



Desarrollo de software para la estandarización y gestión automatizada de datos de actividad biológica

**Adrián Pérez García**

Máster de bioinformàtica y bioestadística

Àrea 3

**Antolín Barrena Rico**

**Elisabeth Ortega Carrasco**

**15 de Diciembre de 2018**

[Llicència de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)  
Aquesta obra està subjecta a una llicència deReconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons

Table of Contents

[**1.** **Introducción del proyecto** 3](#_Toc532562551)

[**1.1** **Contexto y justificación del trabajo** 3](#_Toc532562552)

[**1.2** **Objetivos del proyecto** 4](#_Toc532562553)

[**1.3 Enfoque y método a seguir** 4](#_Toc532562554)

[**1.4 Planificación de trabajo** 10](#_Toc532562555)

[**1.5 Sumario de productos obtenidos** 14](#_Toc532562556)

[**1.6 Descripción del resto de capítulos de la memoria** 15](#_Toc532562557)

[**2.** **Automatización de la creación y carga de la base de datos Uniprot** 15](#_Toc532562558)

[**2.1 Campos Uniprot** 15](#_Toc532562559)

[**2.2 Estructura de la base de datos Uniprot** 18](#_Toc532562560)

[**2.3 Creación y carga de la base de datos Uniprot** 19](#_Toc532562561)

[**2.3.1 Estructura del programa** 20](#_Toc532562562)

[**2.3.2 Funcionalidad del programa** 20](#_Toc532562563)

[**3.** **Automatización de la extracción de resultados** 23](#_Toc532562564)

[**3.1 Proceso de estandarización y extracción de los resultados** 23](#_Toc532562565)

[**3.1.1 Estandarización de archivos según formato** 23](#_Toc532562566)

[**3.1.2 Búsqueda de patrones** 24](#_Toc532562567)

[**3.2 Funcionalidad del programa** 27](#_Toc532562568)

[ **Extracción de otros resultados (Pasos 9-12 de la figura 12)** 32](#_Toc532562569)

[**3.3 Análisis gráfico y estadístico del proceso de automatización** 35](#_Toc532562570)

[**4.** **Conclusiones** 41](#_Toc532562571)

[**4.1 Análisis de la planificación** 41](#_Toc532562572)

[**4.2** **Objetivos y alcance final del proyecto** 43](#_Toc532562573)

[**4.3** **Conclusiones generales** 45](#_Toc532562574)

[**4.4** **Líneas de trabajo futuras** 45](#_Toc532562575)

[**5.** **Bibliografía** 46](#_Toc532562576)

[**6.** **Glosario** 47](#_Toc532562577)

|  |  |
| --- | --- |
| **Título del trabajo:** | Estandarización de datos de perfil biológico a través de la creación de software |
| **Nombre del autor:** | *Adrián Pérez García* |
| **Nombre del consultor/a:** | *Antolín Barrena Rico, Elisabeth Ortega Carrasco* |
| **Nombre del PRA:** | *Ferran Padros Carrasco* |
| **Feche de entrega (mm/aaaa):** | *01/2019* |
| **Titulación o programa:** | *Máster de bioinformática y bioestadística* |
| **Área del Trabajo Final:** | *3* |
| **Idioma del trabajo:** | *Castellano* |
| **Palabras clave** | *Automatización, Predictive profiling, Molecular Informatics* |
| **Resumen del Trabajo (máximo 250 palabras):** | |
| Tanto la bioinformática como la bioestadística son cada vez más importantes y usuales en las empresas de los sectores de salud. Su utilización es útil para poder mejorar la calidad de funcionamiento de sistemas y para solventar nuevos retos. El trabajo es desarrollado en la empresa Almirall y se trabaja con archivos que contienen datos de perfil biológico que se reciben de empresas externas. Estos archivos tienen diferentes formatos y los datos que contienen son de diferente carácter. El proyecto se centra en estandarizar los documentos en un Excel final con el mismo formato para cada archivo además de crear bases de datos que contengan información relativa a estos documentos, para ayudar a explotar los datos de manera rápida e intuitiva. Para poder automatizar la estandarización de la extracción de los datos será necesaria la creación de un programa a través del lenguaje de programación VBA (Visual Basic Applications). Por otro lado, la creación de la base de datos también se implementará a través del mismo lenguaje, para organizar de forma ordenada toda la información correspondiente a las proteínas para ser cargados los datos. Se presenta en esta síntesis la explicación del contenido de los programas creados, así como varios análisis gráficos y estadísticos del proceso. El usuario que utilice el programa usará una herramienta mucho más sencilla y rápida para la extracción automática de los datos en los archivos y la búsqueda manual de los datos. | |
| **Abstract (in English, 250 words or less):** | |
| Both bioinformatics and biostatistics are becoming more important and usual in health related company. Its usage is useful to improve the quality and for implementing new features to the current existent systems as well as solve new challenges . The project it is developed in Almirall company where biologic profiling files can be found from external companies. Those files has several formats and the data from them are of different kind. The project focuses in standardize the documents in a final Excel form with the same format for each file. Moreover, databases will be created within information relating that files, in order to operate the data easily and quickly. For automatize the standardization of the data extraction, it is going to be necessary the creation of an application through programming language VBA (Visual Basic Applications). In other hand, the creation of database will also be done with the same language in pursuance of automatization of data as well. In this synthesis is reported the explanation of the program content as well as several graphical and statistical analysis of the process. The program user will be employing a simpler and quicker tool for the daily work than the manual search of the data. | |

# **Introducción del proyecto**

## **Contexto y justificación del trabajo**

El proyecto se centra en la gestión de datos del perfil biológico (“profiling” en inglés) de compuestos químicos de nueva síntesis utilizados en los programas de investigación de candidato a nuevos fármacos. El objeto de estudio es explotar toda la información proporcionada por los informes externos provenientes de diferentes empresas, en un formato ordenado, como puede ser una base de datos (bbdd), para extraer conclusiones útiles sobre los compuestos en investigación y descubrir así sobre qué proteínas pueden actuar. Esta información se utiliza para optimizar su actividad y seguridad de cara a su uso como posibles candidatos a ensayos clínicos.

La complejidad del proyecto radica en que estos archivos provienen de diferentes empresas, son de años muy dispares (2004-actualidad), y están en diferentes formatos no siendo fácil para su explotación. Por tanto es necesario una estrategia y un plan de acción para facilitar a los investigadores una manera ‘*user-friendly*’ de explotar los datos que puedan provenir de estudios externos para la toma de decisiones sobre los mismos en el futuro.

El tema escogido es debido al problema que supone, ya que es recurrente en empresas no solo del ámbito biomédico. El contratiempo se plantea cuando se reciben grandes cantidades de datos de empresas externas, o que se crean en la propia y que necesitan ser estudiados. El problema surge cuando los formatos y la estructura de los datos no están estandarizados y su explotación es compleja debido a estas dificultades. Dicho problema se puede resolver si se cumplen los objetivos del proyecto y por tanto mejorar la eficacia de investigadores y analistas y la eficiencia del proceso, entre otros.

En resumen, el objetivo de este proyecto será la **creación de software bioinformático** que permita la automatización y estandarización de la carga de archivos de **perfil biólogico (“profiling” biológico)** de una molécula, en concreto la inhibición de actividades de proteínas. Este software se creará a través de **Visual Basic Application** (VBAincluyendo también **la creación de bases de datos- a través de SQL Developer en un esquema de ORACLE,** para facilitar la explotación de los datos por el usuario.

## **Objetivos del proyecto**

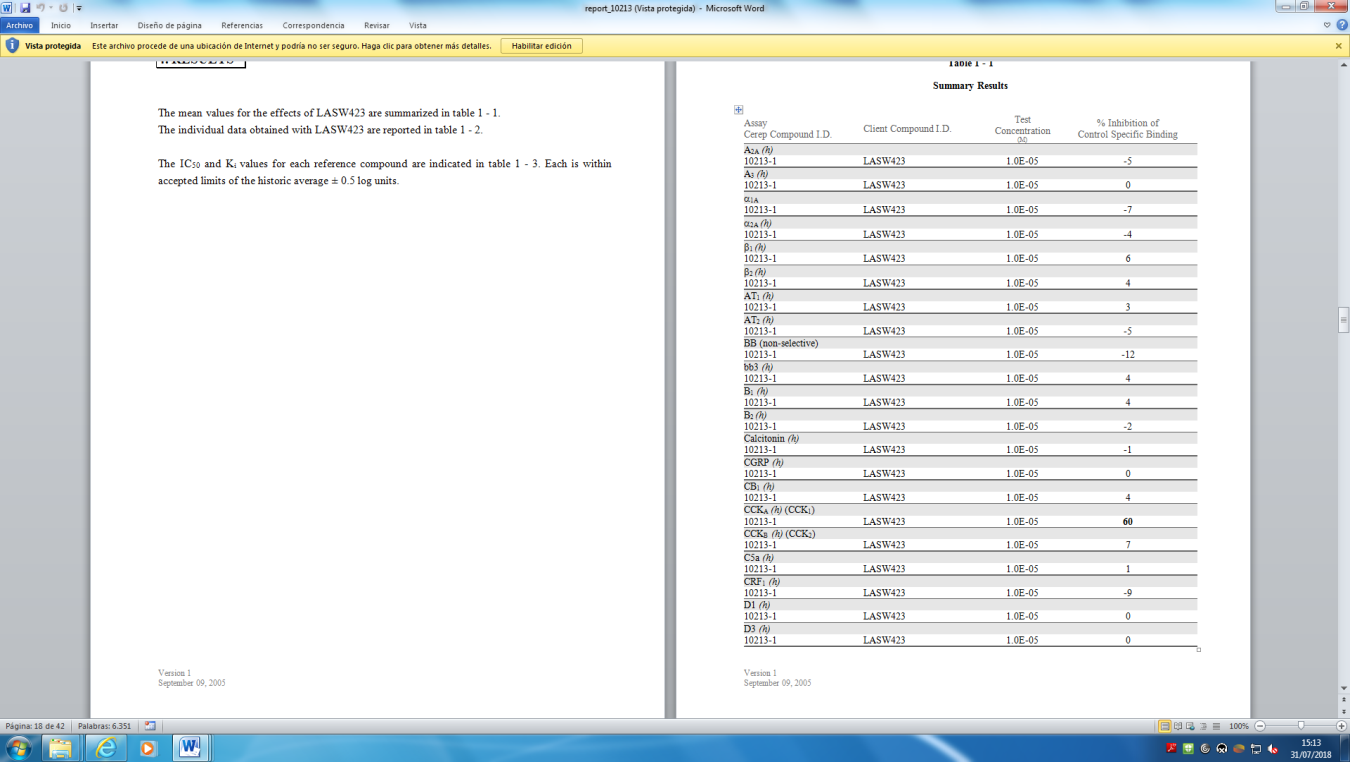
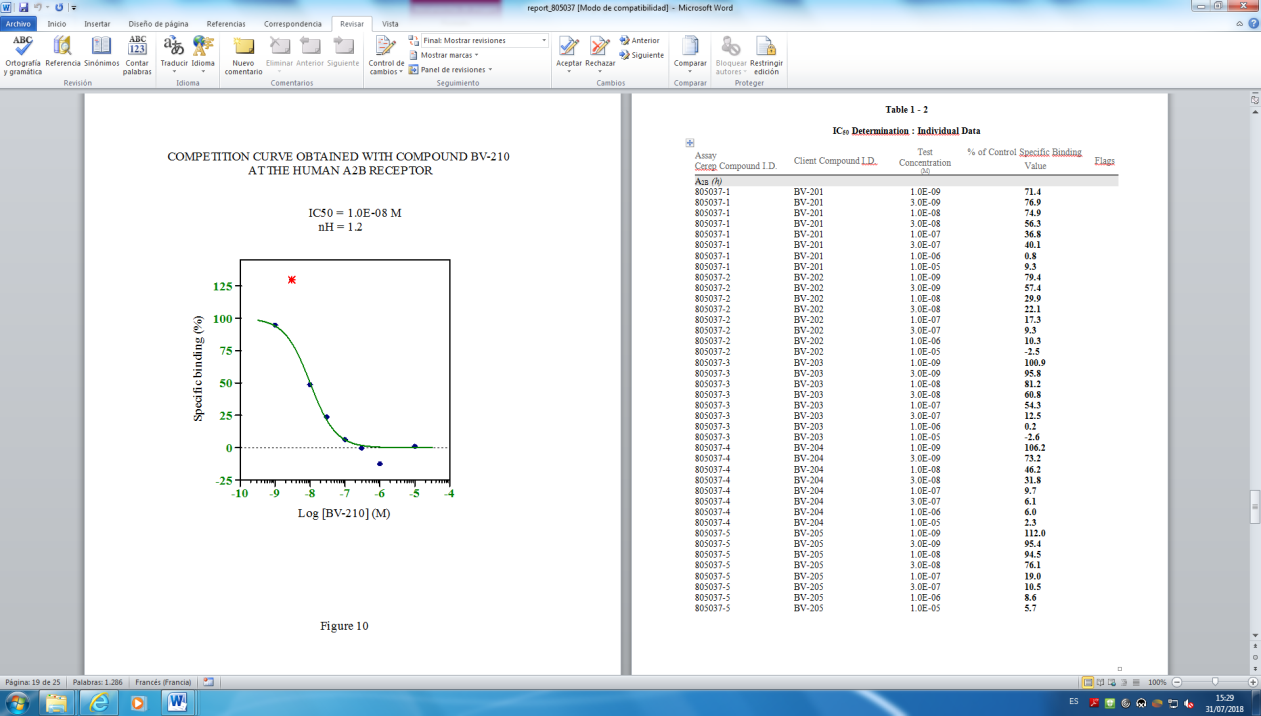
Los principales objetivos de este proyecto consisten en:

1. Estandarizar los archivos de datos de profiling biológico enviados por los diferentes proveedores con los diferentes formatos de archivos, creando un programa que automatice el proceso de estandarización y carga mediante la búsqueda de patrones.

2. Crear bases de datos que simplifiquen el trabajo de investigación y explotación de los datos.

3. Consolidar y ampliar los conocimientos estudiados y aprendidos en el máster de Bioinformática y Bioestadística.

## **1.3 Enfoque y método a seguir**



**Figura 1**. Archivo tipo dosis **Figura 2**. Archivo tipo curva

Se ha solicitado ciertos estudios provenientes de empresas externas, que están almacenados en varios formatos tales como Microsoft Word, Microsoft Excel, y PDF. Estos archivos contienen dichos datos de perfil biológico donde se analiza la actividad de diferentes compuestos sintetizados internamente sobre diferentes proteínas. El objetivo es, por un lado, determinar el tanto por ciento de activación/inhibición sobre dichas proteínas por efecto de los compuestos ensayados a una sola dosis empleada (ver Figura 1); y por otro se determina la concentración capaz de inhibir/estimular el 50% de la actividad de la proteína (IC50/EC50) a diferentes concentraciones del compuesto obteniendo una curva para la misma proteína. (ver Figura 2)

El objeto de estudio es explotar toda la información dada por los informes en un formato ordenado, como puede ser una base de datos (bbdd), para extraer conclusiones útiles sobre los compuestos y descubrir así sobre qué proteínas pueden actuar y en un futuro utilizarlos para mejorar la calidad de vida de las personas aliviando los síntomas o incluso curando enfermedades.

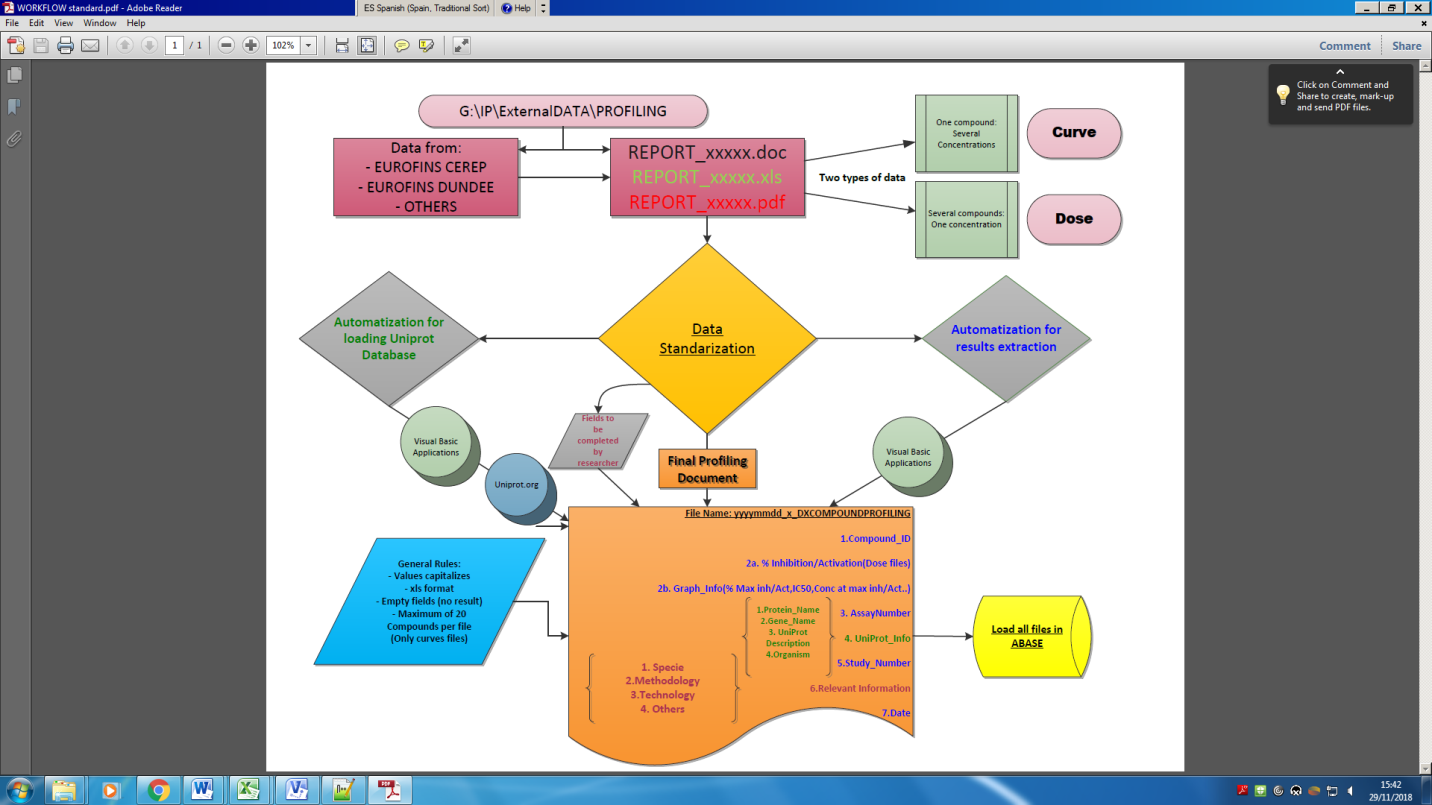
Como se ha comentado en los objetivos del proyecto, se busca estandarizar los datos siguiendo la estrategia de creación de software adaptado a las necesidades específicas de la empresa Almirall y sus usuarios, como la manera más eficaz y eficiente de obtener los resultados relevantes. Automatizar tanto los archivos que ya estén recogidos como para los archivos futuros será necesario y útil para facilitar la explotación de datos a los investigadores. Al haber varios tipos de archivos y diferentes formatos, podrá variar el código utilizado.

Otra de las estrategias que podría haberse utilizado, es estandarizar los documentos manualmente, pero el tiempo de ejecución no sería menor, además de no facilitar la estandarización de los documentos de nueva creación que puedan haber en un futuro.

Las características generales que proporcionará para el programa son las siguientes:

* Versatilidad a la hora de decidir qué tipo de archivo o formato queremos estandarizar.
* Automatización de todos los subprocesos compartidos a lo largo del proyecto, desde la lectura del archivo a la creación del Excel con los datos correctos.
* Optimización en el proceso de extracción de los resultados.
* Aplicación “*user-friendly*” para la toma de decisiones y la explotación de los datos.

En la Figura 3, se muestra un esquema del proceso general del proyecto. La primera parte se ha comentado en el apartado anterior (apartado 1.3), la parte central forma los dos grandes procesos que se han de automatizar haciendo diferentes programas: la extracción de los resultados del proteínas y la carga de la base de datos relativa a la información que proviene de Uniprot. Ésta es el eje central donde de desarrollo de software y del aprendizaje del lenguaje de código para que se lleve a cabo. También será explicada en detalle en los apartados correspondientes al desarrollo de la automatización.



**Figura 3**. Esquema general del proyecto de estandarización.

En cuanto a la estrategia de estandarización, el documento final donde se hallarán todos los datos extraídos de los “*reports*” tendrá la siguiente estructura:

****

**Figura 4.** Formato tipo del documento final

|  |  |
| --- | --- |
| **Object Type** | **Description** |
| O | Object |
| R | Result |
| C | Condition |
| G | Graph |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Order by** | **Object Type** | **Field** | **Units** | **Description** |
| **1** | **O** | **COMPOUND\_ID** |  | ***Code and batch of test compound*, e.g. LASW1234:1** |
| **2** | **R** | **%INHIBITION** | **%** | **%Inhibition** |
| **3** | **R** | **%ACTIVATION** | **%** | **%Activation (positive or negative number)** |
| **4** | **R** | **ADDITIONAL\_INFO** |  | **Any comment. Further information** |
| **5** | **C** | **DOSE** | **M** | **Number of dose** |
| **6** | **C** | **ASSAY\_REF\_NUMBER** |  | ***Catalogue reference number of the assay given by the provider*** |
| 7 | C | UNIPROT\_CODE |  | Page [*www.uniprot.org*](http://www.uniprot.org)  Field: Entry |
| 8 | C | UNITPROT\_DESCRIPTION |  | *Page* [*www.uniprot.org*](http://www.uniprot.org)  *Field: Entry name* |
| 9 | C | PROTEIN\_NAME |  | *Page* [*www.uniprot.org*](http://www.uniprot.org)  *Field: Protein names* |
| 10 | C | GENE\_NAME |  | *Page* [*www.uniprot.org*](http://www.uniprot.org)  *Field: Gene names* |
| 11 | C | ORGANISM |  | *Page* [*www.uniprot.org*](http://www.uniprot.org)  *Field: Organism* |
| 12 | C | SPECIES |  | *Species* |
| 13 |  | SYSTEM |  | *NATIVE, RECOMBINANT (STABLE, TRANSIENT),…* |
| 14 | C | BIOLOGICAL\_MATRIX |  | *CELL LINE, CELL TYPE,…* |
| 15 | C | STIMULUS |  | *STIMULUS 1,2,…* |
| 16 | C | EFFECT |  | *BINDING, AGONISM, INVERSE AGONISM, ANTAGONISM, CATALYTIC,…* |
| 17 | C | METHODOLOGY |  | *cAMP, CALCIUM FLUX, IP1, PHOSPHORYLATION,VOLTAGE CLAMP…* |
| 18 | C | TECHNOLOGY |  | *FLUORESCENCE, HTRF, LUMINESCENCE, RADIOACTIVITY, IW4…* |
| 19 | C | STATE |  | *CLOSED, OPEN, INACTIVATED, USE DEPENDENT (Frequency in Hz)* |
| **20** | **C** | **COMPANY** |  | ***CEREP*** |
| **21** | **C** | **STUDY\_NUMBER** |  | ***Name of study given by the provider*** |
| **22** | **C** | **COMPLETION\_DATE** |  | ***Date of experiment completion***  **e.g. 31/12/2016 (dd/mm/yyyy)** |

Como se puede apreciar, hay varios colores que forman los campos del archivo. Esto es debido a las diversas fuentes de donde se obtendrá la información requerida para la carga de este archivo.

Los campos en azul indican que se extrae la información de los propios ‘*reports*’: el ID del compuesto, el % de inhibición/activación, información del gráfico (en el caso de que lo contenga) el número del estudio, la fecha y el número de asociación con la empresa encargada del estudio. En el caso que el campo no contenga resultado alguno entonces el campo se dejará vacío. El campo se dejará vacío sólo si los resultados (R) no contiene valor. En el caso de las condiciones, si el campo no está rellenado entonces contendrá NA.

Los campos en rojo son datos que el investigador en cuestión podrá rellenar cuando sea necesario, ya que pueden ayudar a detallar el documento. No apareciendo en los archivos, aun así será útil y el campo contendrá el valor NA (No Aplica).

Los campos en verde, son los datos proveniente de UNIPROT. Esta página (www.uniprot.org) contiene los nombres de prácticamente todas las proteínas que aparecen en los documentos. A través del nombre de la proteína, se encuentra el nombre del gen del cual proviene, la especie, una descripción detallada y un código (uniprot) que es un código de referencia aceptado en el ámbito de biología y que por tanto le da calidad objetiva a los resultados.

Te servirá de ayuda para detallar las características del archivo:

Además de contener estos campos, también tiene una serie de estandarizaciones de formato como:

* El formato de la fecha ha de ser siempre DD/MM/AAAA (COMPLETION\_DATE)
* El nombre del archivo siempre contendrá la fecha de realización del report, un número indicando si es el primer archivo de ese report (generalmente habrá 1), a continuación las 2 primeras letras corresponderá al alias del proveedor “ES” y por último la descripción del nombre del ensayo “COMPOUNDPROFILING”.

Ej: 20080610\_1\_ESCOMPOUNDPROFILING.xls

* Los resultados siempre estarán en mayúsculas.
* El formato del archivo siempre será “.xls”.
* En los campos sin resultado, éstos se dejarán vacíos.
* Para no cargar demasiado el archivo, si en los archivos de curva hay más de 20 compuestos, se deberá hacer otro archivo xls para el mismo “*report*”.
* Si hay más de 20 compuestos, se harán archivos extra indicando los compues que hay en cada archivo, manteniendo el documento origen.

Para empezar con la creación de este programa, se ha de buscar un lenguaje de programación apropiado para dicho objetivo. Hay varios lenguajes de código que permiten la creación de dicho programa, pero hay que valorar el uso libre de Software, así como la dificultad al utilizarlo. Por tanto, uno de ellos, de libre uso y sin demasiada dificultad para su aprendizaje es **Visual Basic for Applications (VBA).** Es el [lenguaje](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de [macros](https://es.wikipedia.org/wiki/Macro) de [Microsoft](https://es.wikipedia.org/wiki/Microsoft) [Visual Basic](https://es.wikipedia.org/wiki/Visual_Basic) (VBA) que se utiliza para programar aplicaciones [Windows](https://es.wikipedia.org/wiki/Windows) y que se incluye en varias aplicaciones Microsoft (Flavio Morgado***.*** 2006)

En cuanto a la automatización de la carga de la base de datos con la información proveniente Uniprot, también se ha creado un programa similar, siguiendo las mismas pautas y características que se buscan para el programa de automatización de extracción de resultados.

## **1.4 Planificación de trabajo**

La planificación se ha separado en diferentes etapas, según los objetivos: la introducción, los dos programas de automatización, y la documentación escrita.

Antes de comenzar el proyecto, se pasa por un proceso de adaptación, introducción y formación en materia de las herramientas que fueron útiles más adelante. Por eso, el primer mes es una etapa de asentar las bases y aprender programación, sobre todo en Visual Basic Script (VBS) y en bases de datos. Lo primero es la formación con manuales y clases sobre los programas, además de consolidar lo aprendido con ejemplos y dudas que puedan surgir. Para abordar el proyecto es importante entender el alcance del proyecto, comprender cuál es el objetivo real para generar ideas que permitan progresar en el proyecto y comprender cuáles pueden ser las necesidades y problemáticas que puedan surgir y cómo resolverlas.

También se configura un diagrama de Gantt con las tareas que se han planificado y se van haciendo día a día, lo que permite tener un seguimiento del tiempo real empleado en cada una de ellas. Se traspasa al documento las horas empleadas, en un calendario que actualiza las horas totales para cada tarea, estando éstas en subapartados con los objetivos cortos dentro del proyecto. También es útil tener las horas teóricas que previamente se ha planificado para realizar la previsión del proyecto, y asignar a cada tarea una prioridad (niveles 1-5) gestionando el cumplimiento de cada una de las tareas en el plazo que se ha previsto detectando pequeñas desviaciones una vez que se va avanzado en el proyecto. A su vez, se implementa un gráfico que indica el tiempo estimado en días reales y teóricos, donde se puede ver la diferencia entre ambos, y ver las tareas que se han podido hacer de manera paralela. La importancia de tener un registro de días y otro de horas es debida a que puede haber tareas que se hayan parado durante unos días y luego se hayan vuelto a retomar, por lo que el cálculo de días no será correcto, pero en cambio sí el de las horas empleadas a dichas tareas.

|  |
| --- |
| **Introducción 95h.(total)** |
| Instalación de sistemas 2 h. |
| Diagrama de Gantt 7 h. |
| Presentación 1 h. |
| Meetings de seguimiento de proyecto 30 h. |
| Buscar cómo leer archivos VBScript (pasar a txt) 7 h. |
| Hacer lista con contenido de archivos 2 h. |
| Formación práctica y teórica en VBScript y VBA 27 h. |
| Formación práctica y teórica en BBDD 11 h. |
| Brainstorming para tablas de BBDD 8 h. |

* En el subapartado de ‘Meetings de seguimiento de proyecto’ se tienen en cuenta además de las horas de reunión, las horas dedicas a la preparación de las mismas, vía powerpoint, guión para las intervenciones y repaso de las herramientas que se utilizarán en dichas reuniones.

Por otro lado, como se puede ver en las siguientes tablas, se trabajó en la extracción de resultados, para empezar a resolver cómo se podía automatizar este proceso.

|  |
| --- |
| **Uniprot: como buscar y extraer nombres\_prot en los reports 235h. (total)** |
| Insertar los nombres de las proteínas manualmente 7 h. |
| Automatizar la extracción los nombres de las proteínas VBScript 86 h. |
| Mejorar programa de patrones para las proteínas 12 h. |
| Mejorar sistema de patrones proteínas VBA 59 h. |
| Hacer Excel con variables de los patrones 2 h. |
| Pasar de Word a Excel (unicode) 66 h. |
| Pasar de Pdf/Excel a Excel (Estándar) 3 h. |

|  |
| --- |
| **Extraer el resto de información de los ‘*reports*’ 70 h. (total)** |
| Hacer código VBA para archivos tipo curva y dosis 47 h. |
| Separar la recogida de proteínas para recoger el patrón óptimo 3 h. |
| Separar los patrones de información por grupos 8 h. |
| Poner nombre del Excel final en archivo ‘Summary’ 3 h. |
| Documentar, comentar y repasar código 9 h. |

En este caso, el tiempo para la extracción del nombre, fue mayor del previsto debido a las dificultades que iban sucediéndose en este proceso además del tiempo de adaptación a la programación práctica real. Pero en cuanto a la extracción del resto de información al haber aprendido a gestionar los errores que podían darse, y estar adaptado tanto al lenguaje de programación como al programa en sí, se pudo acabar con el apartado en menos de lo previsto.

Lo siguiente, es ver cómo se han completado los objetivos relacionados con la creación de bases de datos.

Este objetivo, junto con las tareas subordinadas, se ha llevado a cabo añadiendo las mejoras que se han añadido después de la organización no planteadas en el plan de trabajo, además de las mejoras que se han añadido después de la organización.

|  |
| --- |
| **Bases de datos**  **300 h. (total)** |
| Crear base de datos de Prot-Info UNIPROT 25 h. |
| Carga de información y llenado de campos 120 h. |
| Gestión de errores de código 115 h.  Carga de información UNIPROT en Excel final 40h. |
|  |
|  |

Por las actividades previstas que no se han realizado debido al alcance del proyecto, como la automatización de otros formatos o de los documentos tipo curva, se dedicaron a este objetivo. De todas maneras, la mejora del proceso y la carga de información en el Excel final (así como la gestión de errores de código) también implicó horas reflejadas en la tabla.

|  |  |
| --- | --- |
| Documentación de procesos conseguidos | **100h.(total)** |

También hay varias funcionalidades del programa que se han desarrollado a medida que ha surgido nuevos retos a la hora de solventar problemas adoptando un mantenimiento preventivo en la creación del software para alargar su ciclo de vida:

* Creación de un programa de patrones que permita añadir patrones para futuros archivos de una manera sencilla para el usuario
* Bloquear ciertas celdas del programa de patrones que no deben ser cambiadas para evitar fallos en la búsqueda de información
* Al encontrar dos formas diferentes de recoger el nombre de las proteínas de manera igual de óptima, la segunda función utiliza la anterior para saber si esa celda contiene el nombre de una proteína
* Recoger en un archivo de texto diferentes los compuestos LAS que formen parte de cada ‘report’.
* Mejora del gestor de patrones para que el usuario pueda cambiar el origen/destino de varios archivos utilizados/creados a lo largo del proceso, así como la elección del usuario que se conecta a la base de datos.
* Creación y mejora del sistema de análisis de velocidad del proceso por cada archivo, de manera total y parcial por fases.
* Mejora del sistema de gestión de errores y aplicación del mismo en la fase de creación de la base de datos.
* Optimización de procesos para mejorar la velocidad del proceso de automatización.

El primero de ellos, es cambiar el formato de los archivos en PDF/Excel a Excel. Los archivos en PDF están protegidos y son sólo de lectura, no dejando exportar la información desde el PDF al EXCEL.

El segundo de ellos, es extraer la información de los archivos de carácter curva. La dificultad radica en extraer los gráficos de estos archivos y mediante un OCR detectar los gráficos a qué compuestos pertenecen para reflejarlos en el documento estandarizado final . También hay que tener en cuenta que los patrones utilizados para extraer los resultados no es compatible con este tipo de archivos, por lo que habría que pensar en configurar otro patrón diferente.

Por tanto, todas las horas dedicas a este subproceso forman parte de la dedicación para los archivos dosis. El resto de tareas sí se pudieron completar.

Como se ha comentado anteriormente, en la fase de extraer el nombre de los ‘reports’ hubo complicaciones debido a nombres con letras griegas, además de ver que algunos archivos seguían un patrón diferente a la hora de buscar el nombre de las proteínas. Eso supuso una dedicación de tiempo mayor, pero se puedo controlar mediante cambios en el código principal para que un cambio de configuración en el patrón detectara el nombre de las proteínas. En cuanto al tiempo, a la hora de extraer otros resultados supuso menor dedicación de tiempo, por lo que se compensó con la fase anterior.

Se ha trabajado de manera planificada en cuanto al objetivo relativo a las bases de datos, pero no en cuanto a las horas debido a la preparación de la lista de proteínas y su corrección por parte de los expertos antes de cargar toda esta información en la base de datos.

En el siguiente apartado también se tiene en cuenta las tareas que se han realizado y las horas dedicadas para detectar cuáles son los cuellos de botella del programa..

Tareas no programadas que se realizarán:

**Gestión de errores de código: 40h +10h PEC2**

**Optimización del código: 42h**

* Comparar la segunda recogida de proteínas con la primera para optimizar el proceso: 8h
* Mejora y creación de nuevas opciones para el usuario del gestor de patrones: 10h
* Creación del archivo Excel relación LASW-Report: 2h
* Creación de las funciones de optimización del proceso de automatización: 2h
* Mejora del análisis gráfico y estadístico del proceso de automatización: 20h

Estas horas se detallarán en la parte de actividades programadas que no están en el alcance de este proyecto como la automatización de archivos curva o de formato Excel.

## **1.5 Sumario de productos obtenidos**

Los productos obtenidos para resolver el alcance del proyecto son los siguientes:

* Archivo Macro/VBA para la automatización de la extracción de resultados.
* Archivo Macro/VBA para la automatización de la carga de la base de datos Uniprot.
* Diagrama de Gantt con el Plan de trabajo actualizado
* Análisis gráfico/estadístico del proceso de automatización de extracción de resultados

## **1.6 Descripción del resto de capítulos de la memoria**

El capítulo 2 contempla en detalle el proceso de la extracción de resultados y las tareas y procesos subyacentes relativo al mismo. Se desarrolla el funcionamiento del programa, las problemáticas y las soluciones que se han encontrado para resolverlas. También se detalla el análisis gráfico y estadístico de la velocidad del procesamiento, con el objetivo de analizar las funciones que más tiempo ocupan en general.

En el capítulo 3, se explica el proceso de creación y carga de la base de datos Uniprot. Se comentan todos los campos procedentes de Uniprot necesarios para el Excel final estandarizado , la estructura de la base de datos y el desarrollo del código en detalle.

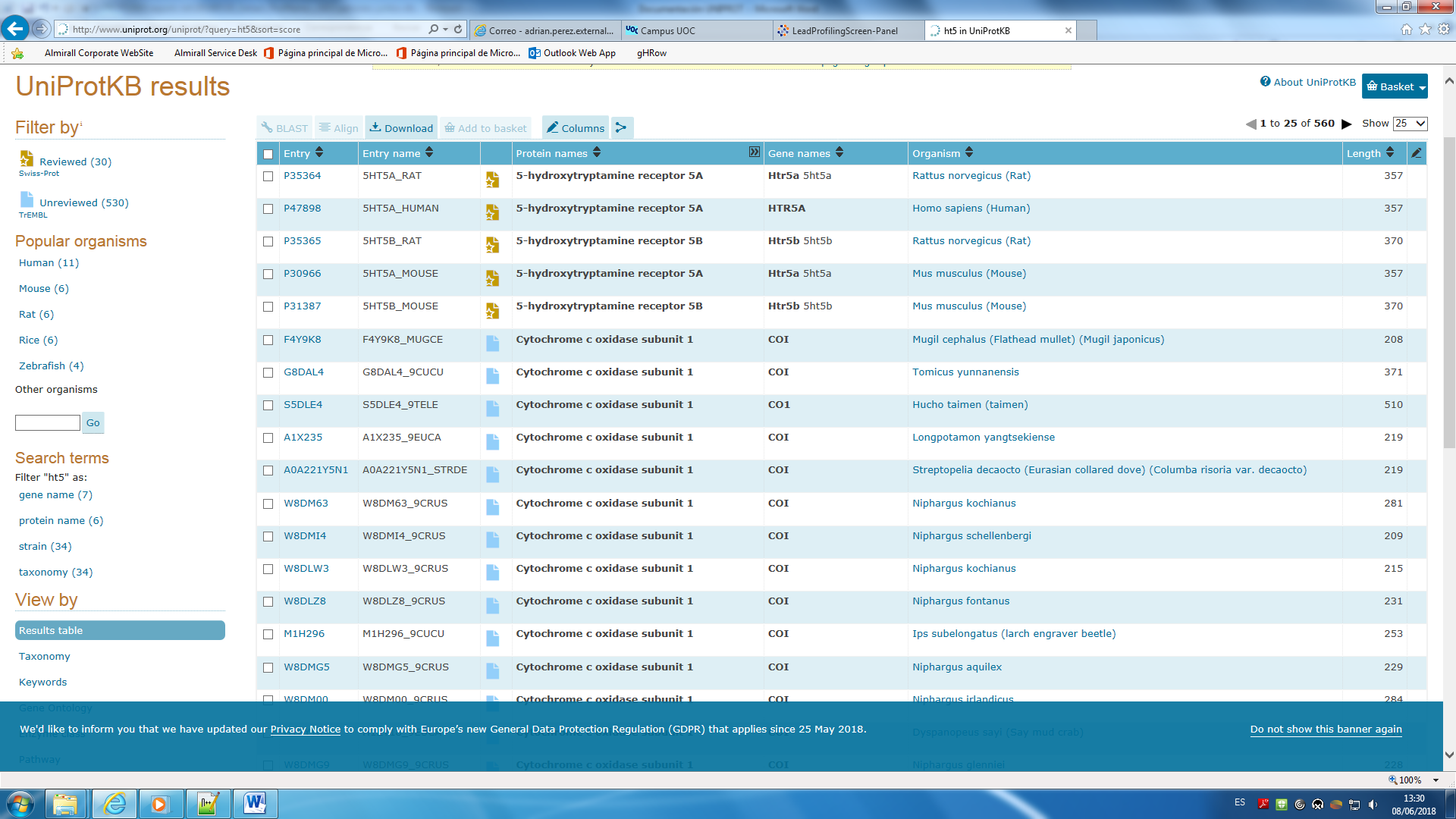
Por último, se detallarán las conclusiones, el glosario, los anexos y la bibliografía.

# **Automatización de la creación y carga de la base de datos Uniprot**

## **2.1 Campos Uniprot**

En este apartado se detalla los campos que se substraerán de la página web [www.uniprot.org](http://www.uniprot.org) con el objetivo de tener información más precisa de las proteínas que aparezcan en el estudio. Además cuentan con el código Uniprot, que es universal y, por tanto, aunque cambie el nombre de ésta siempre tendrá el mismo código.

A través del nombre de la proteína, se podrá consultar toda la información relativa de la misma en la página web, y cargarla en el documento final. Cuando se encuentra la proteína, Uniprot proporciona todos los campos requeridos, además de distintos datos interesantes como la localización subcelular, otros nombres de la proteína y los procesos biológicos que puede llevar a cabo, entre otros.



**Figura 5.** Ejemplo de búsqueda de proteínas en www.uniprot.org

La estrategia es crear una base de datos a modo de diccionario con toda la información detallada de todos estos campos para compararla con el nombre de las proteínas recogidas previamente de los documentos enviados por los proveedores, debido a que la página web como se ha analizado proporciona más información que la que reflejarán los documentos finales de cada documento. Con esta base de datos, si el investigador consulta la BD principal requiere una información más precisa de las proteínas podrá consultar el diccionario de manera sencilla y obtener la información requerida, además de poder conectar con la página web original por si se precisa.

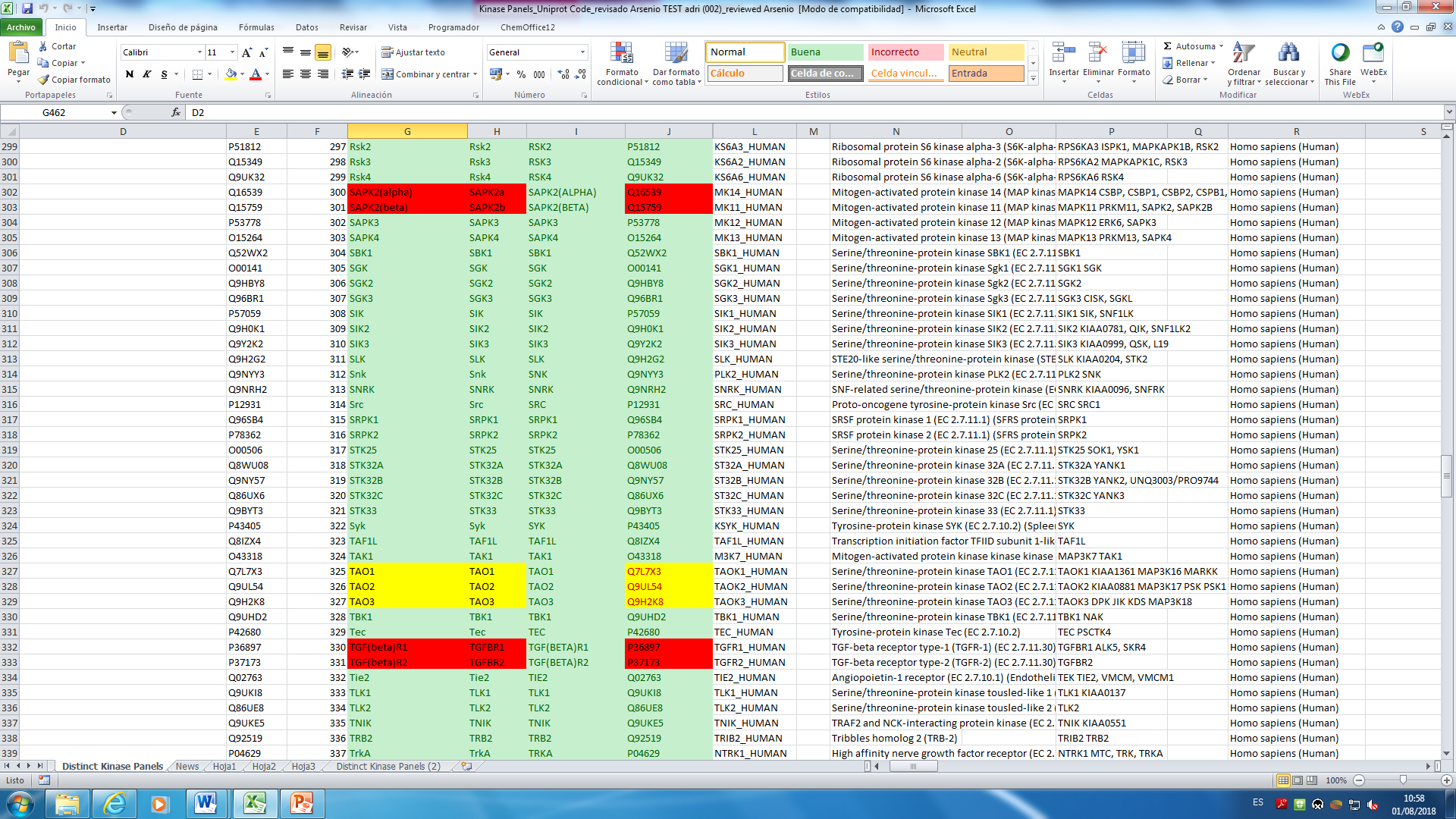
Pero después de ver y estudiar el objetivo, se ha de afrontar las diferentes problemáticas y ciertos aspectos importantes a la hora de construir la base de datos diccionario.

La información relacionada con el nombre de la proteína no siempre es fácil de encontrar , ya que la web de Uniprot no sólo muestra un resultado, debido a varias razones:

* Puede haber diferentes proteínas que sean unidades catalíticas de otras proteínas mayores y tengan diferente Uniprot, pero un nombre muy similar.
* Hay que tener en cuenta las especies de las que proviene la proteína ya que la página puede mostrar las de humano, ratón, rata, entre otras.
* El nombre de la proteína que se ha extraído de los ‘reports’ es menos específico que los resultados de Uniprot (Ejemplo: Glutamate Receptor, y en Uniprot está el Glutamate Receptor 1A, 1B,2A…).
* Hay proteínas que son idénticas salvo por el tipo de ensayo utilizado (Ligando agonista o antagonista) que tendrán el mismo Uniprot pero es importante reflejar esa separación cuando se cree la base de datos.
* Se pueden tener dudas debido a nombres parecidos o de otro tipo.

La solución debe darse a través de ciertos pasos: Se creará un Excel previo a la base de datos con toda la información necesaria que ésta contendrá. Es necesario ya que también se añadirá la información que no sea del todo certera. Para que las proteínas con información dudosa sean las menores posibles, en los documentos hay información que es de ayuda: en todas las proteínas cita la especie de la que proviene, por tanto, se puede filtrar la proteína por especie. También contiene un campo llamado “Reference Compound” que se puede añadir a la búsqueda y así disminuir la cantidad de resultados. Además, Uniprot contiene un campo que indica si los resultados están revisados por la propia página, entonces también se filtrará por dicho campo ya que si no están revisadas puede dar lugar a fallos.

Como se aprecia en la Figura 6 , las proteínas tendrán una clasificación por colores:

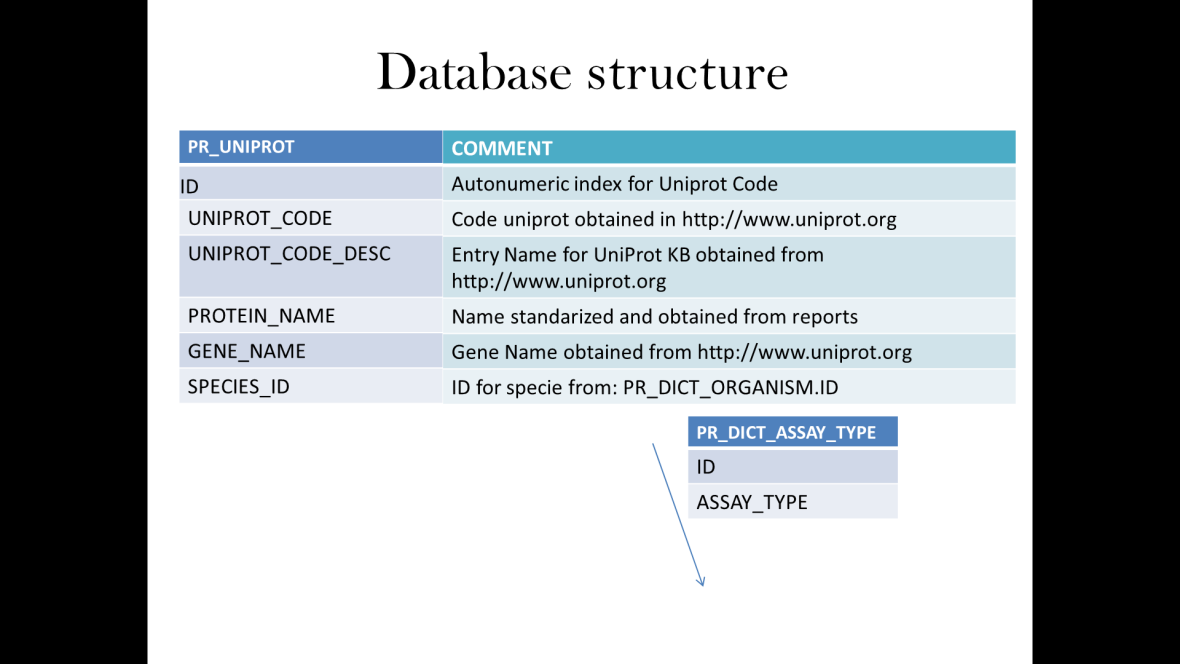
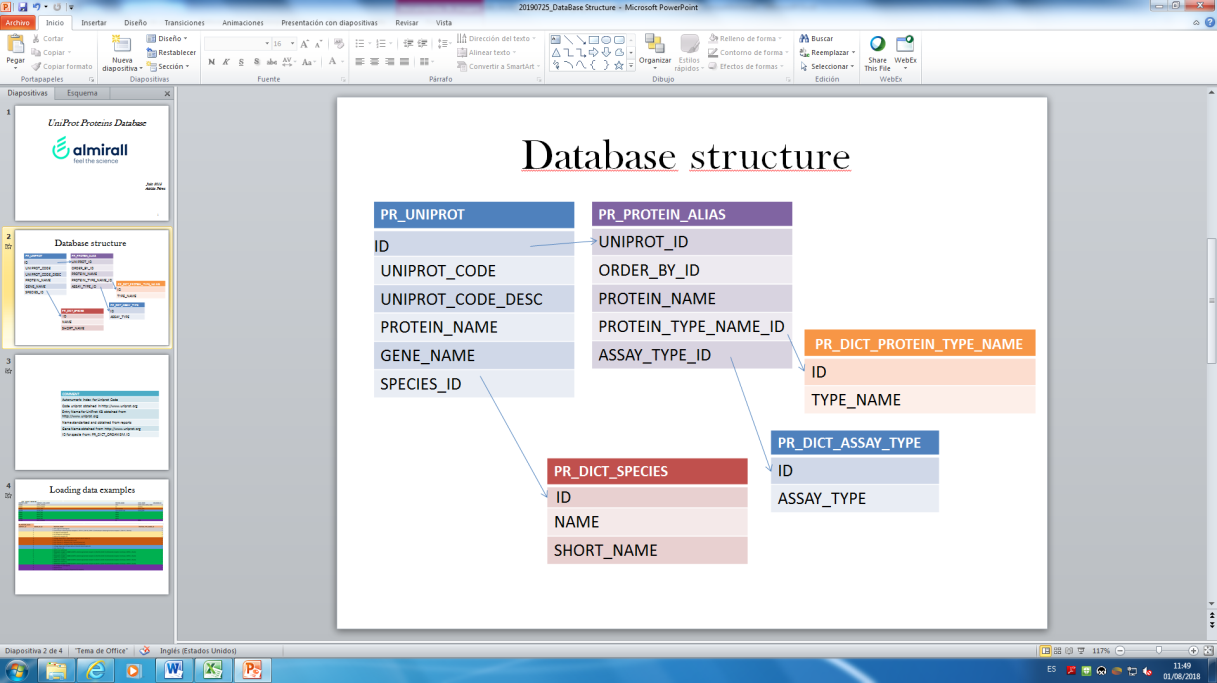
* Verde: la información que es correcta
* Amarillo: la información de la que no está clara que pertenezca a esa proteína, y que por tanto habrá que consultarlo con las personas de referencia
* Rojo: las proteínas de la que no se haya encontrado información o sea muy susceptible de ser errónea.

**Figura 6.** Ejemplo de información de proteínas en el Excel clasificado por colores

Seguidamente, con este Excel creado, se contacta con dichas personas que pueden ayudar a discernir sobre las proteínas correctas y revisar los datos, y por tanto disipar dudas sobre las mismas. Una vez se haya revisado este Excel, ya se podrá cargar con dicha información la base de datos.

## **2.2 Estructura de la base de datos Uniprot**

La base de datos que se creará tendrá varias tablas contenidas con los diccionarios útiles para poder consultar la información detallada de las proteínas.



**Figura 8.** Ejemplo de la tabla PR\_UNIPROT de la base de datos diccionario de proteínas Uniprot

**Figura 7.** Estructura de la base de datos diccionario de proteínas Uniprot

Se puede apreciar en la Figura 7 y Figura 8 la idea que se ha estructurado para la base de datos. Crear 5 tablas que contengan todos los datos.

* La tabla “PR\_UNIPROT” es la tabla principal con todos los campos únicos que se han implementado. Conectará con dos tablas más:
* Tabla PR\_DICT\_SPECIES: que contendrá los ID de las especies, con el nombre original y con un nombre corto, ambos asociados a las especies.
* Tabla PR\_PROTEIN\_ALIAS: que contendrá los alias para los nombres de la proteína (campo PROTEIN\_NAME). Dicha tabla tendrá diferentes alias para cada proteína tales como el que se extrae del report, el nombre o nombres que aparecen en la web de Uniprot y otros que puedan existir. También contendrá un campo “ORDER\_BY\_ID” para poder ordenar los resultados que aparezcan por ese alias, ya que estarán clasificados.
* Habrá otra tabla PR\_DICT\_PROTEIN\_TYPE\_NAME con los ID de los tipos de nombres que puedan aparecer.
* Por último, la tabla diccionario PR\_DICT\_ASSAY\_TYPE con los tipos de ensayo que pueda tener la proteína también conectará con la tabla alias PR\_PROTEIN\_ALIAS, formando así un conjunto consistente donde hacer las consultas.

En cada tabla habrá una sección de comentarios donde indicará información de los campos, por si surgen dudas a la hora de explotarla.

Una vez creada esta base de datos y cargada toda esta información en la base de datos diccionario, se procederá a activar su funcionamiento a través de ActivityBase, una plataforma donde se almacena los resultados generados por Discovery siendo ésta la base de datos corporativos permitiendo el análisis de los datos de una manera “user-friendly”.

Como se ha comentado previamente, en los documentos finales y en la base de datos con todos los datos procedentes de los “reports” habrá campos con información resumida de esta tabla.

Para la gestión de esta base de datos tanto en la creación como en el mantenimiento, se utilizará Structured Query Language (SQL) un lenguaje a través de ‘Queries’ que permite la explotación de los datos de una manera muy sencilla y organizada.

En cuanto a la base de datos, hay dos procesos a destacar para su completa automatización:

1. Extraer del Excel que contiene el panel de quinasas con la información de proteínas (previamente revisado por las personas de referencia) para cargar toda la información en la base de datos, y, cargar la información relativa a cada proteína en los Excel final estandarizados. Una vez se obtengan los demás resultados, se confirma la información a través de la conexión de la BD Además, se han implementado puntos de control que facilitan el control del programa por parte del usuario. Uno de ellos, guarda los avisos de proteínas en un archivo de texto que estén en el panel de quinasas y que no hayan podido ser cargadas en la BD por falta de información, la cual también se refleja junto al nombre de la proteína.
2. Cargar en el Excel estandarizado y guardar el nombre de la proteína que no ha podido ser cargada porque no hay información en la base de datos, además de indicar el archivo dónde se halla esa proteína. De esta manera, se le facilita al usuario la información que falta y en qué punto, para cargarla si es necesario de un modo rápido.

## **2.3 Creación y carga de la base de datos Uniprot**

## 

Se ha creado un programa, separado de la extracción de resultados, para resolver este apartado. Como se ha comentado anteriormente, el objetivo es automatizar la creación y la carga de toda la información correcta que esté en el panel de quinasas para tener una base de datos fiable donde poder consultarla. La creación de las tablas se hace previamente al código con PLSQL (SQL para Oracle), además de implementar un código con el mismo lenguaje para vaciar las tablas en caso de que se requiera.

### **2.3.1 Estructura del programa**

El programa, también sintetizado en VBA, contiene un gestor de configuración para facilitar cualquier cambio al usuario de manera rápida y sencilla.

En él, se encuentran varias tablas que se comentarán a continuación:

* Tablas diccionarios “ASSAY\_TYPE” y “TYPE\_NAME”: Las tablas diccionario (excepto la tabla con las especies que se detallará en el siguiente apartado), serán siempre iguales y son cargadas al inicio del programa (siempre que no estén ya creadas y cargadas). Se pondrán en la hoja del Excel dónde se encuentra el código del programa. Si el usuario quiere cambiar el formato de las tablas, podrá hacerlo (siempre y cuando consulte el cambio) en la hoja de Excel donde se encuentra el gestor de información.
* Tabla de gestor de Información: Esta tabla contiene una serie de información que le indicará al programa como crear las tablas y a qué columnas del mismo Excel (para las tablas diccionario) o del panel de quinasas (para las demás tablas) pertenecen la serie de características relativas a la proteína. Indica la fila y columna donde se encuentran las tablas diccionario comentadas anteriormente, para que el programa las pueda cargar. También , a la hora de la carga en las otras tablas, indicará la columna del panel de quinasas donde se encuentra los datos necesarios para su carga.
* Tabla de condiciones: Tabla con una serie de condiciones necesarias para el tratamiento de nombres especiales de las proteínas. Como ya se comentó, puede ser que un mismo nombre se diferencie en que tenga escrito en el mismo el tipo de ensayo (agonista o antagonista) por tanto esta tabla sirve para tratar estos nombres.
* Usuario de la base de datos: Una celda para poner el nombre del usuario que vaya a utilizar la base de datos, para evitar que se cambie en el código.

### **2.3.2 Funcionalidad del programa**

La figura 9 muestra un esquema de las fases y funciones del código del programa de automatización de la base de datos. Este esquema contiene una parte con el código principal que se ha utilizado, y los pasos que se han seguido, y una parte con los procesos que desde dicho código se llaman (funciones), con un resumen de la tarea que llevan a cabo. Primero de todo, antes de empezar el proceso en sí, se llevan a cabo unos procesos necesarios que el programa necesita (Figura 9, procesos grises). Éstas consisten en funciones que optimizan el proceso, para mejorar la rapidez, igual que la función que se ha utiliza en la automatización de la extracción de resultados. También se comprueba que la configuración del gestor sea la correcta, para evitar fallos en el proceso debido a cambios hechos por el usuario. Por último, el programa se conectará a la base de datos y cargará en un array y/o en diferentes variables la información necesaria del gestor de configuración.

Seguidamente, se procede a la carga de las tablas diccionario (Figura 9, proceso azul). El programa comprobará si dichas tablas ya tienen la información completa, o si necesitan ser cargadas total o parcialmente. Las tablas se pueden modificar desde el gestor de configuración, ya que el programa recoge la información del mismo. El paso posterior es insertar los datos en la tabla diccionario de especies. El funcionamiento de esta carga será diferente a la de las anteriores, ya que en este caso sí es más probable que pueda haber especies nuevas, por lo que el programa lee para cada proteína el nombre de la especie, y compara ésta con las que ya están cargadas en la tabla: si existe, no la carga de nuevo y si no existe, guardará el nuevo registro en la base de datos.

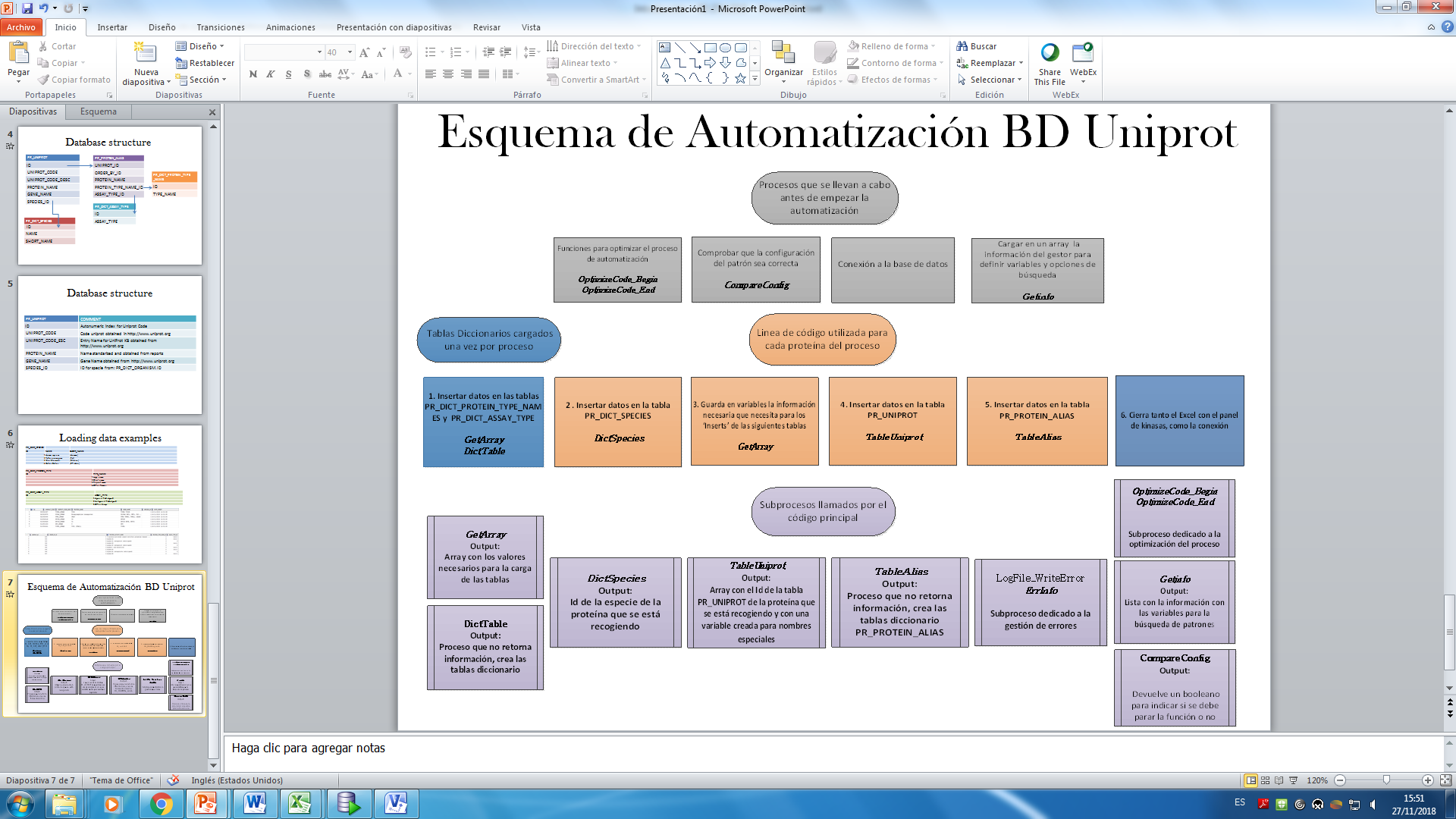
El próximo paso es insertar datos en la tabla ‘PR\_UNIPROT’, no sin antes guardar en variables tanto la información necesaria del panel de quinasas, como la tabla de condiciones explicada anteriormente. El programa comprueba si el código UNIPROT existe en la tabla (ya que los Alias de los nombres de referencia de esta tabla se pondrán en la siguiente tabla), y si no corresponde a ningún código existente lo guardará en la tabla. También se ha de confirmar si el nombre contiene el tipo de ensayo, para guardar en la tabla el nombre de la proteína sin el ensayo (tratando el nombre correctamente), y que este nombre se guarde en la tabla de alias, con su correspondiente índex para el tipo de ensayo.

El paso número 5 es cargar datos en la tabla de alias. Primero se coteja con la base de datos si el nombre existe en la misma. Si no es así, y alguna de las condiciones forma parte del nombre de la proteína, se guarda el índex para dicha condición. Si ninguna de las condiciones aparece en el nombre, se guarda en la tabla como un nombre proveniente del ‘report’. Seguidamente, se guarda el nombre sin el tipo de ensayo, que se recoge del tratamiento que se ha aplicado en la tabla anterior. Para acabar se carga el nombre largo que proporciona Uniprot.

Cuando en el Excel con el panel de quinasas no hay más proteínas, se cierra dicho documento y se corta la conexión con la BD.

Por último, se han creado dos funciones relativas a la gestión de errores: la primera, idéntica al proceso que se sigue en la automatización de extracción de resultados, ya que guarda en un archivo de texto errores que se puedan dar en este proceso, relativos a la conexión con la base de datos, a las rutas de archivo utilizadas, u otros. La segunda, crea otro archivo de texto para las proteínas que no puedan ser cargadas en la BD, avisando al usuario de la información que no es correcta o que falta por insertar en el panel de quinasas, e indicando qué proteína es la que no ha podido ser cargada.

Por tanto, el programa finaliza y la base de datos estará cargada con todas las proteínas del panel de quinasas que han podido ser cargadas.



**Figura 9.** Estructura del programa de automatización de la base de datos.

# **Automatización de la extracción de resultados**

## **3.1 Proceso de estandarización y extracción de los resultados**

Como se ha comentado en los objetivos del proyecto, la idea sería estandarizar los datos en un documento, y para ello, se ha de crear un programa informático para automatizar tanto los archivos que ya estén recogidos como los archivos futuros que puedan haber.

El programa servirá para recoger los campos que en el formato del documento final estaban señalados en azul (apartado 1.3). Al haber tanto varios tipos de archivos, como diferentes formatos, el programa podrá variar según la circunstancia, pero todos estos detalles serán comentados más adelante.

Para empezar con la creación del programa, se ha de buscar una herramienta apropiada para dicho objetivo. Hay varios lenguajes de código que permiten llegar al propósito, pero hay que valorar el uso libre de Software que pueda tener éstos, así como la dificultad que se pueda dar al utilizarlos. Por tanto, uno de ellos, libre y sin demasiada dificultad es **Visual Basic Script (VBS).** Es un lenguaje interpretado por el Scripting Host de Microsoft, utilizado como herramienta de automatización (<https://es.wikipedia.org/wiki/VBScript>). Una vez comenzado el programa, se comprobó que no abarcaba suficientes funcionalidades como para poder crear un programa con las características comentadas anteriormente. Entonces, se decidió cambiar a un lenguaje similar que permitiese acoplar el código creado sin perderlo. **Visual Basic for Applications (VBA)** es un lenguaje similar fácilmente adaptable, habilitado para macros de Excel, con lo que aumenta su funcionalidad de cara a optimizar procesos.

Todos los programas que se utilizarán para el desarrollo del programa son proporcionados por Almirall, con los permisos adecuados y entre los que se encuentran VBS, VBA, Microsoft Office y otros que ayuden a la síntesis y evolución del proyecto en general.

### **3.1.1 Estandarización de archivos según formato**

La mayoría de los documentos que se necesitan procesar son archivos Word, por lo que se resolvería gran parte de la problemática en cuanto al formato de los archivos, y la otra gran parte son Excel, por lo que VBApplications no tiene ningún problema a la hora de trabajar con el mismo.

La idea que se plantea, es transformar el archivo a un formato estándar que se pueda leer fácilmente para poder extraer la información necesaria necesaria obteniendo toda la información de los diferentes formatos, motivo por el cual se desechó la solución inicial en convertir los archivos a ‘txt’ya que la versatilidad del formato disminuía.

Una vez que se obtengan los archivos Excel que provienen de los diferentes tipos de archivo, se extraerá la información de interés de manera automatizada.

### **3.1.2 Búsqueda de patrones**

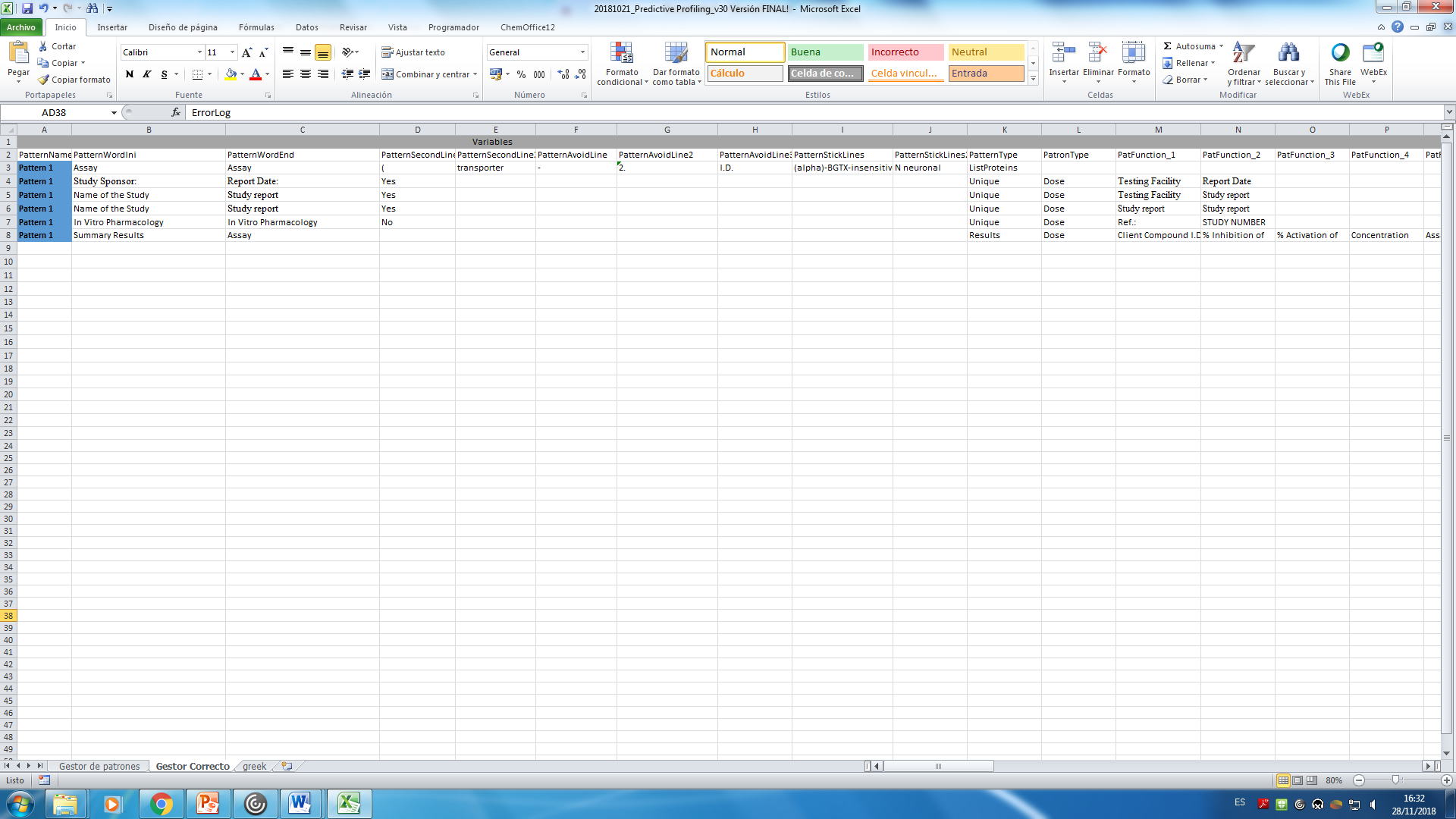
Similarmente al otro programa de automatización, se ha creado un patrón de configuración que se detalla a continuación:

* Sistema de Patrones:

Para encontrar los resultados necesarios, se necesita incorporar en el programa una serie de código para implementar la búsqueda de patrones que permita hacerlo de manera automática. Debido a que los datos requeridos no se encuentran en el mismo lugar del documento, habrá que generar patrones diferentes para encontrarlos. Para ello se creará un gestor de patrones, donde se podrán crear diferentes patrones de búsqueda para extraer la información, tanto la ya existente como los nuevos documentos.

Hay dos tipos de resultados que se han extraído. Los únicos, que se recogen una vez por archivo y los que se deben recoger varias veces por archivo. Para cada archivo habrá habitualmente un patrón de cada tipo. De todas formas el patrón 1 forma un conjunto que engloba ambos tipos de patrón, y hasta varios de un tipo para extraer diferente información de un mismo archivo.

Asimismo, esta herramienta nos permitirá crear variables para facilitar la carga de los datos en el Excel final.

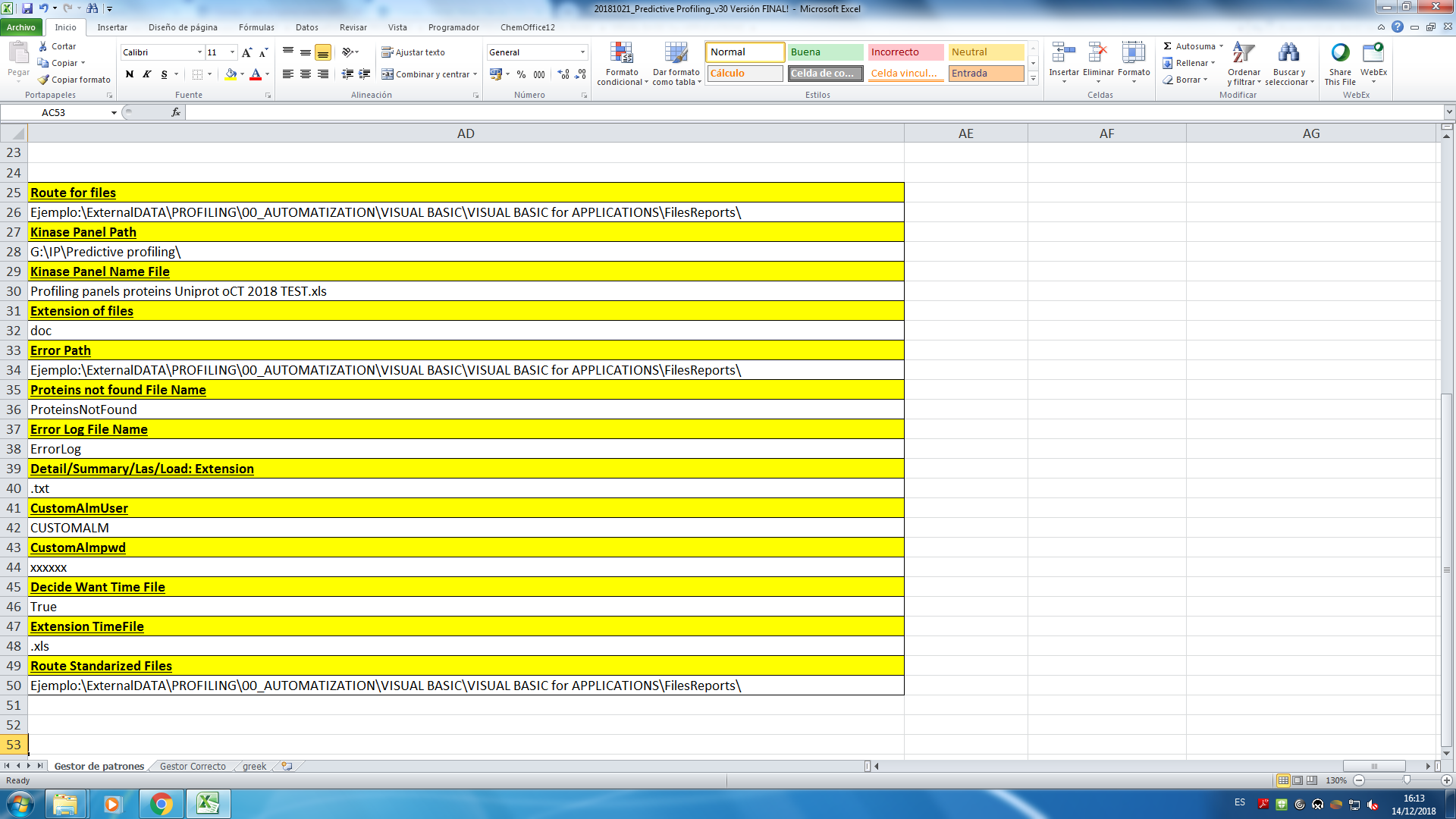


**Figura 10**. Gestor de patrones

Como se aprecia en lafigura 10**,** una serie de variables buscará palabras clave dentro del archivo, y cuando las encuentre extraerá el resultado que interese. Una vez encontrada y contrastada la información que se requiera, para cada “report” se creará un Excel con todos los datos en los campos correctos según el formato del documento final. También se cargará en la base de datos automáticamente para optimizar el proceso y darle una mayor calidad al mismo.

Todas las variables del gestor de patrones que serán necesarias para extraer la información correcta son las siguientes:

* **PatternWordIni**: Palabra/s para indicar que el programa empiece a buscar el patrón.
* **PatternWordEnd**: Palabra/s para indicar que el programa acabe de buscar el patrón.
* **PatternSecondLine(2):** Carácter/es para indicar que el nombre de la proteína contiene más de una línea.
* **PatternAvoidLine(3):** Carácter/es para indicar que el nombre de la proteína no contiene más de una línea.
* **PatternStickLines(2):** Carácter/es para indicar a ciertos nombres de proteínas, que el programa reconoce como diferentes, que forman partes de una sola.
* **PatternType:** Indica si el patrón se utiliza para la lista de proteínas, para los resultados únicos o para los otros resultados del report.
* **PatronType:** Indica si el patrón es para los archivos tipo dosis o tipo curva.
* **PatFunction(8):** Carácter/es necesarios para encontrar cierta información de los documentos.
* **columnID(8):** Número que indica el campo correcto del Excel final donde irá la información que se extrae.
* **Fields:** Nombre de los campos del Excel final.
* **Tabla resumen (**Figura 10**)** que indica qué extensión tiene la diferente información de todos los archivos que existen.
* Tabla de configuración:

Se ha implementado una tabla de configuración similar a la del porgrama de automatización relativo a la BD Uniprot. Incluye las rutas que utilizará el programa, la extensión de los archivos a convertir, el nombre de los archivos de errores y/o de nueva creación, el usuario y contraseña a la base de datos, y la decisión y extensión del archivo que guarda los tiempos del procesado.

**Figura 11.** Tabla de configuración

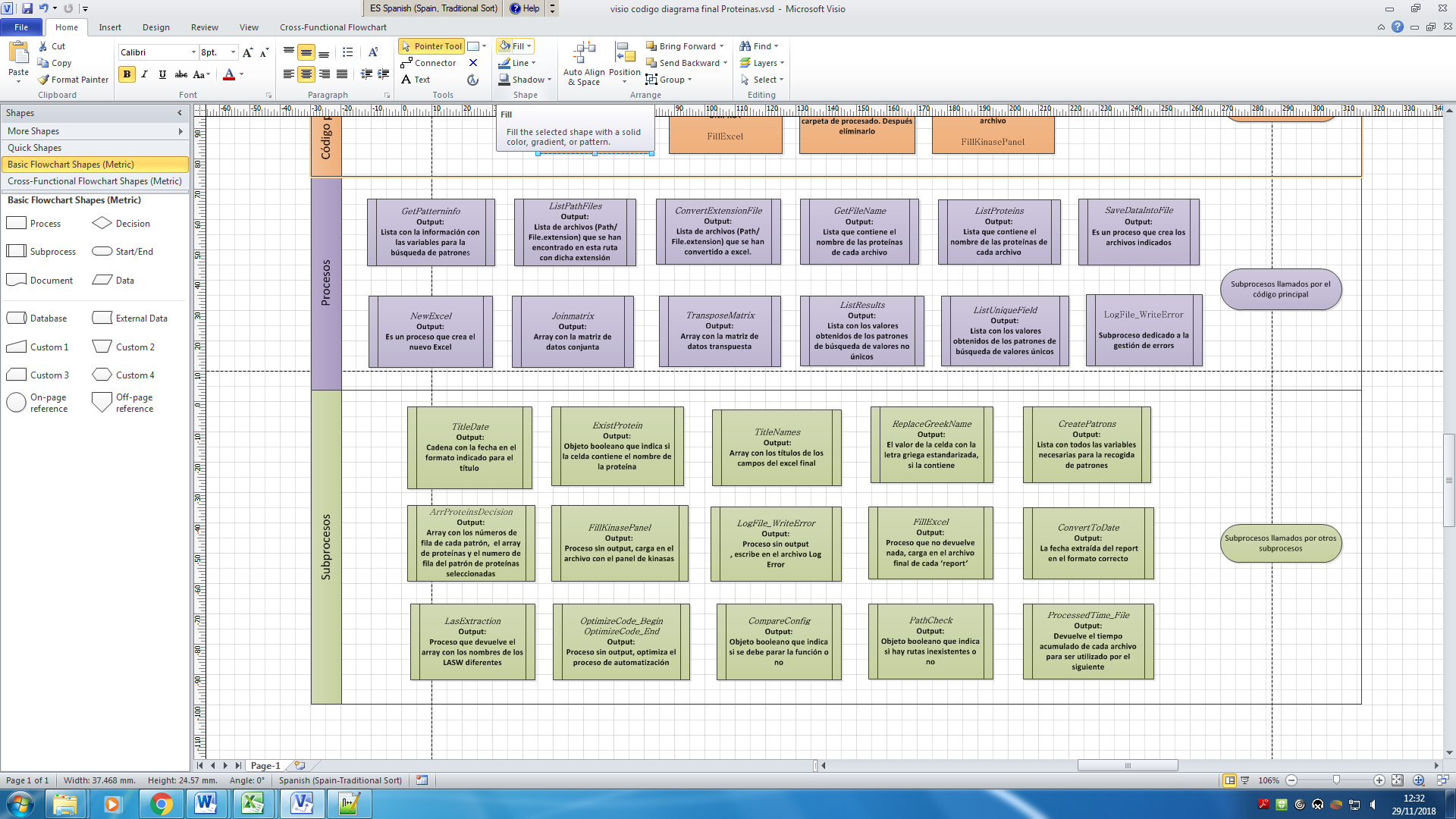
## 

## **3.2 Funcionalidad del programa**

**Figura 12.** Diagrama de flujo del programa de automatización. Código principal.

Para finalizar, se ve en la Figura 12 el proceso de automatización relativa a la extracción de resultados de los documentos. El esquema presenta una serie de procesos con varias características diferenciadas por color, que se aclaran en la leyenda, según si los pasos se hacen:

* Naranja: una vez por archivo
* Azul: por patrón
* Gris: por cada vez que se inicie el programa.
* Verde: los procesos que se llevan a cabo y que son llamados por el código principal
* Lila: subprocesos que son llamados a través de otros procesos que se han implementado antes de la automatización para mejorar la calidad del procesado.

La figura 12 muestra los procesos que son llamados por el código principal (color lila) y los subprocesos (color verde) que son llamados por los procesos anteriores.

**Figura 13.** Procesos y subprocesos del proceso de automatización

Como en el programa de la base de datos, se hacen unas funciones de optimización evitando que la pantalla vaya cambiando cada vez que se activa un documento Word o un Excel, que la comprobación de cambio de valor de las celdas en el Excel donde se encuentra el código y que las fórmulas se calculen en tiempo real. De esta manera, se mejora la velocidad del proceso. También se comprueba que el gestor de configuración esté configurado correctamente (habría que consultar apersonas expertas para añadir un nuevo patrón) y guarda el alfabeto griego en un array, para consultarlo más adelante, como se detallará, además de comprobar que las rutas que se utilizan existan. El programa notificará al usuario si hay alguna ruta incorrecta. Por último, se decidirá si se quiere estudiar los tiempos que necesita cada fase del procesado. En los hexágonos rojos se muestran los puntos donde se recogerá el tiempo en un EXCEL para el análisis gráfico.

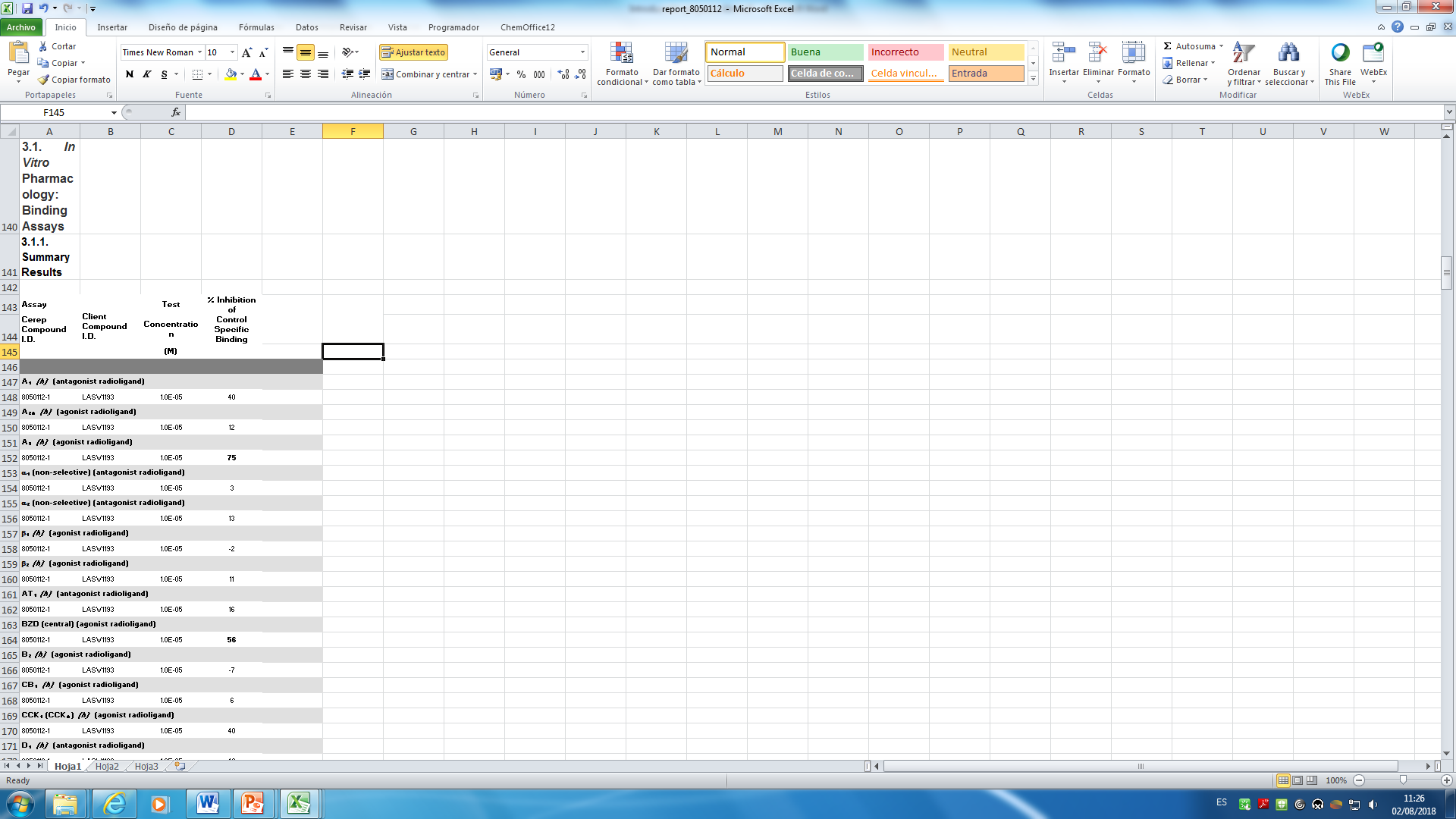
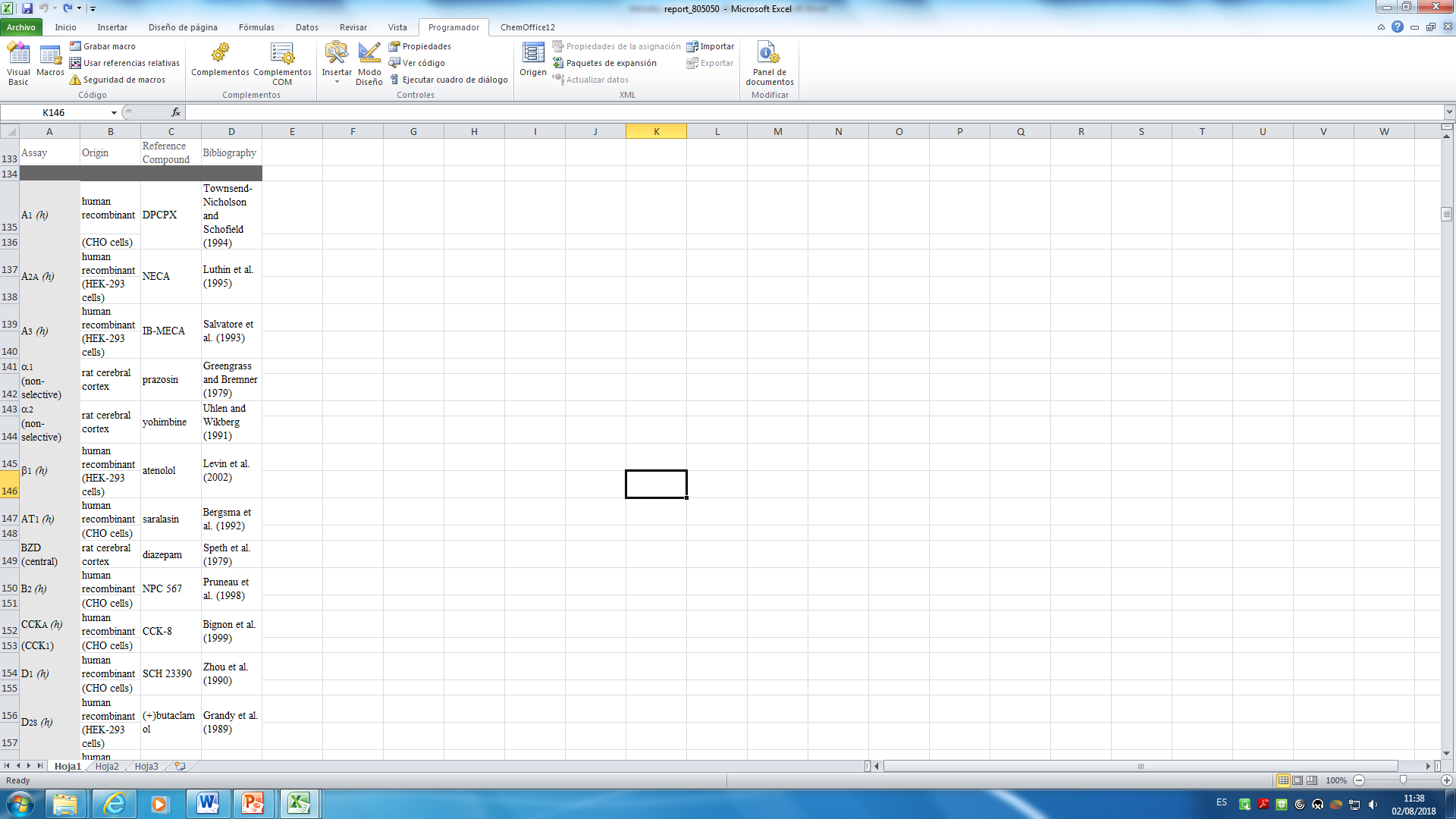
Una vez hecho esto, ya se puede empezar con el proceso.

* **Pasos iniciales (Pasos 1-4 de la figura 12)**

Lo primero será guardar la información de los patrones y del gestor de configuración en variables y/o ‘arrays’. En uno de ellos, se guardarán las rutas de cada archivo (Word, en este caso) que se vayan a procesar. Cada documento, se abrirá para que la información completa se copie en un nuevo archivo Excel, que tendrá el mismo nombre. El documento Word pasará a una carpeta de procesados, y se guardará en otro ‘array’ la ruta de los documentos Excel.

* **Extracción del nombre de las proteínas (Pasos 5-8 de la figura 12)**

Para empezar con el proceso de estandarización se trabajó en la extracción del nombre de las proteínas que participaban en cada archivo. Para ello, había que estudiar los patrones que podían ser útiles y eficaces para tal empresa. Las proteínas no tenían ningunque se utilizan exixa característica en concreto ya que son heterogéneas, pero dentro del archivo hay apartados idénticos donde se muestra el nombre de las proteínas para poder extraerlas. En la imagen siguiente se puede ver varios ejemplos de cómo está estructurada dicha tabla.



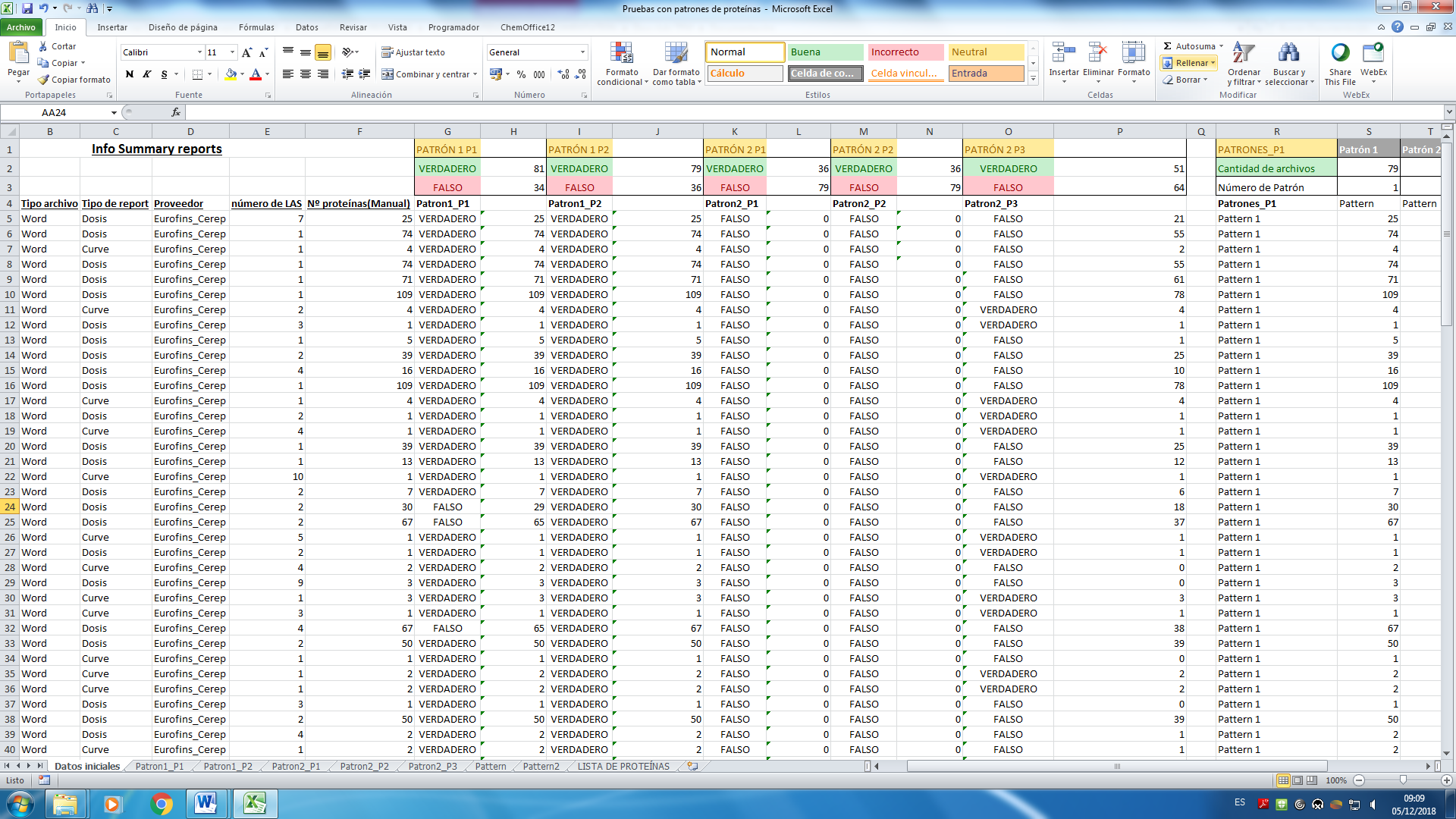
|  |  |
| --- | --- |
| **Figura 14a.** Ejemplo de tablas que contiene el nombre de las proteínas | **Figura 14b.** Ejemplo de tablas que contiene el nombre de las proteínas |
|  |  |

Estas dos tablas representan dos archivos donde la misma información se presenta de forma similar pero no idéntica. En las líneas en negrita (4a) aparece el nombre de las proteínas y en la segunda imagen (Figura 14b) se aprecia que están en la primera columna. Ambas tienen en común que la palabra “Assay” da comienzo a la columna donde se encuentran las proteínas. Por tanto, esta es una variable para que el programa reconozca que debe empezar a buscar la información requerida. A través de recorrer la primera columna del archivo, se generaron restricciones para indicar que guarde tan sólo las líneas que contengan el nombre de la proteína.

De esta manera, se consiguió aislar el nombre de las proteínas en el caso de que el archivo sea del tipo de la 14a.

En el caso de la segunda tabla, el patrón que se ha de crear se complica. Como se puede ver, no resultaría posible establecer el mismo patrón al no existir la línea concordante de la 14a. Además, el nombre entero de la proteína no está en la misma línea como sucede en el primer caso, si no que habría que establecer un sistema para que reconociese que forma parte de la misma proteína, e incluso con proteínas que tienen más de dos líneas.

Es importante recoger toda la información posible del nombre de las proteínas, ya que así se podrá enriquecer el diccionario con toda la información relativa a las mismas. Por ello, es imprescindible confirmar de que el sistema de patrones funciona correctamente. Con tal de facilitar y optimizar el período de ensayo y error del programa, se desarrolló una herramienta que extrae la información que el patrón había recogido en dos archivos .txt: uno con el nombre del archivo, y el nombre de todas las proteínas recogidas. El otro, también contenía el nombre del archivo y además la cantidad de nombres que había recogido. Ambos tenían el patrón correspondiente de donde procedía la información, así como el título contenía la hora de creación de estos para evitar dudas a la hora de hacer pruebas. Ambos documentos, se implementaron en un archivo Excel que servía para mantener un registro de todas las pruebas que se hacían además de compararlo con un recuento manual de las proteínas que había en los archivos para verificar si el número que se había extraído automáticamente era el correcto.

**Figura 15.** Hoja Excel para corroborar que la búsqueda de patrones para las proteínas es correcta

En la imagen superior se aprecia que se crea una columna con el recuento manual de las proteínas. Las columnas siguientes corresponden a diferentes pruebas con los patrones por separado. Indica si coincide el recuento manual con el automático, y recuenta cuántos archivos coinciden y cuántos no lo hacen.

Aunque el recuento manual fuera igual al recogido por el programa, era importante comprobar que el nombre de las proteínas también coincidía, puesto que podía no recogerse de manera correcta.

De esta manera, tanto con los dos documentos como con el Excel era mucho más sencillo y eficiente hacer pruebas para comprobar cómo y qué había recogido el programa, y modificar el código si fuese necesario.

Ahora bien, una de las razones por las que se decidió hacer el cambio de VBS a VBA, fue la dificultad de extraer los nombres de las proteínas. Algunas de éstas contienen letras griegas en sus nombres, y VBS (y el formato .txt que pensamos como formato estándar) no era capaz de retornar correctamente dicha letra, ya que devolvía un interrogante. Al pasar a VBA con el formato Excel, dichos nombres eran retornados correctamente con un tipo de letra llamada “Symbol”. Pero aun así al crear los archivos .txt con los resultados que el programa retornaba, devolvía de nuevo un interrogante. La solución pasó por crear una función que recorriera letra por letra los nombres una vez los recogía, y si encontraba el tipo de letra griega entonces, se decidió estandarizar el nombre de la letra griega poniéndolo entre paréntesis (Ej: (alpha)). Así, cargando en una variable un pequeño diccionario de letras griegas, se consiguió que el nombre de las proteínas se recogiera correctamente. De esta manera se obtienen los nombres de las proteínas extraídos de los archivos.

En el anexo se puede leer las funciones del programa en detalle.

Por tanto, cada patrón recogerá el número de proteínas y se guardará en un array por cada proteína que encuentre. Asimismo, se guarda por cada patrón dos archivos: Uno con el nombre de cada proteína recogida y otro con el número de proteínas recogidas, siempre con el nombre del archivo en otra columna del texto (los mismos archivos utilizados para extraer los datos de la figura 13). Una vez que se hayan pasado todos los patrones, se decidirá cuál es el correcto, y se regirá por el patrón que más proteínas recoja. Por último, se extraerá el valor del array que corresponda al patrón elegido para ser utilizado de aquí en adelante.

* **Extracción de otros resultados (Pasos 9-12 de la figura 12)**

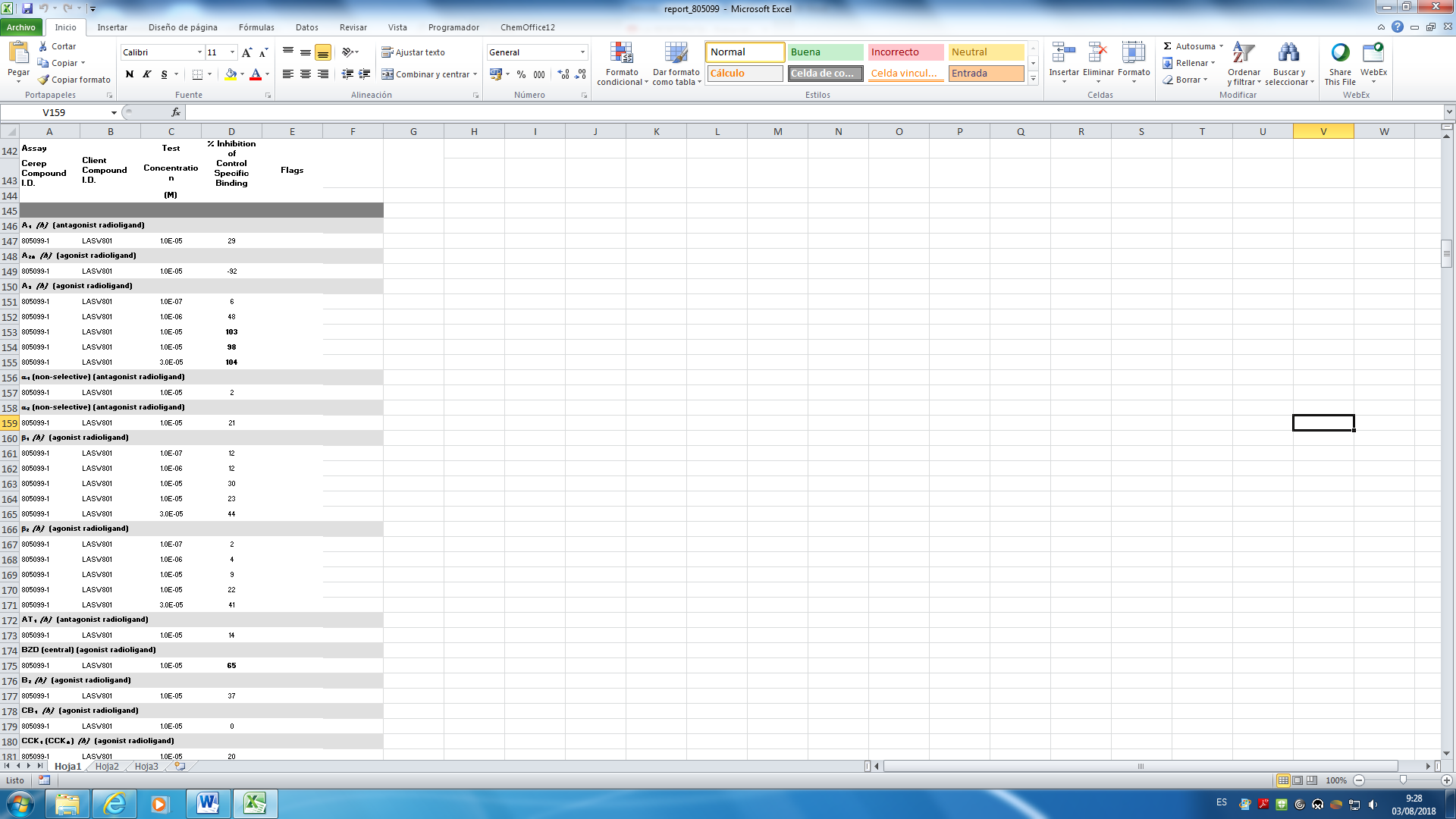
Una vez se tiene resuelta la extracción de proteínas y el valor del array del patrón indicado, se procedió a encontrar patrones para el resto de la información que provenía de los ‘reports’. Siguiendo el mismo procedimiento que en el apartado anterior, se modificó el programa para crear patrones que pudiesen extraer la información requerida. Se crearon dos tipos de patrones: uno para resultados que están asociados al report que se necesitarán tan sólo una vez (fecha, número del estudio y compañia) y otro para resultados que, como el nombre de las proteínas y los resultados asociados a cada una de ellas, necesitan ser recogidos varias veces según el archivo.

Es importante recordar que hay dos tipos de archivos a diferenciar: los archivos dosis que contienen varias proteínas por compuesto, y los archivos curva que contienen gráficas donde el compuesto suele estar a varias concentraciones. Debido a esta gráfica. el tipo de patrón puede cambiar. Por ello, el programa se centra en con los archivos tipo dosis y se creó una variable que discerniese entre estos dos tipos de documentos.

Para los resultados únicos, se creó un patrón que recogiese dichos datos una vez el programa encontrase en las celdas la variable que indicase que debe buscarlos. Se estableció también un rango de búsqueda, con una palabra inicial y una final, para que no encontrase el valor de la variable en otro punto del documento, y así evitar información errónea en los archivos.

En cuanto a los otros resultados, se pudo observar (Figura 14) que las tablas de los documentos dosis donde aparecía la información requerida tenían la misma estructura en todos los documentos, y por tanto se podía establecer un patrón y extraer toda la información. A partir de encontrar el nombre de la proteína, en la siguiente línea del archivo se encontraban todos los resultados relativos a la misma que se necesitaban recopilar.

Cabe destacar que para que tanto el patrón de resultados únicos como del resto se recojan, tienen que coincidir con el patrón previamente escogido para evitar recoger resultados que no corresponden a dicho patrón.



**Figura 16.** Ejemplo de tabla donde se ubican los resultados

Para cada proteína, se busca la información en las siguientes líneas y la guardaba en una variable de tipo array hasta que encontrase la siguiente proteína, donde empezaba el ciclo de nuevo. Para cada dato, buscaba en cada una de las columnas ya que previamente se le indicaba qué información estaba en cada columna.

Una vez había recorrido el rango de búsqueda que se había establecido y guardado para todos los resultados, se necesitaba guardar esa información en un Excel. Era importante tener en cuenta que los resultados debían estar en la columna correcta del archivo final, y por tanto a la hora de guardar los resultados en un array, a través de una variable que provenía del gestor de patrones, se le indicaba en qué columna debía ir la información, tanto la de resultados únicos como el resto. Este lenguaje de programación no permite ir llenando filas mientras progresa la búsqueda de resultados, pero sí de columnas, por lo que se hizo una pequeña adaptación y se agregaron columnas, mientras que el orden correcto de columnas pasaba a ser un orden de filas. Simplemente, cuando se obtuvo todo el contenido completo, se transpuso la matriz para poder tener la información correcta y cargarla en el documento final. Asimismo, previo al paso de cargar la información se unieron los resultados únicos y el resto, para evitar tener que cargar el Excel dos veces y no perder eficacia en el proceso.

Cuando ha acabado el proceso, se crea y se guardan dos archivos útiles para la explotación de los datos: uno contendrá el nombre del archivo Excel relacionado con el nombre del documento del cual proviene y otro con todos los compuestos que se han encontrado en el proceso.

* **Creación y carga del Excel estandarizado final (Pasos 13-15 de la figura 12)**

Seguidamente, se crea el Excel y se carga con los datos correctos para cada archivo. Así, se podrá llegar a tener toda la información de los Word de tipo dosis estandarizada.

Se abre un nuevo Excel y se insertan los títulos de los campos en su correspondiente lugar. Acto seguido, se procede a cargar toda la información que se ha extraído del archivo, en las columnas correctas. Una vez se acaban de cargar los datos, se conectará con la base de datos para comprobar si cada proteína existe en la misma, y si es así, selecciona toda la información de la base de datos relativa a cada proteína recogida. Si la proteína no se encuentra en la BD, se guardará en un archivo esta alerta, indicando también el archivo en el cual no se encuentra. Cuando no hay más filas con proteínas, el archivo se cierra. El Excel con la información copiada del Word pasa a otra carpeta de Excel procesados.

* **Carga de las proteínas en el panel de quinasas (Paso 15 de la figura 11)**

Cuando no haya más archivos para procesar, se guarda en un ‘array’ todas las proteínas que no estaban cargadas en la BD Uniprot. Éstas se cargarán en el panel de quinasas y luego se comprobará si hay nombres duplicados, para evitar que se copie el mismo nombre varias veces.

Antes de cerrar el programa, se anulan los cambios para optimizar el programa, para volver a restaurar los valores por defecto.

* **Gestión de errores**

Por último, se ha implementado un espacio para gestionar los errores que puedan suceder a lo largo del proceso:

* La conexión a la base de datos
* Si no hay archivos en la carpeta para procesar,
* Los Excel de nueva creación no han podido ser creados, por
* Los patrones no tienen la información necesaria para extraer los resultados o si algún patrón no ha encontrado ninguna proteína en el archivo, ya que esto daría que los siguientes pasos no se diesen y podría parar el proceso general.

## **3.3 Análisis gráfico y estadístico del proceso de automatización**

Se han analizado 9 gráficos con características relacionadas:

* Tamaño del documento/Número de datos

**Figura 17.** Gráfico que relaciona el tamaño del documento con el volumen de datos

En este gráfico se puede ver como el tamaño del documento está directamente relacionado con el número de datos. Cuanto más grande es el documento, más datos recoge el programa para cargar en el Excel final. Aunque no es una relación detallada perfectamente, el cálculo de la correlación es de 0.81, por lo que se intuye una clara tendencia.

**Figura 18.** Gráfico que relaciona el tiempo procesado total con el volumen de datos

En el gráfico anterior, se ve también la relación entre el tiempo de procesado de cada archivo y el volumen de datos, aunque no es tan clara como el gráfico anterior ya que el coeficiente de correlación muestra 0.68 . Sobre todo en la parte final, donde claramente los archivos que más tardan, son los que más volumen de datos tienen. En las siguientes gráficas se estudiará en qué parte/s del procesado afecta más este volumen de datos.

**Figura 19**. Gráfico que relaciona el tiempo para convertir en Excel y el volumen de datos

Este tiempo corresponde al que recoge cuando acaba de crear el Excel desde el archivo Word. Como se puede apreciar, la gráfica es muy parecida a la anterior, ya que esta parte es la que necesita prácticamente la totalidad del tiempo. Por tanto, se relaciona de igual manera que la gráfica anterior y la correlación también es muy similar: 0.67

**Figura 20**. Gráfico que relaciona el tiempo para convertir en Excel y el volúmen de datos

Aquí se puede apreciar el tiempo que tarda en cargar la información UniProt en el Excel final. En este caso no está tan relacionado con el volumen de datos, ya que si el nombre de las quinasas está repetido, o bien hay muchas proteínas que no están reflejadas en la base de datos, el programa lo procesa con mayor velocidad. La correlación así lo ratifica, dando un coeficiente bajo de 0,36.

**Figura 21.** Gráfico que relaciona el tiempo para convertir en Excel y el número de datos

También se quiso comprobar si el número de páginas afectaba al tiempo de procesamiento. Como se aprecia no es así, ya que un mayor número no implica que la información que recoja el programa sea mayor). Por tanto, no hay relación aparente entre estos dos datos. Además, la correlación tan sólo llega a ser 0.25.

**Figura 22.** Gráfico que relaciona las páginas y el volumen de datos

A diferencia del gráfico anterior, sí hay una relación clara entre estas variables. Los mayores volúmenes de datos son los que más páginas presentan, por tanto se aprecia una regularidad suficiente como para afirmar que la relación de aumento proporcional existe. Esto es debido a que cuantos más datos, más aumentará el número de páginas. La correlación también indica una relación positiva moderada, ya que el coeficiente es de 0.64.

**Figura 23**. Gráfico que relaciona el tiempo para convertir en Excel y las páginas del archivo

En este gráfico comparado con el anterior, si se aprecia una regularidad más clara. (Correlación positiva de 0.68) Por tanto si se puede relacionar el número de páginas con lo que el programa necesita para convertir el archivo Word en Excel, debido a que copia toda la información, sea necesaria o no extraer a posteriori. Como el código (y el lenguaje de programación) está preparado para automatizar a partir de Excel, será imprescindible copiar todo el contenido del documento.

**Figura 24.** Gráfico que relaciona el tiempo para convertir en Excel y el tamaño del documento

Por último, siguiendo con los gráficos analíticos del tiempo después de convertir en Excel, se estudia la relación con el tamaño del documento. Como se ve, también hay una relación aparentemente clara, ya que los tamaños más grandes, son los que más tardan en copiar la información a Excel . Por tanto, tanto el tamaño como el volumen de datos afectan al tiempo de procesado, en concreto al tiempo que tarda el programa en cambiar el formato. De todas maneras, la correlación indica que la relación positiva no es demasiado elevada: 0,56.

**Figura 25.** Gráfico que contiene el tiempo procesado acumulado

Por último, se ha analizado gráficamente el tiempo procesado para observar si con el paso del tiempo el programa aumentaba el tiempo de procesado parcial exponencialmente o de manera lineal. Como se puede distinguir, el tiempo acumulado aumenta de manera lineal, por lo que no afecta el número de archivos procesados al tiempo procesado parcial de cada ‘report’.En cuanto al análisis estadístico, se han estudiado las mismas relaciones que en el análisis gráfico, pero calculando el coeficiente de correlación de Pearson y representando el gráfico de dispersión de las variables relacionadas, además de una comprobación de la normalidad de los datos. Se observa que todos los coeficientes de correlación son positivos, por lo que la relación de todas las parejas de variables es positiva. Algunas relaciones más moderadas que otras, pero cabe destacar que la correlación más elevada es entre el tamaño del documento y el número elevado de quinasas. Se ha visto que en todos los gráficos, las muestras están muy apareadas en una zona del mismo gráfico, y por tanto se estudió si las muestras tienen una distribución normal, para poder asumir que el estudio estadístico con tests paramétricos es correcto.

Por ello, se ha calculado la distribución de frecuencias de los tiempos de procesado, para ver si siguen la distribución indicada (Figura 25). Se añade también el gráfico ‘PP-plot’ (Figura 26) y el cálculo de la curtosis y la asimetría para tener varias comprobaciones de la normalidad de la muestra. Cualquiera de las pruebas o condiciones de normalidad no se cumplen, como se aprecia en ambos gráficos, ya que hay un marcado aumento de frecuencia de valores en el lado izquierdo tiempos de procesado menores). En cuanto a la curtosis, que se refiere a la manera en que las colas de una distribución difieren de la distribución normal, llega a estar alrededor de 12, cuando un valor correcto para la distribución normal sería alrededor de 0. La asimetría, que calcula el grado de simetría de los datos, también debería estar cercana al 0 (aunque si lo estuviera podría ser simétrica y no implicar normalidad). En el caso de estos datos, es 3,51 por lo que también prueba la falta de normalidad.

Por ello, hay que tener en cuenta que los datos no son paramétricos para futuras elecciones de tests estadísticos.

**Figura 26.** Gráfico de normalidad del proceso de automatización

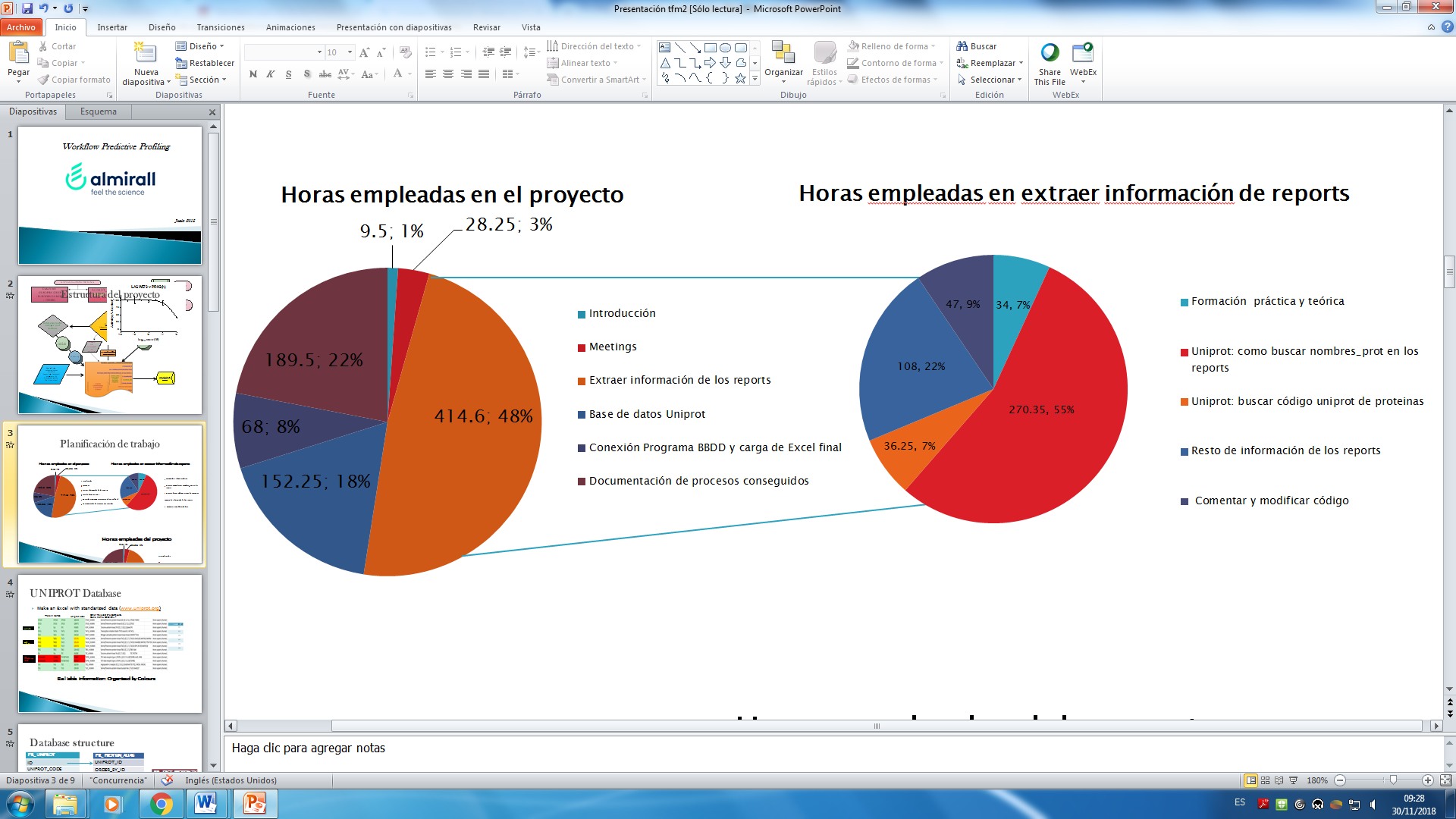
**Figura 27.** Gráfico ‘PP-Plot’ para la comprobación de normalidad

# **Conclusiones**

Una vez concluido el proyecto, se analiza el alcance del proyecto real comparándolo con el teórico para estudiar las diferencias entre ambos, ver cuánto se ha podido cumplimentar, además de reflexionar sobre lo aprendido durante la duración del proyecto. Se analizará la planificación y la metodología del trabajo, para explicar las mejoras y cambios que han sucedido. También habrá un apartado para las líneas de trabajo futuro que se deben implementar en el programa para aumentar las funcionalidades del mismo.

## **4.1 Análisis de la planificación**

Como se ha comentado en el apartado 1.4, la planificación ha ido según su curso de manera general, pero sí que ha habido ciertos cambios relacionados con las tareas que no se han cumplimentado.

**Figura 28.** Gráfico de horas empleadas en el proyecto.

En la figura 28 se puede ver las horas empleadas en el proyecto, y a qué fases se han destinado. En la tarea de extraer nombres de los ‘reports’, se han empleado un 48% de las horas, muchas más de las inicialmente planificadas. Esto se debe a varias causas:

* **Primera tarea para cumplimentar:** al ser la primera fase que se decidió implementar en el programa, la experiencia utilizando el lenguaje de programación era mucho menor, a pesar de haber hecho una etapa de formación. Por ello, la velocidad con la que se resolvían los problemas que podían sucederse en el código requerían más tiempo de dedicación.
* **Diferencias en el formato de los documentos**: Los documentos, tienen formatos diferentes, y por tanto el código para estandarizarlos será diferente. Aunque se probó, no se pudieron estandarizar los documentos Excel ni PDF, ya que también se dedicaron las horas planificadas a otras fases del proyecto. También hay que tener en cuenta que los documentos Word no tienen la misma estructura, por tanto, para extraer el nombre de la proteína se han implementado restricciones y patrones para cada documento, lo que ha aumentado el número de horas.
* **Estandarización del nombre de la proteína**: También hay nombres de proteínas que contienen letras griegas, y se implementó una función para estandarizar los nombres de proteínas ya que el programa no reconocía las letras griegas. Por tanto esto también supuso horas de dedicación.

En la figura 17 se ven las horas teóricas que se habían presupuestado para el proyecto, siendo la fase de extraer los ‘reports’ la que ha aumentado la proporción respecto al total. Las demás fases han disminuido el peso de horas dedicado, excepto la documentación, ya que también se ha creado más funcionalidades del programa y por tanto se han empleado más horas en documentarlo.

**Figura 29.** Gráfico con las horas teóricas que se emplearán en el proyecto.

## **Objetivos y alcance final del proyecto**

En este apartado se realizará un análisis de los objetivos del trabajo, estudiar cuáles se han llegado a completar y cuáles no.

Primero se expondrán los objetivos específicos, y se comentará paso a paso cuál es el estado que se ha conseguido llegar en cada uno de ellos y qué es lo que ha faltado, si procede, para su finalización. También se expondrán los resultados a los cuales se han llegado con dichos objetivos.

**Objetivo principal 1:** Estandarizar los archivos de datos de profiling biológico enviados por los diferentes proveedores con los diferentes formatos de archivos, creando un programa que automatice el proceso de estandarización y carga mediante la búsqueda de patrones.

* Extraer y automatizar toda la información de los reports mediante software VBA para EXCEL: Este paso se **completó**.
* Configurar los parámetros para realizar la búsqueda de patrones: Hay varios matices en este paso. Los parámetros están configurados para realizar la búsqueda de patrones, y añadir nuevos patrones para extraer los resultados. Pero la configuración del patrón específico en los archivos de tipo curva Word, y para los formatos Excel y pdf no se ha podido realizar. Por tanto la cumplimentación final en este caso es **parcial.**
* Carga del documento final estandarizado con la información de UNIPROT. Una vez que se ha creado y cargado la base de datos con los datos de UNIPROT, se ha comprobado si las proteínas recogidas del patrón existen en dicha BD, y se ha rellenado el documento con el resto de información relativa a esa proteína. Para ello, se ha creado y cargado previamente la BD, a través de un programa que automatiza dicha función. Por tanto este paso también está **completo**.

**Objetivo principal 2:**  Crear bases de datos que simplifiquen el trabajo de investigación y explotación de los datos.

* Diseñar un nuevo esquema en la base de datos corporativa de datos biológicos (ABASE) para recoger la información UNIPROT: se hizo un ‘Brainstorming’ para este esquema de la base de datos relativa a UNIPROT, además de consensuar los datos que se requerían en esta BD con las personas de referencia que la utilizarán en el futuro. Finalmente, se tuvo una idea clara y concisa de las tablas que se utilizarán y por tanto este objetivo se **completó**.
* Automatizar la carga de información y llenado de campos del nuevo esquema: una vez se obtuvo el esquema de la BD se completó la creación mediante PLSQL (Structured Query Language for Oracle), además de automatizar, a partir de un archivo con la lista de todas las proteínas, la carga de los datos en la BD. Además, se ha implementado una configuración para el programa que guarda en un archivo la información que puede faltarle a una proteína, para ser cargada en la BD . Este objetivo se ha cumplido en esta fase 2 del trabajo, por lo tanto está **completo**.

**Objetivo principal 3: Consolidar y ampliar los conocimientos estudiados y aprendidos en el máster de Bioinformática y Bioestadística.**

* Adaptar conocimientos específicos a nivel práctico de la asignatura de “Programación para la bioinformática” para el proyecto: este objetivo ha servido para intentar utilizar el lenguaje de programación Python para llevar a cabo este proyecto. Pero debido a un problema de licencias y del uso libre del mismo, se optó por cambiar de lenguaje. De todas maneras, aprender a estructurar cualquier lenguaje (como Python o R) sirvió para aprender más rápidamente a manejar otro lenguaje de programación. Este objetivo está **completo**. Además, en el análisis gráfico y estadístico del procesamiento también se han aplicado algunos recursos básicos de bioestadística.
* Aplicar diversos lenguajes de programación para llevar a cabo el proyecto: este objetivo también está **completo.** A través del cambio que se ha propuesto de lenguaje de programación. Se comenzó con VBS, pero al observar algunas dificultades que durante el proyecto, se decidió usar VBA ya que facilita el proceso y aumenta la calidad del programa. Además también se ha utilizado PLSQL para la creación de BBDD, por lo que este objetivo está **completo**.

## **Conclusiones generales**

Se exponen las conclusiones que se han obtenido del proyecto.

1. La automatización de procesos para la extracción de datos es imprescindible en las empresas que generan grandes cantidades, ahorrando tiempo y dinero en el proceso de explotación de los resultados.
2. El aprendizaje sobre el lenguaje de programación ha sido imprescindible para la estandarización de los datos , perimitiendo así la creación del software.
3. Las bases de datos son una herramienta eficaz para la consulta y explotación de datos. Con ellas, se provee al usuario la información de manera clara, concisa y ordenada. A través de las bases de datos se puede crear toda una red de datos para diferentes tipos de usuarios, de una manera conjunta y sencilla de utilizar.
4. Es imprescindible implementar un sistema de gestión de errores en cualquier programa para evitar fallos inesperados o que paren el funcionamiento del sistema en el momento que el usuario lo esté utilizando.
5. La coordinación interdepartamental es esencial para aumentar la calidad de los resultados que se esperan. Sin ella el funcionamiento de los sistemas sería menos apropiado ya que no se habría consensuado con las personas de referencia sobre la utilización del programa.

## **Líneas de trabajo futuras**

Para acabar, se expondrán las futuras implementaciones que el programa necesita, ya que no se han alcanzado los objetivos del proyecto global.

Estandarización los documentos de otros formatos: Como se ha comentado, no se ha podido llegar a completar la estandarización de los formatos PDF y Excel. Por tanto, uno de los próximos pasos será llegar a automatizar dicha estandarización. Además también se debe recoger los gráficos de los documentos Word que sean de tipo Curva.

* Carga de los datos a Activity Base: Como se comenta en la introducción, una vez estén todos los documentos estandarizados se tendrá que cargar en Activity Base, plataforma que funciona como base de datos.
* Repaso del código principal: Una vez creado todo el código, habrá que repasar las funciones para implementar mejoras que optimicen el proceso.
* Creación de la aplicación para la utilización del programa: para poder explotar los datos, se necesita la creación de una aplicación desktop para que los usuarios puedan utilizar el programa de manera sencilla.
* Formación para los usuarios sobre la aplicación: Por último, una vez se obtenga el programa final, habrá que hacer una breve formación para que los usuarios puedan utilizarla sin ningún problema, además de idear alguna implementación nueva para el mismo.

# **Bibliografía**

* InduSoft**, “VBScript Reference Manual for InduSoft Web Studio”,**InduSoft,, USA, 2007.
* Bill Jelen, Tracy Syrstrad**, “Excel® 2013 VBA and Macros”,** Pearson Education, Inc**..** USA, 2013.

* Flavio Morgado, “**Programming Excel with VBA: A Practical Real-World Guide”,** SPRINGER**,** Rio de Janeiro, Brasil, 2016.
* Rafael Camps Paré, Luis Alberto Casillas Santillán, Dolors Costal Costa, Marc Gibert Ginestà **, “Bases de Datos for UOC**”, Primera edición, Eureca Media S.L., Barcelona, mayo 2005
* [**www.uniprot.org**](http://www.uniprot.org) **(Fecha de última consulta: 14 de Diciembre de 2018)**

# **Glosario**

**Palabras clave (Keywords)**

* Molecular informatics: Campo informático con el objetivo de mejorar y implementar sistemas relacionados con la biología y la química molecular.
* Predictive Profiling: Campo de estudio con el objetivo de predecir la actividad o el perfil de ciertos compuestos.
* Software Development: Desarrollo de un programa mediante lenguaje de programación para la implementación de sistemas.
* Report Extraction: Extracción de los datos y resultados provenientes de los documentos de perfil biológico.
* Standardization: Desarrollo de un método previamente establecido por unas características definidas.
* Databases Creation: Creación de la plataforma donde se podrán explotar los datos provenientes de los archivos.
* Automatization: Proceso de repetición mecánico de un proceso o una actividad.
* Protein’s Activity: Producción de los efectos de una proteína.
* Patterns Searching: Búsqueda de la información mediante unas variables o estrcuturas recurrentes.