseado en el diccionario. Una propuesta de mejora es encontrar la forma para que el diccionario coja todos los valores correctamente.

Para demostrar que la función funciona en otros casos nos iremos por última vez al documento de pruebas personal para poner a prueba la función.

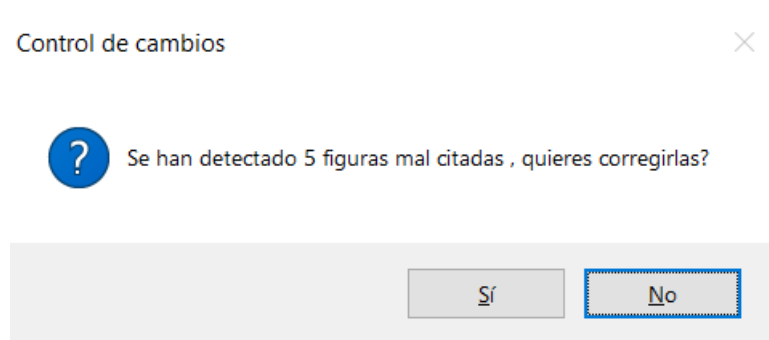


Fig. 6.34 Mensaje respecto a la verificación de los pies de figura.

Procedemos a corregir de tal manera que el formato pase de esta imagen:





Figura 7

Fig. 6.35 Formato de la cita antes de la corrección. Fuente: propia.

A la siguiente:



Figura 2.2

Fig. 6.36 Formato de la cita después de la corrección. Fuente: propia.

El formato es correcto y los números también ya que esta imagen era la segunda imagen del capítulo 2.

En cuanto a las tablas siguiendo la misma lógica y pasando por el control de cambios otra vez:

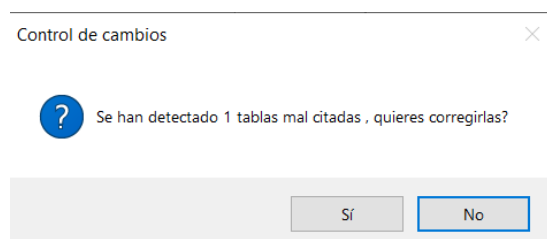


Fig. 6.37 Mensaje respecto a la verificación de las tablas. Fuente: propia.

Por lo tanto, pasamos de esto:

<u>loz</u>	<u>vdvdv</u>	<u>vdvdv</u>
<u>ssvdv</u>	274	42
<u>dvdvd</u>	44	42

Tabla 1

Fig. 6.37 Formato de la tabla antes de la corrección

A la siguiente imagen:

<u>loz</u>	<u>vdvdv</u>	<u>vdvdv</u>
<u>ssvdv</u>	274	42
<u>dvdvd</u>	44	42

Tabla 3.1

Fig. 6.38 Formato de la tabla después de la corrección





## 8. Estudio Económico

Al ser un TFE que se hace completamente a través de un ordenador, el trabajo se tiene en cuenta en forma de tiempo dedicado. Por lo tanto, suponiendo un precio de trabajo de 20€/h se hace el siguiente recuento económico:

Tarea	Horas totales dedicadas (h)	Precio (€)
Aprendizaje y familiarización VBA (Curso)	160	3200
Lectura y recreación de los 2 TFE's relacionados	80	1600
Diseño de nueva interfaz y relacionarlo con el código existente	40	800
Mejoras pies de página	60	1200
Mejoras párrafos	20	400
Nueva función glosario	20	400
Nueva función pies de figura y tablas	60	1200
Cambios y mejoras en el control de cambios	20	400
Mejoras bibliografía	20	400
Nueva función espacios dobles	20	400
Mejoras encabezados	20	400
Mejoras Ortografía	20	400
Mejoras citas	20	400
Mejora índice	20	400
Mejora portada	20	400
Redacción memoria	80	1600
Amortización portátil	-	120
<b>Total</b>	<b>680</b>	<b>13720</b>

Tabla .1 Recuento económico por horas del TFG. Fuente: propia

Para calcular la amortización lo hacemos mediante el método de amortización lineal [45], la versión que se hace mediante un porcentaje que se basa en un coeficiente anual a cada tipo de amortización [46]. Según la referencia, este coeficiente para equipos electrónicos es del 20% pero este coeficiente es anual y como la duración del proyecto ha sido de menos de un año pues este porcentaje se rebajaría a un 15%. Por lo tanto calcularíamos la amortización multiplicando este coeficiente por el valor de adquisición del portátil que fue de 800€ (en realidad fueron 799,99€ pero es para hacer los números redondos). Esto nos daría una amortización de 120€.

## **9. Estudio Ambiental**

Como se ha mencionado antes, al ser un TFG que se desarrolla totalmente en un ordenador no tiene otro impacto ambiental más que el consumo de electricidad para hacer funcionar el ordenador.

Por otro lado, al reducir los tiempos de corrección gracias a la aplicación también se reduce el consumo de electricidad. Otra ventaja es que reduce la posibilidad de impresiones de memorias no deseadas debido a algún error de formato no detectado en su momento que obligue a tener que imprimirla de nuevo y por consiguiente gastar más papel.

## **10. Impacto social**

En cuanto al impacto social, esta herramienta para reducir el tiempo de corrección de una memoria hecha en la ETSEIB y por lo tanto siguiendo el formato que dicta la facultad. Entonces, tendrá un impacto social positivo en la comunidad de la ETSEIB ya que la herramienta les ayudará a verificar y corregir el formato de las memorias en menos tiempo que en el caso de que no dispongan de ella.



## **11. Perspectiva de género**

El desarrollo de la ampliación y mejoras en las funciones ya existentes de este programa no distingue entre hombres y mujeres. Lo puede usar libremente tanto un hombre como una mujer. La aplicación se comporta de la misma forma, lo esté usando una mujer, como lo esté usando un hombre.

## Conclusiones

El motor principal para la realización de este trabajo ha sido mi interés por aprender macros de Excel. Y efectivamente, he aprendido bastante sobre el lenguaje de programación VBA gracias el proceso de aprendizaje e investigación necesario para conseguir los objetivos propuestos.

Podemos afirmar que se han cumplido los objetivos del proyecto ya que se han encontrado e implementado diferentes mejoras a las funciones existentes de la herramienta de corrección y también se han añadido nuevas funciones que en definitiva aumentan la eficiencia del programa y por lo tanto reduce los tiempos de corrección.

Es verdad que el programa aún sigue teniendo algunas limitaciones y sigue necesitando la intervención del usuario en ciertos casos, pero versión tras versión estas limitaciones son cada vez menores ya que ahora la herramienta verifica, corrige (modifica el documento a corregir si así lo quiere el usuario) y las funciones integradas son cada vez más complejas y por lo tanto pueden abarcan más casos y poseer varias funcionalidades extra.

Y, por último, en los apartados de las funciones que creo que pueden mejorar he dejado algunas propuestas de mejora para continuar este proceso de mejora continua versión tras versión de la herramienta de corrección de TFE.

## **Agradecimientos**

Quisiera hacer un ejercicio de agradecimiento. En primer lugar, a mis padres por apoyarme durante toda la carrera y en segundo lugar a mi tutor Manuel Moreno Eguílaz que me propuso algunas ideas de ampliación y mejora para este TFG.

## Bibliografía

- [1] Curso VBA Excel [[\(7\) VBA Excel. Introducción. Vídeo 1.mp4 - YouTube](#)], 2013.
- [2] IGNASI ALÉS ROCA, *Herramienta de verificación de TFE basado en Excel VBA*, 2021.
- [3] PAU PALAU GONZÀLEZ, *Herramienta de corrección de la memoria de un TFE mediante macros de Microsoft Excel*.
- [4] ETSEIB, *Plantilla per la redacció del TFE*, 2021.
- [5] Documentación VBA [[Introducción a VBA en Office | Microsoft Learn](#)]
- [6] Objeto *Sections* [[Objeto Sections \(Word\) | Microsoft Learn](#)]
- [7] Objeto *Headers* [[Propiedad Section.Headers \(Word\) | Microsoft Learn](#)]
- [8] Objeto *Range* [[Objeto Range \(Excel\) | Microsoft Learn](#)]
- [9] Método *ClearContents* [[Método Range.ClearContents \(Excel\) | Microsoft Learn](#)]
- [10] Objeto *InlineShapes* [[Propiedad Document.InlineShapes \(Word\) | Microsoft Learn](#)]
- [11] Método *Count* [[Propiedad InlineShapes.Count \(Word\) | Microsoft Learn](#)]
- [12] Objeto *Userform* [[Objeto UserForm | Microsoft Learn](#)]
- [13] Cuadro de herramientas [[Cuadro de herramientas | Microsoft Learn](#)]
- [14] *VBA Events* [[Event statement \(VBA\) | Microsoft Learn](#)]
- [15] *Microsoft Word Object Library* [[Reference \(Object library reference for Office\) | Microsoft Learn](#)]
- [16] Evento *Click* [[Click \(evento\) | Microsoft Learn](#)]
- [17] Método *TrackRevisions* [[Propiedad Document.TrackRevisions \(Word\) | Microsoft Learn](#)]
- [18] Objeto *Inputbox* [[Función InputBox \(Visual Basic para Aplicaciones\) | Microsoft Learn](#)]
- [19] Objeto *TablesOfContents* [[TablesOfContents object \(Word\) | Microsoft Learn](#)]
- [20] Método *GoTo* [[Instrucción GoTo \(VBA\) | Microsoft Learn](#)]
- [21] Método *Add* [[Método TablesOfContents.Add \(Word\) | Microsoft Learn](#)]
- [22] Objeto *TableOfContents* [[Objeto TableOfContents \(Word\) | Microsoft Learn](#)]
- [23] Objeto *HeadersFooters* [[Objeto HeadersFooters \(Word\) | Microsoft Learn](#)]
- [24] Método *Information* [[Propiedad Selection.Information \(Word\) | Microsoft Learn](#)]
- [25] Operador *Mod* [[Mod \(Operador\) - Visual Basic | Microsoft Learn](#)]

- [26] Objeto *Footers* [[Propiedad Section.Footers \(Word\) | Microsoft Learn](#)]
- [27] Objeto *Left* [[Propiedad ShapeRange.Left \(Word\) | Microsoft Learn](#)]
- [28] Objeto *ShapeRange* [[Objeto ShapeRange \(Word\) | Microsoft Learn](#)]
- [29] Método *Find* [[Propiedad Range.Find \(Word\) | Microsoft Learn](#)]
- [30] Estructura de control de flujo *Do-Until* [[Do...Loop statement \(VBA\) | Microsoft Learn](#)]
- [31] Método *RemoveDuplicates* [[Método Range.RemoveDuplicates \(Excel\) | Microsoft Learn](#)]
- [32] Método *End()* [[Propiedad Range.End \(Excel\) | Microsoft Learn](#)]
- [33] Método *SpellingErrors* [[Propiedad Document.SpellingErrors \(Word\) | Microsoft Learn](#)]
- [34] Método *GetSpellingSuggestions* [[Método Application.GetSpellingSuggestions \(Word\) | Microsoft Learn](#)]
- [35] Método *CheckSpelling* [[Método Range.CheckSpelling \(Word\) | Microsoft Learn](#)]
- [36] Método *Text* [[Propiedad Selection.Text \(Word\) | Microsoft Learn](#)]
- [37] Método *ListString* [[Propiedad ListFormat.ListString \(Word\) | Microsoft Learn](#)]
- [38] Método *Bold* [[Propiedad Font.Bold \(Word\) | Microsoft Learn](#)]
- [39] Método *HyperLinks* [[Propiedad Range.Hyperlinks \(Word\) | Microsoft Learn](#)]
- [40] Objeto *Dictionary* [[Objeto Dictionary \(Word\) | Microsoft Learn](#)]
- [41] Estructura de control de flujo *For – Next* [[For...Next statement \(VBA\) | Microsoft Learn](#)]
- [42] Método *Next* [[Método Selection.Next \(Word\) | Microsoft Learn](#)]
- [43] Objeto *Selection* [[Propiedad Application.Selection \(Word\) | Microsoft Learn](#)]
- [44] Objeto *Tables* [[Propiedad Document.Tables \(Word\) | Microsoft Learn](#)]
- [45] Amortización Lineal [[Qué es la amortización en economía y cómo se calcula - Banco Santander](#)]
- [46] Coeficiente de amortización [[Agencia Tributaria: Amortizaciones](#)]
- [47] Método *Count* para palabras [[Propiedad Words.Count \(Word\) | Microsoft Learn](#)]