



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

**Creación automática de cuadernos
electrónicos en *Evernote*
para la evaluación continua de los
alumnos**

Autor:

Dña. Rebeca Paniagua Herrero

Tutores:

D. Ignacio de Miguel Jiménez

Dña. Noemí Merayo Álvarez

Valladolid, 10 de julio de 2017

**TÍTULO: Creación automática de cuadernos electrónicos en
Evernote para la evaluación continua de los alumnos**

AUTOR: Dña. Rebeca Paniagua Herrero

**TUTORES: D. Ignacio de Miguel Jiménez
Dña. Noemí Merayo Álvarez**

**DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería
Telemática**

TRIBUNAL

PRESIDENTE: Dña. Patricia Fernández del Reguero

SECRETARIO: D. Ramón de la Rosa Steinz

VOCAL: D. Ramón J. Durán Barroso

PRESIDENTE SUPLENTE: Dña. Lourdes Pelaz Montes

SECRETARIO SUPLENTE: D. Javier Aguiar Pérez

VOCAL SUPLENTE: D. Juan Ignacio Asensio Pérez

FECHA: 10 de julio de 2017

CALIFICACIÓN:

Resumen del TFM

En la actualidad, la evaluación continua es vital para la educación. De esta forma, los profesores se encargan de observar y anotar información sobre las actividades realizadas por los alumnos durante las clases, y así poder evaluar el trabajo de una forma más progresiva. El objetivo de este Trabajo Fin de Máster es el desarrollo una aplicación para ordenador que se comuniquen con la cuenta de *Evernote* del profesor (una aplicación de toma de notas) de forma que, a través de esta aplicación desarrollada, sea posible crear automáticamente cuadernos electrónicos (libretas con una serie de notas en su interior) que faciliten al profesor el seguimiento del trabajo de los alumnos en los laboratorios. Estas notas serán creadas con distintos tipos de estructuras y formatos predefinidos por el profesor. Para ello se le facilitan una serie de menús desde los que crearlas y un editor para definir el contenido.

Una vez creados los cuadernos y las notas, el profesor podrá acceder desde cualquier dispositivo en el que inicie sesión en la aplicación de *Evernote*, y desde ahí podrá incluir el contenido deseado en cada nota: texto, imágenes, audio, etc.

Palabras clave

Java, JavaFX, *Evernote*, docencia, innovación, evaluación, automatización, ordenador.

Abstract

Nowadays, continuous assessment is vital for education. In this way, teachers can observe and take notes regarding the activities carried out by the students during their classes, so they can evaluate the work in a more progressive way. The purpose of this Master's Thesis is to develop a computer application that communicates with the teacher's *Evernote* account (an application for taking notes) so that, by using this developed application, it is possible to automatically create electronic notebooks (with a series of notes inside) that make easier for the teachers to follow up the work done by the students in laboratory sessions. These notes will be created with different kinds of structures and formats predefined by the teacher. For that objective, the teacher can use the menus to create them and an editor in order to define the content.

Once the notebook and notes are created, the teacher will be able to access them using any device in which he can log into the *Evernote* application, and from there he can include the content he wants in each note: text, images, sound, etc.

Keywords

Java, JavaFX, *Evernote*, teaching, innovation, assessment, automation, computer.

Agradecimientos

Este Trabajo Fin de Máster se ha enmarcado dentro de los Proyectos de Innovación Docente titulados “Evaluación de la actividad de los alumnos en las sesiones presenciales: Mejora de las aplicaciones EVALCOA y Assessment Notebook Creator” (referencia PID1516_154) y “Evaluación de la actividad de los alumnos en las sesiones presenciales: Evalcoa 3.0 y Assessment Notebook Creator para PC 2.0” (referencia PID1617_107), financiados parcialmente por la Universidad de Valladolid.

A mi familia y amigos por apoyarme siempre,

A Adrián por vivir conmigo tanto los buenos como los malos momentos durante los estudios,

A mis tutores, Nacho y Noemí, por ofrecerme la oportunidad de desarrollar este proyecto.

Tabla de contenido

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura de la memoria	2
2. Descripción general de la aplicación desarrollada	4
2.1. Introducción	4
2.2. Descripción de la aplicación	4
3. Tecnologías empleadas	10
3.1. Introducción y decisiones tomadas	10
3.2. Swing	10
3.3. Java FX	11
3.3.1. Introducción.....	11
3.3.2. Características	11
3.3.3. Estructura en JavaFX.....	12
4. Desarrollo de la aplicación	13
4.1. Recursos utilizados en el proyecto.....	13
4.2. API de Evernote	13
4.2.1. Autenticación.....	13
4.2.1.1. Tokens de desarrollador.....	13
4.2.1.2. Protocolo Oauth	14
4.3. Estructura del proyecto	15
4.4. Flujo de la aplicación	18
4.5. Desarrollo del editor de plantillas	22
4.6. Preferencias	24
5. Manual de uso de la aplicación	25
5.1. Introducción	25
5.2. Inicio de sesión.....	25
5.3. Aplicación	28
5.3.1. Menú inicial.....	28
5.3.2. Menú de creación de notas básicas.....	29
5.3.3. Menú de creación de notas elegantes	30
5.3.4. Menú de creación de notas avanzadas.....	34
5.3.5. Editor de plantillas.....	35

5.3.6. Barra de menús	37
5.3.6.1. Archivo	37
5.3.6.2. Ayuda.....	39
5.3.7. Validación y progreso de creación de notas	41
6. Conclusiones y líneas futuras.....	45
6.1. Conclusiones	45
6.2. Líneas futuras	46
7. Referencias	47

Índice de figuras

Figura 2.1. Inicio de sesión en la aplicación (con la cuenta de usuario de Evernote)	5
Figura 2.2. Menú inicial de la aplicación	6
Figura 2.3. Creación de notas básicas	7
Figura 2.4. Creación de notas elegantes	7
Figura 2.5. Creación de notas avanzadas	8
Figura 2.6. Libreta y sus correspondientes notas creadas en Evernote a partir de nuestra aplicación	9
Figura 2.7. Editor de plantillas	9
Figura 3.1. Jerarquía en JavaFX [7]	12
Figura 4.1. Ejemplo de uso del token	14
Figura 4.2. Flujo del protocolo de autenticación OAuth	15
Figura 4.3. Diagrama de flujo de creación de notas básicas	19
Figura 4.4. Diagrama de flujo de creación de notas elegantes	20
Figura 4.5. Diagrama de flujo de la gestión de la sesión de usuario	21
Figura 4.6. Ejemplo de plantilla elaborada con el editor	23
Figura 4.7. Resultado en Evernote con imagen	24
Figura 5.1. Mensaje de bienvenida a la aplicación	26
Figura 5.2. Inicio de sesión para autorizar Assessment Notebook Creator	26
Figura 5.3. Autorización de acceso a su cuenta	27
Figura 5.4. Actividades y cambios que podrá realizar la aplicación sobre su cuenta de Evernote	27
Figura 5.5. Menú inicial de la aplicación	29
Figura 5.6. Menú de creación de notas básicas	30
Figura 5.7. Resultado en Evernote tras crear notas de tipo básico	30
Figura 5.8. Menú de creación de notas elegantes	32
Figura 5.9. Explorador de ficheros para cargar una plantilla	32
Figura 5.10. Explorador de ficheros para guardar una plantilla	33
Figura 5.11. Resultado en Evernote tras crear notas de tipo elegante	33
Figura 5.12. Menú de creación de notas avanzadas	34
Figura 5.13. Contenido del fichero Equipos.txt	35
Figura 5.14. Resultado en Evernote tras crear notas de tipo avanzado	35
Figura 5.15. Editor de plantillas	36
Figura 5.16. Confirmación al volver a abrir el editor tras haber aplicado una plantilla con imágenes	37
Figura 5.17. Mensaje de advertencia: No se guardarán las imágenes	37
Figura 5.18. Menú de archivo	38
Figura 5.19. Ventana de configuración	38
Figura 5.20. Tipos de codificación posibles	39
Figura 5.21. Menú de ayuda	40
Figura 5.22. Tutorial de la aplicación	40
Figura 5.23. Acerca de la aplicación	41
Figura 5.24. Error en la creación: Información necesaria sin completar	41
Figura 5.25. Error en la creación: Falta seleccionar un fichero de contenido o rellenar un contenido en área de texto	42
Figura 5.26. Error de conexión a internet	42
Figura 5.27. Advertencia límite de notas excedido	43
Figura 5.28. Ventana de confirmación: Número de notas muy elevado	43
Figura 5.29. Mensaje sobre libreta ya existente.	43
Figura 5.30. Progreso de creación de libreta	44
Figura 5.31. Progreso de creación de notas	44
Figura 5.32. Creación de notas completada	44
Figura 5.33. Mensaje tras cancelación de la creación de notas	44

1. Introducción

1.1. Motivación

La evaluación continua de los alumnos tiene un papel muy relevante en el Espacio Europeo de Educación Superior. El profesor debe observar, tomar notas y recopilar evidencias sobre las actividades desarrolladas por los alumnos en las sesiones presenciales (por ejemplo, en un laboratorio) para poder darles realimentación sobre su trabajo y evaluarlo. Una herramienta muy útil para tal fin es la aplicación de toma de notas *Evernote* (<http://www.evernote.com>). Se trata de una aplicación multiplataforma que permite tomar notas y mantenerlas sincronizadas entre distintos dispositivos. De esta forma el profesor puede tomar notas sobre la actividad de sus alumnos en un móvil o tablet y luego acceder a esas anotaciones en su ordenador. Además, no solo permite incluir texto en sus notas, sino que también puede integrarse texto manuscrito, fotos, audio o video, de modo que es muy flexible y se adapta muy bien a distintos escenarios. *Evernote* se estructura en libretas y cada libreta tiene notas. Por tanto, un posible uso de esta aplicación para la evaluación cualitativa consiste en crear una libreta para cada asignatura y una nota (es decir, una hoja de la libreta) por cada grupo o por cada alumno que se vaya a evaluar. Después, en las sesiones de laboratorio o seminario se iría a la nota dedicada a ese grupo o alumno y se anotarían las observaciones del profesor. El inconveniente es que el proceso de crear manualmente una nota para cada grupo, con un formato concreto, y cada una con su nombre de grupo correspondiente, es muy tedioso.

El objetivo de este Trabajo Fin de Máster (TFM) es desarrollar una aplicación para PC capaz de comunicarse con la aplicación de toma de notas *Evernote* para crear automáticamente cuadernos electrónicos (con distintos tipos de estructuras y formatos predefinidos por el profesor) que faciliten al profesor el seguimiento del trabajo de los alumnos en los laboratorios. De este modo, el profesor podrá obtener de una forma cómoda una serie de ficheros estructurados y con el contenido predeterminado deseado (es decir, unas plantillas), a los cuáles podrá acceder de una forma sencilla y editarlos según su objetivo (en este caso con las evaluaciones oportunas o capturas de resultados, etc.). La creación de estas plantillas y de los cuadernos de evaluación se hará desde el ordenador ya que es algo que solo será necesario realizar una vez, por ejemplo, al comienzo de la asignatura, y el ordenador es más cómodo para este tipo de actividad; una vez creado, los profesores podrán utilizar sus tablets para la edición a través de *Evernote* de estas notas.

Este proyecto está asociado a los Proyectos de Innovación Docente de la Universidad de Valladolid (UVa) de las convocatorias 2015/2016 y 2016/2017 titulados, respectivamente, “*Evaluación de la actividad de los alumnos en las sesiones presenciales: Mejora de las aplicaciones EVALCOA y Assessment Notebook Creator*” y “*Evaluación de la actividad de los alumnos en las sesiones presenciales: Evalcoa 3.0 y Assessment Notebook Creator para PC 2.0*”.

1.2. Objetivos

Los objetivos de este Trabajo Fin de Máster se citan a continuación:

- Proporcionar una herramienta para PC que facilite a los profesores la recogida de evaluaciones, comentarios y evidencias sobre la actividad de sus alumnos, fundamentalmente durante el transcurso de sesiones prácticas, sin necesidad de invertir un tiempo muy elevado en la preparación previa para que esto sea posible. En realidad, será la aplicación de toma de notas *Evernote*® [1] quien realice el núcleo de esa función, siendo el verdadero objetivo del proyecto el desarrollo de una herramienta que permita la creación de “cuadernos electrónicos” (libretas) en *Evernote* para las asignaturas, y una hoja (nota) por cada alumno o grupo a evaluar siguiendo una plantilla predefinida por el profesor
- Hacer que la herramienta a utilizar sea cómoda, intuitiva y ágil.
- Desarrollar un editor de plantillas, de modo que el formato de las hojas de los cuadernos que utilice el profesor para la toma de notas sean creados por él mismo y completamente personalizables de una forma sencilla y visual, sin necesidad de utilizar *Evernote Markup Language* (ENML) [2], el cual es el tipo de lenguaje de marcado que utiliza *Evernote* para establecer el contenido de sus notas, ni tener conocimientos acerca del mismo.

1.3. Estructura de la memoria

Este proyecto se estructura de forma que en el capítulo 2 encontramos una descripción general de la aplicación desarrollada *Assessment Notebook Creator* para PC, en la que se explicará cuál es su utilidad y las facilidades que proporciona.

En el capítulo 3 se describirán las tecnologías de desarrollo de la interfaz gráfica que han sido empleadas en la realización de este proyecto, así como las decisiones tomadas para elegir cuál era la forma más correcta para obtener un programa más actualizado, personalizable y homogéneo. La explicación se centrará más específicamente en *JavaFX* [3], ya que ha sido la utilizada finalmente.

El capítulo 4 consta de una descripción de los recursos que han sido utilizados para desarrollar el proyecto (programa y librerías) y una breve introducción a los métodos utilizados para la autenticación en la aplicación. También se proporciona una visión de la estructura del proyecto en *Eclipse* [4], plataforma de software empleada. A mayores se muestran una serie de diagramas de flujo que determinan el comportamiento de diferentes apartados de la aplicación. Por último, se comenta cómo ha sido desarrollado el editor de plantillas del programa, y cómo funcionan las preferencias.

A continuación, el capítulo 5 contiene un manual en el cuál se explica detalladamente el funcionamiento de la aplicación, con todas las posibilidades y opciones que proporciona, acompañado de capturas de pantalla del programa donde podemos

contemplar de forma visual lo que está siendo comentado en cada momento, y ejemplos concretos sobre cómo serían los resultados obtenidos.

Posteriormente, en el capítulo 6 se extraen una serie de conclusiones derivadas del desarrollo de este trabajo. Además, se proporcionan una serie de posibles líneas futuras con las que se podría continuar trabajando.

Finalmente encontramos una serie de referencias que han sido utilizada para poder llevar a cabo el proyecto.

2. Descripción general de la aplicación desarrollada

2.1. Introducción

La aplicación desarrollada (*Assessment Notebook Creator* para PC), pretende ser una herramienta que facilite la actividad docente en el sentido de permitir a los profesores generar de forma automática “cuadernos electrónicos” en *Evernote* formados por un conjunto de hojas (tantas como alumnos o grupos vaya a evaluar el profesor). Utilizando la terminología de *Evernote*, se crearán libretas que contengan una serie de notas con un contenido predeterminado, definido de forma sencilla por el propio profesor. Estas libretas y notas son creadas en la cuenta de *Evernote* del usuario. El objetivo de ellas es que ayuden al seguimiento de las clases y a la toma de notas y evidencias de una forma rápida y cómoda. La aplicación está destinada especialmente a las clases de laboratorio, en las cuales el profesor puede tener una plantilla con contenido predeterminado acerca de las diferentes prácticas que se irán realizando, y durante las sesiones prácticas puede ir complementando los huecos destinado a ello. Esto podría hacerse mediante imágenes sobre lo que el alumno ha desarrollado, o simplemente notas sobre cómo lo han hecho. Esta aplicación, como se ha mencionado anteriormente, es para ordenador y ha sido desarrollada utilizando *Java* como lenguaje de programación.

2.2. Descripción de la aplicación

Assessment Notebook Creator para PC, es una aplicación de ordenador, que facilita la actividad docente a la hora de tomar notas sobre las actividades de sus alumnos durante el transcurso de las clases. En las clases de laboratorio, habitualmente los alumnos tienen que realizar una práctica propuesta y proporcionar los resultados obtenidos una vez finalizada. Es de mucha utilidad que el profesor pueda ir recorriendo la clase a través de los distintos alumnos o grupos de alumnos, y tomar de una forma sencilla notas sobre cómo se está desarrollando su actividad, incluso tomar evidencias como puede ser una foto, y tener todo organizado de una forma correcta y cómoda para su posterior revisión.

La toma de notas e inclusión de fotos o incluso otros elementos como puede ser audio, es posible mediante el uso de la aplicación denominada *Evernote*, aplicación multiplataforma que permite acceder tanto desde móviles, como tablets u ordenadores, manteniendo la información sincronizada en todos los dispositivos que tengan iniciada la sesión con la misma cuenta.

La aplicación desarrollada como fin de este proyecto, se conecta a la cuenta de *Evernote* del usuario, y lo que permite es crear de forma automatizada libretas (grupo de notas) con el nombre deseado, y dentro de ellas tantas notas como se deseen (de nuevo, con el nombre que el usuario necesite) y a su vez estas con el contenido predeterminado solicitado. A continuación, se comentará con más calma, pero en líneas generales, un

profesor podría, por ejemplo, crear de forma automatizada una libreta con el nombre de una de sus asignaturas, y que dentro de esta apareciese el nombre de cada grupo de alumnos que participasen en la asignatura, obteniendo una nota para cada uno de ellos. A su vez el contenido de estas notas creadas podría ser por ejemplo una plantilla que contenga diferentes apartados para cada una de las prácticas que se realizarán a lo largo de la asignatura, espacios para anotaciones acerca de ellas, etc. Además, estas notas se generarían con el formato deseado (posteriormente veremos cómo se podrá hacer esto). Como podemos ver, si el proceso de creación requiriese crear manualmente una a una cada una de las libretas, y a su vez una a una cada una de las notas que contuvieran y su contenido correspondiente, esto supondría una tarea larga y pesada. Por ello, nuestra herramienta podría facilitar mucho el trabajo a los docentes, además de ahorrarles mucho tiempo.

Como se ha mencionado anteriormente, nuestra aplicación se conecta con la cuenta de *Evernote* de la persona que la desee utilizar, de esta forma conseguimos sincronizar su cuenta y actualizarla con los nuevos contenidos que desee generar desde la aplicación. En la Figura 2.1 podemos ver cómo es el inicio de sesión en la aplicación, donde es necesario introducir tanto el usuario como la contraseña utilizados en la cuenta de *Evernote*.



Figura 2.1. Inicio de sesión en la aplicación (con la cuenta de usuario de Evernote)

Una vez iniciada sesión tenemos tres formas diferentes de crear nuestras libretas y notas: Básicas, elegantes y avanzadas. La Figura 2.2 muestra el menú inicial con el que nos encontramos.

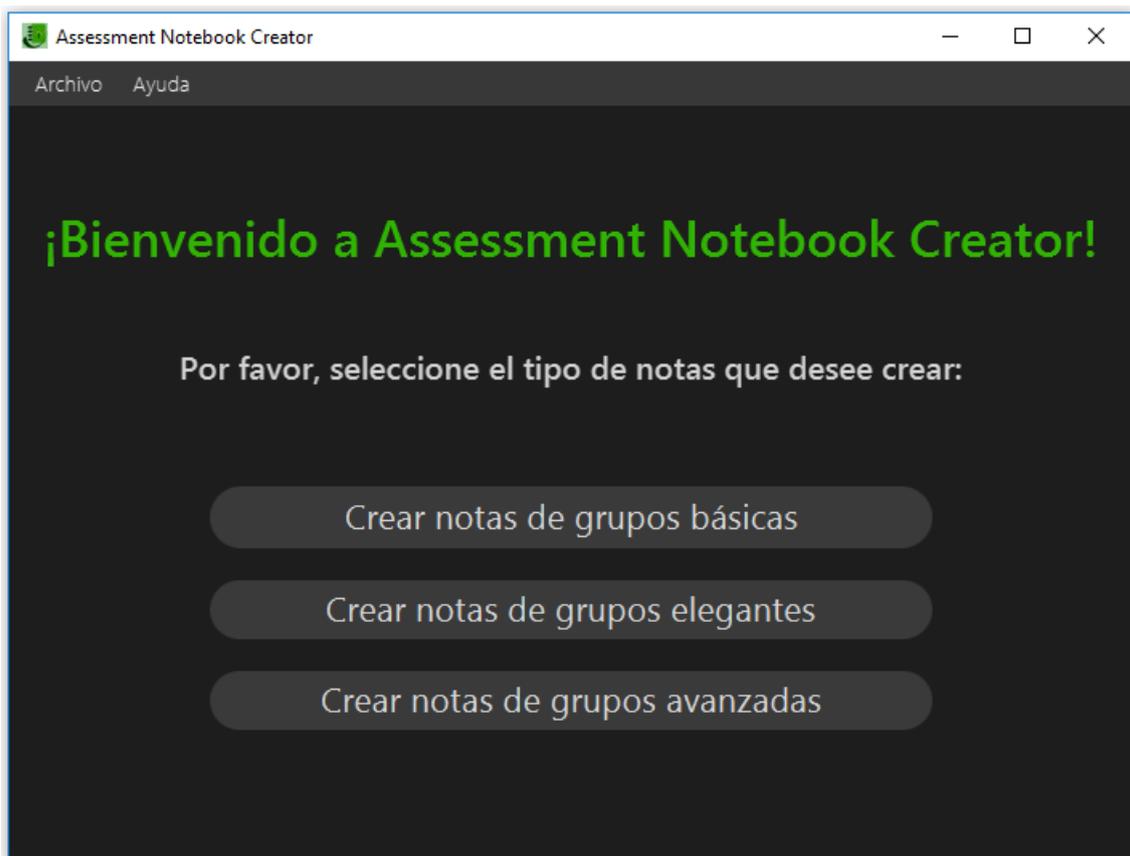


Figura 2.2. Menú inicial de la aplicación

En el tipo de **notas básicas**, tenemos la forma más sencilla de crear nuestra libreta con sus notas correspondientes. En esta únicamente es necesario introducir el nombre de la libreta (por ejemplo, el nombre de la asignatura), el nombre básico de las notas (cada una de las notas se irá generando con ese nombre raíz seguido de un número que comenzará por el 1 e irá avanzando hasta el número total solicitado) y el número de notas a crear. En la Figura 2.3 podemos ver un ejemplo de la interfaz, con cuyos datos obtendríamos una libreta llamada “Comunicaciones Ópticas” que contenga 12 notas nombradas “Grupo01”, “Grupo02”, ... , y “Grupo12”, y que estarán completamente vacías de contenido.

Para el caso de las **notas elegantes**, tendríamos una interfaz en la cual debemos introducir el nombre de la libreta, el nombre básico de las notas y el número de grupos, como en el caso anterior. Pero en este caso, a mayores seleccionaremos una plantilla de contenido, la cual definirá el contenido inicial de cada una de las notas generadas. Una vez seleccionada nos aparecerá el nombre de la plantilla elegida y su contenido en código ENML (el utilizado por *Evernote* para establecer el contenido de las notas). Esta plantilla podrá ser elegida de entre los ficheros residentes en nuestro ordenador, pulsando la opción de cargar fichero, o podrá ser generada mediante un editor, el cual veremos posteriormente. La Figura 2.4 muestra la apariencia de esta opción.

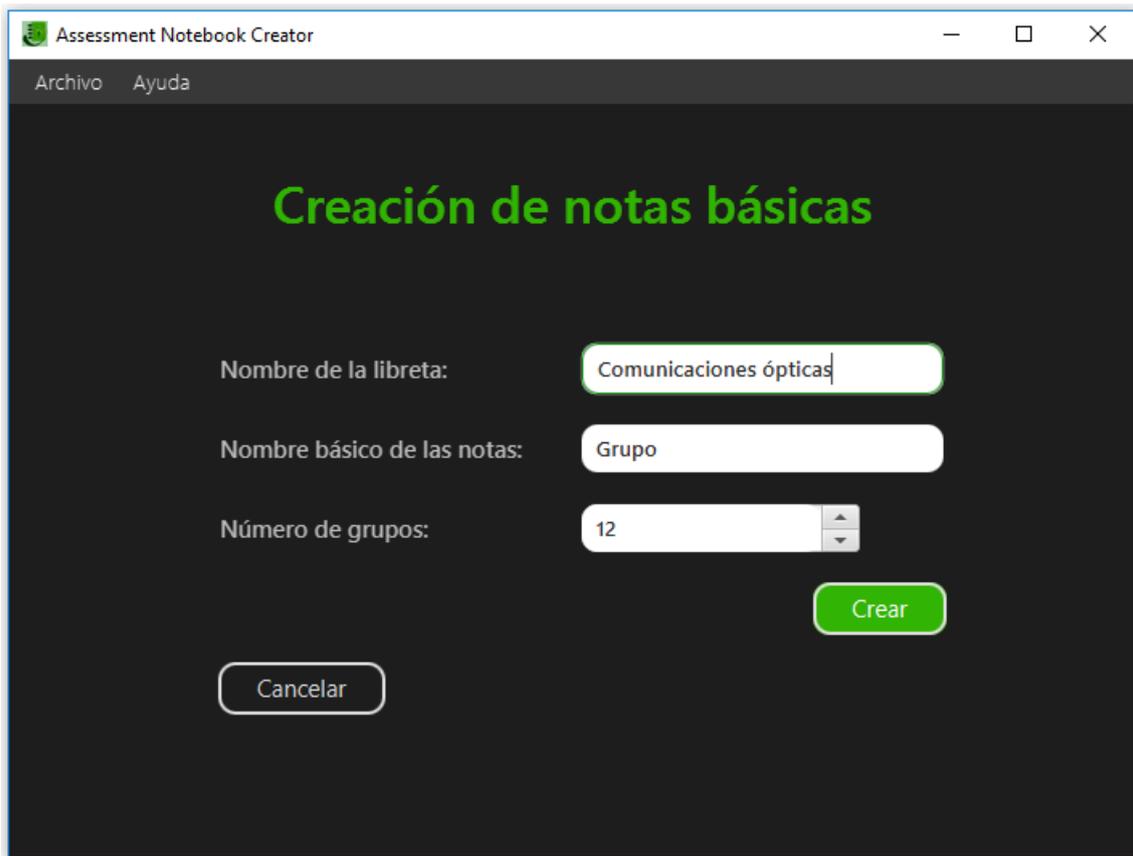


Figura 2.3. Creación de notas básicas

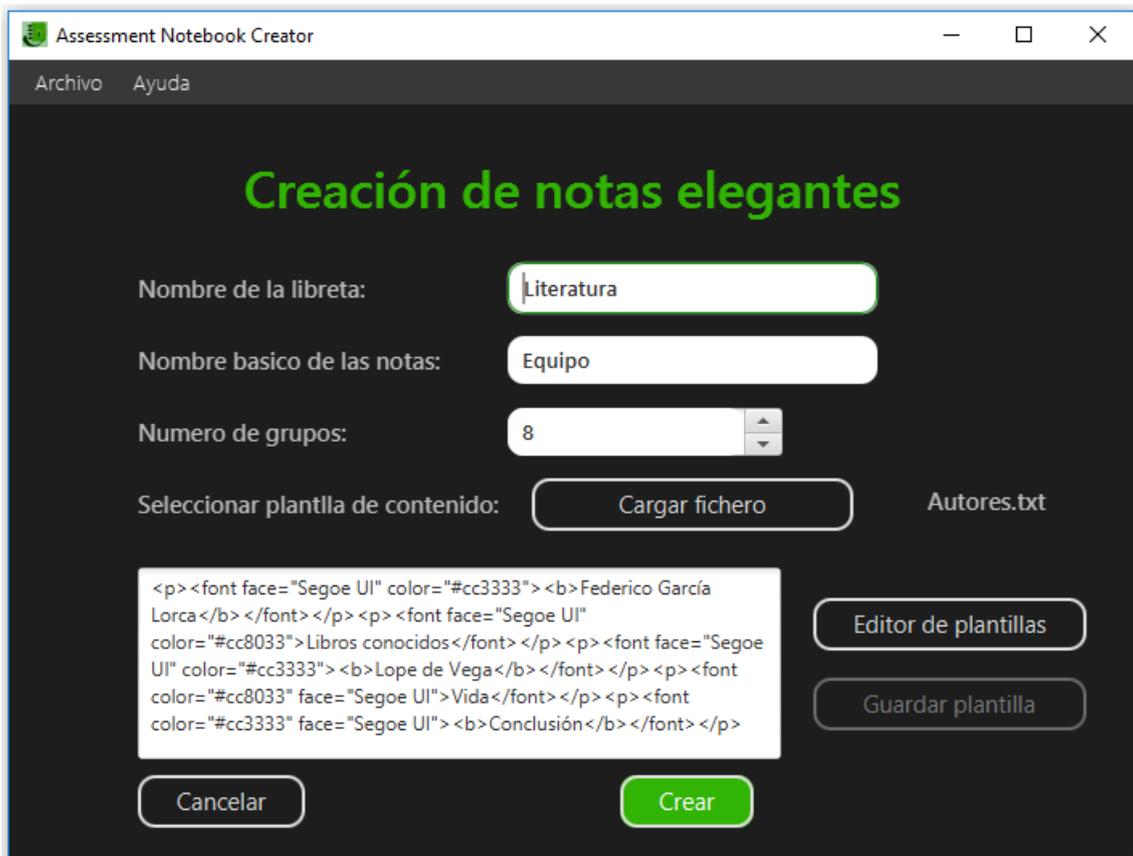


Figura 2.4. Creación de notas elegantes

El último tipo de notas que podemos crear son las **notas avanzadas**, cuyo menú podemos ver en la Figura 2.5. En este último caso, el usuario debe introducir el nombre de la libreta y a continuación seleccionar un fichero que contenga información sobre los grupos o alumnos cuyo trabajo se desea evaluar, de forma que las notas que se crearán dentro de la libreta se llamarán con los nombres contenidos en ese fichero, los cuales va tomando línea a línea. Este contenido se carga en el área de texto destinada a ello, desde la cuál es posible editarlo. También es posible escribir este contenido directamente en ella, sin necesidad de utilizar ficheros. De este modo se crearán tantas notas como líneas contenga el área de texto, y el título de cada una de ellas será el contenido de cada una de las líneas. Posteriormente, podremos elegir si queremos un estilo básico (notas vacías) o elegante (notas con contenido inicial según una plantilla). En el caso de seleccionar el estilo básico, ya tendríamos toda la información necesaria y podríamos crear las notas; sin embargo, en el caso de seleccionar el estilo elegante, se deberá elegir la plantilla de contenido que represente lo que queremos que aparezca en cada una de las notas creadas.

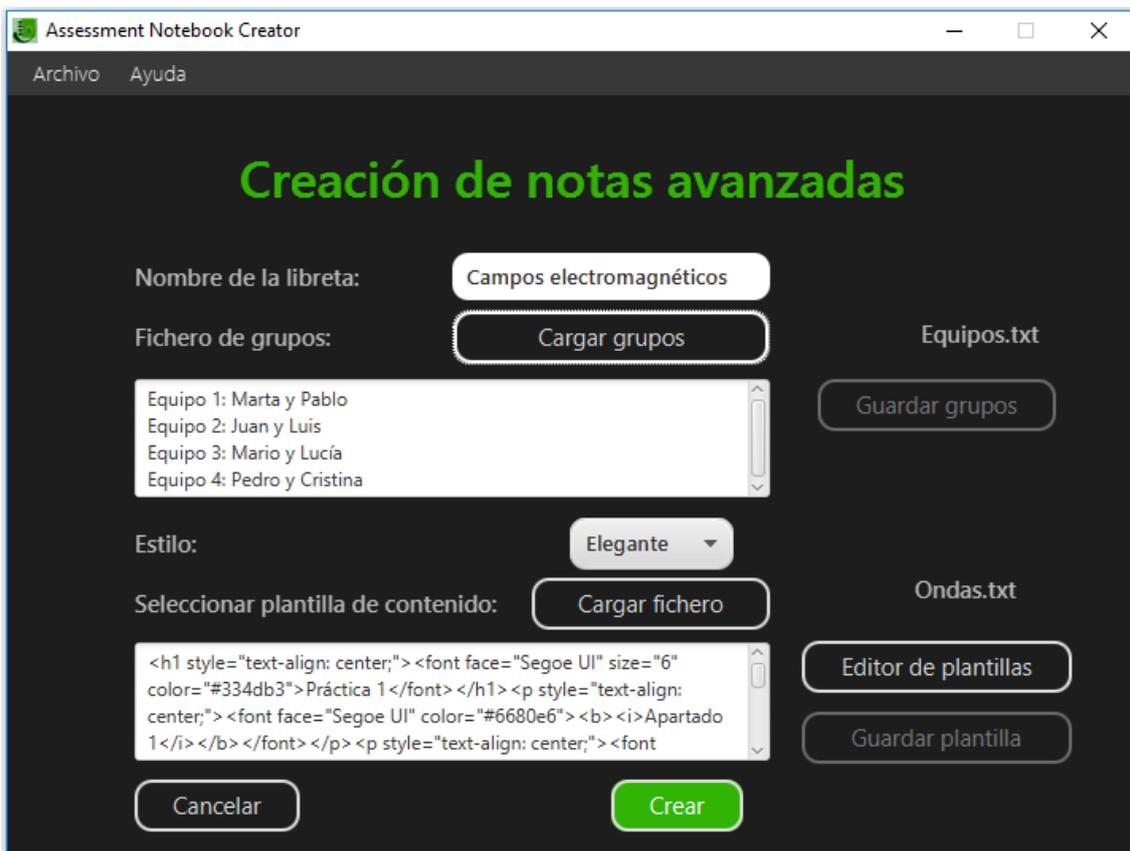


Figura 2.5. Creación de notas avanzadas

El resultado tras crear las notas de cualquiera de los tipos se verá reflejado en nuestra cuenta de *Evernote*, a la que podremos acceder desde cualquiera de nuestros dispositivos en los cuales tengamos *Evernote* instalado. En la Figura 2.6 podemos ver como ejemplo la libreta con las notas y su contenido creadas mediante los datos de la Figura 2.5.

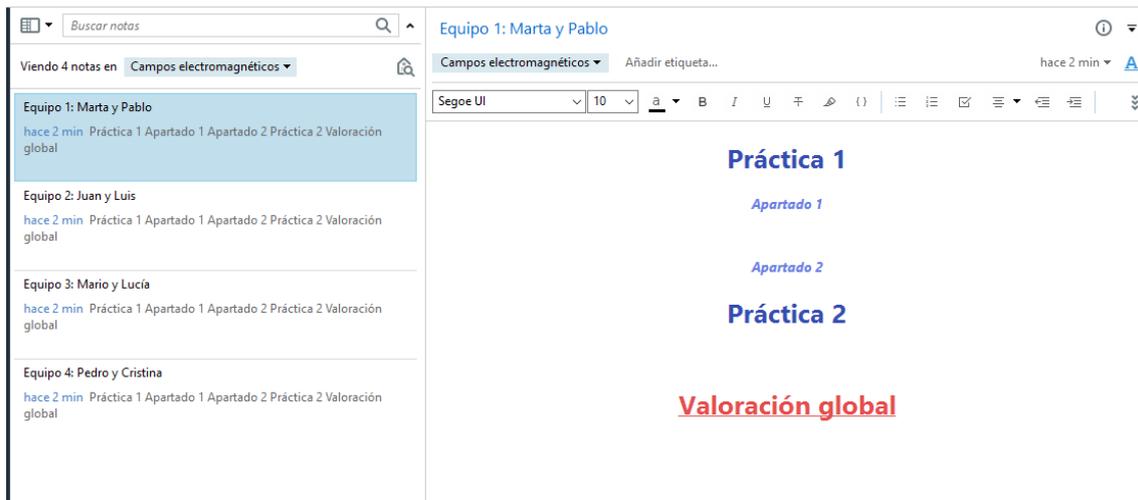


Figura 2.6. Libreta y sus correspondientes notas creadas en Evernote a partir de nuestra aplicación

Además, en el caso de notas elegantes o notas avanzadas con estilo elegante, se ha desarrollado un editor, mostrado en la Figura 2.7, que permite obtener las plantillas a utilizar de una forma muy sencilla e intuitiva, sin necesidad de tener conocimientos sobre el lenguaje ENML. Además, se ha incorporado una funcionalidad para añadir imágenes, aunque con ciertas restricciones como se mencionará en el manual de usuario.

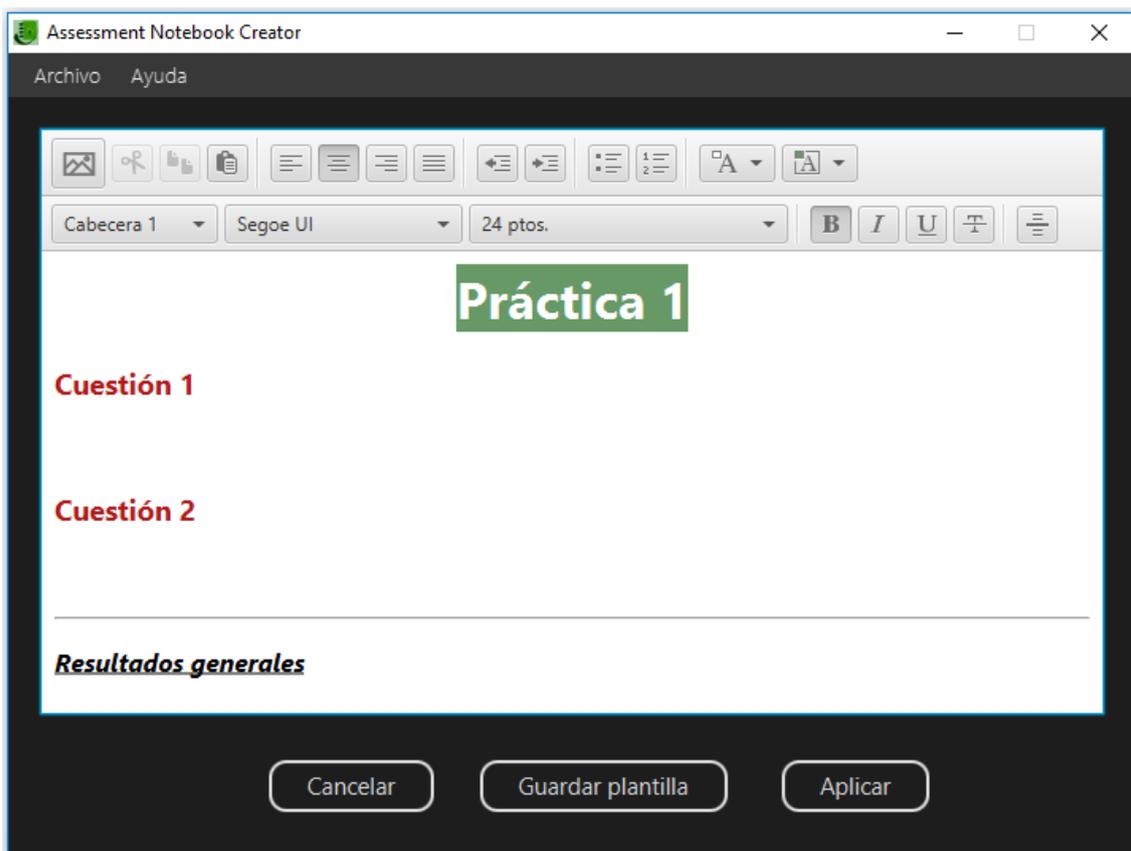


Figura 2.7. Editor de plantillas

Al final de este documento (Apartado 5), se encuentra el manual en el que se detallan todas y cada una de las opciones de la aplicación y cómo funcionan, y se muestra de una forma más clara y concisa los resultados obtenidos.

3. Tecnologías empleadas

3.1. Introducción y decisiones tomadas

Durante el desarrollo de este proyecto, se ha utilizado el lenguaje de programación Java, al ser uno de los cuales permitía utilizar *Evernote* para trabajar con su API correspondiente. Como entorno de desarrollo se ha empleado la plataforma de software *Eclipse*. A mayores, para diseñar el entorno gráfico de la aplicación, se comenzó empleando Swing, aunque posteriormente fue reemplazado por JavaFX, ya que esta última proporcionaba muchas más posibilidades y permitía implementar funcionalidades que no eran posibles mediante Swing. En el proceso de ir incorporando funcionalidades que no se podían realizar mediante Swing, se utilizó una combinación de ambas; de este modo lo que estaba implementado, se mantenía con Swing, y las funcionalidades que no eran posibles con este, se desarrollaban mediante JavaFX. La decisión final fue reemplazar todo utilizando únicamente JavaFX para unificar el proyecto. De esta forma, además, se permitía que la interfaz pudiese ser más personalizable, y facilitaría las posibilidades de posteriormente continuar trabajando con esta aplicación mejorándola o añadiendo nuevas funcionalidades. Por ello, a continuación, se presentará Swing de una forma breve, mientras que se profundizará más en los aspectos de JavaFX.

3.2. Swing

Swing [5] es un conjunto de componentes utilizado para crear interfaces gráficas en Java, para añadir funcionalidades e interactividad a las aplicaciones. Entre estos componentes podemos encontrar botones, etiquetas, etc. Estos componentes son utilizados con las JFC (*Java Foundation Classes*). En concreto las JFC cuentan con las siguientes características:

- Componentes Swing GUI: Incluye diferentes componentes como botones, paneles, tablas, etc.
- Cambio de apariencia (*look-and-feel*): Es posible cambiar la apariencia de los objetos durante la ejecución.
- API de accesibilidad: Permite tecnologías de asistencia para por ejemplo pantallas de Braille.
- API de Java 2D: Permite incorporar de forma sencilla gráficos 2D de alta calidad.
- Internacionalización: Proporciona la posibilidad construir aplicaciones que interactúen con usuarios de todo el mundo en su propio idioma.

3.3. Java FX

3.3.1. Introducción

JavaFX [6] es una tecnología que permite a los desarrolladores crear una interfaz visual avanzada para sus aplicaciones Java, gracias a su conjunto de paquetes. Está disponible en Windows, Mac OS X y Linux. Para poder ejecutar estas aplicaciones es necesario tener el *Java Runtime Environment* (JRE), que incorpora las funcionalidades de JavaFX, instalado en el ordenador.

Con JavaFX es posible personalizar de una forma mucho más amplia la apariencia de las aplicaciones, ya que es posible utilizar CSS (*Cascading Style Sheets*), proporcionando un amplio abanico de posibilidades a la hora de definir la apariencia de la aplicación. Además, permite utilizar FXML, un lenguaje de marcado basado en XML que proporciona la estructura necesaria para construir una interfaz de usuario. Esto da pie al uso de un modelo vista-controlador (MVC), de forma que en el propio FXML definimos únicamente la interfaz de usuario, y posteriormente le asignamos una clase Java como controlador, en la cual definimos la lógica de la aplicación asociada a esa interfaz. Utilizando este método, la aplicación es más fácil de mantener ya que tenemos diferenciadas y separadas las diferentes partes: diseño y lógica. De este modo, se codificaría de forma separada la lógica de la aplicación (clases Java), la interfaz de la aplicación (FXML) y la apariencia de esta (CSS). Además, es posible generar estos ficheros FXML de una forma sencilla mediante una herramienta llamada *Scene Builder*, que permite hacerlo de una forma visual mientras que el código se genera de forma automática.

3.3.2. Características

A continuación, se muestran una serie de características que proporciona JavaFX 8 y sus versiones posteriores:

- **APIs Java:** JavaFX es una librería de Java que consiste en clases e interfaces escritas en código Java.
- **FXML y *Scene Builder*:** Como se ha mencionado anteriormente, FXML es un lenguaje de marcado basado en XML para construir la interfaz de la aplicación. El código se puede generar manualmente o bien mediante la ayuda de la herramienta *Scene Builder*, que permite obtener la interfaz de usuario de una forma interactiva.
- ***WebView*:** Con JavaFX es posible embeber páginas web en las aplicaciones, lo cual es muy útil en el caso de la aplicación que se desarrolla en este proyecto en concreto para el inicio de sesión.
- **Interoperabilidad con Swing:** Aplicaciones ya existentes desarrolladas en Swing pueden ser actualizadas a JavaFX. El contenido Swing puede ser embebido en aplicaciones JavaFX, de forma que en la aplicación coexistan ambas tecnologías.
- **Controles de interfaz y CSS:** Cuenta con controladores de interfaz de usuario que pueden ser personalizados aplicándole un estilo mediante CSS.

- Tema *Modena*: Este es el tema que utiliza por defecto. Reemplaza al tema *Caspian* (utilizado antiguamente), aunque se puede cambiar.
- Gráficos 3D: Cuenta con clases para utilizar figuras 3D.
- API Canvas: Permite dibujar directamente en un área de la escena.
- Soporte multitouch.
- Soporta pantallas Hi-DPI (pantallas con una alta densidad de píxeles).
- Motor de datos de alto rendimiento.
- Modelo de despliegue de aplicaciones autocontenidas.

3.3.3. Estructura en JavaFX

En este apartado se va a comentar la jerarquía existente en las aplicaciones JavaFX. La Figura 3.1 muestra gráficamente cómo es la estructura básica.

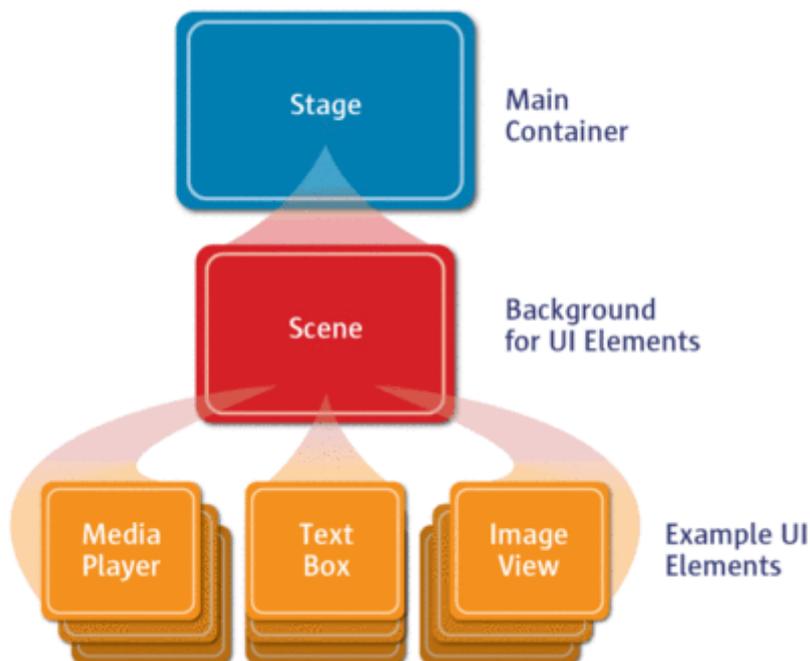


Figura 3.1. Jerarquía en JavaFX [7]

Como podemos observar, en la parte superior se encuentra el *Stage* (escenario). Este actúa como el contenedor principal de la aplicación y, en general, se basa en una ventana con su borde y sus correspondientes botones de minimizar, maximizar y cerrar. Dentro de ella, se puede introducir un *Scene* (escena), que actúa como fondo de los elementos que se utilicen en la interfaz de usuario. De este modo, la escena contiene a su vez estos elementos de la interfaz (nodos JavaFX) que pueden ser cuadros de texto, botones, etiquetas, tablas, y cualquier otro de los que ofrece JavaFX. En base a un *Stage*, podemos utilizar diferentes escenas que cambien dinámicamente en función de qué menú o sección de nuestra aplicación queremos que se muestre en ese momento. Además, se pueden utilizar varios *Stages* si queremos abrir ventanas adicionales.

4. Desarrollo de la aplicación

4.1. Recursos utilizados en el proyecto

La aplicación ha sido desarrollada utilizando el lenguaje de programación Java dentro del entorno de desarrollo *Eclipse*. De este modo, se ha utilizado el JDK (*Java Development Kit*) de Java [8]. Además, ha sido necesario utilizar la API (*Application Programming Interface*) de *Evernote* para Java [9], que contiene las librerías necesarias para trabajar con funciones relacionadas con *Evernote*. Por último, para la autenticación, *Evernote* utiliza el protocolo *OAuth*, por lo que ha sido necesario trabajar con la librería *scribe-1.3.7.jar* [10].

4.2. API de *Evernote*

Para trabajar con la API de *Evernote* es importante comprender algunos conceptos básicos. La API está compuesta por dos servicios lógicos [11]: el *UserStore* y el *NoteStore*. El *UserStore* gestiona las cuentas de usuario de *Evernote*, mientras que el *NoteStore* gestiona el contenido de estas cuentas de usuario. Una vez que el usuario se identifica, se debe obtener acceso a su cuenta de *Evernote*. Para utilizar algún servicio de la API, es necesario crear un objeto que contenga el *NoteStore* y otro que contenga en *UserStore*, e invocar a los métodos desde ellos. El que realmente se utilizará será el *NoteStore*, ya que lo que se hace es modificar el contenido de la cuenta del usuario, y no su cuenta en sí. Algunas funciones posibles serían crear una libreta, listar las notas que contenga, etc.

4.2.1. Autenticación

Existen dos modos para comunicarse con la API de *Evernote*: los tokens de desarrollador u *OAuth* (cuyo funcionamiento se explica en el apartado 4.2.1.2).

4.2.1.1. Tokens de desarrollador

A la hora de desarrollar una aplicación que conecte con la API de *Evernote*, es necesario obtener dos tokens de desarrollador [12]: uno para *sandbox* y otro para producción. Para el comienzo del desarrollo de la aplicación se utiliza el token de *sandbox*, para así poder probar los avances en la aplicación utilizando este servicio en lugar de generar los cambios en nuestra propia cuenta de *Evernote*. Para obtener el token de *sandbox* se puede acceder a través de esta URL: <https://sandbox.evernote.com/api/DeveloperToken.action>. Para el caso de producción la URL a utilizar es: <https://www.evernote.com/api/DeveloperToken.action>. Estos tokens permiten utilizar la API de *Evernote* para acceder a la cuenta de *Evernote* del desarrollador. En el caso de que alguien tenga acceso a nuestros tokens, estos se pueden revocar y generar otros nuevos. Estos tokens únicamente se utilizan durante el desarrollo de la aplicación, para que el desarrollador no necesite loguearse en su cuenta de forma continua, pero a la hora de publicar la aplicación, el modo de acceso es diferente, como se verá posteriormente. La Figura 4.1 muestra un ejemplo de uso del token y que a

continuación se muestre una lista con los nombres de las libretas existentes en la cuenta del desarrollador. En ese caso se está utilizando el entorno de *sandbox*, mientras que para utilizar el de producción habría que sustituir *EvernoteService.SANDBOX* por *EvernoteService.PRODUCTION*.

```
1 String developerToken = "my developer token";
2
3 // Set up the NoteStore client
4 EvernoteAuth evernoteAuth = new EvernoteAuth(EvernoteService.SANDBOX, developerToken);
5 ClientFactory factory = new ClientFactory(evernoteAuth);
6 NoteStoreClient noteStore = factory.createNoteStoreClient();
7
8 // Make API calls, passing the developer token as the authenticationToken param
9 List<Notebook> notebooks = noteStore.listNotebooks();
10
11 for (Notebook notebook : notebooks) {
12     System.out.println("Notebook: " + notebook.getName());
13 }
```

Figura 4.1. Ejemplo de uso del token

4.2.1.2. Protocolo *Oauth*

Para que un usuario pueda utilizar la aplicación desarrollada, es necesario utilizar el protocolo *Oauth* [13], de forma que sea posible acceder a su cuenta de usuario. Una vez que este proceso de autenticación se completa, el usuario accede a la aplicación y a sus funcionalidades.

Antes de utilizar este protocolo, es necesario solicitar a *Evernote* una API key que consta de un *consumer key* y un *consumer secret* (que actúan como si fueran un usuario y contraseña) que distinga la aplicación del resto. Al solicitar esta clave, se debe indicar si se necesita acceso básico o completo [14]. En este caso, se ha utilizado el acceso completo, ya que era necesario para modificar libretas existentes, lo cual es requerido en la aplicación.

Respecto al protocolo *OAuth*, podemos ver su flujo de funcionamiento en la Figura 4.2. En ella se observa que la aplicación desarrollada solicita un *request token* del servidor de autorización de *Evernote*, y este servidor se lo proporciona. A partir de este token podemos extraer la URL que se debe cargar en nuestra aplicación desarrollada (debemos redirigir al usuario a esta), para que este usuario se loguee mediante su usuario y contraseña. Una vez que el inicio de sesión es correcto se devuelve la autorización por un periodo de tiempo a nuestra aplicación. Nuestra aplicación seguidamente envía el token que recibió además de la autorización, de forma que, si todo es correcto, el servidor le devuelve un token de acceso. Utilizando este token de acceso nuestra aplicación puede solicitar los recursos del usuario o enviar recursos al servidor de *Evernote*, de forma que el usuario puede utilizar la aplicación y sus funcionalidades. Esto será posible mientras que el token de acceso siga siendo válido (siempre que el usuario no haya revocado el acceso o se haya alcanzado la fecha de expiración del token).

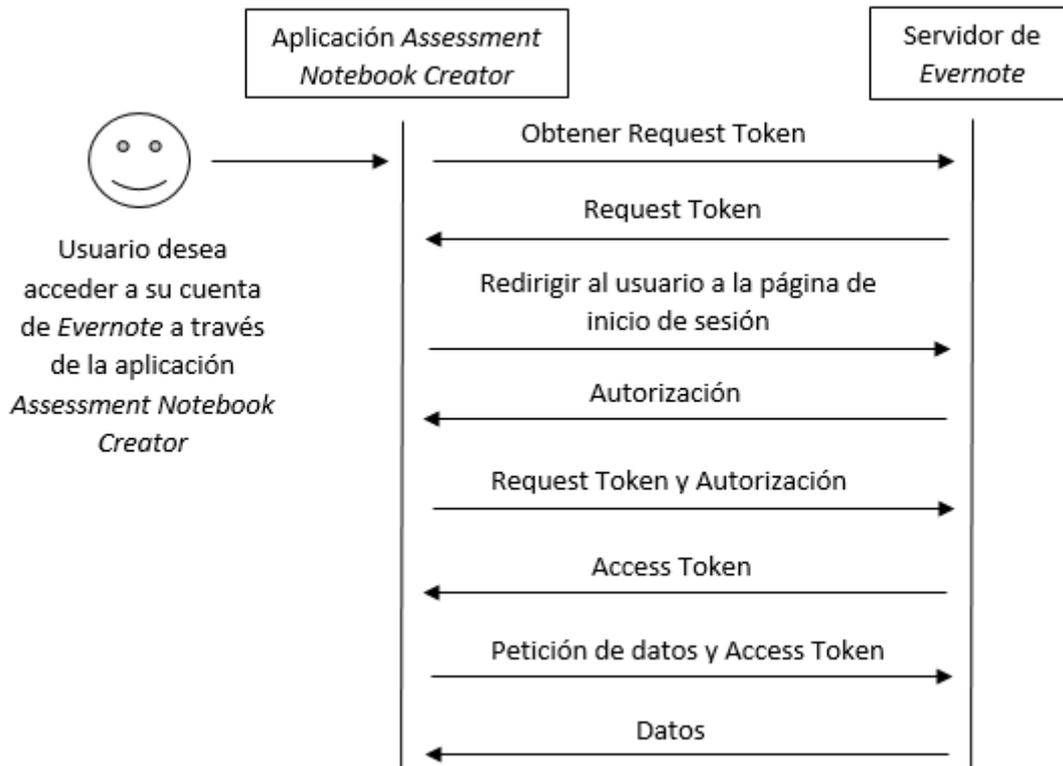


Figura 4.2. Flujo del protocolo de autenticación OAuth

4.3. Estructura del proyecto

El proyecto ha sido desarrollado con Eclipse, en el cuál se ha creado una estructura de directorios con sus correspondientes ficheros que se irá comentando a lo largo de este apartado. Además, en cada fichero de código se encuentran los comentarios correspondientes tanto a cada clase como a cada método de estas clases. A parte de ello se ha comentado de una forma pormenorizada todo el funcionamiento de cada fragmento de código para que quien acceda a este pueda ver de una forma sencilla qué se está haciendo en cada momento. Dentro del proyecto encontramos lo siguiente:

Directorio src:

Dentro de este directorio tenemos todos los paquetes y ficheros que engloban el código desarrollado para la aplicación. Concretamente dentro de este directorio se encuentra el paquete *root*, que a su vez contiene a los paquetes *root.common* y *root.views*.

- **Paquete *root*:** Este contiene el fichero principal de la aplicación: *MainApplication.java*.
 - ***MainApplication.java*:** Esta es la clase principal de la aplicación, la cual extiende de la clase *Application* de *JavaFX*, de forma que lo primero de lo que se encarga es lanzar la aplicación. En ella se encuentra un método que la inicializa (el cual sobrescribe el método *start* de la clase de la que extiende). Allí se establecen valores como el título de la aplicación, su icono, la escena, etc. Además, aquí se realiza la comprobación de si el usuario tiene una sesión

iniciada o no, de forma que, si no la tiene, invoca al método que se encarga del inicio de sesión (el cuál se encuentra en esta misma clase); mientras que, si ya la ha iniciado, llama al método encargado de mostrar el menú inicial, el cual será la pantalla principal de nuestra aplicación. A mayores, esta clase cuenta con los métodos correspondientes a la carga de las diferentes escenas de la aplicación, no solo la de inicio, y también para la ventana de progreso que aparece mientras se están creando notas. Aquí también se define un evento para que al cerrar la aplicación nos muestre un mensaje de confirmación y el método encargado del cierre de sesión.

- **Paquete *root.common***: En este paquete encontramos 3 clases, que serán comentadas un poco en profundidad:

- ***CreateNotebookTask.java***: Esta clase cuenta con un método que es ejecutado en un segundo hilo, para la creación de libretas y notas, permitiendo mostrar al mismo tiempo la ventana de progreso. Para ello la clase extiende de la clase *Task* de *JavaFX* y se sobrescribe el método *call*. Nuestra clase cuenta con 3 constructores diferentes, utilizando sobrecarga de métodos, de modo que en función de los parámetros que se le pase utilizará el correspondiente a notas básicas, elegantes o avanzadas. Posteriormente, en el método *call* se gestiona la creación de la libreta, la obtención de los nombres de las notas en el caso de que no sean de tipo avanzadas y la creación de notas. Además, desde ahí se obtiene el número de notas creadas para pasarle ese parámetro a la ventana de progreso.
- ***Logic.java***: Esta clase cuenta con una serie de métodos que se utilizan en la aplicación, los cuales no tienen relación con la interfaz gráfica. Estos son llamados desde las otras clases para realizar diferentes tareas. Como ejemplos, cuenta con métodos para obtener la url de inicio de sesión, para obtener el token del usuario, crear una libreta, crear una nota, crear y borrar recursos (para las imágenes del editor), generar el título de una nota, separar parámetros del token, comprobar si el usuario está autenticado, obtener el usuario que tiene iniciada su sesión en ese momento, guardar una plantilla de texto, cargar una plantilla de texto, comprobar si existe alguna libreta con un nombre dado, etc.

- **Paquete *root.views***: Este paquete contiene todos los ficheros relacionados con la interfaz gráfica (de extensión *.fxml*) y sus correspondientes controladores. Además, contiene el fichero *.css* utilizado para la estética de la aplicación.

Los ficheros *.fxml* contienen la estructura y colocación de los elementos contenidos en la escena correspondiente, mientras que los controladores cuentan con una serie de funciones que gestionan los eventos que puedan llevarse a cabo en estas escenas, como pueden ser las funciones a realizar cuando se pulsa cierto botón

- ***About.fxml***: Define la apariencia gráfica de la ventana correspondiente a “Acerca de..”.
- ***AdvancedMenu.fxml***: Define la apariencia gráfica del menú de creación de notas avanzadas.
- ***BasicMenu.fxml***: Define la apariencia gráfica del menú de creación de notas básicas.
- ***ConfigurationMenu.fxml***: Define la apariencia gráfica de la ventana correspondiente a “Configuración”.

- *Editor.fxml*: Define la apariencia correspondiente al editor de plantillas que se puede utilizar desde notas elegantes o avanzadas.
- *ElegantMenu.fxml*: Define la apariencia gráfica del menú de creación de notas básicas.
- *MainMenu.fxml*: Define la apariencia gráfica del menú inicial.
- *Navigator.fxml*: Define el navegador utilizado en el momento de inicio de sesión, ya que este debe ser realizado a través de la web.
- *ProgressWindow.fxml*: Define la apariencia gráfica de la ventana de progreso.
- *RootLayout.fxml*: Define la apariencia gráfica del marco principal de la ventana de la aplicación, incluyendo la barra de menús superior.
- *Tutorial.fxml*: Determina la apariencia del tutorial mostrado en las opciones de ayuda.
- *Welcome.fxml*: Define la ventana de bienvenida mostrada al usuario la primera vez que ejecuta la aplicación.
- *AboutController.java*: Controlador para la ventana correspondiente a “Acerca de..”, definida en el fichero *About.fxml*.
- *AdvancedMenuController.java*: Controlador para el menú de creación de notas avanzadas, definido en el fichero *AdvancedMenu.fxml*.
- *BasicMenuController.java*: Controlador para el menú de creación de notas básicas, definido en el fichero *BasicMenu.fxml*.
- *ConfigurationMenuController.java*: Controlador para la ventana correspondiente a la ventana de configuración, definida en el fichero *ConfigurationMenu.fxml*.
- *EditorController.java*: Controlador del editor de plantillas definido en el fichero *Editor.fxml*.
- *ElegantMenuController.java*: Controlador para el menú de creación de notas elegantes, definido en el fichero *ElegantMenu.fxml*.
- *MainMenuController.java*: Controlador para el menú inicial, definido en el fichero *MainMenu.fxml*.
- *NavigatorController.java*: Controlador para el navegador utilizado durante el inicio de sesión, definido en el fichero *Navigator.fxml*.
- *ProgressWindowController.java*: Controlador para la ventana de progreso, definida en el fichero *ProgressWindow.fxml*.
- *RootLayoutController.java*: Controlador para el marco principal de la ventana de la aplicación, definido en el fichero *RootLayout.fxml*.
- *TutorialController.java*: Controlador correspondiente al tutorial que se muestra desde las opciones de la barra de menús, definido en el fichero *Tutorial.fxml*.
- *WelcomeController.java*: Controlador para la pantalla de bienvenida mostrada al usuario la primera vez que utiliza la aplicación.
- *Style.css*: Fichero .css que contiene el estilo de todos los componentes de la aplicación: fondos, botones, etiquetas, barras de menús, etc.

Directorio resources:

Dentro de este directorio se encuentran los recursos utilizados en el proyecto. Este contiene un directorio *images*, un fichero denominado *tutorial.html* y otro fichero denominado *about.html*.

- **Directorio *images***: Cuenta con las imágenes necesarias para el desarrollo de la aplicación: el icono de aplicación, un icono que se utiliza para el botón de añadir imágenes en el editor de plantillas y todas las imágenes utilizadas en el tutorial que se muestra en la aplicación.
- **Fichero *tutorial.html***: Dispone del código HTML necesario para mostrar el contenido del tutorial de la aplicación en un contenedor de tipo *webView*. Desde este código se cargan las imágenes necesarias para el tutorial, contenidas en el directorio *images* mencionado previamente.
- **Fichero *about.html***: Dispone del código HTML necesario para mostrar el contenido del apartado “*Acerca de...*” de la aplicación en un contenedor de tipo *webView*. Desde este código se cargan el texto y las URLs mostradas en ese apartado.

4.4. Flujo de la aplicación

En este apartado se mostrarán diagramas de flujo de la aplicación en los casos de creación de notas básicas y elegantes. Para las notas avanzadas el funcionamiento sería una mezcla de ambas teniendo en cuenta el estilo que el usuario seleccione (básico o elegante). Además, se muestra el flujo de la gestión de la sesión de usuario. Concretamente en la Figura 4.3 se muestra el caso de notas básicas, el progreso desde que el usuario selecciona esta opción de creación desde el menú inicial.

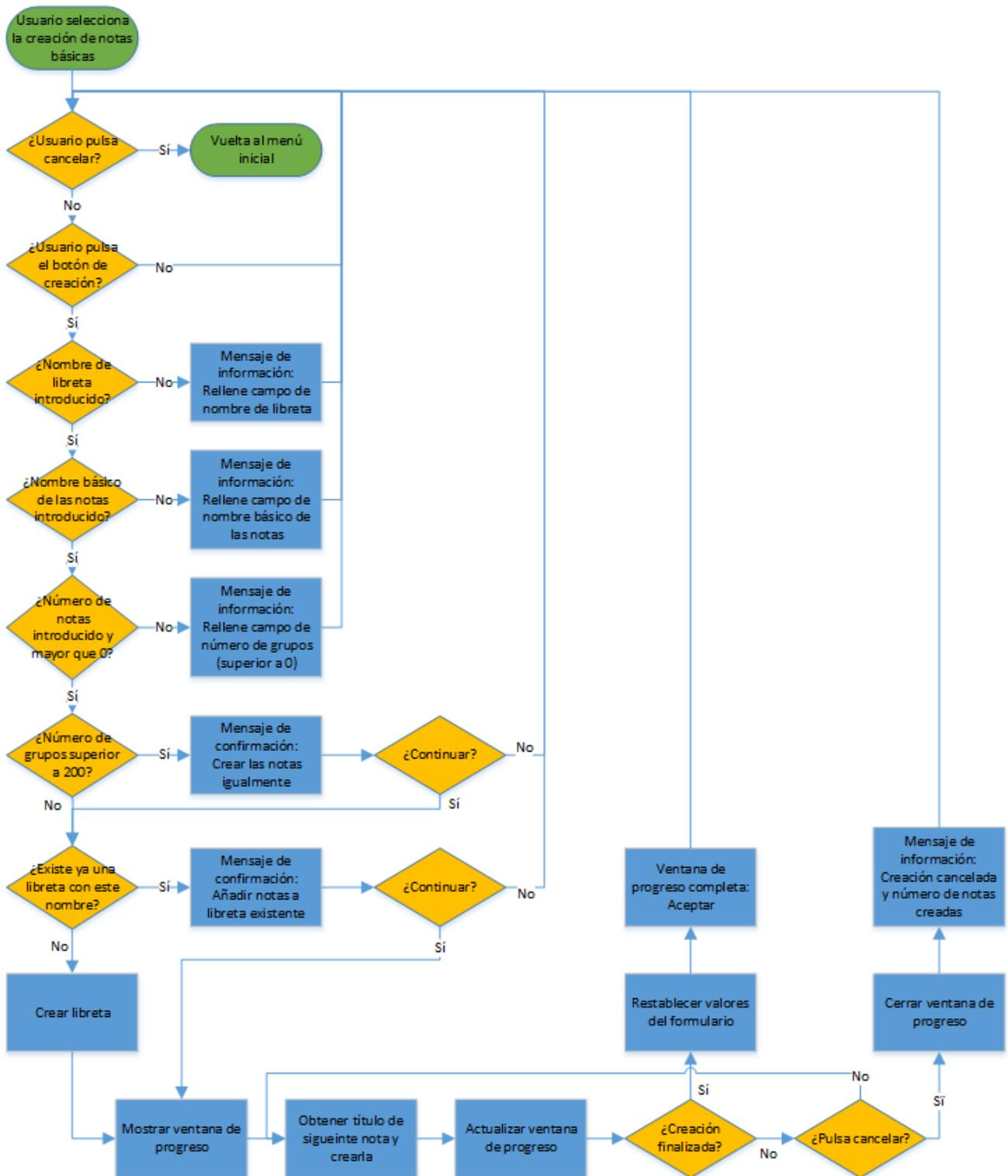


Figura 4.3. Diagrama de flujo de creación de notas básicas

Posteriormente, en la Figura 4.4 podemos ver el flujo para el caso de selección de notas elegantes, donde se puede ver el progreso de la aplicación ante las diferentes acciones posibles.

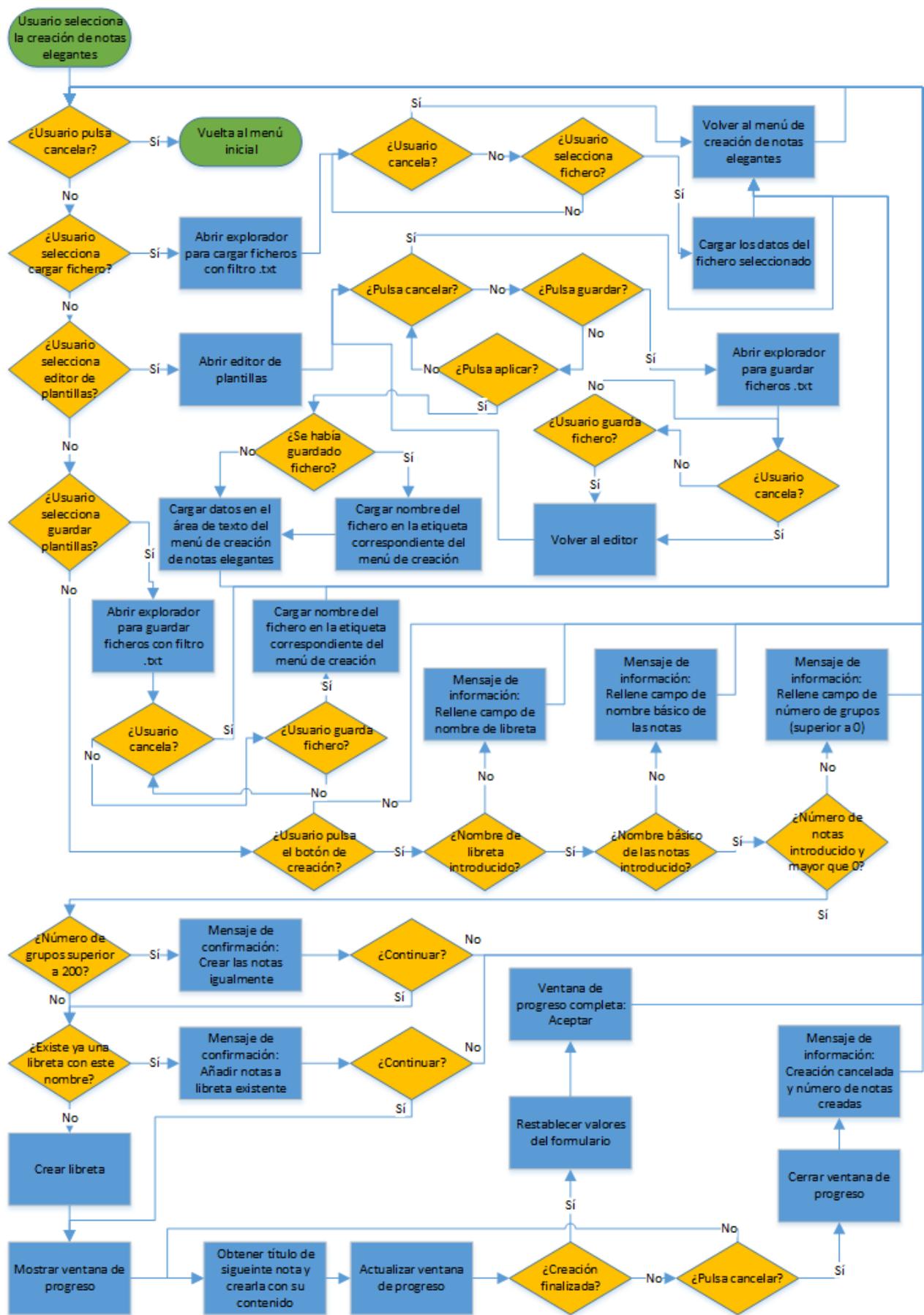


Figura 4.4. Diagrama de flujo de creación de notas elegantes

Por último, la Figura 4.5 muestra el diagrama de flujo correspondiente a la sesión, donde se puede observar qué pasa en el caso de que la sesión se cierre, expire, etc.

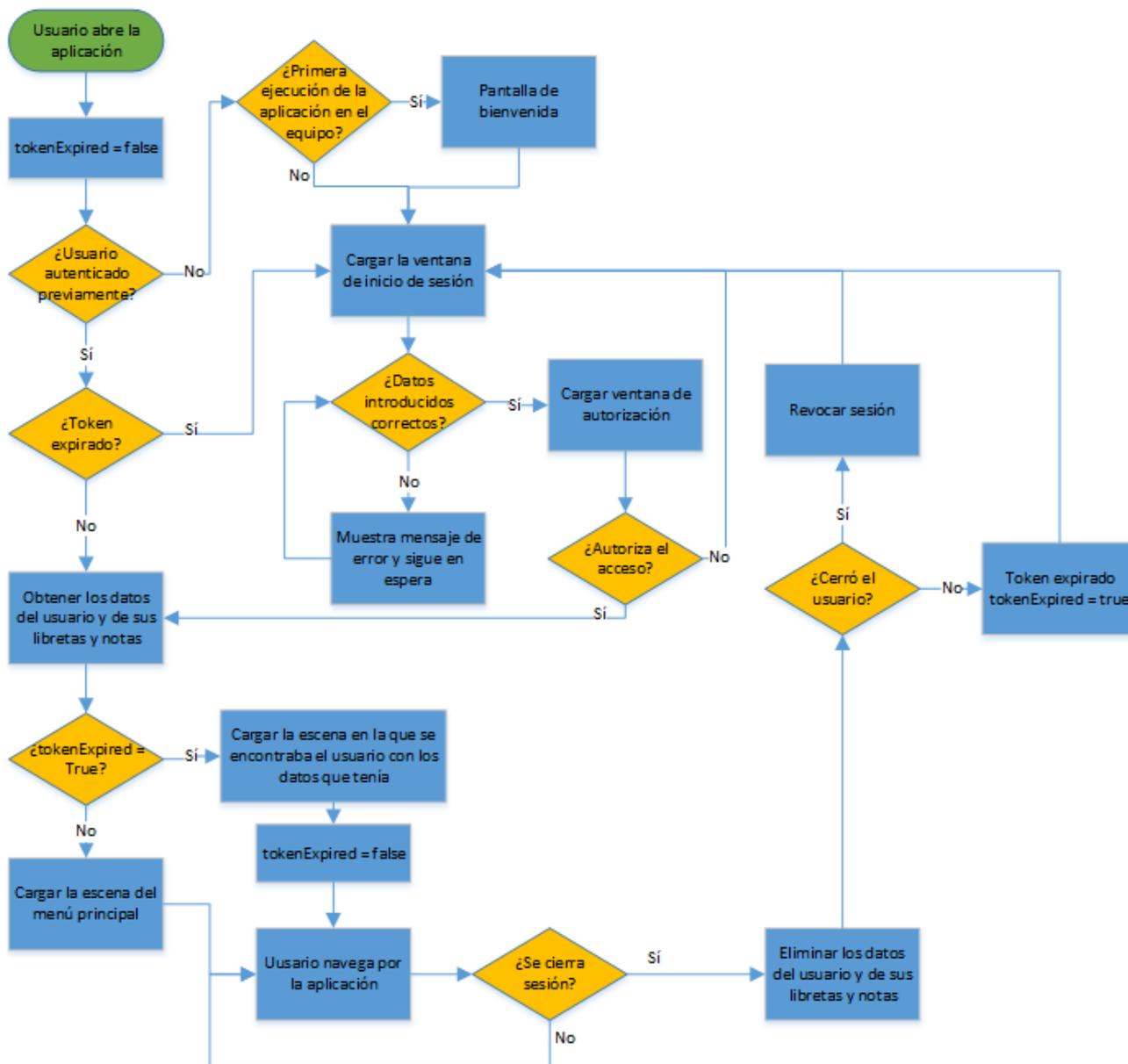


Figura 4.5. Diagrama de flujo de la gestión de la sesión de usuario

A través de esta serie de diagramas mostrados durante este apartado, se puede tener una visión general de cómo funciona el flujo de la aplicación.

4.5. Desarrollo del editor de plantillas

La aplicación cuenta con un editor de plantillas, utilizado para generar contenido que posteriormente puede ser añadido en las notas que se deseen crear. Para la realización de este editor, se ha utilizado un elemento con el que cuenta JavaFX llamado *HTMLEditor*. Este es un editor que permite escribir y editar este texto de una forma visual mediante botones para cambiar el color del texto, el tipo de fuente, el tamaño, etc. A partir de este es posible obtener el código HTML que generaría ese contenido. De este modo, se ha utilizado el contenido HTML generado por el editor, y mediante una serie de cambios se ha adecuado para que este pueda ser reconocido por *Evernote*, ya que para elaborar el contenido de las notas es necesario utilizar ENML. Entre estos cambios se incluye la eliminación de las cabeceras correspondientes al lenguaje HTML y añadir las correspondientes a ENML, el cerrado de algunas etiquetas que no son necesarias en HTML pero sí en ENML, y especialmente la adaptación de lo que se comentará posteriormente: las imágenes.

Como se ha mencionado, a este editor se le ha añadido un botón a mayores, que permite cargar imágenes. De este modo se pueden complementar las plantillas generadas con el uso de imágenes. Este añadido ha supuesto una dificultad a la hora de integrarlo en el editor debido al funcionamiento de la carga de imágenes u otro tipo de recursos como pueden ser audios, en *Evernote*. El problema en este caso radica en que las plantillas de contenido se guardan en ficheros de texto con marcado ENML. Para añadir un recurso local a *Evernote* (en este caso imágenes), es necesario crear un nuevo objeto de tipo recurso. Cada recurso posee el contenido binario del fichero y una serie de metadatos como es el nombre del fichero. Este recurso se debe adjuntar a la nota en el momento de crearla, además de añadir en el fichero de contenido ENML una referencia a ese recurso mediante marcado. Esta referencia contendría un *hash* en formato hexadecimal (que consiste en un resumen obtenido a partir del contenido binario) y el tipo de recurso que estamos utilizando. Por lo tanto, si quisiésemos guardar una plantilla en ENML que contuviese recursos, solo se guardaría la etiqueta en el fichero de texto que hace referencia a ese recurso, pero no el objeto recurso a utilizar, por lo que al utilizarlo posteriormente no funcionaría. Es por ello que cuando se guarda una plantilla que contiene recursos se avisa al usuario de que estos no serán almacenados. A su vez las etiquetas correspondientes a las referencias a recursos son eliminadas ya que dejan de tener utilidad.

A continuación, se muestra un breve ejemplo de cómo sería el editor, y el código ENML correspondiente a lo que se muestra en él.

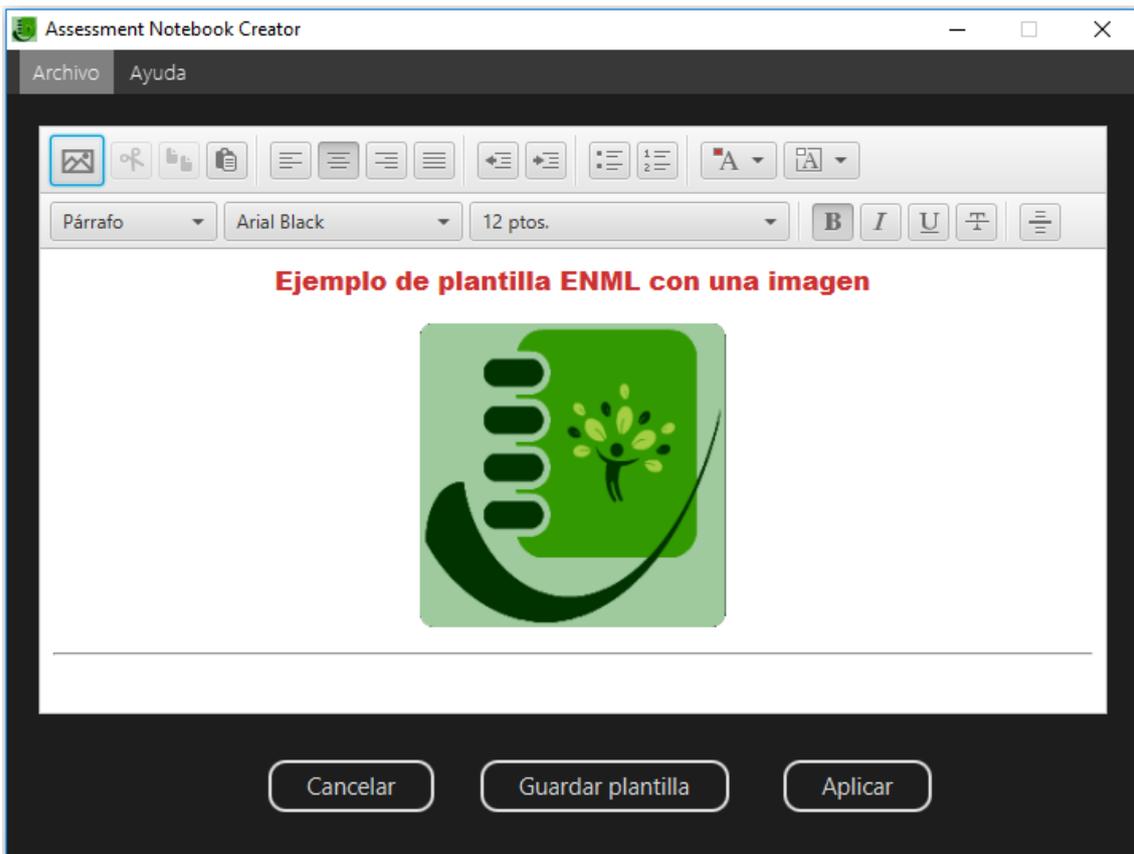


Figura 4.6. Ejemplo de plantilla elaborada con el editor

El código HTML correspondiente a la plantilla mostrada en la Figura 4.6 sería el siguiente:

```
<html dir="ltr"><head></head><body contenteditable="true"><p style="text-align: center;"><font face="Arial Black" color="#cc3333"><b>Ejemplo de plantilla ENML con una imagen</b></font></p><p style="text-align: center;"><font face="Arial Black" color="#cc3333"><b><br></b></font></p><hr></body></html>
```

Sin embargo, el código ENML que se genera al aplicar la plantilla mostrada en la Figura 4.6 se observa a continuación. En él se incluyen las cabeceras, aunque en el cuadro de texto de la aplicación estas no serían mostradas, sino que se aplican a posteriori al realizar la creación:

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE en-note SYSTEM "http://xml.evernote.com/pub/enml2.dtd"><en-note><p style="text-align: center;"><font face="Arial Black" color="#cc3333"><b>Ejemplo de plantilla ENML con una imagen</b></font></p><p style="text-align: center;"><font face="Arial Black" color="#cc3333"><b><en-media type="image/png" hash="38ddfd0a801e830cdf6b591ab57ee808"/><br></b></font></p><hr></hr></en-note>
```

Estos códigos HTML y ENML definen los párrafos, el tipo y color de la fuente, la imagen que se ha incluido, y la barra horizontal. Algunas diferencias, a parte de las cabeceras, se pueden observar en que las etiquetas `<hr>` y `
` no se cierran en el código HTML, mientras que en ENML es necesario. Haciendo especial énfasis en la parte de la

imagen, podemos ver que para ella en el ENML se ha generado una etiqueta *en-media*, de tipo imagen, en la que se indica el *hash* mencionado anteriormente, que hace referencia al recurso que se ha creado. En cambio, se puede observar que, en el código de marcado HTML, esta era una etiqueta de tipo *img* que mostraba la ubicación de la imagen.

Si generamos una libreta con este contenido obtenemos un resultado como el mostrado en la Figura 4.7.

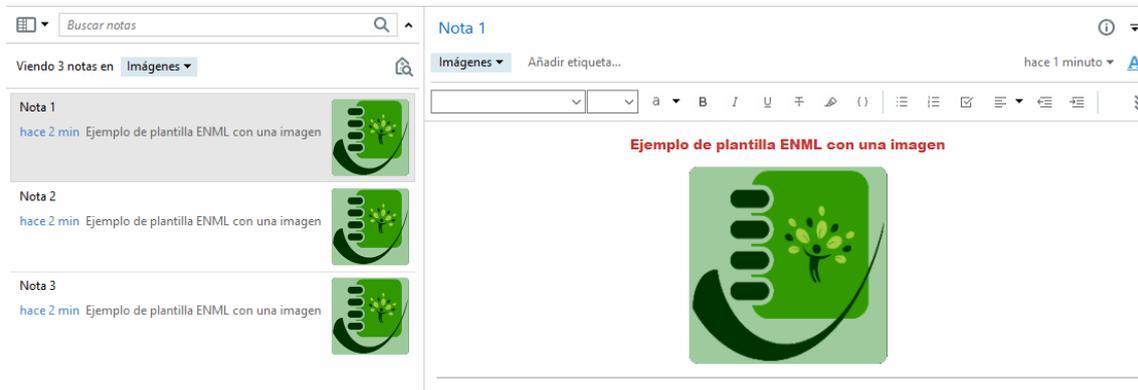


Figura 4.7. Resultado en Evernote con imagen

4.6. Preferencias

La aplicación desarrollada, utiliza preferencias para almacenar los datos referentes al usuario que ha iniciado sesión: su nombre de usuario y su token de acceso. Además, estas son usadas para guardar los datos de configuración seleccionados este: el tipo de codificación a utilizar y si desea completar el nombre de las notas con ceros. Estas son almacenadas en el ordenador del usuario de forma local. Para poder verlas, es necesario iniciar *regedit*, una herramienta que permite editar el registro del sistema operativo. Una vez dentro, se accede a estas preferencias siguiendo la ruta `HKEY_CURRENT_USER\Software\JavaSoft\Prefs`.

5. Manual de uso de la aplicación

5.1. Introducción

En este capítulo se describirá detalladamente el funcionamiento de la aplicación desarrollada durante este Trabajo Fin de Máster.

Assessment Notebook Creator es una aplicación para ordenador cuya utilidad reside en la creación automática de libretas y notas en la cuenta de *Evernote* del usuario. De este modo, es posible evitar la tediosa tarea de crear una a una todas las notas que necesitemos cuando se trata de una gran cantidad de ellas. En concreto, esta aplicación está orientada a proporcionar una ayuda en el entorno de la docencia, proporcionando la posibilidad a los profesores de crear libretas para sus asignaturas, y dentro de ellas tantas notas como alumnos o grupos de alumnos necesite, con sus correspondientes nombres y con el contenido que desee. Es una herramienta fácil de usar, y un requisito indispensable para hacerlo es disponer de una cuenta de usuario en *Evernote*, ya que será la que se utilice para iniciar sesión. Las libretas y notas creadas estarán disponibles en esa misma cuenta de *Evernote*.

Una vez creadas estas notas, el usuario puede editarlas desde su cuenta de *Evernote* como desee. De esta manera, en el caso de su utilización para la docencia, el profesor tiene la posibilidad de editar las notas incluyendo comentarios u otro tipo de elementos acerca de las diferentes actividades con las que cuente la asignatura para la cual hayan sido creadas, de forma que posteriormente podrá consultar este material para la evaluación del alumno o grupo de alumnos.

5.2. Inicio de sesión

Al utilizar por primera vez la aplicación, se muestra por pantalla un mensaje de bienvenida, como se muestra en la Figura 5.1 en el que se indica que para utilizarla se debe iniciar sesión mediante la cuenta de *Evernote* del usuario y proporcionar permiso para que la aplicación pueda comunicarse con dicha cuenta. En ese caso, tras seleccionar iniciar el proceso, o si ha utilizado la aplicación con anterioridad, pero ha finalizado su sesión o esta ha expirado, será necesario identificarse para iniciar una nueva sesión. Esto se realizará a través de nuestra cuenta de *Evernote*, por lo que se introducirá nuestro correo electrónico o nombre de usuario y nuestra contraseña, como se puede observar en la Figura 5.2. Una vez introducida esta información, se solicitará la autorización para que la aplicación pueda acceder a su cuenta de *Evernote* además de por cuánto tiempo (1 día, 1 semana, 30 días o un año), como se muestra en la Figura 5.3. En esa misma ventana, en la parte inferior, aparecerá una lista de las actividades o cambios que la aplicación podrá / no podrá hacer en su cuenta, las cuáles aparecen listadas en la Figura 5.4. Una vez que autoricemos a la aplicación a realizar estas actividades, seremos redirigidos a la página principal de la misma. En el caso de que abramos la aplicación sin haber cerrado

anteriormente nuestra sesión y sin que esta haya expirado, la aplicación se iniciará directamente en el menú inicial con nuestra cuenta.

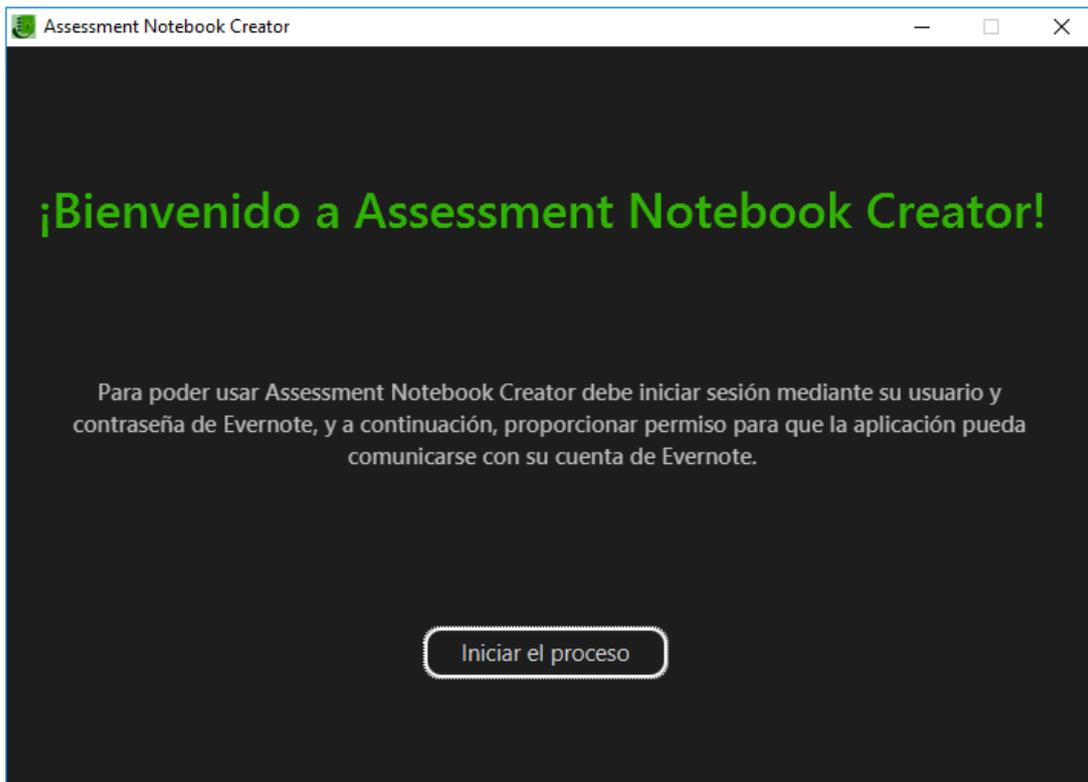


Figura 5.1. Mensaje de bienvenida a la aplicación

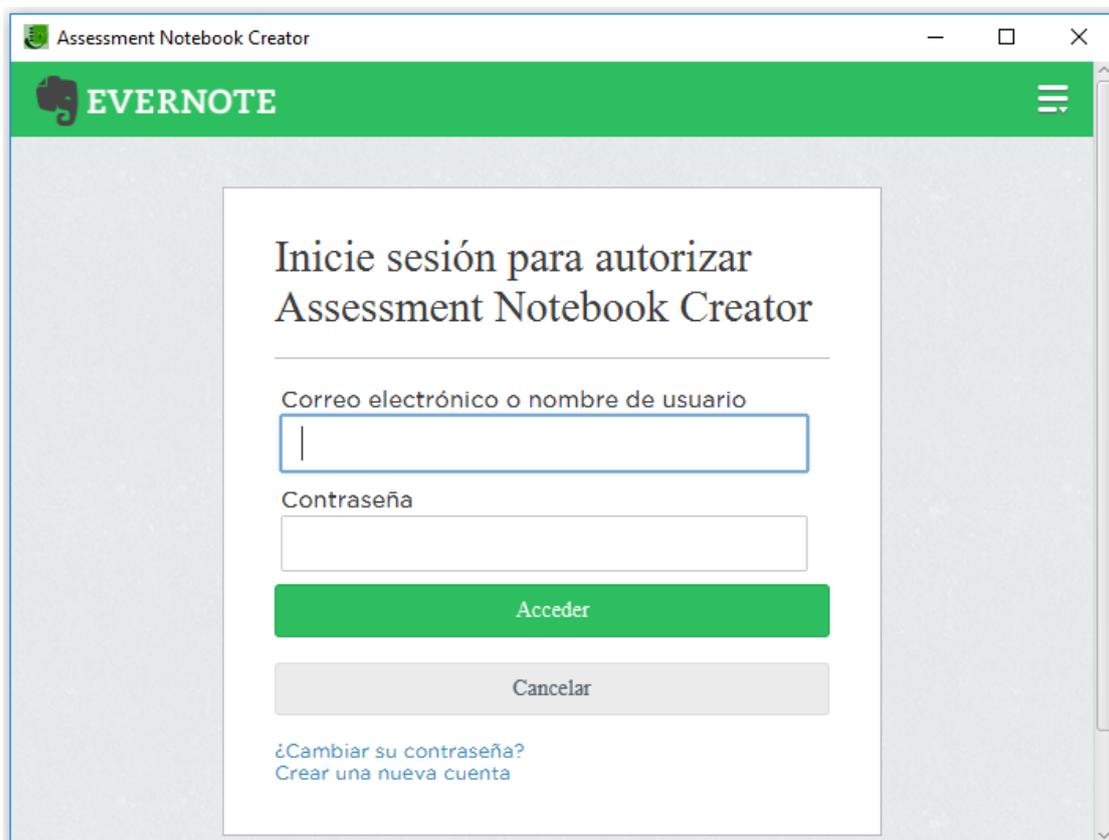


Figura 5.2. Inicio de sesión para autorizar Assessment Notebook Creator

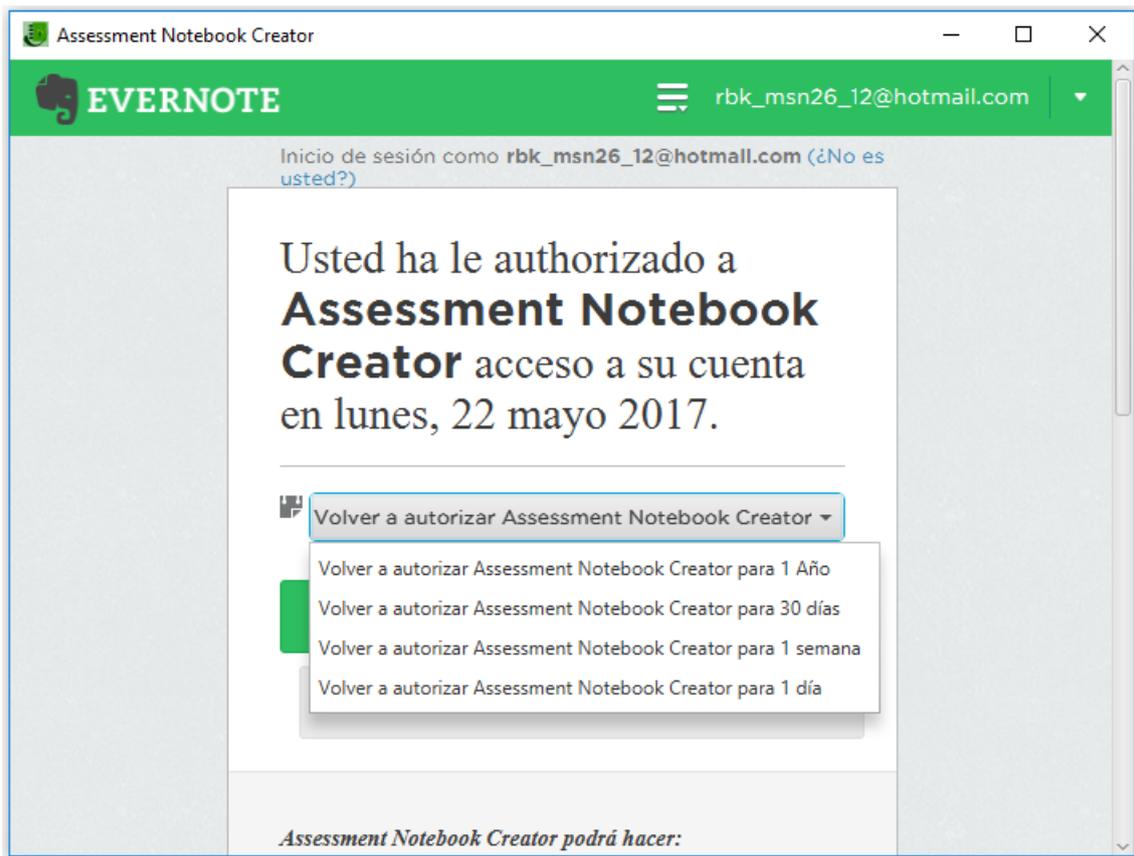


Figura 5.3. Autorización de acceso a su cuenta

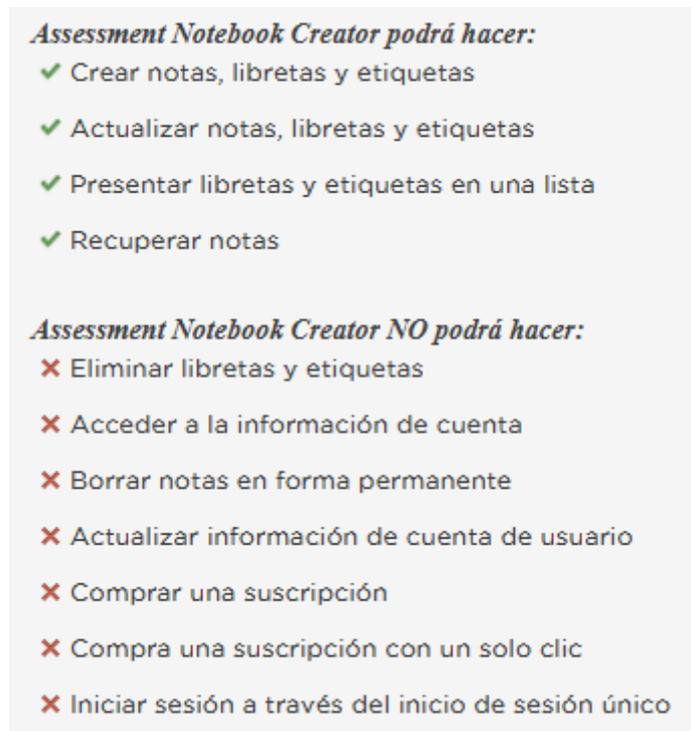


Figura 5.4. Actividades y cambios que podrá realizar la aplicación sobre su cuenta de Evernote

Cabe destacar que en el caso de que el token, utilizado para mantener la sesión y trabajar en ella, expire mientras la aplicación está siendo utilizada, el usuario será redirigido a la página de inicio de sesión para que se vuelva a identificar y permita el

acceso por un nuevo periodo de tiempo. Si esta autenticación se realiza de forma correcta, el usuario es redirigido al punto en el que se encontraba, manteniéndose los datos con los que estuviese trabajando.

5.3. Aplicación

En este apartado se verán las diferentes ventanas con las que cuenta nuestra aplicación.

5.3.1. Menú inicial

Este menú es el que aparece una vez tenemos nuestra sesión iniciada. Como se puede observar en la Figura 5.5, muestra un mensaje de bienvenida y proporciona tres opciones de crear notas diferentes:

- Crear notas de grupos **básicas**: Este tipo genera una libreta con el nombre proporcionado por el usuario, dentro de la cual se incluyen tantas notas sin contenido como este haya indicado. Las notas tendrán como nombre un nombre base (indicado por el usuario) seguido de un número que comenzará por el 1 y continuará hasta el número de notas a crear. En el apartado 5.3.2 se verá este caso con más detalle y con un ejemplo.
- Crear notas de grupos **elegantes**: Este caso generará una libreta y sus correspondientes notas de la misma forma que la indicada en el caso de notas básicas, con la diferencia de que aquí se podrá seleccionar una plantilla de contenido que defina la información que aparecerá en cada una de las notas. Además de esta opción, se puede escribir directamente el contenido deseado utilizando el lenguaje ENML en el área de texto destinada a ello, o bien utilizar el editor de plantillas a través del cual se puede generar el contenido de una forma mucho más sencilla, intuitiva y visual. Posteriormente, en el punto 5.3.3, se proporcionará información más detallada acerca de esto.
- Crear notas de grupos **avanzadas**: Esta opción es la más completa. En ella se proporciona el nombre de la libreta, pero para el caso del nombre de las notas, se proporcionará un fichero de texto que incluya línea por línea los nombres que se desean asignar. En cuanto al contenido de cada nota, podremos elegir que sean notas vacías eligiendo el estilo básico, o con contenido eligiendo el estilo elegante. En el apartado 5.3.4 se podrá ver de una forma más precisa.

En todos los casos si la libreta indicada ya existe, se ofrecerá al usuario la posibilidad de añadir las notas a la libreta ya existente o de cancelar la actividad.

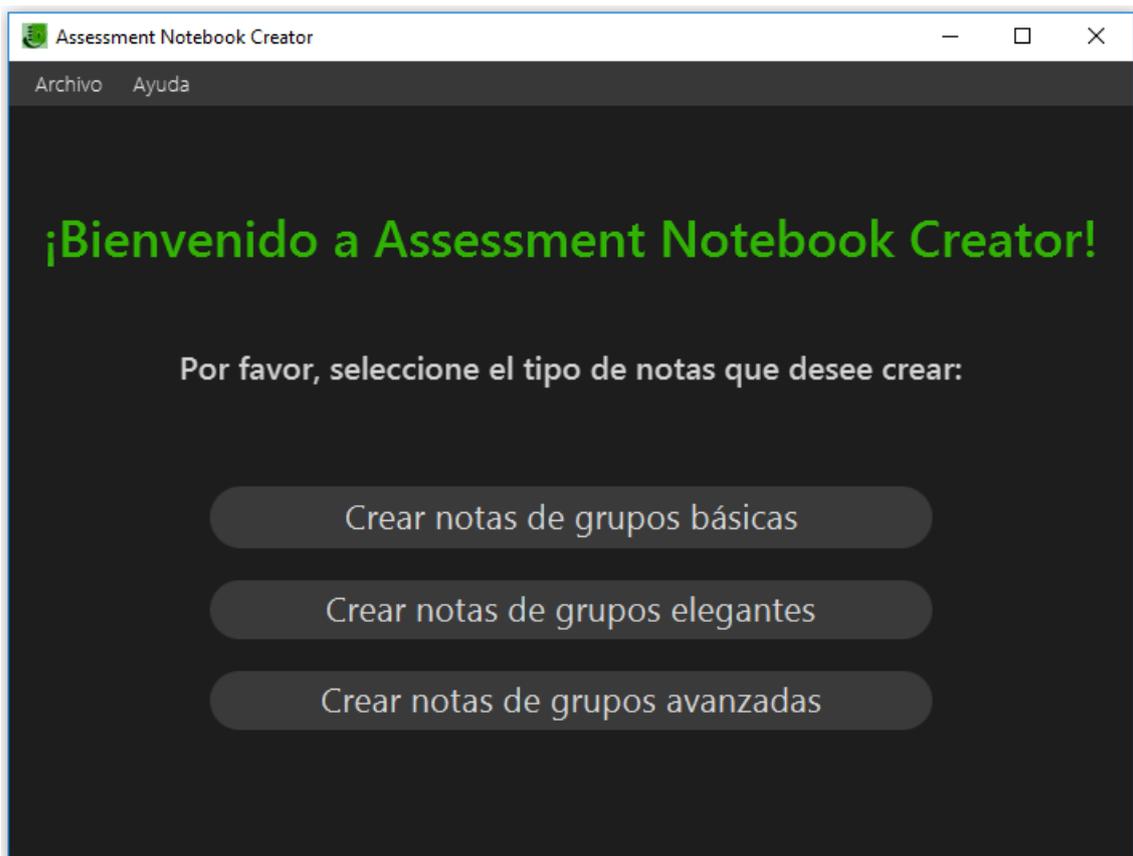


Figura 5.5. Menú inicial de la aplicación

5.3.2. Menú de creación de notas básicas

Si seleccionamos la primera opción “Crear notas de grupos básicas”, nos aparecerá el menú correspondiente a esta opción, como se puede ver en la Figura 5.6. En el campo “Nombre de la libreta” introduciremos el nombre de la libreta a crear, el campo “Nombre básico de las notas” determinará el comienzo del nombre de cada una de las notas y por último “Número de grupos” será el número de notas a crear. En el ejemplo mostrado en la Figura 5.6 si pulsamos el botón “Crear” se generaría una libreta llamada “Comunicaciones ópticas”, dentro de la cual se crearían 12 notas llamadas “Grupo01”, “Grupo02”, ..., “Grupo12” o bien “Grupo1”, “Grupo2”, ..., “Grupo12”. Esto último dependerá de la opción que tengamos seleccionada en el menú de configuración que se mostrará posteriormente, en el apartado 5.3.6.1. La Figura 5.7 muestra el resultado que obtendríamos con este ejemplo en nuestra cuenta de *Evernote*.

Además, el menú de creación de notas dispone de un botón “Cancelar” a través del cual podremos volver al menú inicial si no deseamos realizar la creación de notas.

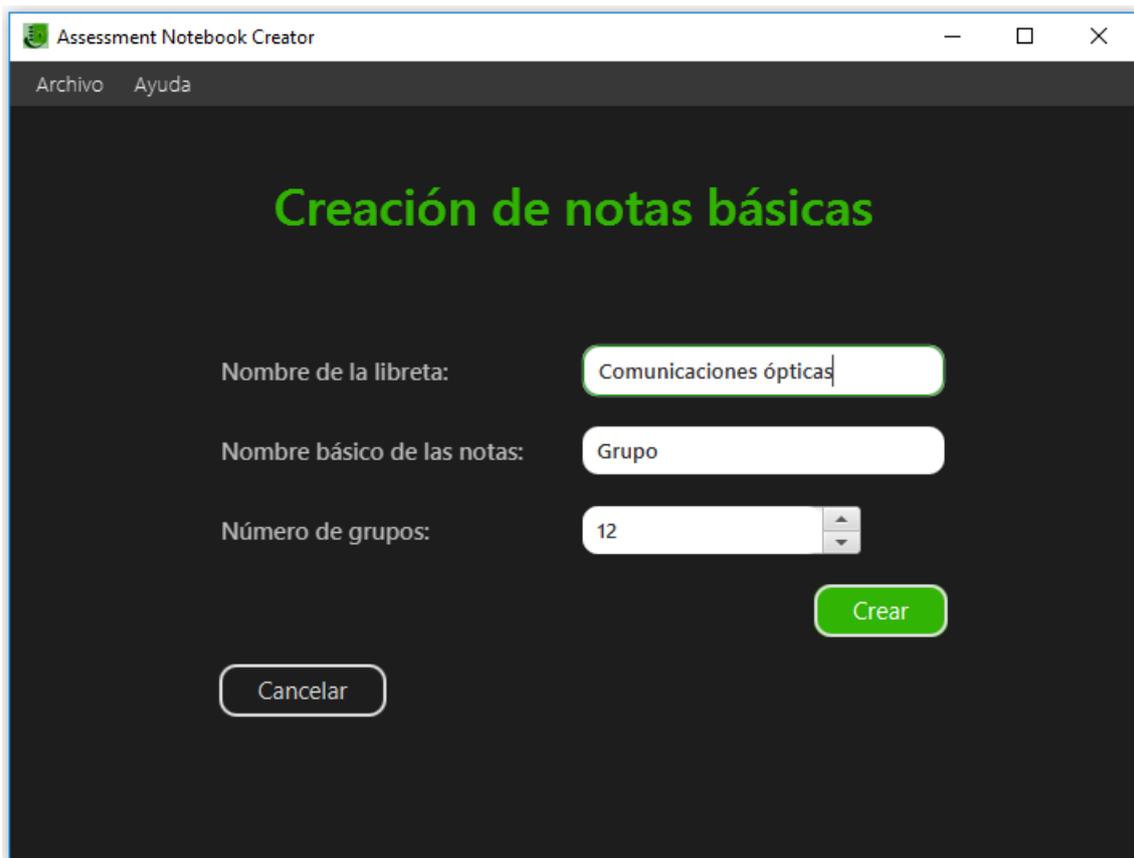


Figura 5.6. Menú de creación de notas básicas

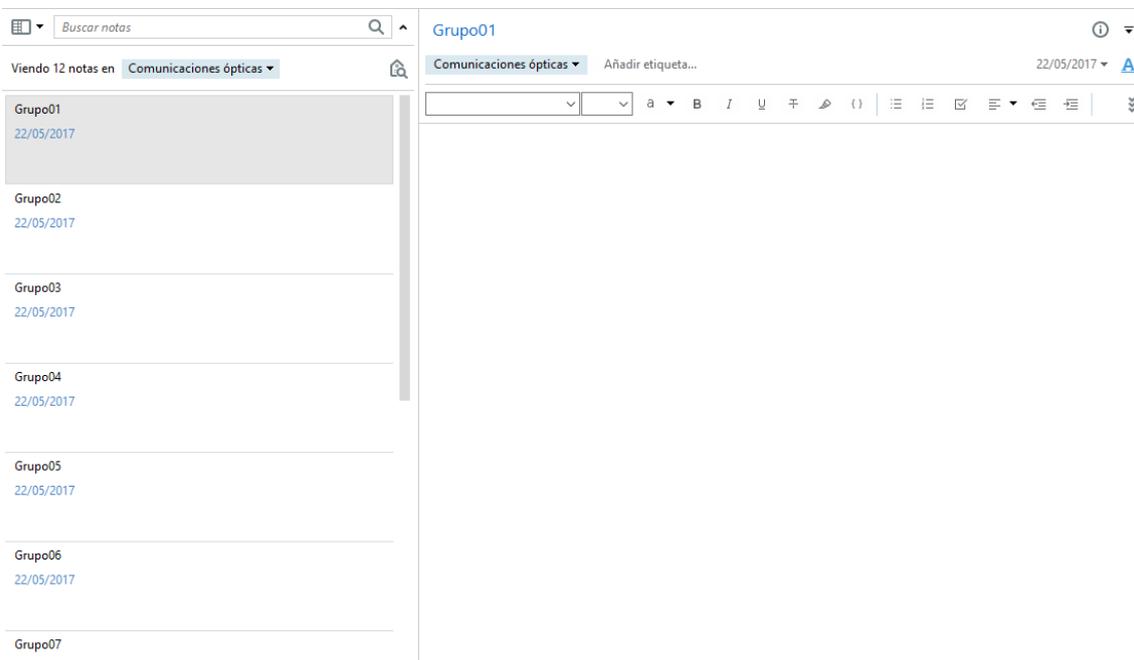


Figura 5.7. Resultado en Evernote tras crear notas de tipo básico

5.3.3. Menú de creación de notas elegantes

En el caso de seleccionar en el menú inicial la opción “Crear notas de grupos elegantes”, se mostrará por pantalla un menú como el de la Figura 5.8. Los campos equivalentes a los del menú de creación de notas básicas, funcionan de la misma forma.

La diferencia es que en este caso tenemos a mayores la posibilidad de añadir contenido a las notas, en lugar de crearlas vacías. Para añadir este contenido tenemos diferentes opciones:

- **Seleccionar plantilla de contenido:** Si pulsamos el botón “Cargar fichero” correspondiente a esta opción, se abrirá un explorador de archivos como el mostrado en la Figura 5.9. Este se sitúa en el directorio raíz del usuario y permite únicamente seleccionar ficheros de extensión “.txt” ya que será la necesaria para utilizar nuestras plantillas. La plantilla deberá contener la información deseada siguiendo los estándares de ENML, ya que es el lenguaje de marcado utilizado por *Evernote* para ello. Esta plantilla definirá el contenido que deseamos que se aplique en nuestras notas. Una vez seleccionado un fichero, el nombre de este aparecerá al lado del botón (para que sepamos cuál tenemos seleccionado), y el contenido del fichero se mostrará en el área de texto destinada a ello.
- **Escribir el contenido directamente en el área de texto:** También es posible, en lugar de seleccionar un fichero que incluya el contenido deseado, escribir este directamente en el área de texto que aparece. Al igual que en el caso anterior, el contenido deberá ser definido mediante ENML para que *Evernote* pueda reconocerlo.
- **Utilizar el editor de plantillas:** Si pulsamos el botón “Editor de plantillas”, se abrirá un editor en el que podremos definir el contenido de nuestra plantilla de forma visual, y que generará automáticamente el código ENML necesario. Una vez que lo aplicásemos, al igual que en el caso de cargar un fichero, el código obtenido aparecerá en el área de texto, y si además hemos guardado la plantilla, aparecerá el nombre de esta. La apariencia y uso del editor se verá posteriormente en el apartado 5.3.5.

En este modo de creación (elegante), es obligatorio introducir un contenido para las notas. En el caso de que no se deseen crear con contenido, deberemos seleccionar el modo de creación básico.

En nuestro menú de creación, también podemos observar un botón de “Guardar plantilla”. Este nos permitirá guardar una plantilla a partir del contenido que tengamos en el área de texto, siempre que este no esté vacío y que el contenido sea diferente al del último fichero cargado (ya que en ese caso sabemos que ya existiría un fichero con ese mismo contenido). Este botón estará deshabilitado en los casos en los que no sea posible guardar una plantilla, mientras que en el resto de casos se mostrará habilitado y se podrá pulsar. Al pulsar este botón se abre otro explorador de ficheros, en este caso con la opción de guardado, que a su vez únicamente permite guardar ficheros con extensión “.txt”. La Figura 5.10 muestra este otro explorador.

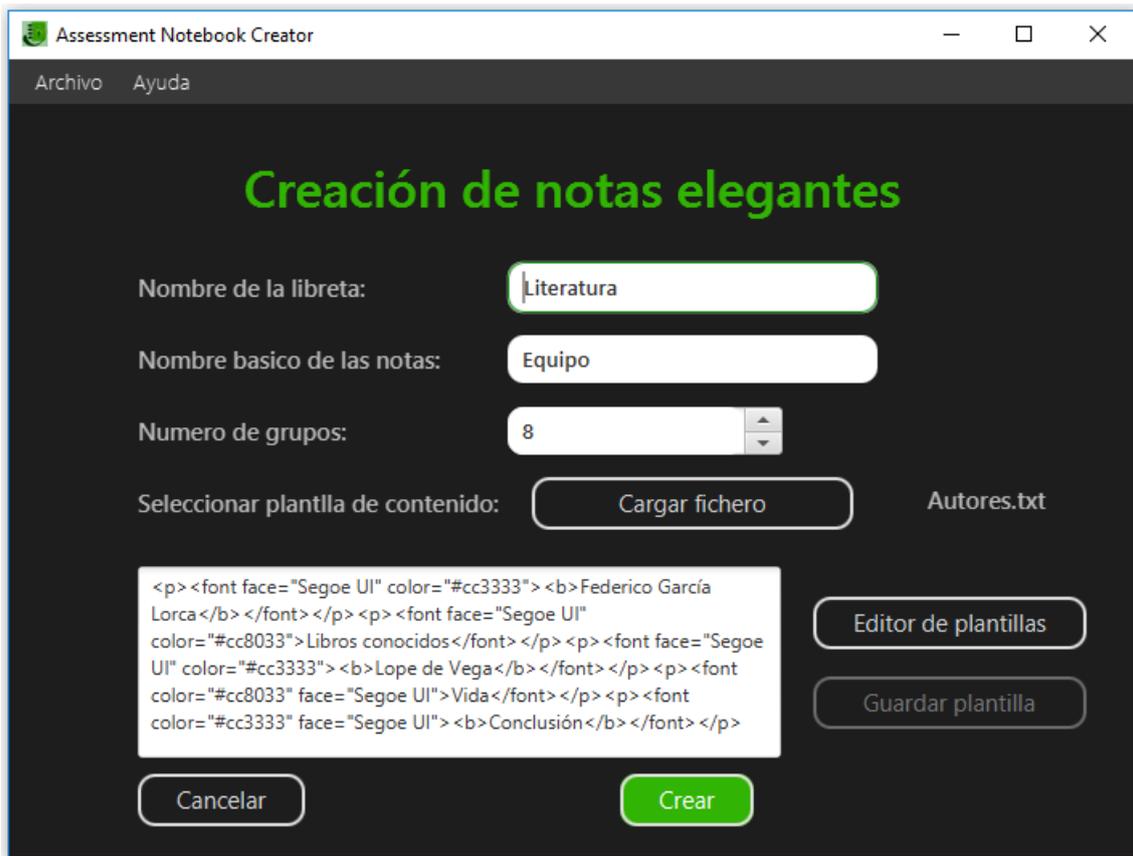


Figura 5.8. Menú de creación de notas elegantes

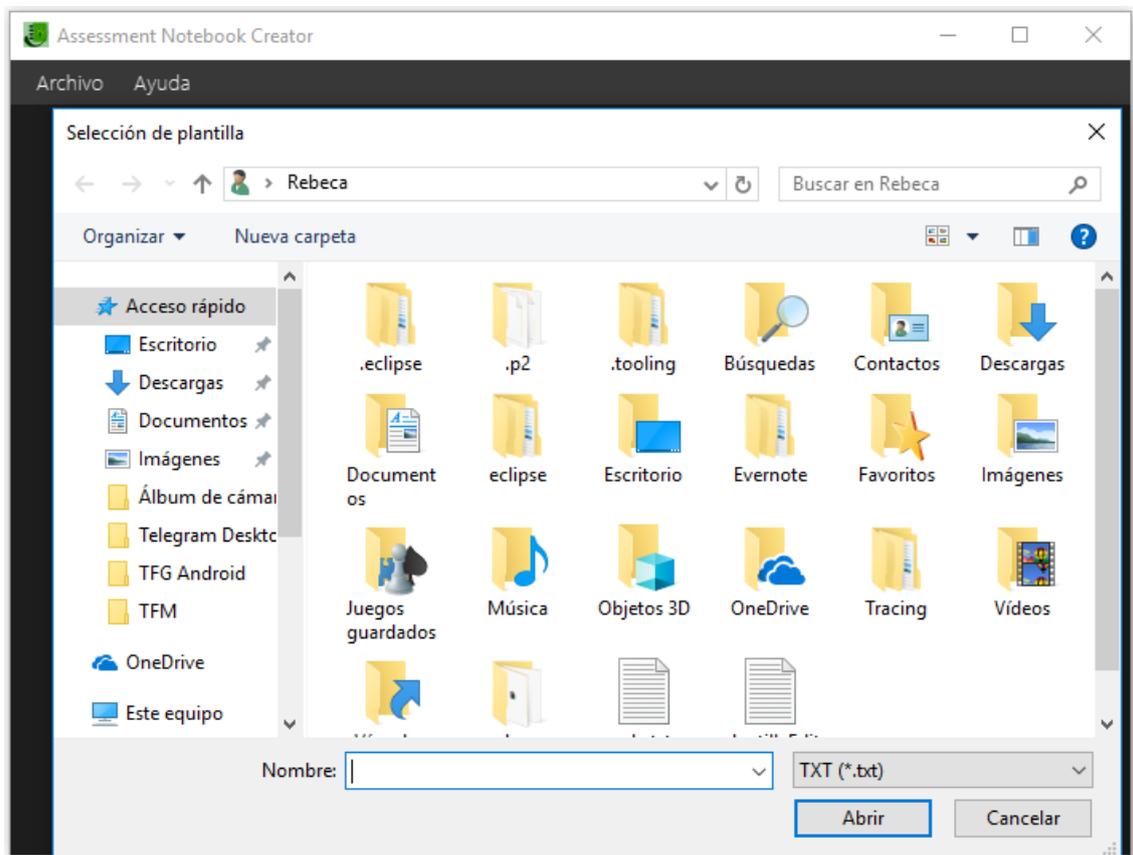


Figura 5.9. Explorador de ficheros para cargar una plantilla

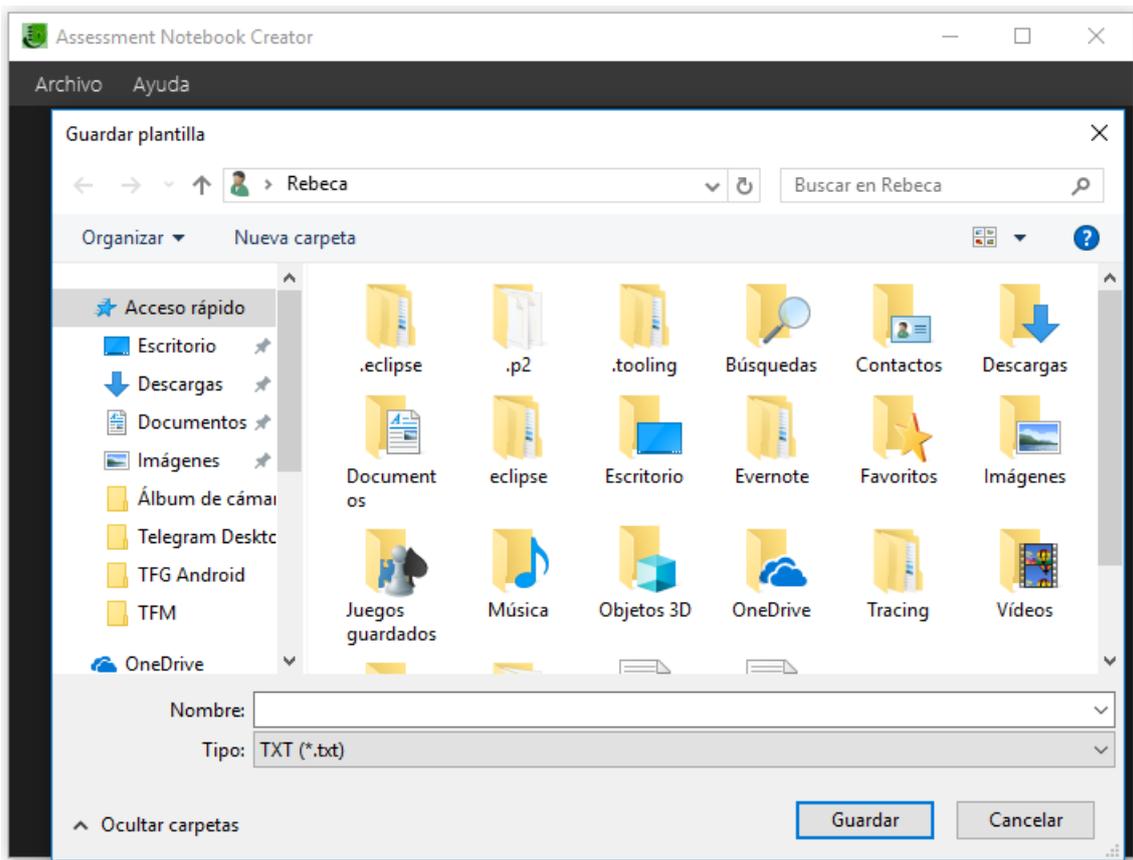


Figura 5.10. Explorador de ficheros para guardar una plantilla

El resultado en nuestra cuenta de *Evernote* tras ejecutar la creación de notas mostrada en el ejemplo de la Figura 5.8 se puede observar en la Figura 5.11. Se creó una libreta llamada Literatura, con 8 notas llamadas “Equipo(nº de nota)”. Además, cada una de las notas tiene el contenido especificado en la plantilla que se utilizó.

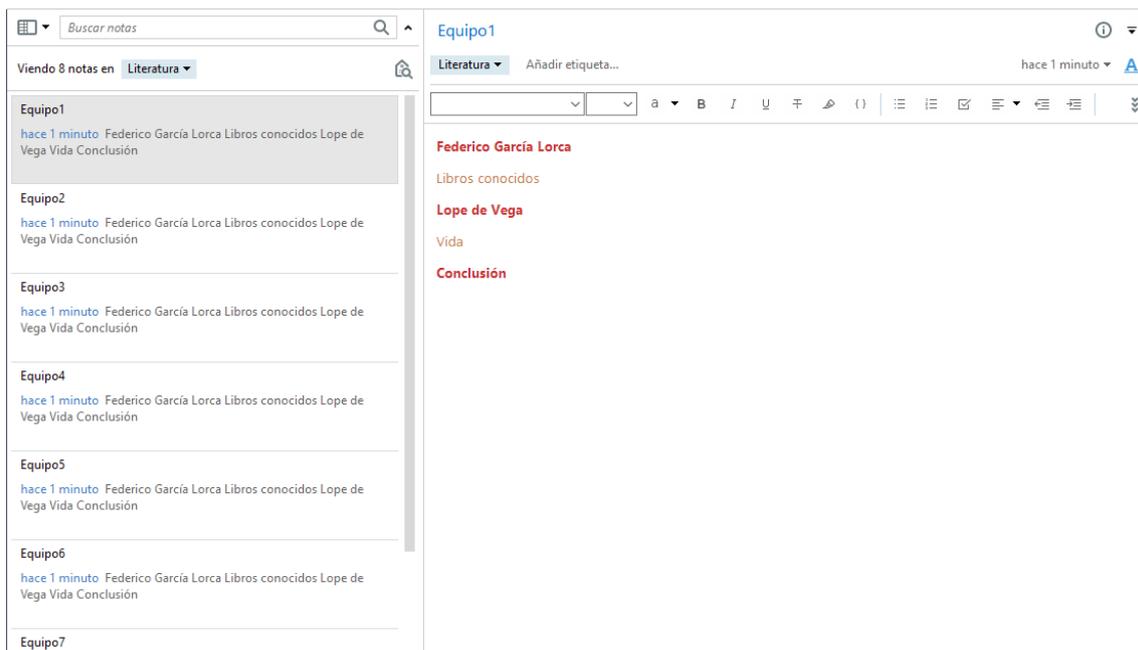


Figura 5.11. Resultado en Evernote tras crear notas de tipo elegante

Como en el caso de notas básicas, a parte del botón para crearlas, tenemos otro botón para cancelar en el caso de que decidamos que queremos volver al menú inicial.

5.3.4. Menú de creación de notas avanzadas

Para la opción “Crear notas de grupos avanzadas”, nos encontramos con el menú mostrado en la Figura 5.12. En este caso se debería introducir como siempre el nombre de la libreta.

En cuanto al nombramiento de las notas, este se realiza de una forma diferente; en lugar de seleccionar un nombre básico y un número de notas, en este caso se selecciona un fichero “.txt” que se cargará en el área de texto destinada a ello, o bien se escribirá directamente en esta. De este modo, el área de texto se lee y se crean tantas notas como líneas contenga. El nombre de esas notas será el contenido de cada una de las líneas. A modo ilustrativo, la Figura 5.13 muestra el contenido del fichero “Equipos.txt”, el cual es utilizado en el ejemplo de la Figura 5.12. En este caso se crearían 4 notas, ya que el fichero contiene 4 líneas, y el nombre de cada nota sería del tipo: para la primera nota “Equipo 1: Marta y Pablo”, y así con cada línea para cada una de las notas.

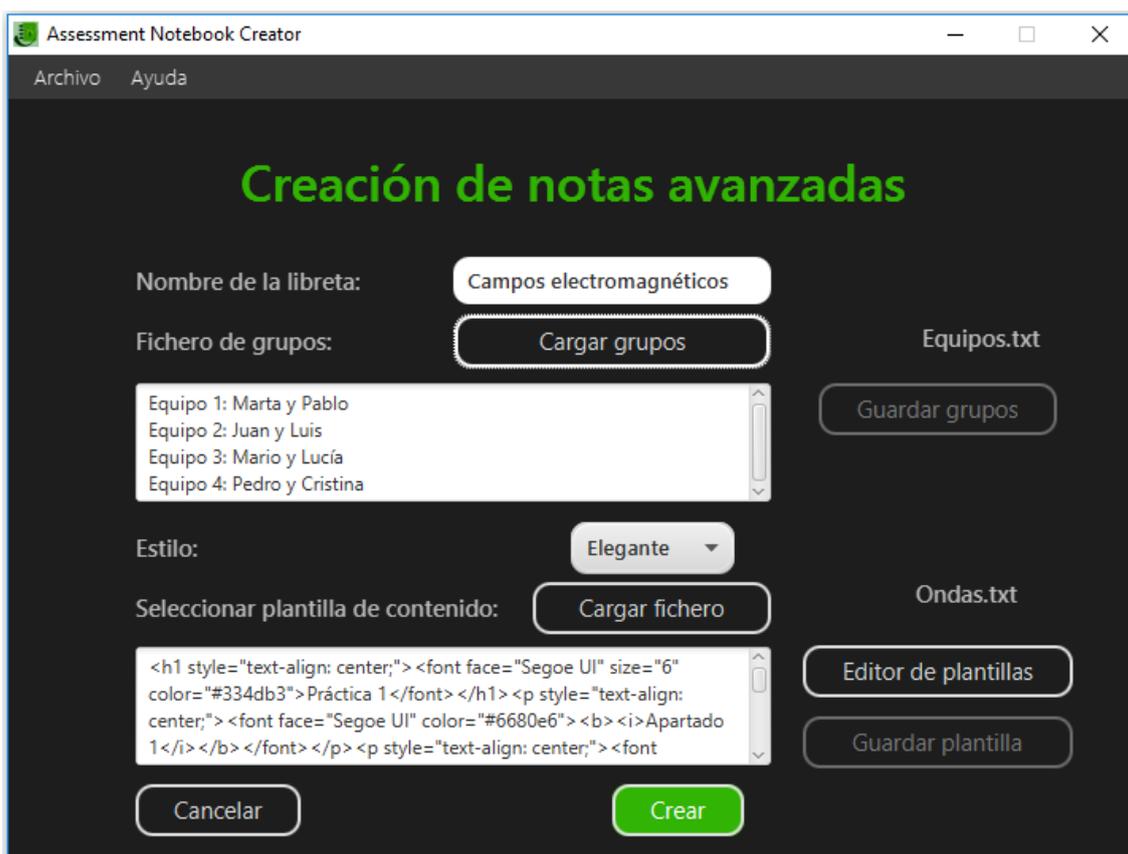


Figura 5.12. Menú de creación de notas avanzadas

Posteriormente se encuentra la opción de estilo, en la cual podremos seleccionar “Básico” o “Elegante”. Con la opción de básico, todas las opciones relacionadas con el contenido de las notas se deshabilitarían, ya que, en este modo, las notas se crearían sin contenido, vacías. En cambio, en el caso de elegir el estilo elegante, tendremos todas las

opciones que teníamos en el caso de notas elegantes, para seleccionar el contenido de nuestras notas.

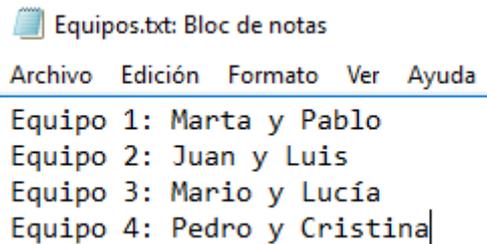


Figura 5.13. Contenido del fichero Equipos.txt

El resultado de la creación de notas con los datos del ejemplo de la Figura 5.12, se muestran en la Figura 5.14. Como podemos observar se ha creado una libreta con el nombre “Campos electromagnéticos” con 4 notas cuyos nombres se corresponden con las líneas del fichero utilizado para tal fin. Además, el contenido de cada nota se corresponde con el un fichero llamado “Ondas.txt”.

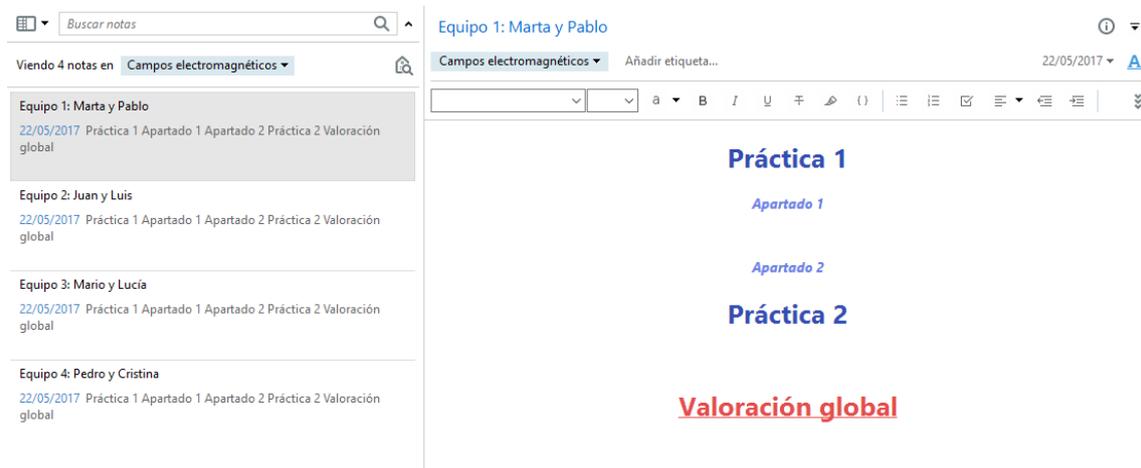


Figura 5.14. Resultado en Evernote tras crear notas de tipo avanzado

5.3.5. Editor de plantillas

Como se mencionó anteriormente, desde la creación de notas elegantes y avanzadas, podemos acceder a un editor de plantillas, para generar el contenido que deseemos para que sea aplicado en nuestras notas. El editor, como se puede observar en la Figura 5.15. Dispone de diferentes botones y desplegados que permiten: recortar, copiar, pegar (aunque estas acciones también se pueden realizar mediante los accesos rápidos del teclado Ctrl+X, Ctrl+C y Ctrl+V), establecer el alineamiento del texto (izquierda, centrado, derecha o justificado), establecer tabulaciones hacia atrás o hacia adelante, insertar listas bien con puntos sin numerar o con numeración, establecer el color del texto y de su fondo, elegir si nuestro texto es un párrafo o algún tipo de cabecera, el tipo de fuente de la letra, el tamaño de letra, configurarla como negrita, cursiva, subrayado o tachado, y añadir líneas de separación. Adicionalmente cuenta con un botón que permite introducir imágenes.

Al utilizar este editor, tenemos en su cuadro de texto una representación visual de lo que obtendríamos, de modo que es muy fácil de utilizar. Desde este editor, podremos guardar una plantilla con el contenido generado, la cuál será almacenada en un fichero de extensión “.txt” con el nombre que el usuario desee. Esta plantilla contendrá el texto ya convertido al formato ENML para que posteriormente pueda ser utilizada en nuestra aplicación. Desde el editor además podemos pulsar el botón “Aplicar” de forma que volveríamos al menú de creación de notas en el que estábamos, con la diferencia de que ahora en la parte del contenido de las notas, aparecería la información correspondiente a la plantilla aplicada. Esta puede ser aplicada habiendo sido guardada previamente (por ejemplo, en los casos en los que tengamos intención de volver a utilizarla en un futuro) o podemos hacerlo sin guardar, de forma que únicamente se aplicaría para este caso. En el caso de haberla guardado previamente a aplicarla, el nombre del fichero utilizado también aparecerá en nuestro menú de creación de notas al igual que cuando lo cargábamos desde la opción de cargar plantillas. También se cuenta con un botón para cancelar para el caso en que hayamos decidido que no deseamos utilizar la plantilla que estábamos creando mediante el editor.

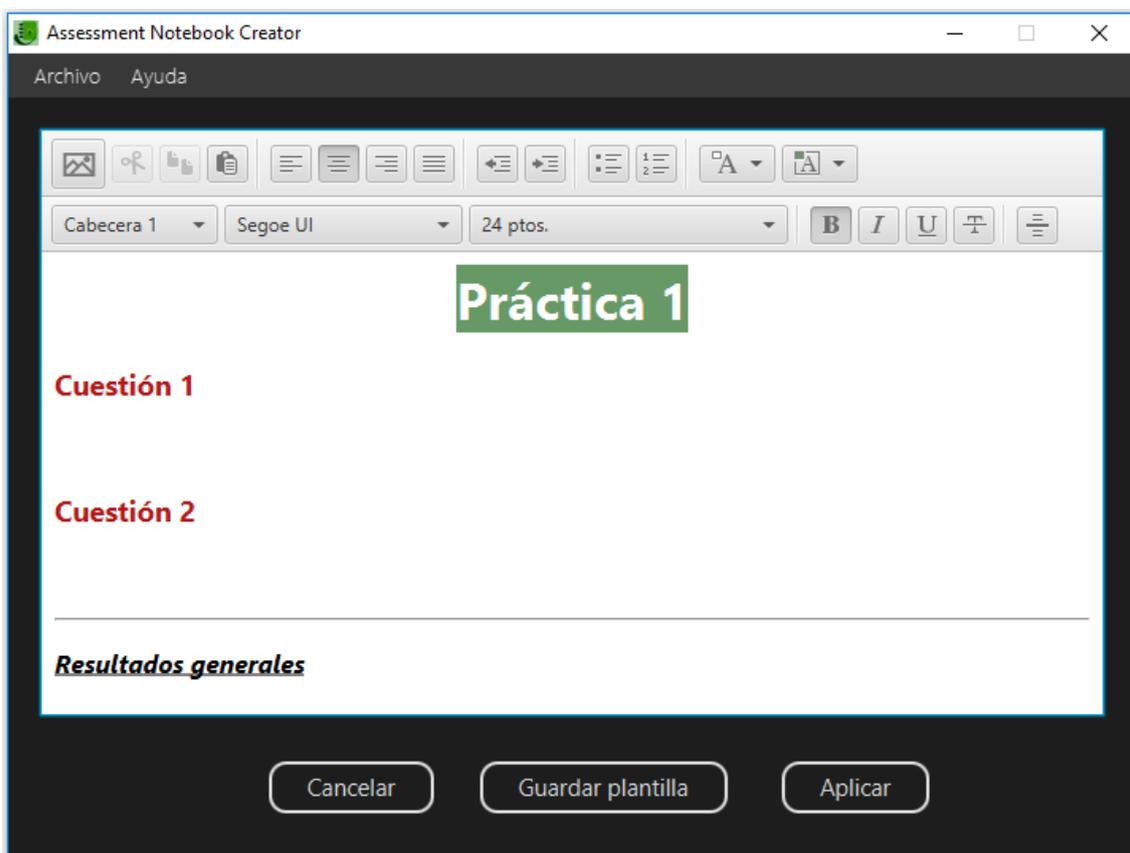


Figura 5.15. Editor de plantillas

En el caso de que cuando abramos nuestro editor, tuviésemos un contenido en el área de texto del menú de creación de notas correspondiente, este será representado en el área de texto del editor de una forma visual, así podremos ver qué es exactamente lo que contiene ese código.

Respecto a la opción de añadir imágenes (el botón de la parte superior a la izquierda), esta es una función adicional que únicamente se puede utilizar bajo ciertas

restricciones. Si lo pulsamos obtendremos un explorador de ficheros exclusivo para imágenes, donde podremos seleccionar la imagen deseada que será incluida en nuestra plantilla. Una vez pulsemos aplicar, obtendremos el código correspondiente y podremos crear nuestras notas con sus correspondientes imágenes. Se debe tener en cuenta que una vez que se ha aplicado la plantilla del editor, no se puede volver a entrar al editor y pulsar aplicar, ya que las imágenes se perderían. Si se desea volver a editar desde este editor, será necesario incluir de nuevo las imágenes y posteriormente volver a aplicar. Por ello, si se intenta abrir de nuevo el editor una vez aplicada una plantilla con imágenes, aparecerá un mensaje de confirmación como el que se muestra en la Figura 5.16.

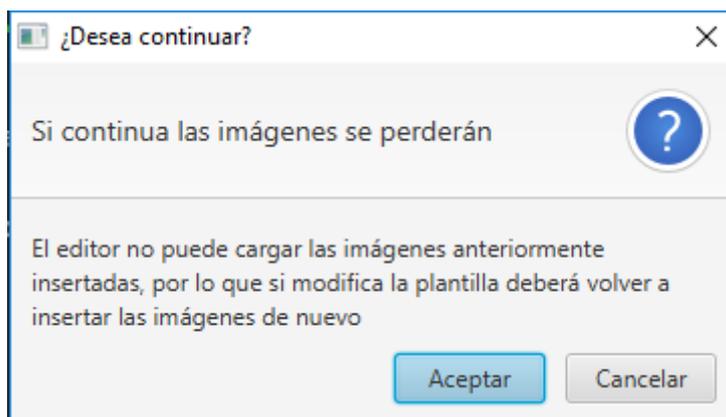


Figura 5.16. Confirmación al volver a abrir el editor tras haber aplicado una plantilla con imágenes

También se debe tener en cuenta que, si una plantilla en la que se han incluido imágenes es guardada, las imágenes no se guardarán, por lo que el usuario deberá volver a introducirlas si desea usarla posteriormente. Para ello se avisa al usuario con un mensaje como el mostrado en la Figura 5.17.

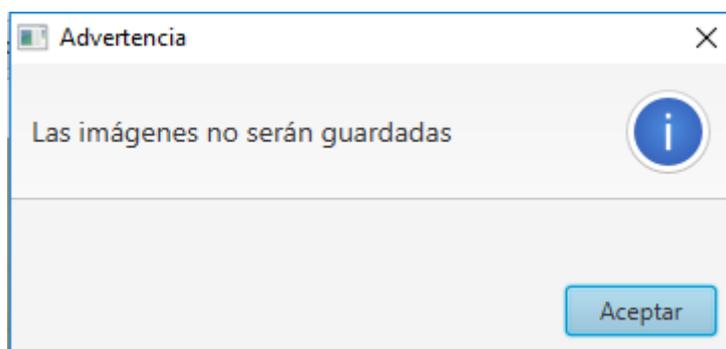


Figura 5.17. Mensaje de advertencia: No se guardarán las imágenes

5.3.6. Barra de menús

La aplicación cuenta con una barra de menús en las que tenemos las opciones "Archivo" y "Ayuda".

5.3.6.1. Archivo

Si desplegamos el menú archivo, nos encontramos con las opciones "Configuración" y "Cerrar sesión", como se puede observar en la Figura 5.18.

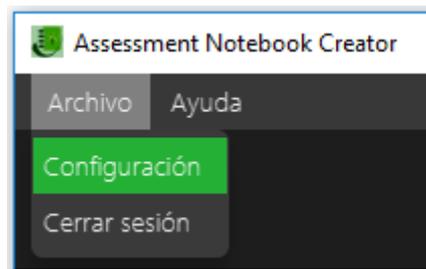


Figura 5.18. Menú de archivo

En el caso de seleccionar la opción cerrar sesión, nuestra sesión finalizará, de modo que el usuario será redirigido a la página inicial de inicio de sesión.

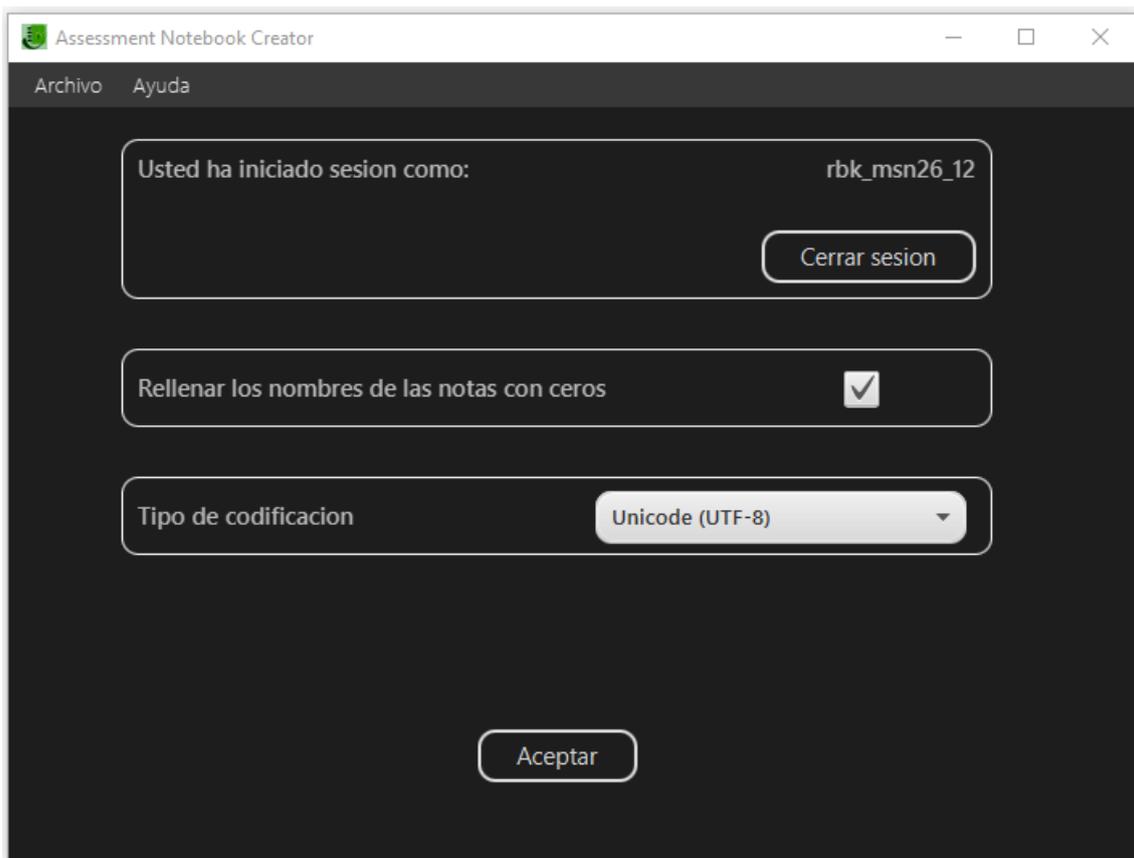


Figura 5.19. Ventana de configuración

Si seleccionamos la configuración, nos aparecerá una ventana como la mostrada en la Figura 5.19. En ella se muestra el usuario con el que se ha iniciado sesión, además de ofrecer desde ahí también la posibilidad de cerrar la sesión para, por ejemplo, en el caso de que no esté iniciada con el usuario que queríamos. A mayores, podemos ver una opción que dice “Rellenar los nombres de las notas con ceros”. Esto, como fue mencionado previamente, permite que, en los casos de creación de notas básicas o elegantes, la denominación de las notas creadas sea de una forma u otra. Para obtener una mayor claridad en este aspecto, veamos un par de ejemplos.

Como primer ejemplo, si un usuario utiliza como nombre básico de las notas la palabra “Alumno” y decide crear 15 notas.

- Si tiene la opción de rellenar con ceros desmarcada, se crearán notas tituladas “Alumno1”, “Alumno2”, ..., “Alumno14” y “Alumno15”.
- Si la opción de rellenar con ceros estuviera marcada, el nombre de todas las notas tendría el mismo número de caracteres ya que las que no lo tengan se completarían con ceros, es decir, tendríamos notas llamadas “Alumno01”, “Alumno02”, ..., “Alumno14” y “Alumno15”.

A modo de segundo ejemplo, si el número de notas a crear fuese 105 y el nombre básico fuera el mismo.

- Si tiene la opción de rellenar con ceros desmarcada, se crearán notas tituladas “Alumno1”, “Alumno2”, ..., “Alumno20”, “Alumno21”, ..., “Alumno104” y “Alumno105”.
- Si la opción de rellenar con ceros estuviera marcada, el nombre de todas las notas tendría el mismo número de caracteres ya que las que no lo tengan se completarían con ceros, es decir, tendríamos notas llamadas “Alumno001”, “Alumno002”, ..., “Alumno020”, “Alumno021”, ..., “Alumno104” y “Alumno105”.

Como última opción de la configuración, tendríamos la opción relativa al tipo de codificación. Aquí se muestra una lista con diferentes opciones de codificación, de las cuales la seleccionada será la utilizada a la hora de leer los ficheros de texto que se carguen y guardar los que sean creados (plantillas). En la Figura 5.20 podemos ver la lista de diferentes codificaciones posibles.

Para salir de la ventana de configuración, pulsamos el botón de aceptar y la aplicación vuelve al menú en el que se encontrase el usuario, con los mismos datos que tuviese.

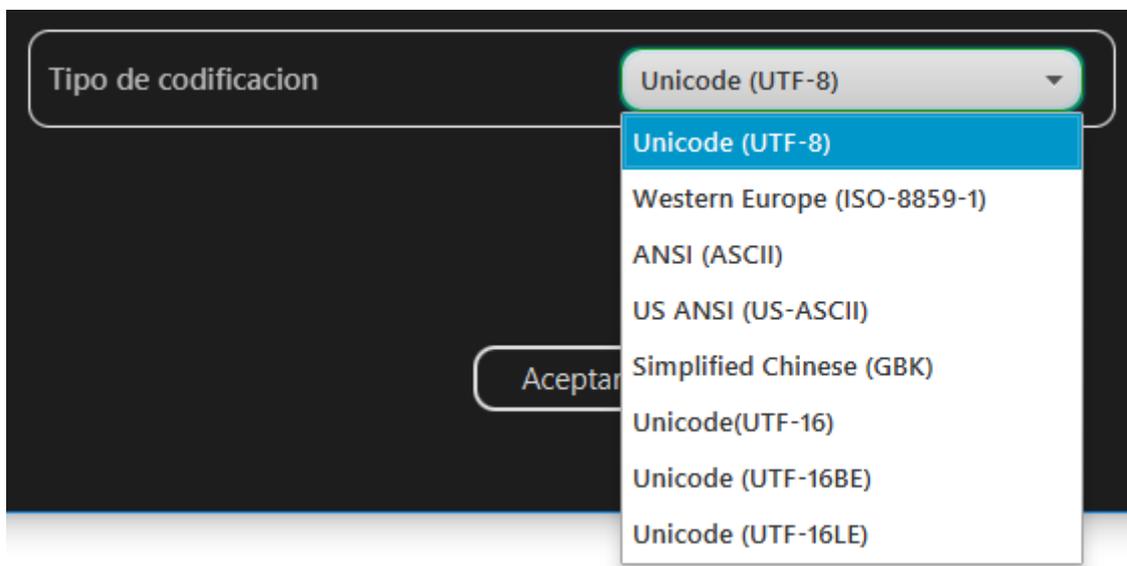


Figura 5.20. Tipos de codificación posibles

5.3.6.2. Ayuda

En el menú de ayuda tendríamos otras dos opciones, que serían “Tutorial” y “Acerca de”, como se puede observar en la Figura 5.21.

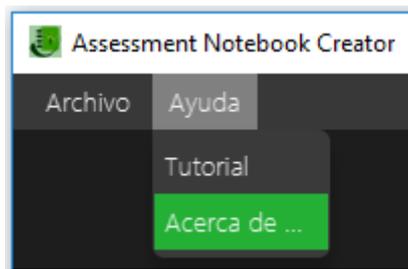


Figura 5.21. Menú de ayuda

En la opción de tutorial, se muestra una breve explicación de todas las funcionalidades de la aplicación (como se puede observar en la Figura 5.22) de una forma más breve que en este tutorial; mientras tanto, en la opción “Acerca de..” se muestra información relativa a la aplicación, como se ve en la Figura 5.23. En ambos casos, tras pulsar el botón de aceptar, volveríamos al menú en el que nos encontrásemos previamente, con los datos con los cuales estuviésemos trabajando (si procede).

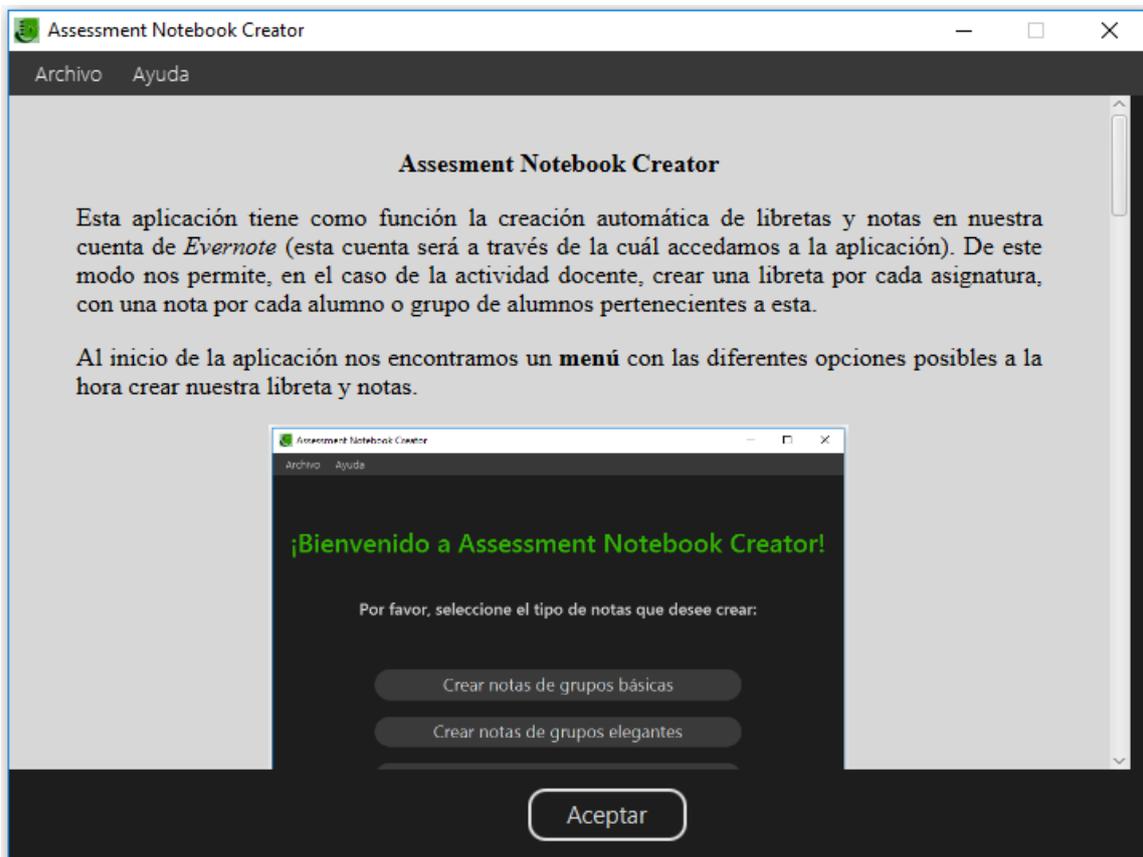


Figura 5.22. Tutorial de la aplicación

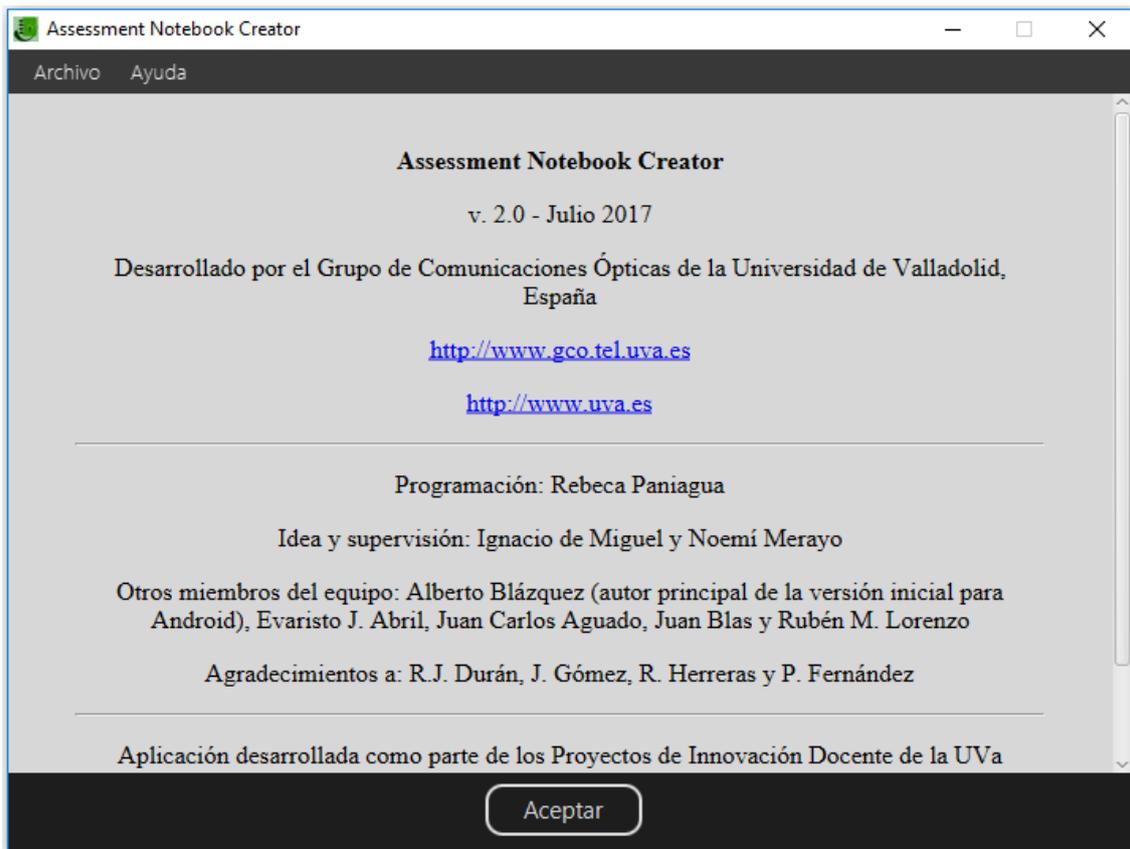


Figura 5.23. Acerca de la aplicación

5.3.7. Validación y progreso de creación de notas

A la hora de pulsar el botón de creación para cualquier tipo de notas que deseemos crear, se realizan una serie de comprobaciones para asegurar que todos los campos necesarios están completados y que cumplen las condiciones que deben, por ejemplo, que tienen el formato adecuado. Si algún campo obligatorio no está completo o no cumple las condiciones necesarias, aparecerá un mensaje de aviso que indique al usuario cuál es el problema, de modo que pueda solucionarlo. Por ejemplo, la Figura 5.24 muestra un mensaje que aparecería si el usuario intenta crear notas y el campo de nombre de la libreta está vacío.

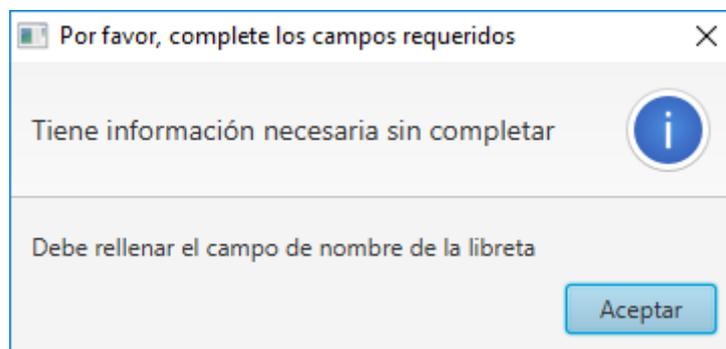


Figura 5.24. Error en la creación: Información necesaria sin completar

Otro ejemplo podría ser si un usuario intenta crear notas de estilo elegante si haber seleccionado ningún fichero de contenido ni haber escrito este directamente en el área de texto. En ese caso el usuario recibiría un mensaje como el mostrado en la Figura 5.25.

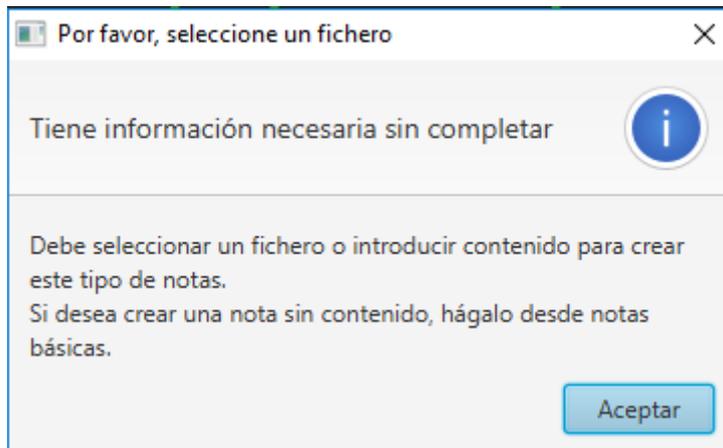


Figura 5.25. Error en la creación: Falta seleccionar un fichero de contenido o rellenar un contenido en área de texto

Si el usuario intenta crear notas sin conexión a internet, recibirá un mensaje de error como el que se muestra en la Figura 5.26.

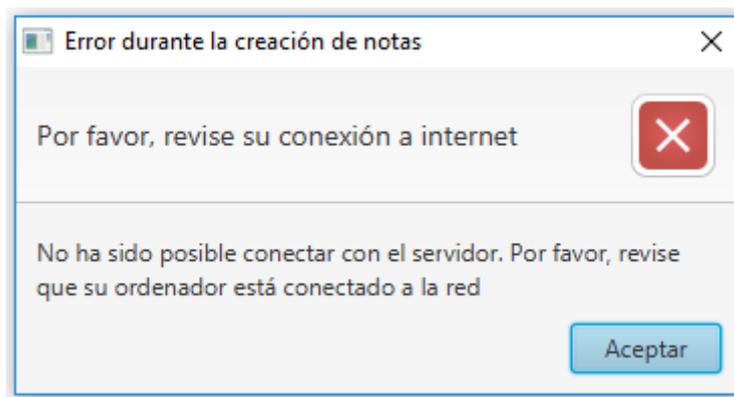


Figura 5.26. Error de conexión a internet

Además, el usuario tiene una limitación máxima de creación de 1000 notas, de forma que, si introduce un número superior a este, el número será automáticamente cambiado a 1000 y aparecerá por pantalla un mensaje como el de la Figura 5.27. A mayores, si el número de notas a crear supera las 200 notas, aparece un mensaje de confirmación preguntando si desea continuar ya que el número de notas es muy elevado, como vemos en la Figura 5.28. Si el usuario confirma que desea continuar, se crearán las notas solicitadas, y si cancela volverá a su menú de creación.

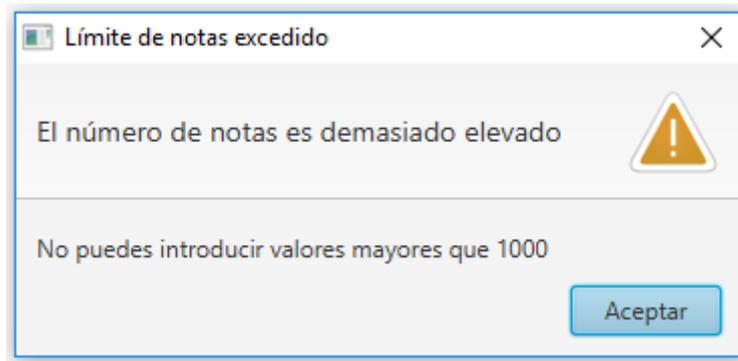


Figura 5.27. Advertencia límite de notas excedido

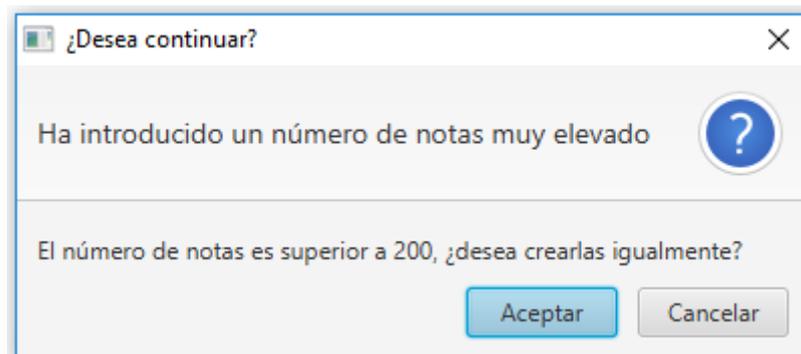


Figura 5.28. Ventana de confirmación: Número de notas muy elevado

En el caso de que la libreta que hayamos indicado ya exista, aparecerá un mensaje como el de la Figura 5.29, que nos indicará la situación. Si elegimos pulsar aceptar, las notas serán añadidas a la libreta existente, manteniendo intactas las que ya existiesen en ella. Si por el contrario pulsamos cancelar, no se creará ninguna nota nueva, por lo que la libreta se mantendrá con el estado que tuviera hasta el momento.

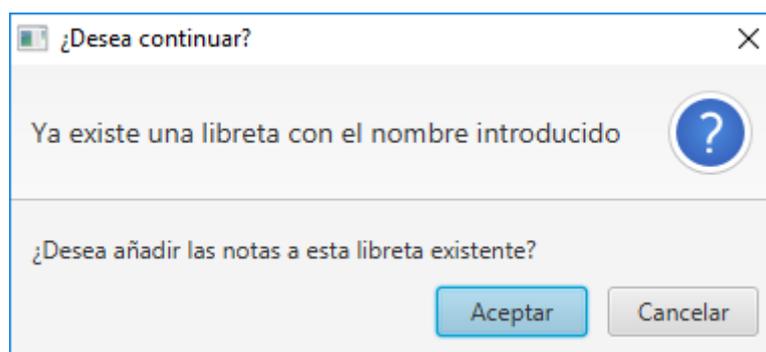


Figura 5.29. Mensaje sobre libreta ya existente.

Una vez que todo esté correcto, tras pulsar el botón de creación, aparecerá una ventana de progreso, que mientras se crea la libreta será como la de la Figura 5.30 y posteriormente esta mostrará el porcentaje de notas que han sido creadas hasta el momento, y el número de notas que se corresponde con ello, como se muestra en la Figura 5.31. Una vez finalizada la creación, la ventana de progreso se mostrará como en la Figura 5.32.

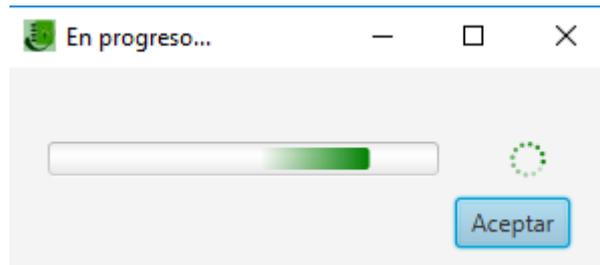


Figura 5.30. Progreso de creación de libreta

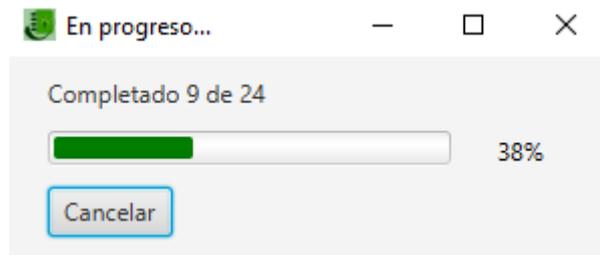


Figura 5.31. Progreso de creación de notas

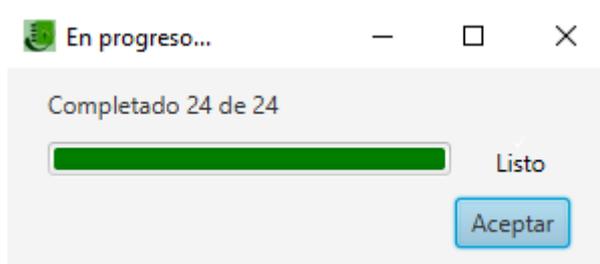


Figura 5.32. Creación de notas completada

Si durante el progreso de creación de notas, decidimos cancelarla, aparecerá un mensaje informativo que confirme que la creación ha sido cancelada. Además, como podemos ver en la Figura 5.33, se indicará el número de notas que ha sido creado exitosamente antes de la cancelación.

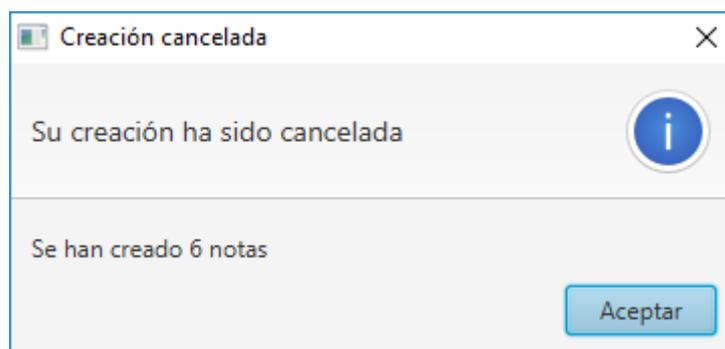


Figura 5.33. Mensaje tras cancelación de la creación de notas

6. Conclusiones y líneas futuras

6.1. Conclusiones

La aplicación desarrollada durante este Trabajo Fin de Máster tiene como función mejorar y facilitar los métodos docentes, de forma que permita que la tarea que los profesores desempeñan sea menos lenta y tediosa, mediante la automatización. Sin embargo, esta aplicación también puede ser utilizada en otros ámbitos en los cuáles puedan ser útiles las facilidades que proporciona.

Como se ha visto durante los anteriores capítulos de esta memoria, la función de la aplicación desarrollada consiste en automatizar la creación de libretas y notas en nuestro espacio de trabajo de *Evernote*. De esta forma, los profesores podrán crear libretas correspondientes a distintas asignaturas, con notas correspondientes a cada uno de los alumnos o grupos de alumnos que forman parte de ella, de forma automatizada y muy rápida. Para ello, únicamente deberán introducir el nombre de libreta deseado y los nombres de las notas a crear, o en su defecto, un fichero que contenga los nombres de las notas a crear. Además, las notas podrán tener un contenido predeterminado, mediante la carga de una plantilla que mediante código ENML defina este contenido. Debido a que el código ENML puede no resultar sencillo para escribir lo que se desee, se cuenta con un editor visual en el que se puede diseñar el contenido y obtener posteriormente el código correspondiente de forma automática, sin necesidad de que el usuario conozca cómo funciona este. Respecto a la aplicación, existía una versión menos avanzada desarrollada en Android por el alumno de la Universidad de Valladolid Alberto Blázquez Sánchez [15], pero para el desarrollo de esta nueva aplicación no ha sido apenas utilizada debido a que las APIs a utilizar eran diferentes (de modo que no se podían utilizar los mismos métodos) y se consideró más sencillo iniciar de cero. Como se ha mencionado, se desarrolló una aplicación similar para ordenador en lugar de para móvil o Tablet (ya que la creación de estas libretas y notas se hace de una vez y es más cómodo realizarlo desde un ordenador) y se añadieron nuevas funcionalidades sobre lo que tenía esta, como el editor visual para las plantillas.

Como resultado además se ha obtenido una interfaz sencilla, agradable visualmente y de fácil interacción con el usuario, de forma que cualquier persona puede ser capaz de utilizarla sin problemas.

El desarrollo de este proyecto me ha proporcionado la oportunidad de obtener nuevos conocimientos sobre Java, permitiendo ampliar los adquiridos durante los estudios; y en especial sobre la parte de desarrollo de interfaz gráfica de usuario, ya que era algo completamente nuevo. Así se han adquirido conocimientos de Swing y JavaFX. Además, se ha ejercitado la capacidad de resolución de problemas, buscando soluciones a los inconvenientes o dificultades que se planteaban en el proceso de desarrollo y se afrontado la responsabilidad de toma de decisiones sobre cuál es el modo más correcto para ello.

Este proyecto también puede animar a otros alumnos y profesores a plantear y desarrollar nuevas aplicaciones con funcionalidades que permitan integrar la tecnología en las aulas de forma que se agilice la forma de realizar alguna actividad, o se utilicen otras para que sea más ameno.

6.2. Líneas futuras

A raíz del desarrollo de esta aplicación, contamos con una serie de funcionalidades que cumplen el objetivo planteado desde el inicio. Sin embargo, siempre existe la posibilidad de mejorarla añadiendo nuevas funcionalidades o mejorando las ya existentes.

Posibles líneas futuras para esta aplicación serían:

- Incluir una opción de idioma que permita cambiar el idioma en el que trabajamos.
- Incluir adaptabilidad a la aplicación, de forma que permita redimensionar la ventana de trabajo y todos los elementos sean recolocados y redimensionados acordeamente.
- Añadir nuevas funcionalidades en el editor, como por ejemplo insertar tablas de las dimensiones deseadas.
- Mejorar la funcionalidad para añadir imágenes, permitiendo que se guarden las plantillas con las imágenes.

7. Referencias

- [1] Evernote Corporation, «Evernote,» [En línea]. Available: <https://evernote.com/intl/es/>. [Último acceso: Junio 2017].
- [2] Evernote Corporation, «ENML. Understanding the Evernote Markup Language,» [En línea]. Available: <https://dev.evernote.com/doc/articles/enml.php>. [Último acceso: Junio 2017].
- [3] Oracle, «JavaFX,» [En línea]. Available: docs.oracle.com/javafx/. [Último acceso: Junio 2017].
- [4] The Eclipse Foundation, «Eclipse,» [En línea]. Available: <https://eclipse.org/>. [Último acceso: Junio 2017].
- [5] Oracle, «The Java™ Tutorials - Lesson: Getting Started with Swing,» [En línea]. Available: <http://docs.oracle.com/javase/tutorial/uiswing/start/index.html>. [Último acceso: Junio 2017].
- [6] Oracle, «JavaFX Overview - What Is JavaFX?,» [En línea]. Available: <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>. [Último acceso: Junio 2017].
- [7] M. Jakob, «JavaFX 8 Tutorial - Part 1: Scene Builder,» Marzo 2015. [En línea]. Available: <http://code.makery.ch/library/javafx-8-tutorial/part1/>. [Último acceso: Junio 2017].
- [8] Oracle, «Java SE Development Kit 8 Downloads,» [En línea]. Available: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. [Último acceso: Junio 2017].
- [9] Evernote Corporation, «API Evernote,» [En línea]. Available: <http://dev.evernote.com/doc/reference/>. [Último acceso: Junio 2017].
- [10] MvnRepository, «Scribe OAuth Library » 1.3.7,» [En línea]. Available: <https://mvnrepository.com/artifact/org.scribe/scribe/1.3.7>. [Último acceso: Junio 2017].
- [11] Evernote Corporation, «Core Concepts - Evernote Cloud API Basics,» [En línea]. Available: https://dev.evernote.com/doc/articles/core_concepts.php#patterns. [Último acceso: Junio 2017].

- [12] Evernote Corporation, «Developer Tokens - Authenticating with the Evernote Cloud API using Dev Tokens,» [En línea]. Available: https://dev.evernote.com/doc/articles/dev_tokens.php. [Último acceso: Junio 2017].
- [13] Evernote Corporation, «Authentication - Authenticating with the Evernote Cloud API using OAuth,» [En línea]. Available: <https://dev.evernote.com/doc/articles/authentication.php>. [Último acceso: Junio 2017].
- [14] Evernote Corporation, «API Key Permissions,» [En línea]. Available: <http://dev.evernote.com/doc/articles/permissions.php>. [Último acceso: Junio 2017].
- [15] A. Blázquez Sánchez, «Empleo de tablets para evaluar la actividad de los alumnos en las sesiones presenciales: Desarrollo de una aplicación para Android. Trabajo Fin de Grado (Grado en Ingeniería de Tecnologías Específicas de Telecomunicación, mención en Ingeniería Telemática),» E.T.S.I. de Telecomunicación, Universidad de Valladolid, 2015.