

Trabajo de Fin de Grado

## Grado en Ingeniería en Tecnologías Industriales

# Herramienta automática de generación de exámenes para Moodle en lenguaje Python a partir de ficheros de texto

### MEMORIA

**Autor:** Sergio Tiscar Ortega  
**Director:** Manuel Moreno Eguílaz  
**Departamento:** EEL - Departamento de Ingeniería Electrónica  
**Convocatoria:** Enero 2022



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





## Resumen

Este trabajo de fin de grado consiste en el desarrollo de una herramienta automática de generación de exámenes para Moodle a partir de ficheros de texto desarrollada totalmente en lenguaje Python. Más concretamente, el trabajo se centra en los exámenes de la asignatura de 'Electrónica' del Grado en Ingeniería en Tecnologías Industriales (GETI) de la ETSEIB, los cuales siguen un modelo basado en un cierto número de preguntas verdadero o falso, seguido de preguntas tipo test con 5 opciones. El objetivo final es la generación de una base de datos con preguntas que permitan la generación de cuestionarios aleatorios en Atenea.

Este proyecto surge de la evolución forzada a la que se ha visto sometida la educación a raíz de la pandemia que se vive actualmente y que ha impedido realizar los exámenes de manera convencional estos últimos meses. La creación de esta herramienta pretende agilizar un proceso que manualmente sería mucho más largo y tedioso, además de presentar una mayor probabilidad de error.

Por lo tanto, en el presente documento se refleja todo el proceso de diseño y desarrollo de la herramienta, así como también se da el contexto necesario para entender los entornos en los que se desarrolla todo el trabajo, los cuales son Moodle, Microsoft Word y los lenguajes Python y XML.

En primer lugar, se realiza una breve introducción a la problemática del proyecto y se explican las motivaciones que han llevado a realizar dicho proyecto, así como los objetivos y alcance del mismo. En segundo lugar, se verán proyectos realizados anteriormente y que comparten objetivo con éste, aunque están realizados con lenguajes de programación y herramientas diferentes. De este modo, se compararán con el propio para ver ventajas que éste puede aportar sobre los existentes. Seguidamente, se explica de forma detallada el funcionamiento de Moodle y de sus ramas más utilizadas en este trabajo, como son los diferentes tipos de *quiz* que se pueden crear y los bancos de preguntas. También se explica tanto el funcionamiento del lenguaje Python y de las bibliotecas utilizadas para la creación de la herramienta como el lenguaje XML, que permite la importación de preguntas a Moodle. Finalmente, se detalla qué es Microsoft Word y para qué sirven las macros de este programa.

Una vez terminada esta parte más teórica, se pasa a describir la elaboración de esta herramienta automática de creación de cuestionarios para Moodle, exponiendo, paso a paso, el proceso y razonamiento realizado para llegar al resultado final. Por último, se presenta una pequeña guía de usuario para que cualquier persona que desee usar esta herramienta tenga unas breves indicaciones de cómo hacerlo correctamente.



# Sumario

<b>RESUMEN</b>	<b>3</b>
<b>SUMARIO</b>	<b>5</b>
<b>1. GLOSARIO</b>	<b>7</b>
<b>2. INTRODUCCIÓN</b>	<b>9</b>
2.1. Motivaciones.....	9
2.2. Objetivos del proyecto.....	9
2.3. Alcance del proyecto.....	9
<b>3. ANÁLISIS DE ANTECEDENTES</b>	<b>10</b>
<b>4. ENTORNOS DE DESARROLLO</b>	<b>12</b>
4.1. Entorno Moodle.....	12
4.1.1. Cuestionarios.....	13
4.1.2. Bancos de preguntas.....	15
4.1.3. Tareas.....	18
4.1.4. Chat.....	19
4.1.5. Foros.....	19
4.1.6. Glosarios.....	19
4.1.7. Consultas.....	19
4.2. Lenguaje XML.....	19
4.3. Lenguaje Python.....	20
4.3.1. Biblioteca docx.....	22
4.4. Microsoft Word y macros.....	23
<b>5. HERRAMIENTA DE GENERACIÓN DE EXÁMENES PARA MOODLE A TRAVÉS DE PYTHON</b>	<b>24</b>
5.1. Descripción básica y funcionalidad.....	24
5.2. Macros.....	24
5.3. Función sustituir.....	25
5.4. Función subíndices y superíndices.....	26
5.5. Función imágenes.....	27
5.6. Código para preguntas VF.....	28
5.7. Código para preguntas test.....	29
5.8. Funcionamiento general de la herramienta.....	32
5.9. Resultados.....	33

<b>6. DATOS TÉCNICOS DE LA HERRAMIENTA</b>	<b>37</b>
6.1. Bibliotecas utilizadas .....	37
6.2. Funciones creadas .....	38
<b>7. TEST Y VALIDACIÓN</b>	<b>39</b>
<b>8. GUÍA DE USUARIO</b>	<b>41</b>
8.1. Pasos previos .....	41
8.2. Pasos de utilización de la herramienta .....	43
<b>9. IMPACTO AMBIENTAL</b>	<b>48</b>
<b>10. PRESUPUESTO</b>	<b>49</b>
<b>11. PLANIFICACIÓN DEL PROYECTO</b>	<b>50</b>
<b>CONCLUSIONES</b>	<b>51</b>
<b>BIBLIOGRAFÍA</b>	<b>53</b>
Referencias bibliográficas.....	53

# 1. Glosario

*API*: Conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.

*ASCII*: Código de caracteres basado en el alfabeto latino.

*Big data*: Hace referencia a conjuntos de datos tan grandes y complejos que precisan de aplicaciones informáticas no tradicionales de procesamiento de datos para tratarlos adecuadamente.

*Blockchain*: Es un sistema de transferencia digital basado en la distribución de la información en multitud de nodos independientes que registran y validan dicha información.

*Data analytics*: Es una aplicación más concreta y precisa del *Data Science*. Trata de encontrar tendencias y métricas que sirvan a las compañías para tomar decisiones más acertadas y obtener mejores resultados.

*Data mining*: Proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos.

*Data science*: Es una rama del *Big Data* cuyo objetivo es extraer e interpretar la información que se deriva de una gran cantidad de datos recopilada.

*ETSEIB*: Escuela Técnica Superior de Ingeniería Industrial de Barcelona.

GETI: Grado en Ingeniería en Tecnologías Industriales.

IA: Inteligencia Artificial.

ISO: International Organization for Standardization.

*Scraping*: Técnica utilizada mediante programas de software para extraer información de sitios web.

UPC: Universidad Politécnica de Cataluña.

VBA: Visual Basic para Aplicaciones.

VF: Verdadero/Falso.

XML: eXtensible Markup Language.





## 2. Introducción

### 2.1. Motivaciones

La motivación principal que ha impulsado el desarrollo de este proyecto es el interés generado en los primeros cursos del Grado en Ingeniería en Tecnologías Industriales (GETI) por la informática, y más concretamente, por el lenguaje Python. Este atractivo siempre ha suscitado interés por la materia, de seguir profundizando y a modo de construcción intelectual, es por ello que este proyecto supone un reto y una posibilidad más de aprendizaje.

Además del interés propio, es cierto también que si finalmente se consigue generar una herramienta como la que se pretende con este trabajo, puede llegar a facilitar parte del trabajo a los docentes que quieran emplear dicho método, puesto que, tras la situación epidemiológica ocasionada por el Covid-19, su carga profesional se ha visto ciertamente en aumento.

### 2.2. Objetivos del proyecto

Los objetivos de este proyecto son los mencionados a continuación:

- Generar una herramienta sencilla e intuitiva para el usuario, que reciba un archivo de Microsoft Word y devuelva un archivo XML listo para importar preguntas a Moodle.
- Fácil uso de esta herramienta para automatizar el proceso de generación de preguntas en Moodle a través de esta herramienta creada en lenguaje Python.
- Crear un banco de preguntas que permita la generación de cuestionarios aleatorios en Moodle.

### 2.3. Alcance del proyecto

Este proyecto abarca todos los exámenes proporcionados por el tutor, desde el año 2013 hasta la actualidad, de la asignatura de 'Electrónica' impartida en la Escuela Técnica Superior de Ingeniería de Barcelona (ETSEIB). Todos ellos, ya sean exámenes parciales, finales o reevaluaciones, poseen una misma estructura basada en preguntas verdadero/falso (VF) y preguntas tipo test con cinco opciones para elegir.

Tras extraer todas las preguntas de todos estos exámenes se obtendrá una base de datos de aproximadamente 760 preguntas VF y 680 preguntas de tipo test, lo que proporciona una gran variedad de preguntas para la creación de cuestionarios aleatorios en Moodle.

### 3. Análisis de antecedentes

En la ETSEIB se han realizado recientemente dos trabajos de final de máster que tienen objetivos semejantes al de este proyecto.

El primero de ellos, *Tool Based on Excel VBA to Generate Test in Moodle* de Blanca Ràfols [1], tiene exactamente el mismo propósito que es la obtención de un archivo XML, listo para ser importado a Moodle a partir del archivo Microsoft Word de un examen de 'Electrónica'.

El otro de los proyectos semejantes es el de Valentín González: *Herramienta basada en VBA para generación de bancos de preguntas en Moodle* [2], pero en este caso lo que se busca es la importación de preguntas de cualquier asignatura -a través de XML-, al banco de preguntas de Moodle, siempre que las cuestiones sean de tipo VF, tipo test de 5 opciones o de respuesta numérica. La diferencia es que la creación del archivo XML no se basa en un archivo Microsoft Word previo, sino que es necesario rellenar información en ciertos campos.

Ambos proyectos están desarrollados en VBA (Visual Basic para Aplicaciones), que es el lenguaje de macros usado para programar aplicaciones en Microsoft Windows, sobre todo en Microsoft Excel. Este lenguaje es usado habitualmente para automatizar tareas cotidianas, así como para crear aplicaciones y servicios de bases de datos.

Analizando más en profundidad el proyecto de Blanca Ràfols, se puede comprobar que el objetivo principal es la creación de una aplicación que permita, en unos pocos y sencillos pasos, pasar de un documento Microsoft Word que contenga un examen de 'Electrónica', a uno XML listo para importar preguntas a Moodle. Éste se lleva a cabo de forma satisfactoria y, a pesar de que funciona correctamente, tiene ciertos inconvenientes, como por ejemplo que no detecta las ecuaciones e importa las imágenes en baja calidad. Dado que la funcionalidad del proyecto desarrollado en el presente documento es exactamente la misma que en el proyecto mencionado, en las conclusiones se desarrollará una pequeña comparativa para analizar las ventajas e inconvenientes de cada uno de los proyectos.

En el caso del trabajo de Valentín González, el propósito es algo diferente, ya que lo que pretende es la creación de una aplicación que permita la importación a Moodle de preguntas de cualquier asignatura, pero no es una herramienta que automatice el proceso, sino una aplicación que facilita la creación del documento XML. Es decir, en su proyecto, lo que se ha creado es una herramienta que permite al docente crear preguntas rellenando unos simples campos como son el tipo de pregunta, el enunciado, las posibles respuestas... De manera que se trata de una herramienta más general que la que se propone en este trabajo, pero que

requiere de más esfuerzo, es decir, no es una aplicación que automatice el proceso, sino que lo agiliza y simplifica.

## 4. Entornos de desarrollo

En este capítulo se presentan todos los entornos en los que se desarrolla la herramienta de generación de exámenes. Estos son: Moodle, la plataforma en la que se basa Atenea y que funciona como el campus virtual utilizado por la UPC; XML, el lenguaje utilizado para importar preguntas a Moodle; Python, el lenguaje de programación que ha permitido la creación de la herramienta; y, finalmente, Microsoft Word, el programa ofimático en el que están editados los exámenes de 'Electrónica' que se procesan en este proyecto.

### 4.1. Entorno Moodle

Moodle es una plataforma de aprendizaje diseñada para proporcionar a educadores, administradores y estudiantes, un sistema integrado único, robusto y seguro para crear ambientes de enseñanza personalizados. Posee una interfaz simple, características de arrastrar y soltar, así como recursos bien documentados que, junto con mejoras continuas en usabilidad, hacen de Moodle una herramienta fácil de entender y usar. Además, al ser un código abierto, Moodle es continuamente revisado y mejorado para adecuarse así a las necesidades actuales y cambiantes de sus usuarios [3].

Moodle está construida por módulos, lo que permite una gran flexibilidad a la hora de agregar y quitar funcionalidades en muchos niveles. A continuación, se dará una breve descripción de la funcionalidad de los módulos más útiles, una descripción algo más extensa y una explicación sobre cómo usar los módulos necesarios en este proyecto.

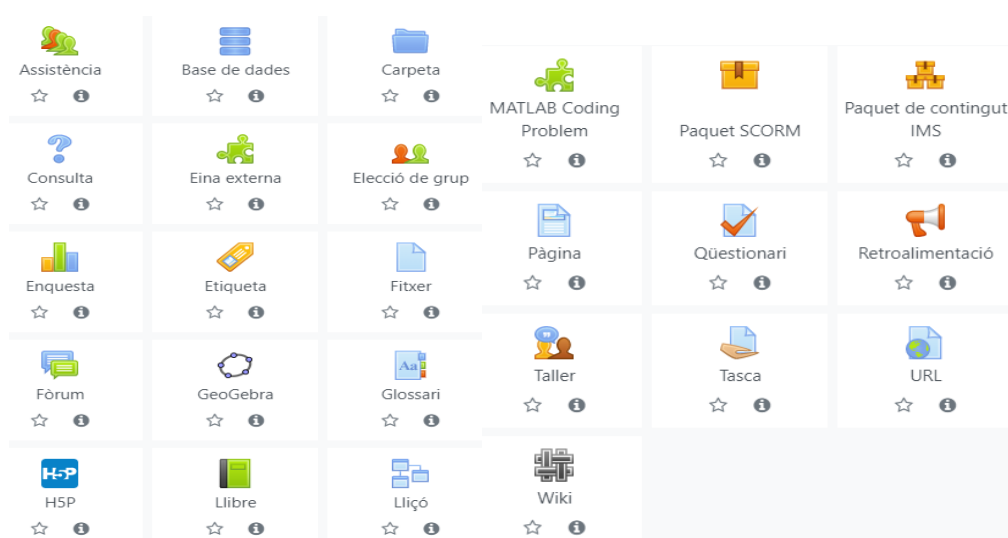


Figura 1: Módulos y funciones que se pueden añadir a Moodle. Fuente: propia.

#### 4.1.1. Cuestionarios

Este módulo permite diseñar y plantear cuestionarios de diferentes tipos, por ejemplo, la opción múltiple, VF y respuestas numéricas, entre otros. Las preguntas están organizadas por categorías que el propio profesor puede crear en una base de datos, además, éstas pueden ser reutilizadas en cualquiera de los temas del curso correspondiente. Los cuestionarios pueden permitir múltiples intentos, cada intento es calificado y permite que el profesor pueda mostrar algún mensaje o las respuestas correctas al finalizar el cuestionario. El hecho de que este módulo tenga capacidad de calificación, permite que pueda utilizarse para la realización de exámenes no presenciales de las asignaturas, que es la situación que se presenta en este proyecto.

Para crear un cuestionario en Moodle lo primero que hay que hacer es activar el modo edición de Moodle y, una vez activado, al pasar por encima de cualquiera de los temas del curso en el que se encuentre, aparecen los siguientes botones:

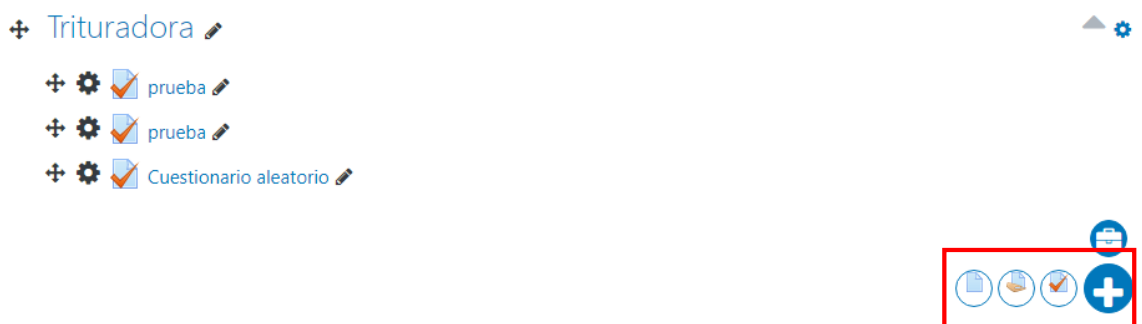


Figura 2: Botones de creación de módulos. Fuente: propia.

Con estos botones (Figura 2) podemos añadir al tema -en este caso sería el tema Trituradora-, un fichero, una tarea, un cuestionario o abrir un desplegable que nos muestra todos los módulos existentes.

Una vez seleccionado el botón para añadir el cuestionario, aparece la siguiente pantalla:

S'està afegint l'activitat o el recurs següent: Qüestionari a Trituradora ?

▶ [Expandeix-ho tot](#)

Aquest formulari conté camps obligatoris marcats amb **i** .

**Paràmetres generals** Característiques avançades activades

Nom **i**

Descripció

↕
A
✎
B
I
☰
☰
🔗
🔄

🖼️
📄
🎤
📹
📎
H-P

[Mostra'n més...](#)

**Temporització**

Obre el qüestionari **i**  de       Habilita

Tanca el qüestionari  de       Habilita

Límit de temps **i**    Habilita

[Mostra'n més...](#)

▶ [Qualificació](#)

▶ [Aspecte](#)

▶ [Comportament de la pregunta](#)

▶ [Opcions de revisió ?](#)

▶ [Aparença](#)

▶ [Restriccions extres durant els intents](#)

▶ [Retroacció global ?](#)

▶ [Paràmetres comuns de mòduls](#)

▶ [Restriccions d'accés](#)

▶ [Etiquetes](#)

Desa els canvis i torna al curs
Desa els canvis i visualitza
Cancel·la

Aquest formulari conté camps obligatoris marcats amb **i** .

Figura 3: Modificació de paràmetres del qüestionari. Fuente: propia.

Aquí puede añadirse el título o una descripción, así como también pueden modificarse parámetros como el tiempo de realización, el aspecto y la calificación, entre otros. Todo esto hace que sea una herramienta muy útil para crear exámenes no presenciales.

Una vez creado el cuestionario se pueden añadir preguntas de diferentes formas: la primera es manualmente, seleccionando el tipo de pregunta y redactándola; por otro lado, se pueden agregar preguntas a partir del banco de preguntas o, por último, mediante la incorporación de preguntas aleatorias, también desde el banco de preguntas.

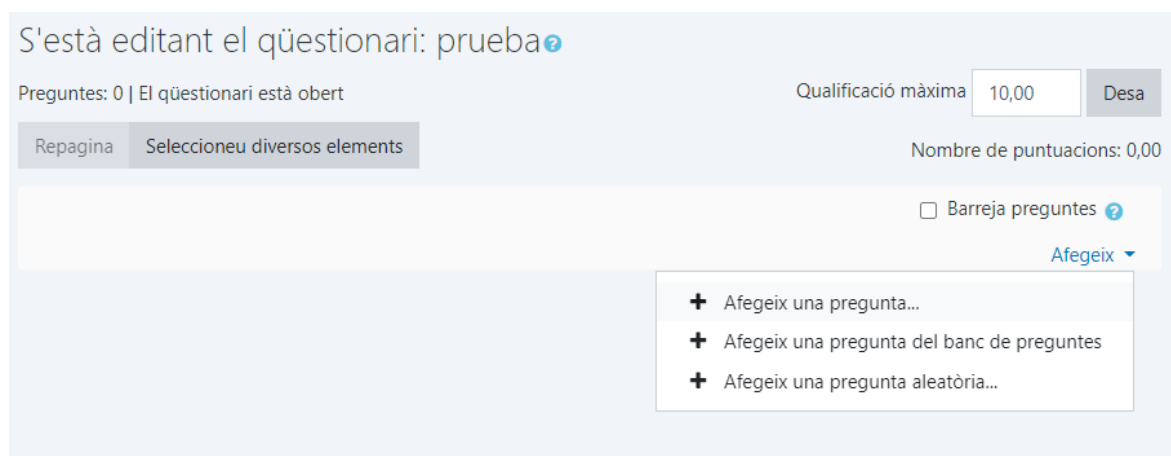


Figura 4: Adición de preguntas al cuestionario. Fuente: propia.

Una vez añadidas todas las preguntas que se deseen, se guarda y se obtiene el cuestionario completado con todas las características adaptadas a las necesidades del examen.

#### 4.1.2. Bancos de preguntas

Este módulo permite crear un banco de preguntas en el que se pueden editar, exportar e importar preguntas, así como también es factible clasificarlas por categorías. Existe la posibilidad de limitar el uso de las categorías a todo el curso, a un tema o al nivel del examen, esto será en función de cada profesor. Las preguntas del banco pueden añadirse a una actividad mediante el proceso explicado anteriormente.

TFG\_TFM

CAMPUS VIRTUAL UPC / Les meves assignatures / bd6cd34fea1cd0f304644ef64e93e35d / Banc de preguntes /

Preguntes Categories Importa Exporta

## Banc de preguntes

Tria una categoria: Categoria per defecte en bd6cd34fea1cd0f304644ef64e93e35d

La categoria per defecte per a preguntes compartides en el context «bd6cd34fea1cd0f304644ef64e93e35d».

No hi ha cap filtre d'etiqueta aplicat

Filter by tags...

Mostra el text de la pregunta a la llista de preguntes

Opcions de cerca

Mostra també les preguntes de les subcategories.

Mostra també les preguntes antigues

Crea una nova pregunta...

Figura 5: Banco de preguntas. Fuente: propia.

En este proyecto, el objetivo es importar preguntas a esta base de datos mediante el documento XML resultante tras utilizar la herramienta creada. Desde el propio archivo XML que proporciona la herramienta, se crea una categoría nueva, con el nombre del archivo Microsoft Word original, en el banco de preguntas.

Las preguntas están organizadas por categorías, aunque inicialmente cada curso tiene solamente una categoría llamada 'Por defecto'. Es una buena práctica el crear más categorías para organizar las preguntas. Es por esta razón que dicha herramienta crea una categoría por cada examen que se haya importado al banco, lo cual resulta un método sencillo para encontrar las preguntas y, además, permite usarlas aleatoriamente sin problemas. También se puede crear una jerarquía de categorías, ya que se pueden crear subcategorías comprendidas dentro de categorías superiores.

La pantalla para editar preguntas las muestra desde la categoría actualmente seleccionada. Desde esta pestaña es posible elegir el apartado a través del menú desplegable y, al usar las casillas de selección de debajo de dicho menú, se determina si se desea que aparezcan las preguntas de todas las subcategorías inferiores.



Preguntes Categories Importa Exporta

## Banc de preguntes

Tria una categoria:

No hi ha cap filtre d'etiqueta aplicat

Mostra el text de la pregunta a la llista de preguntes

Opcions de cerca ▾

Mostra també les preguntes de les subcategories.

Mostra també les preguntes antigues

1 2 3 4 5 6 7 8 »

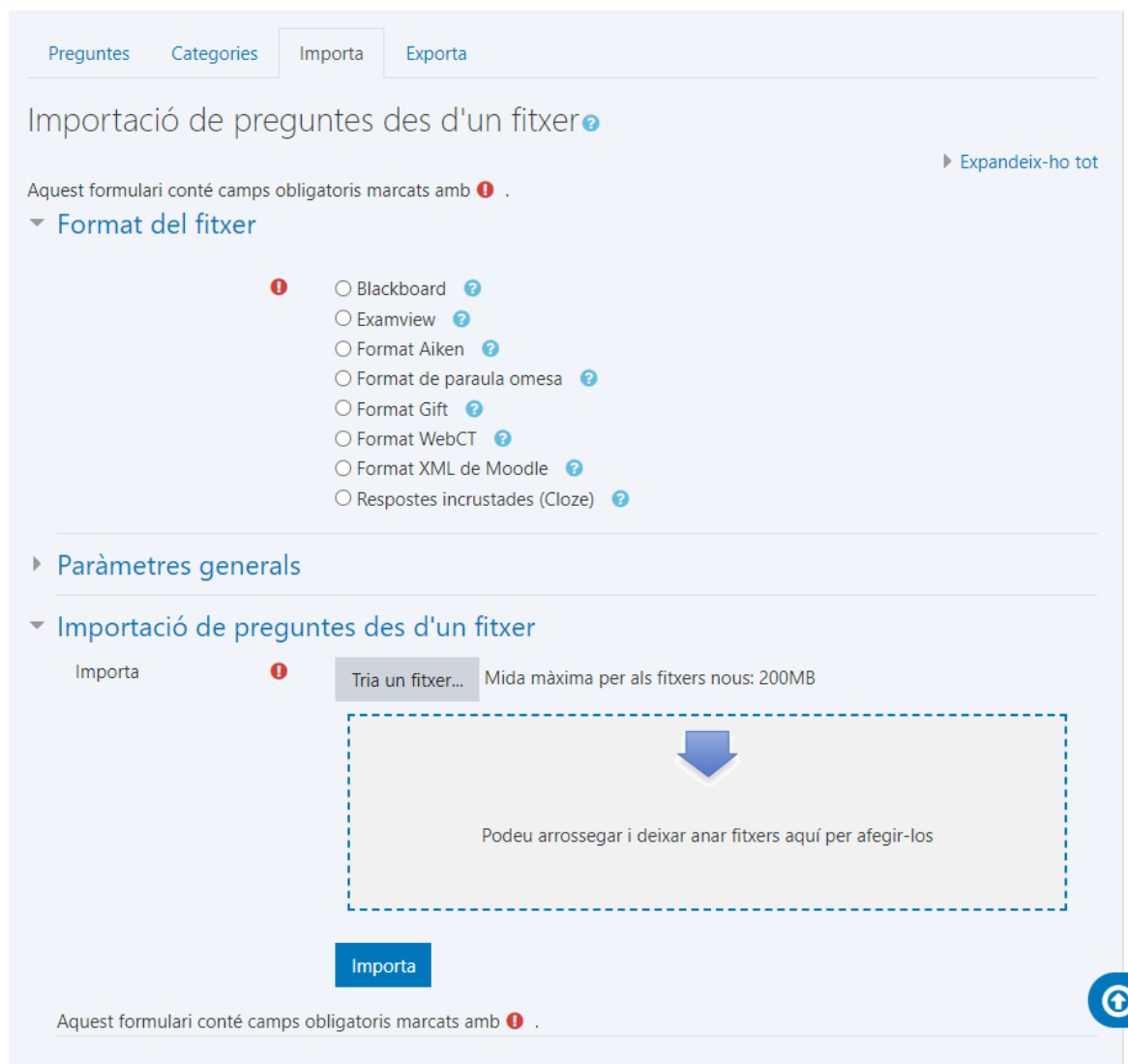
<input type="checkbox"/> T ▲ Pregunta	Accions	Creada per	Última modificació feta per
<input type="checkbox"/> Nom de la pregunta / Nombre ID		Nom / Cognoms / Data	Nom / Cognoms / Data
<input type="checkbox"/> Pregunta 50		Tiscar Ortega Sergio 10 de novembre 2021, 16:20	Tiscar Ortega Sergio 10 de novembre 2021, 16:20
<input type="checkbox"/> Pregunta 50		Tiscar Ortega Sergio 12 de novembre 2021, 17:25	Tiscar Ortega Sergio 12 de novembre 2021, 17:25

Figura 6: Visualizador de preguntas por categorías. Fuente: propia.

Si una pregunta es eliminada cuando todavía está siendo usada en algún examen, entonces no es removida de la base de datos, sino que simplemente se configura a 'oculta'. El 'También mostrar preguntas antiguas' sirve precisamente para mostrar estas preguntas silenciadas.

Entrando en la pestaña 'Importa' aparece un apartado en el que se debe seleccionar el formato del fichero que se va a introducir -en este caso sería XML de Moodle-, ya que, tras analizar todas las alternativas posibles, se ha determinado que, a pesar de ser algo más complicado en cuanto a sintaxis -cosa que el usuario no notará-, tiene menos limitaciones.

Hay otra pestaña de 'Parámetros generales', en la que podemos seleccionar la categoría a la que importar las preguntas y una última pestaña en la que debemos seleccionar el archivo a importar.



The screenshot shows a web interface for importing questions. At the top, there are tabs for 'Preguntes', 'Categories', 'Importa', and 'Exporta'. The 'Importa' tab is active. The main heading is 'Importació de preguntes des d'un fitxer'. Below this, there is a message: 'Aquest formulari conté camps obligatoris marcats amb [red exclamation mark]'. A section titled 'Format del fitxer' is expanded, showing a list of file formats with radio buttons and help icons: Blackboard, Examview, Format Aiken, Format de paraula omesa, Format Gift, Format WebCT, Format XML de Moodle, and Respostes incrustades (Cloze). Below this is a section for 'Paràmetres generals', which is also expanded. Under 'Importació de preguntes des d'un fitxer', there is a sub-section 'Importa' with a red exclamation mark icon. It contains a button 'Tria un fitxer...' and a note 'Mida màxima per als fitxers nous: 200MB'. Below this is a large dashed blue box with a downward arrow and the text 'Podeu arrossegar i deixar anar fitxers aquí per afegir-los'. At the bottom of this section is an 'Importa' button. At the very bottom of the form, there is another message: 'Aquest formulari conté camps obligatoris marcats amb [red exclamation mark]'. A blue circular icon with an upward arrow is located in the bottom right corner of the form area.

Figura 7: Pestaña para importar preguntas. Fuente: propia.

Aquí se puede arrastrar directamente el archivo XML obtenido al usar la herramienta, con lo cual, se importan todas aquellas preguntas del examen que previamente se han seleccionado.

### 4.1.3. Tareas

El módulo de tareas permite al profesor asignar un trabajo a los alumnos que deberán preparar en alguno de los formatos permitidos y remitir, subiéndolo al servidor. Las tareas que se suelen incluir son informes, ensayos, proyectos, etc. Además, en este módulo se permite también la calificación de la tarea.

#### 4.1.4. Chat

El módulo de *Chat* permite a los participantes mantener conversaciones por escrito y en tiempo real a través de conexión a Internet. Esto es una manera útil de tener una mayor interacción y una mejor percepción de la opinión del resto de participantes en posibles debates. El módulo de *chat* contiene varias herramientas que permiten al profesor administrar las conversaciones actuales y revisar las pasadas.

#### 4.1.5. Foros

En los foros es donde se dan la mayor parte de los debates, aunque en este caso, no se participa en vivo junto al resto de participantes. Los foros pueden estructurarse de diferentes maneras, ya que puede plantearse un tema en concreto y que cada alumno aporte su opinión, incluso puede crearse un debate en que es posible responder tanto al tema planteado como a la opinión del resto de participantes. Además, se puede incluir la evaluación de cada mensaje por parte del resto de participantes. En los mensajes enviados a este módulo pueden incluirse archivos adjuntos e imágenes incrustadas.

#### 4.1.6. Glosarios

El módulo denominado “glosarios” permite a los participantes crear una lista de definiciones - semejante a un diccionario- y, posteriormente, las entradas de éste pueden ser exportadas de un glosario a otro dentro del mismo curso por el profesor.

#### 4.1.7. Consultas

La ‘consulta’ consisten en que el profesor hace una pregunta y especifica una serie de opciones, entre las cuales deben elegir los alumnos.

### 4.2. Lenguaje XML

XML es el acrónimo de ‘Extensible Markup Language’, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. El lenguaje de marcado se refiere a un conjunto de códigos que pueden aplicarse en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos, para crear un formato y generar un lenguaje personalizado [4].

Un archivo XML se divide en dos partes: *prolog* y *body*. La parte *prolog* consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. La parte del *body* se compone de dos partes: estructural y de contenido (presente en los textos simples) [4].

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web. Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que ayudan en el procesamiento de datos XML [4].

El lenguaje XML proporciona varias ventajas como la simplificación en el intercambio de datos, uno de los desafíos más complejos para los desarrolladores. Este intercambio a través de XML reduce en gran medida la dificultad del proceso, ya que los datos pueden ser leídos por diferentes aplicaciones incompatibles. De manera similar, también facilita el intercambio de plataforma puesto que los datos XML se almacenan en formato de texto. Esto supone una comodidad en la expansión o actualización a nuevos sistemas de información, nuevas aplicaciones o nuevos navegadores sin pérdida de datos [4].

Algo de gran importancia a la hora de usar XML es definir en qué formato de codificación de caracteres se va a escribir. Para la realización de este proyecto se utilizará la norma ISO-8859-1, que es una norma de la *International Organization for Standardization* (ISO) que define la codificación del alfabeto latino, incluyendo los diacríticos (como letras acentuadas, ñ, ç), y letras especiales (como ß, Ø), necesarios para la escritura de las lenguas originarias de Europa occidental y, sobre todo, fundamentales para la realización de los exámenes de 'Electrónica', ya que contienen tanto diacríticos como letras especiales [5].

Esta norma pertenece al grupo de juegos de caracteres de la ISO que se caracterizan por poseer la codificación ASCII en su rango inicial (128 caracteres) y otros 128 caracteres para cada codificación, empleando un total de 8 bits [5].

### 4.3. Lenguaje Python

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como 'Java' o '.NET', se trata de un lenguaje interpretado, es decir, no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado 'intérprete'. Python es un lenguaje sencillo de leer y escribir, debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto, es decir, gratuito, por lo que permite desarrollar software sin límites. Con el paso del tiempo, Python ha ido ganando popularidad gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos tiempos, ya que facilita trabajar con inteligencia artificial (IA), *big data*, *machine learning* y *data science*, entre muchos otros campos en auge.

Python es un lenguaje de programación multiplataforma, algo que permite desarrollar aplicaciones en cualquier sistema operativo con gran facilidad. Existen una gran cantidad de tecnologías que se relacionan muy bien con Python debido a su sencillez y a su gran potencia para el tratamiento de datos, lo que ha hecho subir exponencialmente el uso de este lenguaje a nivel laboral. De hecho, son cada vez más las empresas que solicitan expertos en Python [6].

- **Data analytics y big data:** El uso de Python está muy extendido en dos áreas que están ganando cada vez más importancia: el análisis de datos y el *big data*. Su simplicidad y su gran número de bibliotecas de procesamiento de datos hacen que Python sea ideal a la hora de analizar y gestionar una gran cantidad de datos en tiempo real. Está siendo utilizado en la actualidad por muchas empresas, tanto de forma directa, como indirecta, ya que detrás de los distintos softwares de *data analytics*, muchas veces está este lenguaje interpretado de alto nivel. Analizar una gran cantidad de datos para transformarlos en información útil para el *big data* es una de las especialidades de Python [6].
- **Data mining:** La minería de datos o *data mining* es un proceso que permite analizar grandes bases de datos con el objetivo de predecir futuras tendencias. Se trata de un proceso complejo que Python puede simplificar a través de la limpieza y organización de datos y del uso de algoritmos de aprendizaje automático que facilitan el análisis de datos [6].
- **Data science:** Tras la creación de los motores numéricos como “Pandas” o “NumPy”, Python está compitiendo con programas como MATLAB. La razón es la misma que en los anteriores apartados; la sencillez y la potencia para trabajar con un gran número de datos, unidos al gran número de bibliotecas existentes, hacen que Python sea ideal para este tipo de tareas [6].
- **Inteligencia artificial:** Durante los últimos años se ha visto incrementado el interés por la IA y parte de su avance se debe a Python. La facilidad de escritura y robustez han convertido a Python en el complemento ideal para el desarrollo de la IA. Su capacidad de plasmar ideas complejas en pocas líneas, unidas al gran número de marcos de trabajo existentes, han hecho que Python sea uno de los lenguajes de programación más usados en el ámbito de la IA [6].
- **Blockchain:** La base de datos distribuida *Blockchain*, conocida por ser la base sobre la que se sustentan las criptomonedas, también funciona bien junto a Python. Como lenguaje versátil, seguro y rápido, es muy útil para formar cadenas de bloques, e incluso permite a los desarrolladores crear una cadena de bloques sencilla en menos de 50 líneas de código, haciendo sencillo algo muy complejo [6].

- **Machine learning:** El *machine learning* o aprendizaje automático es otra de las tecnologías que está revolucionando el mundo de la tecnología. La robótica y la IA son ahora capaces de aprender por sí mismas, a medida que van procesando más y más datos. De esta forma, obtienen información cada vez más relevante, lo que les permite tomar las decisiones adecuadas [6].
- **Desarrollo web:** Python también permite desarrollar webs complejas en menos líneas de código, lo que permite que éstas sean más ligeras y optimizadas. 'Django' es uno de los marcos de trabajo de Python más populares de la actualidad, que puede ser utilizado para crear webs dinámicas y seguras. Python es también muy utilizado para hacer *scraping*, es decir, para obtener información de todo tipo de webs, tal y como lo hacen Netflix, Instagram o Pinterest [6].
- **Juegos y gráficos 3D:** Python también posee una gran capacidad para manejar gráficos 3D, gracias a la gran cantidad de marcos de trabajo y herramientas existentes. 'PyGame', 'Blender' o 'Arcade' son algunos de los más conocidos. Uno de los juegos más populares desarrollado con Python es 'Battlefield 2', un juego de acción bélica lanzado en 2005 en el que el motor gráfico, las animaciones y sus distintas funcionalidades fueron desarrolladas con Python [6].

Todas estas razones son las que han hecho que el interés por este lenguaje haya ido en aumento en los últimos tiempos. Python es un lenguaje de programación sencillo de leer y comprender, algo poco habitual en este tipo de lenguajes. Asimismo, posee un potencial de crecimiento enorme, gracias al gran número de bibliotecas existentes, número que irá en aumento, ya que, al ser un código abierto y gratuito, los usuarios tienen total libertad para crear nuevas bibliotecas con nuevas funcionalidades.

#### 4.3.1. Biblioteca docx

La biblioteca *docx* [7] de Python sirve para crear, modificar o analizar archivos de Microsoft Word. Todo el código detrás de la herramienta desarrollada para la generación de exámenes está basado en el uso de esta biblioteca. Es por esto que, a continuación, se van a detallar los aspectos esenciales para entender el funcionamiento de dicho código.

Como en este caso solo se desea analizar los archivos Microsoft Word, lo primero es abrirlos con el módulo *Document*, el cual genera un objeto iterable que engloba el contenido de todo el documento. Una vez obtenido este objeto *Document*, se puede dividir en objetos *paragraphs*, los cuales contienen el texto de cada párrafo del Microsoft Word. Es decir, con cada salto de línea que detecta, genera un nuevo objeto. Este nuevo objeto también es

divisible en otros nuevos objetos llamados *runs* y estos separan por palabras el contenido del objeto *paragraphs*. Estos objetos *runs* permiten saber más detalles del formato del texto, como por ejemplo, si el texto se encuentra subrayado, en negrita o cursiva, así como si hay subíndices o superíndices.

#### 4.4. Microsoft Word y macros

Microsoft Word es un programa informático orientado al procesamiento de textos que utiliza un formato nativo cerrado y muy utilizado, comúnmente llamado *docx* (utiliza la extensión de archivo *.docx*). Por la amplia difusión de Microsoft Word, este formato se ha convertido en el estándar con el que pueden transferirse textos con o sin formato. Es por eso que la propia biblioteca de Python utilizada para el tratamiento de estos documentos se llama *docx* e igualmente se trata del formato utilizado para la redacción de los exámenes de 'Electrónica' que se tratan en este proyecto. Pero para el correcto tratamiento de los exámenes por parte de esta biblioteca, y por lo tanto de la herramienta desarrollada, es necesaria la ejecución previa de dos macros. Cabe mencionar que las macros son automatismos creados en Microsoft Word para automatizar un conjunto personalizado de acciones, por lo que la creación de una macro adecuada puede ahorrar mucho tiempo al usuario en acciones rutinarias que requieren siempre de las mismas acciones.

## 5. Herramienta de generación de exámenes para Moodle a través de Python

### 5.1. Descripción básica y funcionalidad

Con la creación de esta herramienta se busca automatizar el proceso de generar nuevas preguntas y añadirlas al banco de preguntas, ahorrando así mucho tiempo a los docentes -puesto que hasta ahora lo han hecho de forma manual-.

La herramienta de generación de exámenes recibe un archivo Microsoft Word, que será el examen del que se extraigan las preguntas que proporcionará un archivo en formato XML, donde aparecerán todas estas preguntas listas para ser importadas a Moodle con el método explicado anteriormente.

El primer paso, antes de poder usar la herramienta, es ejecutar dos macros en el archivo de Microsoft Word que se quiera importar. Esto es debido a que, en caso contrario, no procesa correctamente la mayoría de las imágenes de los exámenes, ya que en general -en estos documentos-, las imágenes no suelen ser un único objeto, sino que son un conjunto de figuras. Otro problema que solucionan es la detección de ciertos símbolos como el de infinito o suma exclusiva. Una vez aplicadas estas macros, se puede proceder al uso de la herramienta desarrollada en Python.

Para poder entender cómo funciona, se va a proceder a explicar la creación y la funcionalidad de cada parte del código creado en Python de forma separada, para finalmente mostrar cómo se complementan y se relacionan todas ellas para, en conjunto, formar la herramienta de creación de exámenes de Moodle. En primer lugar, se va a exponer el funcionamiento y la utilidad de las funciones auxiliares que se han creado para el correcto uso de la herramienta de generación de exámenes y, seguidamente, se expondrán las funciones principales.

### 5.2. Macros

Como se ha mencionado anteriormente, para el correcto funcionamiento de la herramienta es necesaria la ejecución previa de dos macros de Microsoft Word.

La primera de ellas unifica en una única imagen el conjunto de elementos que habitualmente componen cada una de las figuras de los exámenes, permitiendo así una exportación a XML mucho más sencilla. La segunda macro sustituye ciertos símbolos que no son detectados por Python -como es el caso del infinito y de la suma exclusiva-, y son



reemplazados por otros que sí son reconocibles, para que más tarde puedan ser intercambiados por el código XML correspondiente con la función 'sustituir' -que se detalla a continuación-.

### 5.3. Función sustituir

La primera de las funciones auxiliares es la función 'sustituir' y esta permite relevar los símbolos que el lenguaje XML no detecta directamente y que han de ser sustituidos por el código correspondiente que XML sí reconoce. Una vez importado a Moodle, estos códigos se visualizan como el propio símbolo. Por ejemplo, el símbolo '<' es cambiado por el código '&lt;' y una vez importado a Moodle, este código se visualiza como el propio símbolo '<'.

Para el uso de esta función son necesarias tres entradas: *line*, la línea a tratar, la cual debe ser un *string*; 'símbolos', que debe ser una lista con los símbolos que se desean sustituir; y finalmente *codes* que debe ser una lista en la que se encuentren los códigos que deben reemplazar cada uno de los símbolos. Estas dos listas deben estar en orden para que la correspondencia símbolo-código sea correcta. El funcionamiento se basa en iterar la lista de símbolos analizando si cada uno de los signos que se recorre se encuentra en la línea que se está tratando y, de ser así, se utiliza la función *replace* para cambiar este símbolo por su correspondiente código. Se puede ver el funcionamiento de forma más sencilla en el diagrama de flujo siguiente:

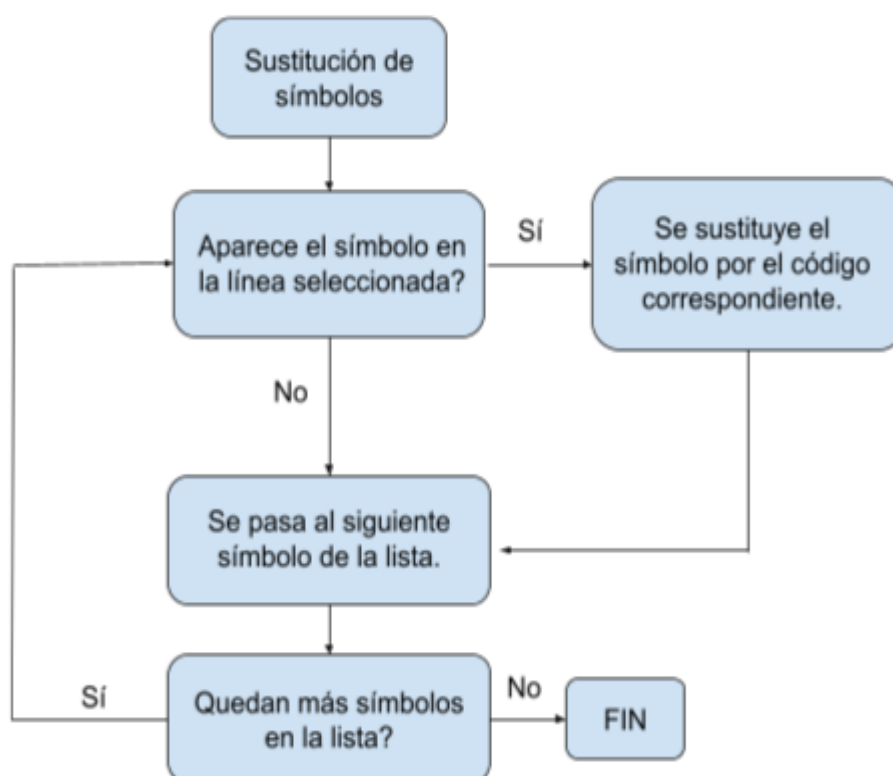


Figura 8: Diagrama de flujo función de sustitución de símbolos. Fuente: propia.

No es necesario que el usuario defina ninguna de las tres entradas de esta función, ya que ésta será llamada por el programa principal cuando sea necesario. Las listas de símbolos y códigos están definidas internamente y creadas a partir del análisis de los símbolos que aparecen en los diferentes exámenes.

#### 5.4. Función subíndices y superíndices

La siguiente función auxiliar no es una función externa que el programa principal deba llamar. Son unas líneas de código que se han tenido que añadir a la función principal para que ésta detecte y escriba en el archivo XML de manera correcta, tanto en subíndices como en superíndices.

Como se verá a continuación, la manera de funcionar del código se basa en almacenar de forma ordenada en tres listas diferentes las preguntas, las posibles respuestas y las respuestas correctas. Lo que se consigue con las nuevas líneas de código añadidas es que el texto se almacene en listas con el formato XML correcto para subíndices y superíndices.

Este código permite analizar los *runs* en busca de subíndices y superíndices y, cuando

detecta uno de ellos, lo que hace es aplicar la función ‘sustituir’ al *paragraph* que se está analizando, para posteriormente añadir “<sub>” delante de los caracteres que sean subíndices y “</sub>” detrás de estos caracteres. Con los superíndices es el mismo proceso, pero añadiendo “<sup>” y “</sup>” respectivamente.

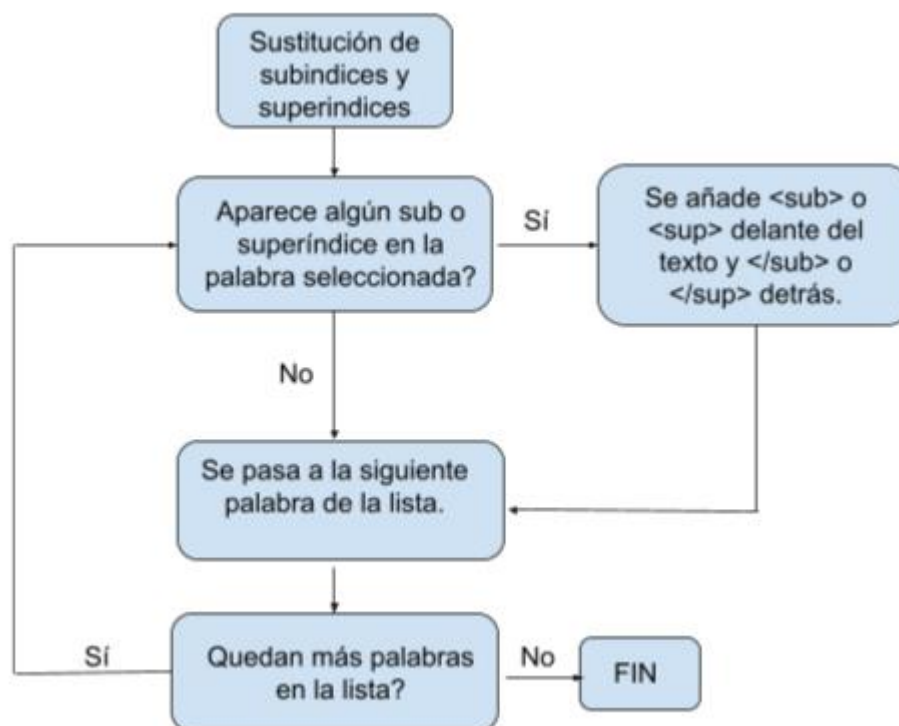


Figura 9: Diagrama de flujo función de sustitución de subíndices y superíndices. Fuente: propia.

La razón por la que se aplica la función ‘sustituir’ antes de analizar si hay subíndices o superíndices es que si se hace posteriormente se detectan los símbolos ‘<’, ‘>’ y éstos son sustituidos por el código correspondiente y en este caso necesitamos que no suceda esta sustitución, ya que sirven para indicar un tipo de formato y no forma parte del propio texto que aparecerá en las preguntas.

## 5.5. Función imágenes

En este caso, para conseguir incluir las imágenes en el archivo XML resultante, se ha hecho uso de 3 funciones auxiliares, así como de añadir ciertas líneas de código nuevas al programa principal.

La primera función auxiliar creada es *get\_images*, un código extraído de internet [8]. Se encarga de extraer y depositar en el lugar indicado las imágenes del examen. Esta función extrae las imágenes en formato EMF, que es el formato habitual de los dibujos de *Microsoft*

*Paint*. Para poder insertar imágenes en el archivo XML de Moodle es necesario que tengan formato PNG y para cumplir con este requisito se ha creado la función 'formato', que guarda las imágenes EMF extraídas del examen en formato PNG en la propia carpeta indicada anteriormente. Esta función no transforma el formato de las imágenes, sino que crea unas nuevas en el formato adecuado.

Finalmente, para poder añadir las imágenes al XML y que se vean correctamente en Moodle, éstas deben estar codificadas en el sistema de codificación *base64*, que es la utilidad que aporta la última función, llamada *encode*.

## 5.6. Código para preguntas VF

Si se analiza el formato de los exámenes que vamos a procesar, lo primero que hay es un encabezado en que se indica la asignatura de la que se trata y el año y mes de dicho examen.

**ELECTRÒNICA**

SEGONA PROVA PARCIAL CURS TARDOR 2013  
28 de NOVEMBRE del 2013  
TOTS ELS GRUPS

**INSTRUCCIONS**

1. Ompliu totes les vostres dades personals en els espais a l'efecte.
2. Poseu el vostre nombre del DNI en el rectangle superior esquerre del full de respostes.
3. Poseu l'identificador de la prova **240-00072-02-0-00** en el rectangle superior central del full de respostes (centre - assignatura - parcial - permutació - grup).
4. **Empreu la numeració de la dreta de les respostes** (numeració amb possibilitat d'anul·lació). Responeu cada pregunta en la casella associada al seu número identificador. Marqueu només una alternativa per pregunta.
5. Utilitzeu **bolígraf o retolador negre**, omplint bé tota la casella. Per fer correccions, empreu preferiblement les files d'anul·lació en comptes d'esborrar les marques de les files principals: **la resposta que doneu és la que està marcada a la fila principal (superior) i no està marcada a la fila d'anul·lació (inferior)**. Si heu d'esborrar marques, empreu *Tipp-Ex*.
6. No es pot emprar cap tipus de documentació ni cap aparell de comunicació, ni de càlcul, ni d'emmagatzematge (telèfon mòbil, PDA, calculadora, tauleta, reproductor d'àudio, etc.).
7. La prova consta de dues subproves. La primera subprova consta de 15 preguntes amb 2 respostes (preguntes 1 a 15). La segona subprova consta de 15 preguntes amb 5 respostes (preguntes 16 a 30). Els valors de N1 i N2 són 15 i 15 respectivament. Els valors de P1 i P2, corresponents als pesos de cada tipus de pregunta, són 1/6 i 0,5 respectivament. Es verifica  $P1 N1 + P2 N2 = 10$ .
8. En la correcció de la primera subprova se sumarà un punt per cada resposta correcta i es descomptarà 1 punt per cada resposta incorrecta, obtenint-se un valor Q1. En la correcció de la segona subprova se sumarà un punt per cada resposta correcta i es descomptaran 0,25 punts per cada resposta incorrecta, obtenint-se un valor Q2. La qualificació de la prova vindrà donada per la suma ponderada  $P1 Q1 + P2 Q2$ , la qual es considerarà 0 si resulta negativa.
9. La durada de la prova és d'una hora i tres quarts. Mentre duri la prova no es poden treure els enunciats de l'aula d'examen i el vostre DNI ha de romandre ben visible damunt la taula.

Figura 10: Primera pàgina del examen de noviembre de 2013. Fuente: propia.

Seguidamente se encuentran las instrucciones necesarias para realizar el examen de forma

presencial y, tras esto, vienen las preguntas VF. Estas preguntas siempre tienen la misma estructura, que consiste en la afirmación, sobre la cual el alumno debe decidir si es VF, seguida de A si es verdadera y B si es falsa.

3. L'ordre d'un filtre és el grau del denominador de la seva funció de transferència. A
4. En un comparador amb histèresi, l'amplificador operacional està realimentat negativament. B

Figura 11: Dos preguntas VF del examen parcial de otoño 2013. Fuente: propia.

Por tanto, el código creado analiza cada una de las palabras y, en el momento en que detecta algún carácter subrayado, escribe en el archivo XML la línea completa, excepto la última letra, que sería la respuesta. Según si esta última letra es A o B, escribe en el archivo XML si la afirmación es verdadera o falsa:

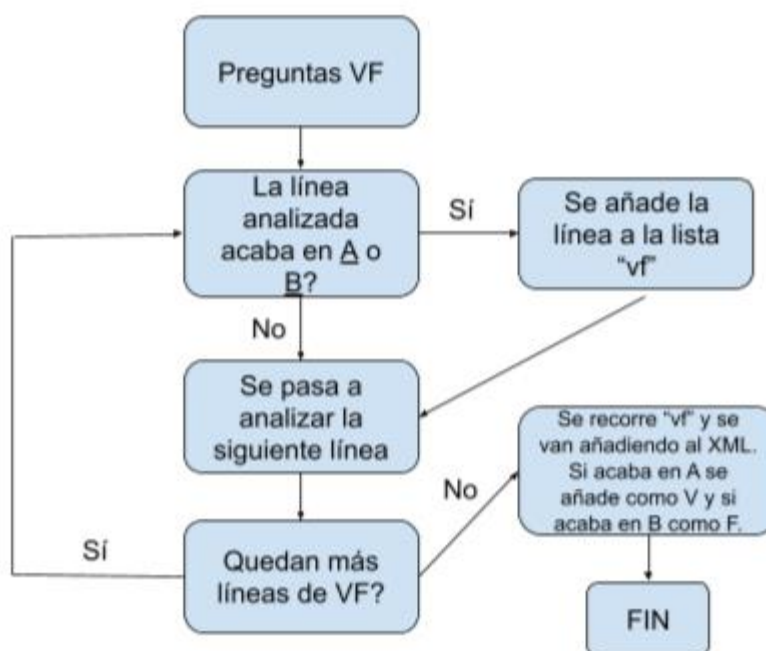


Figura 12: Diagrama de flujo función de preguntas VF. Fuente: propia.

## 5.7. Código para preguntas test

Si se continúa con el análisis de la estructura de los exámenes de la asignatura 'Electrónica', tras las preguntas VF lo que nos encontramos es una línea que anuncia el inicio de las preguntas test, con el siguiente texto:

PREGUNTES ABCDE (preguntas 41 a 60)

41. Indiqueu quina de les següents afirmacions és falsa en relació al paquet de CAD electrònic OrCAD de Cadence.
- A) Inclou un editor d'esquemàtics anomenat Capture.
  - B) Té com a simulador el software PSpice.
  - C) Té com a visualitzador el software Probe.
  - D) Per simular un circuit cal utilitzar un perfil de simulació.
  - E) Les biblioteques de models són fitxers amb extensió olb.

*Figura 13: Pregunta tipo test del examen de reevaluación de 2018. Fuente: propia.*

Por lo que puede deducirse que este código empieza a funcionar una vez detecta el *string* "PREGUNTES" y que, a partir de aquí, se diseña una serie de condiciones que permite almacenar de forma ordenada en una lista llamada "j" todos los enunciados de las preguntas test del examen. En otra lista llamada "p" se almacenan -también de forma ordenada-, todas las posibles respuestas de cada una de las preguntas. Finalmente, en una lista llamada "d" se almacenan las respuestas correctas a cada una de las preguntas.

Almacenar todo ordenadamente en las listas permite que, iterando la lista "j" vayamos añadiendo al XML las preguntas. Para incorporar las posibles respuestas se recurre a un bucle y un contador que se inicializa a 0 en cada iteración de "j" y que suma 1 con cada posible respuesta añadida al XML. Antes de integrar las opciones de cada pregunta, éstas se comparan con las almacenadas en "d" para determinar si es correcta o no. Cuando el contador llega a 5, que es el número de opciones que hay en cada pregunta, este bucle termina y se pasa a la siguiente iteración de "j", es decir, a la siguiente pregunta del examen. Conforme se van añadiendo las opciones al archivo de salida, se van eliminando de la lista "p" así como de "d" y, cuando estas listas están vacías, el código termina.

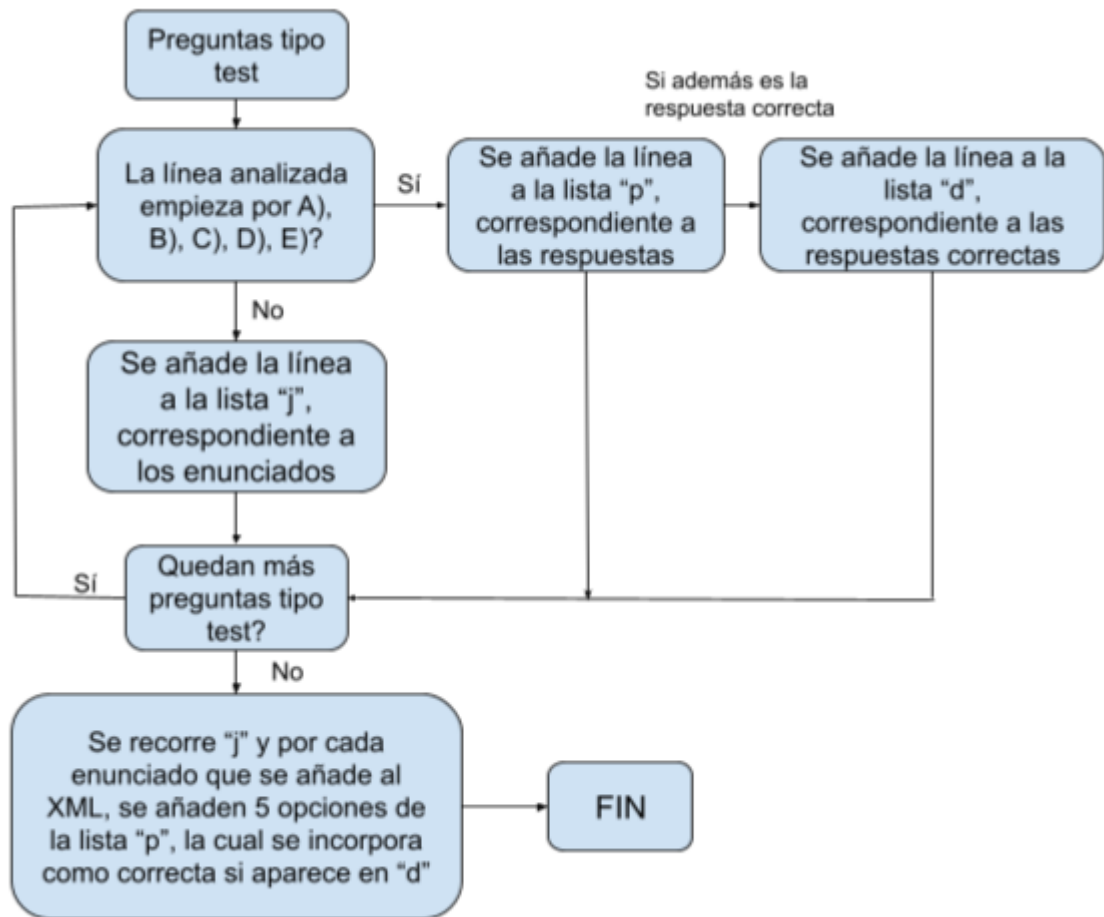


Figura 14: Diagrama de flujo función de preguntas tipo test. Fuente: propia.

También es importante mencionar que esta función se ha programado de manera que en los casos en que las preguntas pertenecen a un mismo problema, se mantiene el enunciado inicial -marcado con un recuadro rojo en la Figura 15- en todas ellas para poder entender el contexto de la pregunta. Sucede lo mismo con los problemas de varias preguntas que tienen una figura en común, la herramienta está programada para que la figura se incluya en todas las preguntas pertenecientes al problema:

**PROBLEMA** (preguntes 16 a 19)

Es desitja implementar un autòmat síncron de Mealy que detecti la seqüència de bits 0101 sense solapaments (el darrer 01 no serveix com a primer 01 d'una altra seqüència), en una línia X síncrona amb el rellotge de l'autòmat. La sortida Y de l'autòmat s'ha d'activar ( $Y = 1$ ) durant un sol període del rellotge cada cop que es detecta dita seqüència. Concebeu el diagrama d'estats de l'autòmat amb el mínim d'estats, codifiqueu aquests estats de forma que, interpretant en binari natural la codificació de l'estat, l'estat inicial sigui el 0 i des de l'estat Q només es pugui passar als estats  $Q^+ \leq Q+1$ , obteniu les funcions de sortida i de transició de l'autòmat i responeu les següents quatre preguntes.

16. Indiqueu quina de les següents afirmacions sobre l'autòmat és falsa:

- A) Un estat recorda la introducció d'una seqüència de 0 bits.
- B) Un estat recorda la introducció d'una seqüència de 1 bit.
- C) Un estat recorda la introducció d'una seqüència de 2 bits.
- D) Un estat recorda la introducció d'una seqüència de 3 bits.
- E) Un estat recorda la introducció d'una seqüència de 4 bits.

17. Indiqueu quina de les següents afirmacions sobre el graf de transicions de l'autòmat és falsa:

- A) De l'estat 1 només es pot passar als estats 1 i 2.
- B) De l'estat 2 només es pot passar als estats 2 i 3.
- C) De l'estat 3 es pot passar a l'estat 0.
- D) De l'estat 3 es pot passar a l'estat 1.
- E) Marqueu aquesta resposta si totes les afirmacions anteriors són certes.

18. Considerant que l'autòmat s'implementa amb biestables D amb sortides Q i  $\bar{Q}$ , la sortida Y es pot implementar amb una porta:

- A) XOR2      B) OR2      C) NOR3      D) AND3      E) NAND2

Figura 15: Problema del examen final de primavera 2015. Fuente: propia.

## 5.8. Funcionamiento general de la herramienta

Ahora que se conoce el funcionamiento individual de todas las funciones creadas, se va a proceder a la explicación sobre cómo se relacionan para crear la herramienta propuesta en este proyecto.

Lo primero que se encuentra al ejecutar la herramienta, sea cual sea el examen que se desea transformar a XML, es el inicio de la función para preguntas VF que funciona hasta que se detecta el *string* "PREGUNTES", momento a partir del cual termina la función para preguntas VF y se inicia la función para preguntas tipo test. Durante el periodo en que estas dos funciones están activas, paralelamente también están en uso las funciones auxiliares de sustitución de símbolos, de subíndices y superíndices que aplican los cambios necesarios antes de añadir el texto a las listas correspondientes. En cambio, las tres funciones auxiliares creadas para el tratamiento de las imágenes únicamente actúan en paralelo con la función de preguntas tipo test. Después de pasar por estas funciones



auxiliares, las preguntas se añaden al archivo XML y una vez no quedan líneas que analizar, la herramienta finaliza:

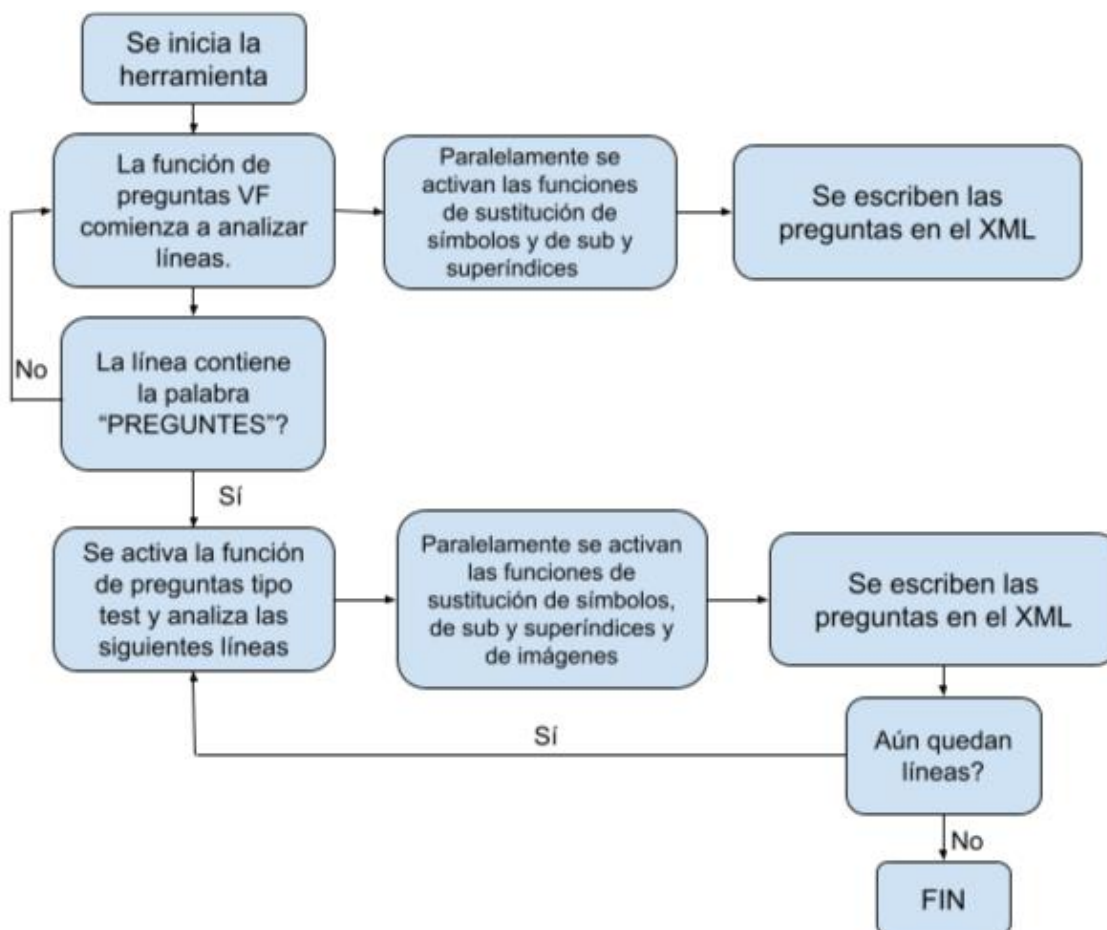


Figura 16: Diagrama de flujo del funcionamiento completo de la herramienta. Fuente: propia.

## 5.9. Resultados

En este apartado se va a proceder a mostrar cómo se aprecian las preguntas una vez se importan a Moodle, primero ejemplificando con una pregunta VF y seguidamente una tipo test. Para finalizar, también se podrá ver un cuestionario de 20 preguntas aleatorias creado a partir del banco de preguntas que se ha ido generando durante el desarrollo de este proyecto.

En la siguiente figura se muestra una pregunta VF, en la que se puede ver que este tipo de preguntas se importan sin ningún problema:

Pregunta 1  
No s'ha respost  
encara  
Puntuat sobre  
1,00

Per reduir l'efecte del soroll, s'utilitzen senyals diferencials en combinació amb parells trenats.

Triu-ne una:

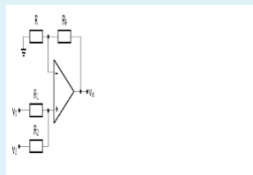
- Vertader  
 Fals

Figura 17: Pregunta VF ya importada a Moodle. Fuente: propia.

A continuación, se ejemplifica una pregunta de tipo test en la que se puede apreciar como la herramienta funciona correctamente con imágenes y subíndices. Además, también puede comprobarse cómo la herramienta es capaz de mantener el enunciado y la figura para todas las preguntas que componen un problema, de manera que todas las preguntas contienen el contexto necesario para ser contestadas. Esto mismo puede apreciarse en la Fig. 18, última pregunta de las dos que componen el problema:

Pregunta 1  
No s'ha respost  
encara  
Puntuat sobre  
1,00

Considerant que l'amplificador operacional treballa a la zona lineal, analitzeu el circuit de la figura i responeu les següents dues preguntes. Indiqueu quin valor ha de tenir la relació  $R_F/R$  per tal que la tensió a la sortida del circuit valgui  $v_O = v_1 + 2 v_2$ :



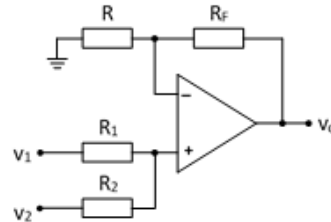
Triu-ne una:

- ERR C) 1  
 ERR A) 0,25  
 ERR B) 0,5  
 ERR D) 2  
 ERR E) 3

Figura 18: Pregunta tipo test ya importada a Moodle. Fuente: propia.

**PROBLEMA** (preguntes 19 a 20)

Considerant que l'amplificador operacional treballa a la zona lineal, analitzeu el circuit de la figura i responeu les següents dues preguntes.



19. Indiqueu quin valor ha de tenir la relació  $R_2/R_1$  per tal que la tensió a l'entrada no inversora de l'amplificador operacional valgui  $(1/3)v_1 + (2/3)v_2$ :
- A) 0,25      B) 0,5      C) 1      D) 2      E) 3
20. Indiqueu quin valor ha de tenir la relació  $R_F/R$  per tal que la tensió a la sortida del circuit valgui  $v_0 = v_1 + 2v_2$ :
- A) 0,25      B) 0,5      C) 1      D) 2      E) 3

Figura 19: Problema correspondiente al segundo examen parcial de otoño 2014. Fuente: propia.

Finalmente, se muestra un cuestionario aleatorio -como se puede ver en la Figura 21- creado únicamente con preguntas importadas a Moodle por la herramienta. Se ha seleccionado que el cuestionario contenga 20 preguntas, pero se puede seleccionar cualquier otro número y Moodle creará automáticamente el cuestionario.

**Pregunta 1**

No s'ha respost encara

Puntuat sobre 1,00

Marca la pregunta

Edita la pregunta

Pregunta 4

Una memòria d'accés aleatori és un dispositiu de caire seqüencial capaç d'emmagatzemar  $2n$  informacions binàries de  $m$  bits, les quals poden ser objecte de lectura i d'escriptura mitjançant operacions que presenten una durada curta i del mateix ordre de magnitud.

Trieu-ne una:

Vertader

Fals

NAVEGACIÓ PEL QÜESTIONARI

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20				

Acaba l'intent...

Comença una previsualització nova

Pàgina següent

Figura 20: Primera pregunta del cuestionario creado a partir del banco de preguntas. Fuente: propia.


































Pàgina 1	 1	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 2	 2	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 3	 3	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 4	 4	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 5	 5	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 6	 6	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 7	 7	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 8	 8	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 9	 9	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 10	 10	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>
Pàgina 11	 11	  Aleatòria (Exàmens Electrònica i les subcategories) <a href="#">(Vegeu les preguntes)</a>

Figura 21: Resumen de las preguntas del cuestionario. Fuente: propia.

## 6. Datos técnicos de la herramienta

Una vez revelado el funcionamiento de la herramienta y los resultados que pueden obtenerse con ésta, seguidamente se procederá a la aportación de los datos técnicos que han hecho posible la invención del proyecto. De este modo, no solamente se podrá comprender el funcionamiento a nivel interno del mecanismo, sino que también se pondrá en manifiesto la complejidad y el nivel de dificultad que presenta.

### 6.1. Bibliotecas utilizadas

Lo primero a tener en cuenta es que, a pesar de únicamente nombrar la biblioteca *docx* durante los apartados anteriores, se han usado 4 bibliotecas más, todas ellas incluidas de forma predeterminada en las versiones más recientes de Python. Estas son las bibliotecas *re*, *os*, *PIL* y *base64*.

La biblioteca *re* [10] -son las siglas en inglés de 'expresión regular'- . Las expresiones regulares especifican un conjunto de cadenas que coinciden con ella; las funciones de este módulo permiten comprobar si una determinada cadena coincide con una expresión regular dada o si una expresión regular dada coincide con una determinada cadena, que se reduce a lo mismo. Esta biblioteca ha sido usada en la función *get\_pictures* que, a pesar de ser extraída de internet [8], se le han hecho pequeñas modificaciones y se ha estudiado su funcionamiento.

La biblioteca *os* [11] -en este caso son las siglas en inglés de sistema operativo-, sirve para usar funcionalidades dependientes del sistema operativo, como leer o escribir un archivo, manipular rutas, crear archivos temporales y directorios, etc. Esta biblioteca es usada en las funciones *get\_pictures* -para detectar si existe el directorio en que se han de guardar las imágenes, de no ser así, lo crea-; preguntas VF y preguntas tipo test -para separar o terminar de escribir líneas en el XML-.

La biblioteca *PIL* [12], de la cual se ha usado únicamente el módulo *Image*, permite la edición de imágenes y soporta una gran variedad de formatos. Éste módulo es el que ha permitido a la función 'formato' poder abrir la imagen en formato EMF y guardarla en PNG.

La biblioteca *base64* [13] como su nombre indica, ha sido usada para poder codificar las imágenes en *base64* dentro de la función *encode*, lo que permite poder importar las figuras a Moodle.

Finalmente, la biblioteca *docx* [7] como ya se ha mencionado anteriormente, es la biblioteca principal para el desarrollo de la herramienta y sirve para el tratamiento de documentos Word, ya sea para la creación de uno nuevo o la modificación de uno existente.

## 6.2. Funciones creadas

Durante el desarrollo de la herramienta se han creado un total de 5 funciones independientes, aunque una de ellas se podría dividir en 3 subfunciones diferentes. La suma de todas ellas genera un total de 345 líneas de código escritas desde cero para dar lugar a la herramienta.

De las 5 funciones creadas, 4 de ellas son auxiliares, siendo la primera la correspondiente a la función 'sustituir' y las tres restantes las necesarias para el tratamiento de las imágenes. Estas funciones no contienen en su interior ninguna otra subfunción ya que son notablemente más sencillas que la función principal, la cual contiene en su interior 3 subfunciones. Por lo tanto, la función principal llamada 'extreutest' está conformada principalmente por dos subfunciones que son las correspondientes al código necesario para la extracción de preguntas VF y preguntas tipo test, pero durante el funcionamiento de estas, se activa una subfunción para generar los subíndices y superíndices siempre que sea necesario. Para poner en magnitud el valor de todas estas subfunciones se ha de mencionar que la función principal está compuesta por 274 líneas de las 345 totales, lo que supone que, junto con sus 3 subfunciones, correspondan al 80% del código de la herramienta.

## 7. Test y Validación

Para la creación de la herramienta han sido necesarias muchas pruebas que han permitido ir depurando los errores que se cometieron en la redacción inicial del código Python. Como se ha explicado con anterioridad, la creación de la herramienta se ha llevado a cabo por partes, de modo que las pruebas también se han realizado por separado.

Inicialmente, se redactó el código de las preguntas VF y para comprobar su funcionamiento, se ejecutaba el programa y se comprobaba si el archivo XML resultante era correcto y contenía las preguntas correspondientes al examen que se deseaba. El proceso de comprobación era primero visual, ya que a simple vista se puede constatar si el número de preguntas era el correcto y si se incluían todos los enunciados -dado que un fallo recurrente era la creación de preguntas sin enunciado-. Cuando se detectaba un error, se creaban líneas de código extras para solucionarlo de manera universal y no solo para ese examen concreto. Una vez el XML pasaba la inspección visual, se trataba de importar el archivo a Moodle, donde en un primer momento aparecieron varios errores de formato en el XML como por ejemplo si había algún símbolo no reconocido o si faltaba algún *tag*. Estas primeras inspecciones a través de Moodle fueron de gran utilidad para la redacción de las líneas de código que escriben los *tags* del XML y que son comunes para todos los exámenes y preguntas, además de servir de soporte en la creación de la función de sustitución de símbolos.

Para todas estas comprobaciones, se utilizó una base de 10 exámenes, que son los correspondientes a 2014 y 2015, y se realizaron múltiples pruebas en cada uno de ellos hasta obtener un código depurado y sin errores. Una vez no se obtuvo ningún tipo de error en ninguno de los 10 exámenes comprobados fue cuando se pasó a la creación del código de preguntas tipo test.

El proceso para comprobar el correcto funcionamiento del código Python correspondiente a las preguntas tipo test es el mismo que el anterior, aunque en este caso se encontraron muchos más fallos y, por lo tanto, se requirió de muchas más pruebas. Uno de los fallos más recurrentes era que el programa no detectaba alguno de los enunciados, por lo que a partir de esta pregunta los enunciados no se correspondían con las respuestas. Otro de los fallos más comunes era que el programa no detectaba -en este caso-, alguna de las respuestas, dando lugar al mismo problema: la no correlación entre enunciado y respuestas. El método de comprobación también es igual al explicado, primero una comprobación visual -en este caso más rigurosa, ya que se debía comprobar la correspondencia enunciado-respuestas-, y, seguidamente, se importaba a Moodle para examinar que todo era correcto y las preguntas se podían importar sin problema.

En este caso, se hizo la comprobación con 13 exámenes, los 10 mencionados anteriormente y 3 de 2016. En estos exámenes no se detectó ningún error grave, pero sí que se observó que había errores puntuales, como no ser capaz de detectar el contenido de las ecuaciones. Estos errores aislados no han sido posibles de resolver por cuestiones de limitación temporal.

Para depurar las funciones auxiliares se ha utilizado el mismo método que en las funciones explicadas anteriormente, sin embargo, en este caso las pruebas no se hacían en exámenes completos, sino que se creó un Word con únicamente una pregunta que contenía una imagen, un enunciado con varios subíndices y superíndices, y algunos símbolos. Al usar este Word, la inspección visual era mucho más rápida, ya que no había que buscar entre las preguntas de un examen hasta encontrar alguna que contuviera alguno de estos elementos. Una vez esta única pregunta fue exportada correctamente con la figura, símbolos e índices, la comprobación pasó a ser en exámenes completos.

Tras la importación a Moodle de las preguntas de los exámenes utilizados en la depuración del código, se ha seguido ampliando el banco de preguntas a través del uso de la herramienta. En total se han procesado 45 ficheros Word, dando como resultado la creación de un banco de preguntas de más de 1200 preguntas entre VF y tipo test.

Es importante destacar –al menos en lo que se refiere al proceso que se ha experimentado durante la redacción de todas estas funciones-, que, en ocasiones, el reto no se encuentra tanto en la redacción de un código correcto, más bien, lo que verdaderamente presenta una mayor dificultad es el proceso que se debe llevar a cabo para conseguirlo. Con ello se quiere dar a entender que más allá de la aparente complejidad del código, la dificultad y el trabajo radica en el proceso. En encontrar, primero, un procedimiento a seguir para conseguir el objetivo deseado, para posteriormente realizar numerosas pruebas sobre este para determinar si el procedimiento pensado puede ser funcional o si por el contrario, se ha de buscar uno nuevo. Una vez encontrado un procedimiento que podría ser funcional, este tiene que ser depurado a través del número de pruebas que sea necesario, lo cual puede suponer un minuto o una semana. En suma, obtener el código resulta laborioso porque se trata de lograr una idea, un resultado, que, para llegar a él debe existir un proceso, que uno mismo debe construir a partir de su propia invención y es ahí donde reside la dificultad de este proyecto.



## 8. Guía de usuario

(Todos los archivos necesarios para el uso de la herramienta se encuentran en el anexo de este proyecto.)

### 8.1. Pasos previos

#### Paso previo 1:

El primer paso para poder hacer uso de la herramienta es instalar una consola capaz de ejecutar Python. En caso de ya tener instalado cualquier medio que permita ejecutar Python vaya al paso previo 2.

A través de la web de Python [9], se debe descargar la última versión de Python clicando el botón 'Download Python 3.x'

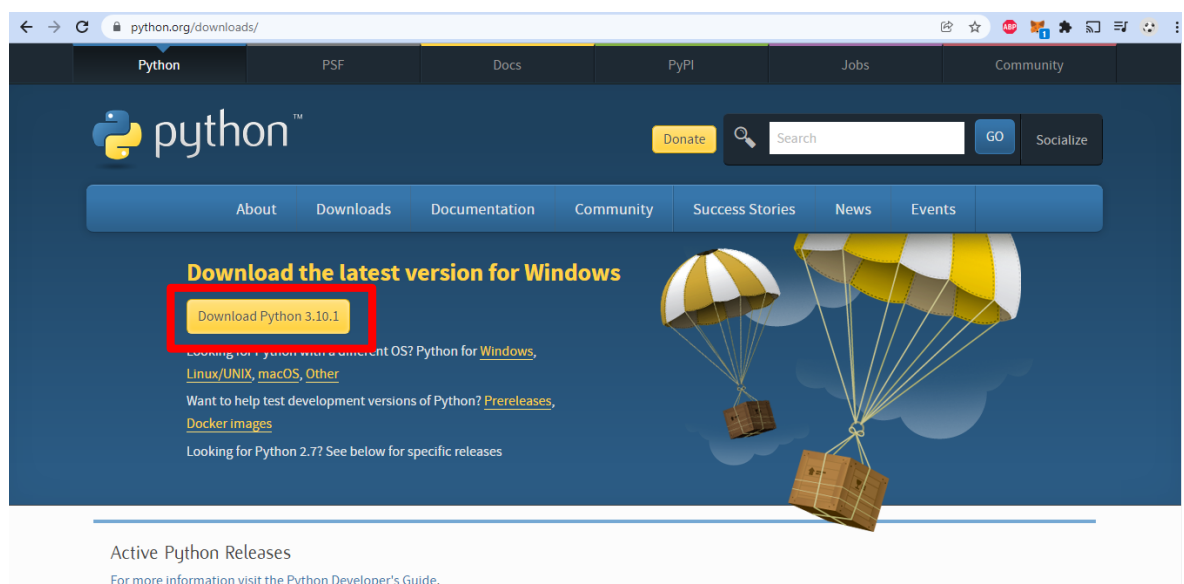


Figura 22: Web de descarga de Python. Fuente: propia.

Se descargará un archivo, que se ha de ejecutar y se abrirá una ventana nueva en la que se debe hacer clic en el botón 'Install now'.

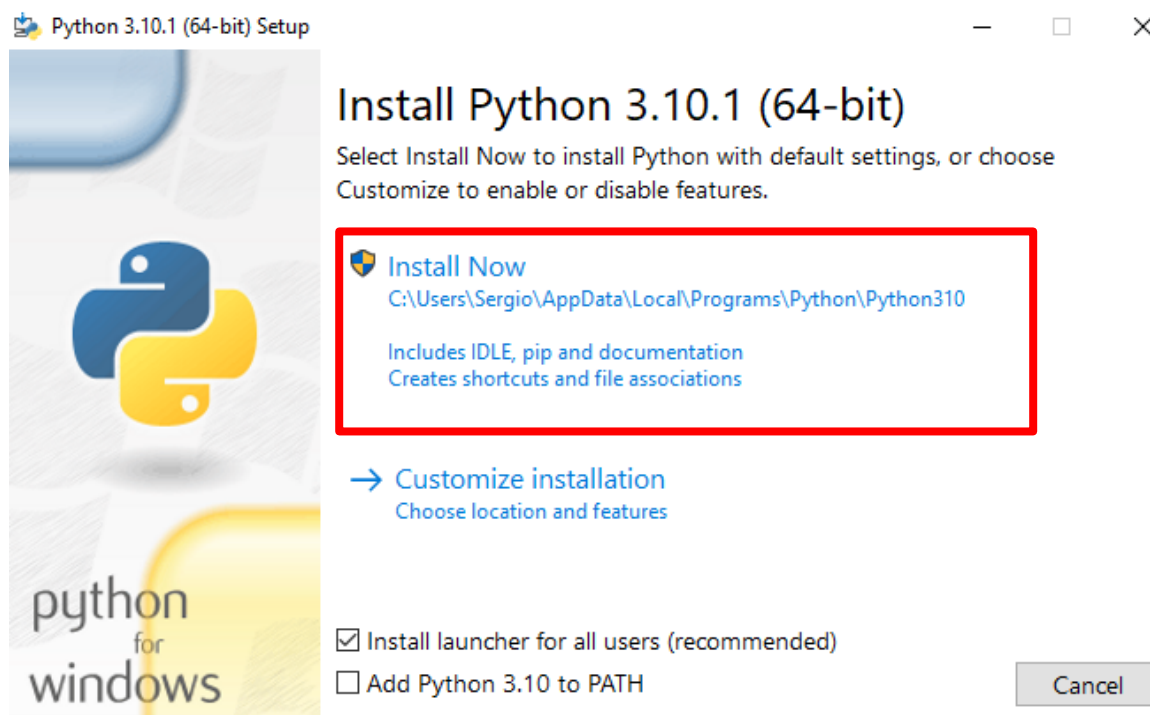


Figura 23: Pantalla de instalación de Python. Fuente: propia.

Aparecerá una barra de carga durante el proceso de instalación, y una vez completada la instalación, aparecerá una ventana que notifica que el proceso ha sido correcto.

### Paso previo 2:

El siguiente paso es instalar las bibliotecas necesarias para el funcionamiento de la herramienta. Para ello, es necesario buscar en la barra de inicio del ordenador 'Símbolo del sistema' y al abrirlo aparecerá una ventana con el siguiente aspecto:

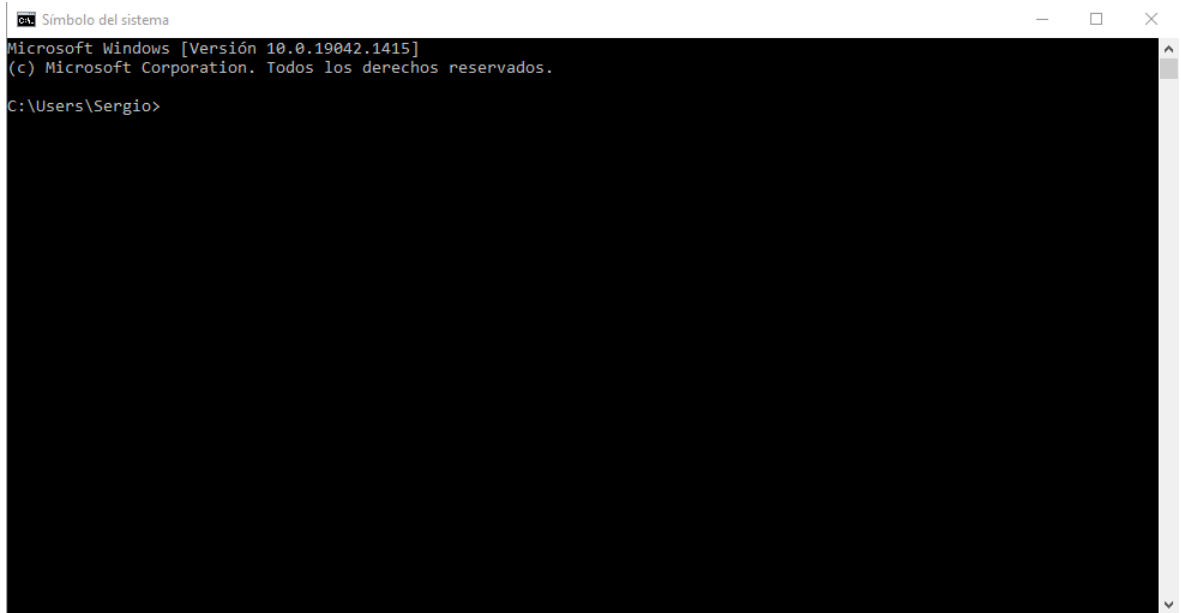


Figura 24: Pantalla de Símbolo del sistema. Fuente: propia.

Una vez aquí se debe de escribir lo siguiente: `'pip install python-docx'` y tras presionar la tecla *Enter*, empezará el proceso de instalación de la biblioteca.

Aquí finalizan los pasos previos, pero si el usuario ya tenía instalado Python previamente y únicamente ha instalado la biblioteca, puede que necesite alguna biblioteca complementaria, ya que las versiones actuales de Python incluyen todas las necesarias para que la herramienta funcione, pero con versiones anteriores podría ser diferente. Para instalar cualquier biblioteca el proceso es el mismo que el explicado anteriormente, pero escribiendo: `'pip install (nombre de la biblioteca)'`.

## 8.2. Pasos de utilización de la herramienta

### Paso 1, Microsoft Word:

Para empezar a usar la herramienta de generación de exámenes lo primero que se tiene que hacer es aplicar las macros necesarias al archivo Microsoft Word del que se quieran extraer las preguntas. Para ello, una vez abierto el Microsoft Word, nos vamos al apartado “Vista” de la barra de tareas y seguidamente se presiona el botón “Macros”.

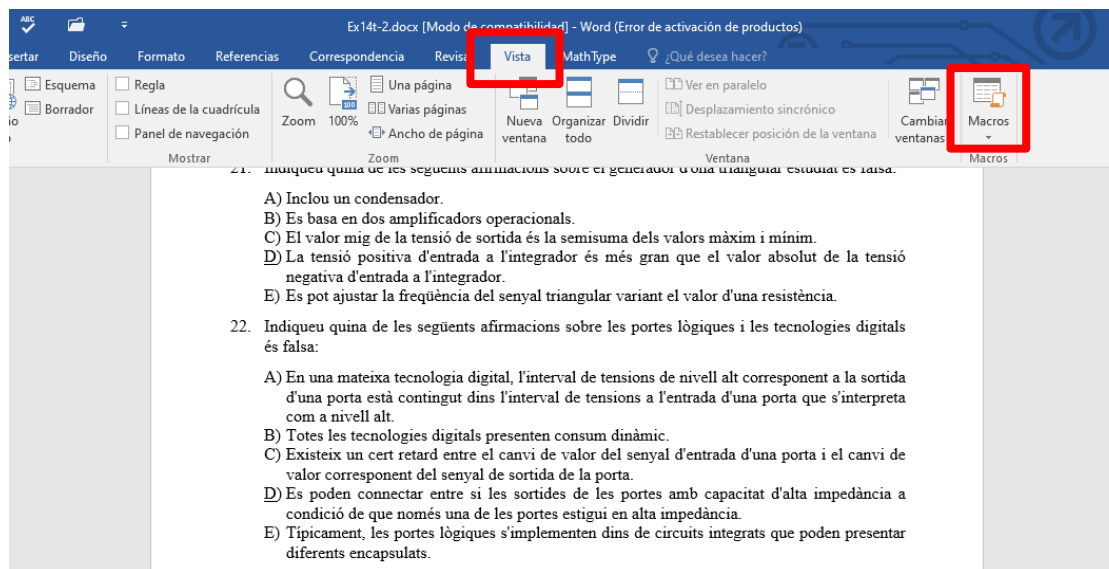


Figura 25: Localización de macros dentro de Word. Fuente: propia.

Se abrirá una ventana emergente en la que se debe seleccionar la macro que se desea aplicar, en este caso será primero la macro “símbolos”. Una vez seleccionada la macro se presiona “Ejecutar” y Microsoft Word comienza a procesar y a realizar los cambios correspondientes. Seguidamente se hace el mismo proceso con la macro *Get\_Images*.

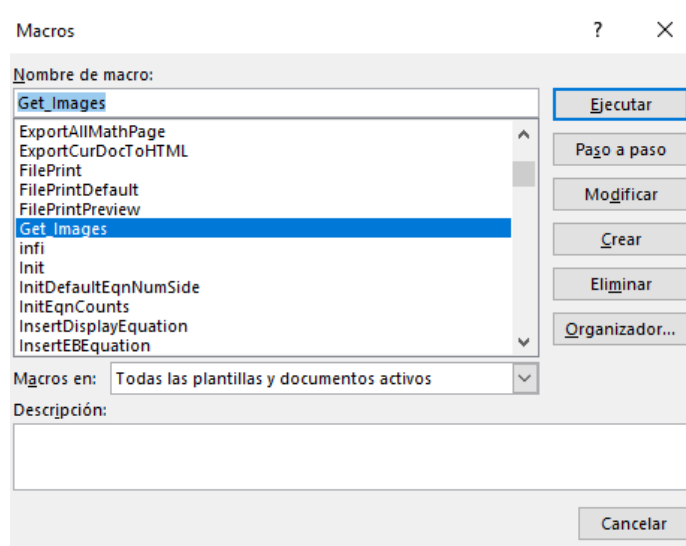
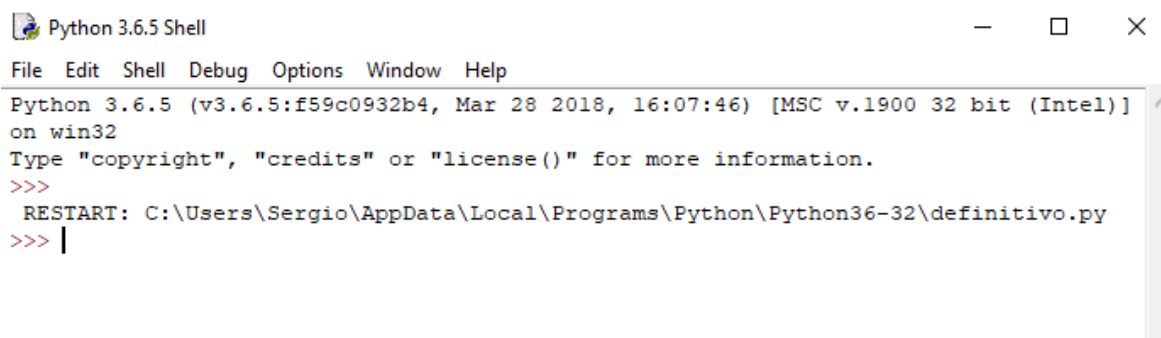


Figura 26: Ventana donde seleccionar la macro. Fuente: propia.

Una vez realizados estos pasos es importante guardar los cambios, ya que, si no, Python seguirá sin detectarlos. También es importante guardar el archivo de Microsoft Word en la carpeta en la que trabaja Python.

## Paso 2, Python:

En este punto ya se inicia la herramienta de generación de exámenes y para ello, es necesario ejecutar el archivo “*definitivo.py*” en la consola de Python que se desee.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Sergio\AppData\Local\Programs\Python\Python36-32\definitivo.py
>>> |
```

Figura 27: Pantalla de Python tras iniciar la herramienta. Fuente: propia.

El primer paso, una vez ejecutado el archivo, es llamar a la función *extreutest*, que requiere de tres entradas, la primera de ellas es *file* y corresponde a un *string* con el nombre del archivo *docx* del examen del cual se quieren extraer las preguntas. La siguiente entrada es “xml” y corresponde al nombre que se le quiera dar al archivo XML resultante. Finalmente, la última entrada es *direccion* y aquí es necesario indicar la dirección en que deseamos almacenar las imágenes extraídas del examen y ésta debe ser la misma carpeta en la que se encuentra el archivo de Microsoft Word.

Con esta información ya podemos llamar a la función de la forma: “*extreutest*(“Ex13t-1.docx”, “archivoresultante.xml”, “C:/Users/Sergio/AppData/Local/Programs/Python/Python 36-32”)”. Cuando la función termine, lanza por pantalla el siguiente mensaje: “Su archivo XML ha sido creado en la dirección indicada”, por lo que se puede proceder a la importación de las preguntas a Moodle.

## Paso 3, Moodle:

A través del botón “Preguntas”, que se puede ver en la siguiente figura, se accede al menú donde se importan las preguntas.

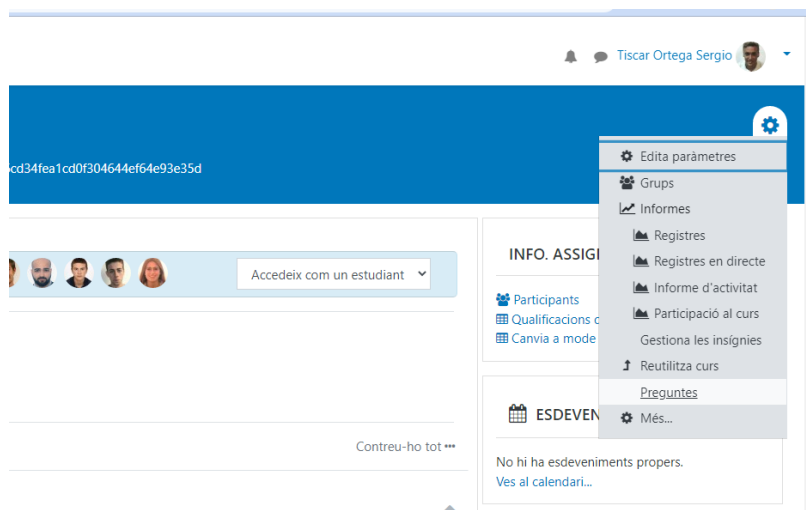


Figura 28: Localización del banco de preguntas dentro de Moodle. Fuente: propia.

Una vez dentro, en la pestaña “Importa”, se selecciona el formato de fichero a importar como XML y tras arrastrar el archivo creado por la herramienta al cuadro correspondiente, se presiona el botón “Importa” y automáticamente se importarán todas las preguntas, tanto VF como tipo test, del examen seleccionado:

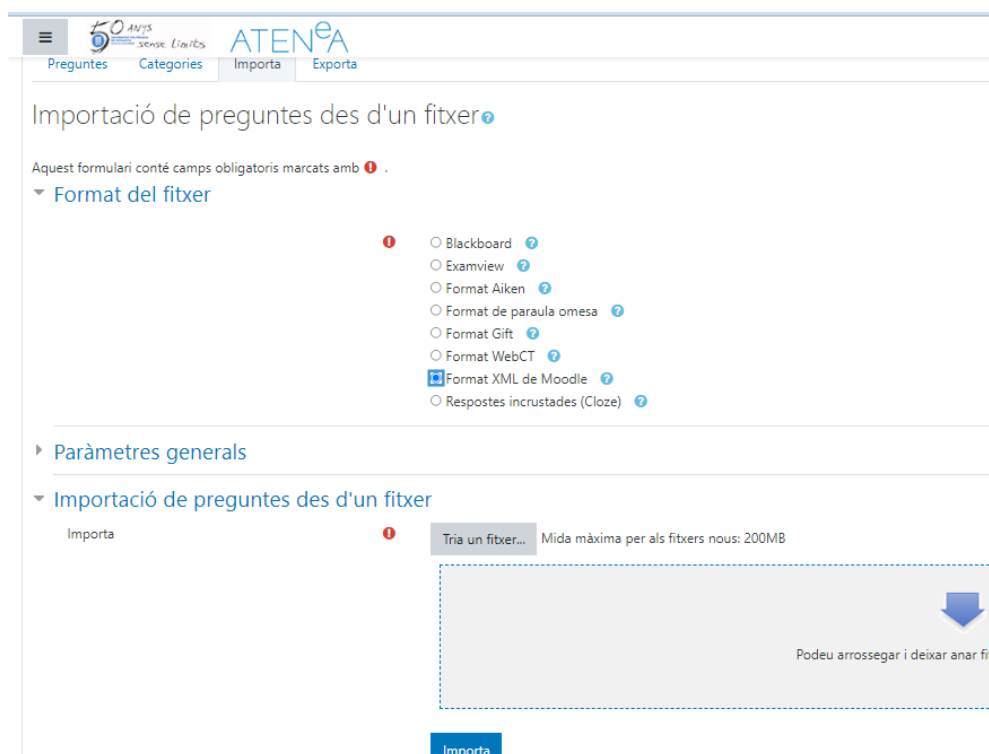


Figura 29: Proceso de importación de preguntas dentro de Moodle. Fuente: propia.

Todas ellas estarán incluidas en una subcategoría nueva, con el mismo nombre que el

Microsoft Word seleccionado inicialmente, dentro de la categoría superior “Exàmens Electrònica”.

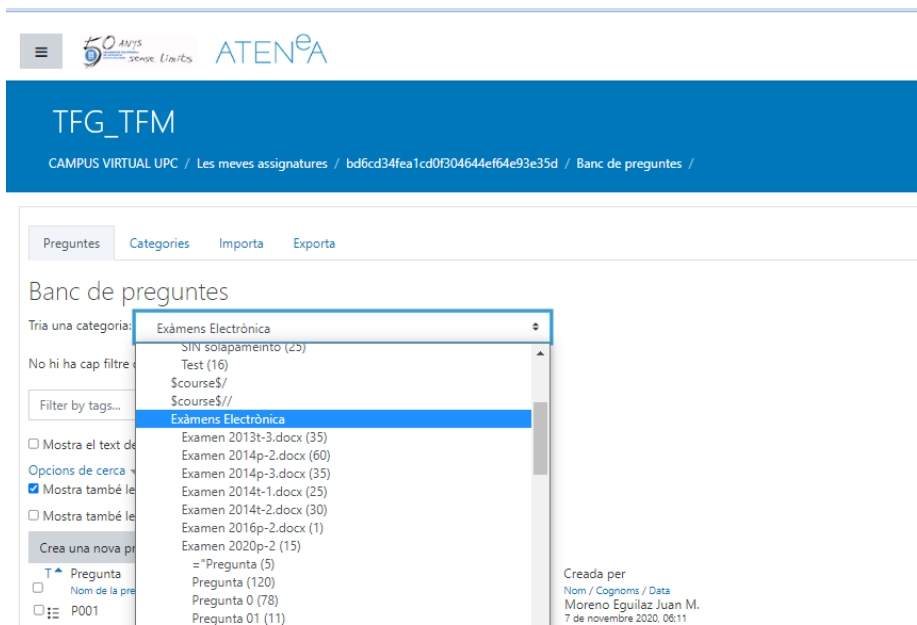


Figura 30: Categorías creadas por la herramienta Fuente: propia.

Una vez el banco de preguntas sea suficientemente amplio se podrán crear cuestionarios de forma aleatoria utilizando las preguntas de una subcategoría concreta o todas las incluidas en la categoría superior “Exàmens Electrònica”.

## 9. Impacto ambiental

Este proyecto, al no requerir de la creación de ningún prototipo, no tiene un impacto ambiental relacionado con materiales o procesos de construcción, por lo tanto, únicamente se relacionaría con el consumo eléctrico durante la fase de creación de la herramienta y el uso de ésta.

Por otro lado, esta herramienta ha sido creada con el objetivo de ahorrar tiempo a los docentes en el proceso de creación de exámenes no presenciales, por lo que el tiempo de estos usando un ordenador puede verse reducido y, por consiguiente, se consigue disminuir el consumo eléctrico. Además, esto facilita los exámenes en línea, lo que hace posible reducir la contaminación por parte de los estudiantes, que no han de trasladarse hasta la universidad para realizar el examen.

En definitiva, el desarrollo de este proyecto tiene un impacto ambiental mínimo, puesto que solamente se requiere de consumo eléctrico en la creación de la herramienta y, con un buen uso, se pueden alcanzar mejoras relevantes no solo en el aspecto ambiental, relacionadas al uso del transporte público debido a los desplazamientos por la presencialidad, sino en la calidad de vida del profesorado y alumnado.



## 10. Presupuesto

En este apartado se detalla el presupuesto total necesario para llevar a cabo el presente proyecto. Debido a que este trabajo no requiere la construcción de ningún prototipo, el gasto total proviene principalmente de dos grupos: los costes materiales y el uso de herramientas ofimáticas, y el coste de los recursos humanos. También se añaden otros conceptos -en los que se incluyen costes generales-, que corresponde a un 10% del total de las partidas. Este gasto cubre posibles imprevistos en la realización del proyecto, así como los gastos de suministros en electricidad e internet. El concepto beneficio industrial, que corresponde al 13% del total de las partidas, es el beneficio que recibe el creador del proyecto. Finalmente, se incluye la partida de impuestos, donde únicamente se encuentra el 21% de IVA. Para calcular la amortización del material utilizado se ha considerado una duración del proyecto de 0,4 años, es decir, de aproximadamente 5 meses:

Concepto	Cantidad	Precio unitario	Amortización	Precio total
<b>Coste de Recursos Humanos</b>				
Documentación	50 h	35,00 € / h	-	1.750 €
Aprendizaje de software	50 h	35,00 € / h	-	1.750 €
Desarrollo de la herramienta	200 h	35,00 € / h	-	7.000 €
Redacción de la memoria	80 h	35,00 € / h	-	2.800 €
<b>Coste de recursos Materiales y Ofimáticos</b>				
Ordenador portátil	1	899 €	0,4	359,60 €
Microsoft Office 365	1	69 €	0,4	28 €
<b>Otros conceptos</b>				
Gastos generales	10% sobre el total	-	-	1.369 €
Beneficio industrial	13% sobre el total	-	-	1.779 €
<b>Precio antes de impuestos</b>				
Suma de todas las partidas	-	-	-	16.835 €
<b>Impuestos</b>				
IVA	21% sobre el total	-	-	3.535 €
<b>TOTAL</b>				<b>20.371 €</b>

Tabla 1: Presupuesto. Fuente: propia.

## 11. Planificación del proyecto

En este apartado se expone la planificación seguida a lo largo de este proyecto, desde los primeros contactos con el tutor hasta la semana de entrega del documento, lo que representa una duración de 20 semanas aproximadamente.

Entre el 21/05/2021 y el 15/09/2021 transcurren las vacaciones de verano, por lo que no hay avances ni modificaciones en la planificación del proyecto.

En cuanto a la presentación del proyecto aún no se determina la fecha exacta, ya que aún no se puede saber, pero si se sabe que será el último paso del proyecto.

ASUNTO	FECHA DE INICIO	FECHA DE FINALIZACIÓN
Definición del proyecto	14/05/2021	21/05/2021
Estudio previo y búsqueda en trabajos anteriores	15/09/2021	22/09/2021
Definición de los objetivos y alcance del proyecto	22/09/2021	24/09/2021
Estudio y aprendizaje de Python	24/09/2021	01/10/2021
Creación del código para preguntas VF	01/10/2021	09/10/2021
Creación del código de sustitución de símbolos	09/10/2021	10/10/2021
Creación del código de subíndices y superíndices	10/10/2021	23/10/2021
Creación del código para incluir imágenes	23/10/2021	10/11/2021
Creación del código para preguntas tipo test	10/11/2021	25/12/2021
Redacción de la memoria	25/12/2021	24/01/2021
Presentación del trabajo	-	-

Tabla 2: Planificación. Fuente: propia.

## Conclusiones

Como conclusión a este proyecto se va a comprobar el nivel de cumplimiento de los objetivos propuestos al inicio de este, así como valorar posibles puntos a mejorar.

El objetivo principal de este proyecto era la creación de una herramienta que procese un archivo Microsoft Word y genere automáticamente un archivo XML listo para ser importado a Moodle, meta que ha sido resuelta con éxito. Se ha creado un mecanismo que, una vez realizados los procesos previos -que se explican en la guía de usuario-, requiere de pocos y sencillos pasos para la importación de un XML a Moodle.

Otro de los objetivos principales se centraba en crear una herramienta cuyo funcionamiento fuera sencillo de emplear para poder así automatizar y agilizar el proceso de generación de preguntas en Moodle a través de este código creado en lenguaje Python. Este objetivo se ha cumplido, aunque no de la manera más eficiente, ya que el uso de esta requiere de más pasos de los esperados. Sin embargo, para compensar esto, es cierto que se ha creado una detallada guía de usuario con la que se debería permitir el uso de la herramienta a cualquier usuario, sea cual sea su nivel de informática.

El último objetivo consistía en la creación de un banco de preguntas lo suficientemente amplio como para permitir la generación de cuestionarios aleatorios en Moodle. Este objetivo también ha sido completado, ya que únicamente en el proceso de depuración del código Python se ha obtenido un banco de preguntas con más de 200 preguntas VF y otras 200 preguntas tipo test. Posteriormente, se han seguido importando preguntas a Moodle y actualmente cuenta con aproximadamente 760 preguntas VF y 680 preguntas tipo test, que podría verse ampliada sumando alguno de los exámenes más recientes -que aún no están integrados-.

A pesar del cumplimiento general de los objetivos, la herramienta tiene problemas generales a mejorar, puesto que no es capaz de detectar las ecuaciones, no procesa bien las preguntas en que las posibles respuestas son figuras, así como la resolución de las imágenes una vez importadas a Moodle podría ser mejor. Además, tiene ciertos errores puntuales en algunos exámenes, ya que a pesar de tener todos el mismo formato, hay ciertos exámenes donde, por ejemplo, las respuestas están incluidas en una tabla y no son detectadas, otros en que las imágenes son extraídas en un formato diferente al EMF programado y tampoco son detectadas, así como otros errores puntuales en exámenes concretos. Es decir, a pesar de poseer el mismo formato, tienen diferencias puntuales cada uno de ellos que se han tenido que programar expresamente, pero por limitaciones temporales no se ha podido programar la herramienta para ser funcional en cada uno de los exámenes. Con respecto a esto, para un funcionamiento totalmente correcto, se

deberían marcar unas pautas a la hora del diseño de los exámenes para poder depurar el código de la herramienta y que sea totalmente funcional para los exámenes hechos con estas directrices más estrictas. A pesar de lo mencionado anteriormente, la herramienta siempre es capaz de extraer las preguntas VF de cualquier examen, por lo que, aunque al ejecutar el código salte algún error, si se abre el XML creado siempre contendrá las preguntas VF.

Finalmente, se comprueba que la herramienta creada por Blanca Ràfols tiene los mismos errores generales que se han encontrado en este proyecto, pero no se mencionan errores puntuales. En cuanto a la facilidad de implementación, el proyecto de Blanca Ràfols probablemente tenga más facilidades, ya que las probabilidades de que el usuario de la herramienta tenga instalado Microsoft Excel son mucho más altas que en el caso de Python. En cuanto a la dificultad de utilización se estima que aproximadamente ambas herramientas requieren del mismo tiempo para la creación del XML.

En líneas generales, se considera que el proyecto se ha llevado a cabo con éxito, a pesar de que la limitación temporal no ha permitido crear una herramienta completamente libre de errores. Sin embargo, aún con estos errores, la herramienta creada puede ser de gran utilidad y ahorrar tiempo a los profesores que decidan emplearla.

# Bibliografía

## Referencias bibliográficas

- [1] UPCommons, *Portal de acceso abierto al conocimiento de la UPC*, Enero 2021  
[\[https://upcommons.upc.edu/handle/2117/340104?show=full\]](https://upcommons.upc.edu/handle/2117/340104?show=full)
- [2] UPCommons, *Portal de acceso abierto al conocimiento de la UPC*, Enero 2021  
[\[https://upcommons.upc.edu/handle/2117/339788\]](https://upcommons.upc.edu/handle/2117/339788)
- [3] MOODLE, *Acerca de Moodle*, Noviembre 2020  
[\[https://docs.moodle.org/all/es/Acerca\\_de\\_Moodle\]](https://docs.moodle.org/all/es/Acerca_de_Moodle)
- [4] IVAN DE SOUZA, *Rockcontent, XML: ¿qué es y para qué sirve este lenguaje de marcado?*, Julio 2019  
[\[https://rockcontent.com/es/blog/que-es-xml/\]](https://rockcontent.com/es/blog/que-es-xml/)
- [5] WIKIPEDIA, *Norma ISO/IEC 8859-1*, Enero 2022  
[\[https://es.wikipedia.org/wiki/ISO/IEC\\_8859-1\]](https://es.wikipedia.org/wiki/ISO/IEC_8859-1)
- [6] SANTANDER UNIVERSIDADES, *Python: qué es y por qué deberías aprender a utilizarlo*, Abril 2021  
[\[https://www.becas-santander.com/es/blog/python-que-es.html\]](https://www.becas-santander.com/es/blog/python-que-es.html)
- [7] STEVE CANNY, *Python docx 0.8.11 documentation*, Mayo 2021  
[\[https://python-docx.readthedocs.io/en/latest/#\]](https://python-docx.readthedocs.io/en/latest/#)
- [8] PROGRAMADOR CLIC, *Python cómo extraer imágenes en Word*, Diciembre 2021  
[\[https://programmerclick.com/article/60051411802/\]](https://programmerclick.com/article/60051411802/)
- [9] PYTHON, *Python downloads*, Diciembre 2021  
[\[https://www.python.org/downloads/\]](https://www.python.org/downloads/)
- [10] DOCS-PYTHON, *re – Operaciones con expresiones regulares*, Enero 2022  
[\[https://docs.python.org/es/3/library/re.html\]](https://docs.python.org/es/3/library/re.html)
- [11] DOCS-PYTHON, *os - Interfaces misceláneas del sistema operativo*, Enero 2022  
[\[https://docs.python.org/es/3.10/library/os.html\]](https://docs.python.org/es/3.10/library/os.html)

- [12] PILLOW, *Pillow documentation*, Enero 2022  
[\[https://pillow.readthedocs.io/en/stable/\]](https://pillow.readthedocs.io/en/stable/)
- [13] DOCS-PYTHON, *base64 - Codificaciones de datos Base16, Base32, Base64, y Base85*, Enero 2022  
[\[https://docs.python.org/es/3/library/base64.html\]](https://docs.python.org/es/3/library/base64.html)