Creación de Herramientas de Software para la Gestión de Proyectos de Investigación

Silvia Marisa Rampello[1], Adriana Haydee Narváez[1], Sebastián Garber[1], Juan José Marenco[1], Gabriel Eduardo Pousada[1], Maira Daiana Gómez[1], Juan Roger[1], Eduardo Daniel Ferrero[1], Clara Cinquegrani[1], Ricardo Zappala[1]

[1]Universidad Nacional de La Matanza {srampello@unlam.edu.ar, ahnarvaez@yahoo.com, sgarber@unlam.edu.ar, jjmarenco@gmail.com; gpousada@edenor.com; mdaianagomez@gmail.com; rogerdocencia@gmail.com, , eferrero@unlam.edu.ar, clara_cinquegrani@hotmail.com, szappala@unlam.edu.ar}

Resumen

El presente proyecto pretende analizar y mejorar el Sistema de Gestión de la Información dentro de la Secretaría de Investigaciones del Departamento de Ciencias Económicas de la Universidad Nacional de La Matanza.

*Para ello se presentan en esta ponencia la parte del proyecto dirigido al desarrollo de un software que opera sobre los currículums vitae provenientes del SIGEVA[1], utilizando programación orientada a objetos y aplicada sobre los currículums vitae entregados por el docente en formato *.pdf. Se priorizará que los mismos se ajusten a los criterios de Open Source[2] para permitir la mejora constante que otros usuarios quieran lograr. En una segunda etapa, se llevará adelante una base de datos de los proyectos de investigación, consistente en el desarrollo de un programa de gestión interna de proyectos aplicable a los proyectos y programas incluidos en los PROINCE[3] y CYTMA2[4]. Cabe destacar que el proyecto inicial preveía operar sobre los currículums CVar, pero el formato actual de descarga del mismo no suministra los datos necesarios usándose por tal motivo los documentos entregados tanto por el SIGEVA-UNLaM como el SIGEVA-CONICET.*

[1] El SIGEVA, Sistema Integral de Gestión y Evaluación, es un conjunto de aplicaciones informáticas a las que se puede acceder de forma segura a través de una plataforma Web. Comenzó a gestarse en 2004 y nace en 2005, siendo un desarrollo de la Dirección de Informática de la Gerencia de Organización y Sistemas del CONICET.

[2] Open Source: Código abierto

[3] Programa de Incentivos de la Secretaría de Políticas Universitarias

[4] Programa CYTMA2 de la Universidad Nacional de La Matanza.

1. Introducción

La presente ponencia se basa en avances del proyecto PROINCE 55 B 207[5]. La investigación referida se divide en tres partes: una primera parte constará de una revisión exhaustiva del estado del arte acerca de la gestión de proyectos, particularmente enfocada en la aplicación de nuevas tecnologías. Una segunda parte, en el desarrollo de un programa de extracción de datos a partir del SIGEVA y una tercera parte en el desarrollo de un programa de gestión para el circuito de los proyectos como base de un sistema de información para la gestión de proyectos dentro de la Secretaría de Investigaciones del Departamento de Ciencias Económicas de la UNLaM.

Administrativamente un currículum vitae es un elemento básico de la gestión de recursos humanos dentro de cualquier tipo de organización, sea ésta pública o privada. Aspectos como la cantidad de perfiles existentes, capacitación deseada, o excedente, o faltante, o la determinación de múltiples ratios, serán entre otros, información resultante de las bases de datos en que se almacenen los SIGEVA, estando dicha información sujeta, como en la mayoría de los procesos administrativos, a algunas instancias excluyentes.

La gestión administrativa de proyectos de investigación supone un cúmulo de tareas y relaciones institucionales. Cada proyecto de investigación realiza un circuito administrativo interno dentro de la Universidad, pero también, en determinados momentos supone contar con información detallada de sus integrantes para

[5] Radicado en del Departamento de Ciencias Económicas de la UNLaM – Inicio 01/01/2018 y finalización 31/12/2019 – con título similar al de la presente ponencia.

suministrarla anualmente a la Secretaría de Ciencia y Tecnología e Innovación productiva de La Nación.

Gestionar los proyectos, implica gestionar una gran cantidad de información, la cual puede apoyarse en las nuevas tecnologías.

2. Metodología

Se utilizará PYTHON® como lenguaje de programación. Éste cuenta con muy buenas herramientas de distribución gratuita para la lectura de pdf y posee un manejo muy ágil sobre archivos de texto plano (*.txt) que formarán la columna central del desarrollo informático. Además satisface el requisito propuesto por los investigadores de ser de código abierto, permitiendo que cualquiera efectúe mejoras o lo flexibilice a sus propias necesidades.

La definición del lenguaje de programación a usar en desarrollos de entorno de open-source como el propuesto en esta investigación implica el reconocer las capacidades particulares que tienen los mismos. Realizar un loop básico es posible tanto en lenguajes de alto como de bajo nivel; tendrá la misma base de acción reacción sea un lenguaje fuertemente tipado[6] o no, al igual los resultados serán los mismos si el lenguaje posee o no una fuerte o débil librería. Un IF o un FOR se resolverá de igual forma si se utiliza C++ o C o JAVA-SCRIPT.

Por ello el lenguaje usado pasa por elementos particulares que PYTHON® posee y que se adaptan a los principios teóricos que se postulan en el diagrama de flujo que se reproduce más arriba, en este mismo acápite.

PYTHON® presenta las siguientes características:

2.1. Es simplificado y rápido ya que con pocas líneas de script se logran resultados. No solamente el lenguaje en el intérprete cumple con esta condición ya que los scripts ejecutables en DOS también cumplen con las expectativas de desarrollo.

2.2. Es flexible: su antes citada capacidad de tipado hace que el trabajo con múltiples variables se organice casi en el momento de la escritura del script.

2.3. Es un lenguaje definido hacia los objetos: esto hace que los desarrollos no pierdan nunca de vista la objetivación. En trabajos como el propuesto por esta investigación, esta quizás sea la característica más importante tanto desde el desarrollo de las líneas de programación como en la direccionalidad a la obtención de datos concretos.

2.4. Es ideal para el manejo de texto plano: aunque la mayoría de los lenguajes funcionan correctamente con la captura de texto, PYTHON® posee una muy buena performance tanto para la captura de texto en diferentes codificaciones (UTF-8, UTF-7, etc.) como en su adaptación sencilla a la extracción del mismo. Esto se cumple también cuando se buscan salidas de texto.

2.5. Es un lenguaje fácil de entender cuando los scripts se concluyen: cualquier persona con conocimientos básicos de programación orientada a objetos puede entender el uso y manejo del entorno, lográndose así, una forma de poder tener un desarrollo open-source adaptable a las correcciones o adaptaciones propuestas por los usuarios. Una característica que lo hace 'rápidamente entendible' es la obligatoriedad de tipiar identaciones[7] correctas para que los scripts se "ejecuten".

2.6. Es portable y multi-platafoma: adaptando ciertos parámetros básicos, PYTHON® se desenvuelve correctamente tanto en MAC®, Linux® o Windows® en comparación con otros lenguajes. Las mejores plataformas son las dos primeras pero solo por la característica que ambas ya contienen PYTHON® en forma nativa.

2.7. Un sistema de Módulos[8] muy amplio: aunque la mayoría del código escrito para esta investigación será definido a la captura y manejo de texto plano con herramientas propias, la apertura, escritura y listado de archivos se harán a través de módulos estándar de PYTHON® (os, glob, etc.).

2.8. Es un lenguaje con crecimiento continuo: el script que será el desarrollo final de este trabajo, puede formar parte el día de mañana de un módulo específico de uso global.

3. Resultados parciales

Se presentan los avances del proyecto (el mismo finaliza el 31/12/19). En tal sentido, dado que lo que lo que se busca en la investigación es lograr un "complejo de administración de datos" se consideró correcto en el grupo de investigación destacar no solo la actividad de extracción de datos sino que quedara definido el "manejo" (administración) de los mismos en una base de datos. Así surgió la idea que los archivos generados para cualquier actividad (extracción, manejo y corroboración)

[6] Un lenguaje será tipado cuando no se permiten violaciones de los tipos de datos o para cambiar las variables debe garantizar y aceptar su nuevo valor. Así, el valor de una variable de un tipo concreto, no se puede usar como si fuera de otro tipo distinto a menos que se haga una conversión.

[7] es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores. En ciertos lenguajes de programación como Haskell, Occam y Python, se utiliza para delimitar la estructura del programa permitiendo establecer bloques de código.

[8] Un módulo es un archivo PYTHON que generalmente tiene solo definiciones de variables, funciones y clases. Normalmente tienen como objetivo el actuar sobre archivos y ejecuciones específicas, como por ejemplo el módulo xlwt que genera archivos -xls

tuvieran destacado el término "administrar". De esta manera surgió el nombre de Administrador de Proyectos Científico-Académicos y cuyas siglas se resumen en APCA.

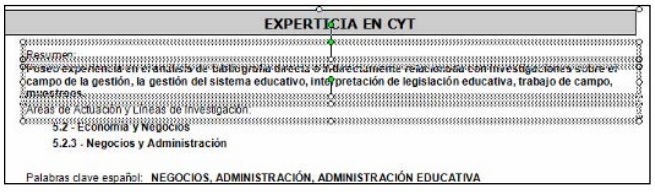
3.1 El origen de los datos usados

En su origen el trabajo fue planeado para usar como datos de origen los documentos entregados por el CVar tanto en formato PDF (Portable Document Format) como Documento de Texto (.doc, docx o .rtf), dado que durante el desarrollo de la primera etapa el formato de descarga del CVar cambió a “mi cvar impreso”, el cual no suministra la totalidad de los datos requeridos para la gestión de la Secretaría, se decidió continuar el desarrollo tomando los currículums vitae que surgen de SIGEVA, tanto de la versión SIGEVA CONICET, como de la versión SIGEVA propio de la Universidad, pudiendo aplicarse a cualquier Universidad que lo requiera, no es excluyente para la UNLaM. Con esta nueva organización, la investigación se definió por el uso tanto del SIGEVA(UNIV) como el SIGEVA CONICET, siendo los cambios generados en ambos documentos adaptables al script que se desarrolló en los orígenes del proyecto. Debido a todos estos cambios y adecuaciones a los que el trabajo ha sido llevado, los script definitivos solo orientan al uso de archivos .pdf ya que los docentes tienden a enviar este tipo de archivo en vez de los archivos .doc bajando así los tiempos de programación invertidos.

3.2 Los diferentes documentos de texto

En todos los casos los formatos respetan una regla general que es la posición de los textos incluidos dentro de "cajas" definidas por cuadros de textos. Nótese cómo en la Figura 1, el documento muestra los diferentes textos en dichos cuadros.

Figura 1. Cajas de texto



Solo se han encontrado algunos formatos no dispuestos en cuadros en algunos SIGEVA's que generalmente provienen de una transformación de *.pdf a pdfs gráficos o de usuarios que han creado cambios en el texto por medio de una transformación de *.doc a *.pdf. Es importante tomar esto en cuenta cuando se les pide a los docentes sus CV insistiendo en que estos sean guardados sin cambios después de generados. Estas transformaciones tienen que ser detectadas por el programa y descartar esos archivos.

Aprovechando este formato en cuadros, todo el sistema de extracción se define desde la posibilidad de encontrar los ejes "X" e "Y" que circunscriben el ángulo superior-izquierdo del mismo. La mejor extensión que presenta este formato "X/Y" es el *.xml ya que en sí es un archivo de tipo "marca".

Un archivo de texto en formato xml muestra el texto sin formato basándose en:

- Posición
- Formato del texto
- Texto a mostrar

Esto motivó que la primera tarea a realizar fuera la transformación de los documentos con extensión pdf a xml. Este pasaje de formatos se realiza con la aplicación pdftohtml.exe de uso abierto y manejable por PYTHON®.

El segundo proceso es el que deja solo un xml por persona: El algoritmo propuesto obtendrá el xml más reciente en su fecha de impresión y el que mejor responda al formato original del SIGEVA (o sea documentos no manipulados por fuera del que entrega el consorcio). todos aquellos xml's repetidos irán junto a los defectuosos al directorio "'RECICLAR". En tanto que todo xml que sea el último capturado por su fecha de impresión se pasará al directorio "XMLS". Seguidamente se parametriza cada texto en formato "Y-X" (posición en columna y posición en fila).

Luego de definir cuál es el xml que continuará el proceso por cada persona, la secuencia primero elimina encabezados y pie de página para evitar datos redundantes.

El tercer proceso se basa en la obtención de un archivo de extensión *.tmt (texto manipulado tercero[9]). La parte de mayor proceso y transformación del xml original se cumple en este proceso en donde si recorre el script se verá que se muestran los textos y sus secuencias por separado según el título que lo contenga en su CV-SIGEVA.

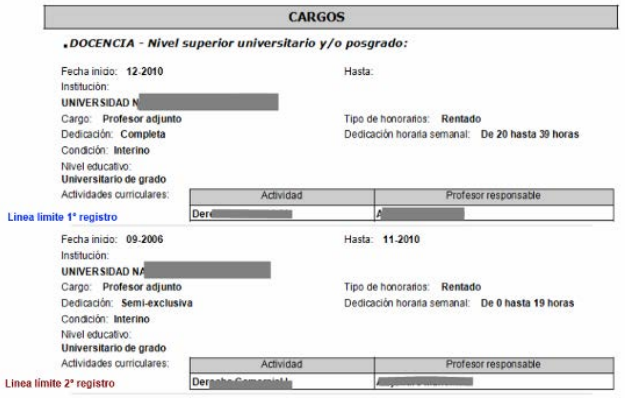
Se puede notar además que las páginas dejan de serlo ya que como se trata de un seguimiento vectorial no importa ya su presentación "formal" sino su posición de manera continua (como si todo el documento contara con solo una página). Un texto en posición "Y=200" en la página 10 y uno "Y=200" en página 2 deben tener en ese valor "y" algo que los diferencia en cuanto a su secuencia. Este sistema continuo hace que el número Y=200 de la página 2 sea tomado como el número 20200 ([nro_pag*10000]+Y) y el de página 10 sea Y=100200. Quedará claro que el primero está muy "arriba" en comparación al segundo. De esta manera se generan a

[9] son siglas que se eligieron sin otra idea a la de definir orden de actividad. Son simplemente archivos de textos con el orden correcto de "concepto->respuesta_usuario" para su posterior extracción. ejemplo : si un usuario puso en su CV-SIGEVA Apellido: García este *.tmt ordena de arriba hacia abajo primero que esté la palabra "Apellido:" y luego abajo "Garcia"

medida que el proceso continua una lista de "valor X, valor Y, texto extraído" separado por sus títulos.

Cuando se analizan todos los archivos *.tmt se observa que cada repetición (registro) tiene un primer CONCEPTO que limitará cada "ítem" generado por el usuario. Para clarificar más la idea, en la Figura 2 se muestra la captura de pantalla del documento original en el apartado CARGOS donde se ve que hay dos cargos de docencia superior y donde "Fecha Inicio" da comienzo al ítem particular (o sea cada futuro registro de la tabla específica).

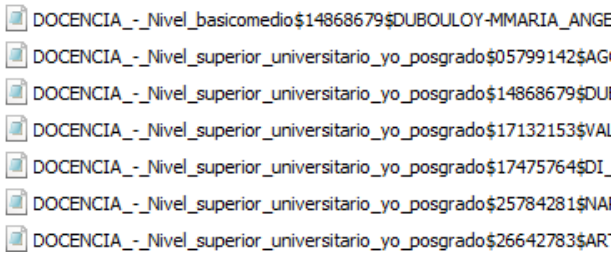
Figura 2. Delimitación de registros en un mismo título



Cada registro estará en este y todos los casos representado por una línea de registro en el documento pero, en el caso del *.tmt, ese límite estará dado por la repetición del CONCEPTO inicial del título analizado.

En la Figura 3 se puede observar lo importante que es contar con archivos *.tmt que tengan una secuencia inequívoca y repetible en todos los CV-SIGEVA's para cada título a analizar. Fíjese la siguiente ilustración para ver como el directorio guarda los archivos tmt.

Figura 3: nombres de los archivos *.tmt



Note que cada archivo contendrá los datos ordenados solo de "docencia…." para cada docente del que se analice su CV-SIGEVA.

El resultado final del script es lograr que esos *.tmt que contienen todos datos referidos a un título principal del CV-SIGEVA para cada persona puedan ser agrupados en un archivo *.txt que simbolice ese título principal.

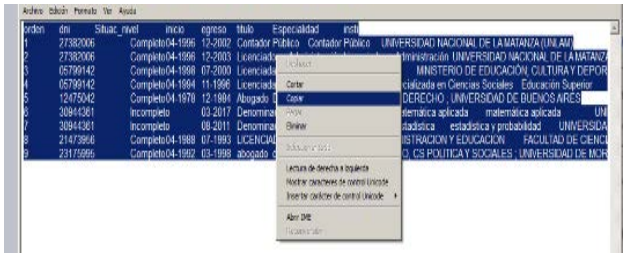
De este agrupamiento se crearán las tablas (archivos txt) para cada título principal. Los nombres de archivos y su explicación se pueden observar en la Figura 4:

Figura 4. Tablas obtenidas de los archivos *.tmt

ORDEN	ARCHIVO	SIGNIFICADO EN TITULOS DEL SIGEVA
1	act_docenciaCURS.txt	DOCENCIA - Cursos'
2	act_docenciaNBYM.txt	DOCENCIA - Nivel básico/medio'
3	act_docenciaNSTERC.txt	DOCENCIA - Nivel superior terciario'
4	act_docenciaNSUYOP.txt	DOCENCIA - Nivel superior universitario y/o posgrado'
5	activ_ACTEVPCYTYJDT.txt	ACTIVIDADES DE EVALUACION - Evaluación de personal CyT y jurado de tesis y/o premios'
6	activ_ACTEVPPIDYOEX.txt	ACTIVIDADES DE EVALUACION - Evaluación de programas/proyectos de I+D y/o extensión'
7	activ_ACTEVTREVCYT.txt	ACTIVIDADES DE EVALUACION - Evaluación de trabajos en revistas CyT'
8	cargos_ACTDIV.txt	ACTIVIDADES DE DIVULGACION'
9	cargos_GICYT.txt	CARGOS EN GESTION INSTITUCIONAL DE CYT'
10	categorizacion.txt	CATEGORIZACION DEL PROGRAMA DE INCENTIVOS'
11	cursextracurr.txt	FORMACION COMPLEMENTARIA - Cursos de posgrado y/o capacit. extracurriculares'
12	datos_personales.txt	DATOS PERSONALES
13	est_basicos.txt	'FORMACION ACADEMICA - Nivel básico'
14	est_espec.txt	FORMACION ACADEMICA - Nivel Universitario de Posgrado/Especialización'
15	est_maest.txt	'FORMACION ACADEMICA - Nivel Universitario de Posgrado/Maestría'
16	estudios_docto.txt	'FORMACION ACADEMICA - Nivel Universitario de Posgrado/Doctorado'
17	estudios_grado.txt	FORMACION ACADEMICA - Nivel Universitario de grado'
18	fin_art-lib-partlib-public-nopubl.txt	es unas sola base para LIBROS','PARTES DE LIBRO','ARTICULOS','TRABAJOS EN EVENTOS CIENTIFICO-TECNOLOGICOS PUBLICADOS','TRABAJOS EN EVENTOS CIENTIFICO-TECNOLOGICOS NO PUBLICADOS','DEMAS TIPOS DE PRODUCCION C-T','TESIS','SERVICIOS CIENTIFICO - TECNOLOGICOS','SIN TITULO DE PROPIEDAD INTELECTUAL','INFORME TECNICO'
19	financiacy_t.txt	FINANCIAMIENTO CIENTIFICO Y TECNOLOGICO'
20	formac_BECARIO.txt	FORMACION DE BECARIOS'
21	formac_DETESIS.txt	FORMACION DE TESIS'
22	formac_ECPODsalud.txt	FORMACION COMPLEMENTARIA - Especialidad certificada por organismo/s de salud'
23	formac_ECposdocto.txt	FORMACION COMPLEMENTARIA - Posdoctorado'
24	formac_IDIOM.txt	FORMACION COMPLEMENTARIA - Idiomas'
25	formac_INVESTIGADOR.txt	FORMACION DE INVESTIGADORES'
26	participUorganizCYT.txt	PARTICIPACION U ORGANIZACION DE EVENTOS CIENTIFICO-TECNOLOGICOS'
27	DESIGNACIONES.txt	Usa el cargo docente actual de la universidad que a ud interesa y define cargo, dedicación y año de extracción del dato

A modo de ejemplo, en la Figura 5 se muestra un archivo est_grado.txt y la manera que presenta datos donde se puede observar que cada registro posee una separación de elementos por medio de una tabulación:

Figura 5: datos dentro de una tabla en *.txt



4. Pruebas al script

El script ha sido probado con casi 500 archivos de texto en todas las extensiones posibles de documentos electrónicos. Aún los archivos que se descargaron sin extensión, pudieron ser transformados en *.xml.

En las primeras pruebas debió realizarse una selección para poder contar con archivos que respondieran a dos características:

4.1. fueran bajados directamente de la página CV-SIGEVA.(sin enmiendas o cambios).

4.2. no fueran documentos de MI CVar IMPRESO[10] u otros como viejas fichas CONEAU.

Pese a todas las protecciones, aún se trabaja sobre el segundo script para que solo pase el proceso archivos que sean bajados fehacientemente del sitio oficial. Las pruebas más importantes han sido llamadas hacia el interior del equipo de investigación como "pruebas de homologación de resultados".

5. Resultados de la homologación / Homologación 2018

Con el término "homologación" se quiere hacer referencia a la actividad de comparación de los datos presentes en el documento CV-SIGEVA y los obtenidos por el software APCA.

El procedimiento se ha decidido que sea manual por medio de dos personas, las cuales efectúan la comparación teniendo una de ellas el documento tal y como ha sido entregado por el usuario CV-SIGEVA y la otra con lo extraído en las tablas generadas por el programa. Para facilitar la actividad del operador que analiza las tablas se decidió que cada usuario contara con un archivo *.xls solo con sus valores para las diferentes tablas.

Esto último se ha hecho con motivos de agilizar la tarea ya que en las tablas el dominio del registro lo da el número DNI y eso llevaría a recordar dicho número en cada tabla abierta, llevando a posibles malas interpretaciones. De la manera propuesta, existiría un archivo llamado "JUAN_JOSE_MARENCO.xls" que contendría todos los datos extraídos desde las tablas particulares. Solo es necesario así que si se homologan los datos del usuario antedicho, solo hace falta contar con un numero documento original que pertenezca a este usuario y el archivo antes citado que presenta los datos de las tablas.

5.1 Resultados de la Homologación.

Dado el tamaño del universo (340 CV-SIGEVA sin repetir) y la cantidad de registros posibles en cada archivo *.xls, se consideró que era aconsejable comparar un 10% de ese total como primer acción y si aparecieran dudas o errores, se los subsanara en los scripts del APCA y se volvieran a analizar otro 10% y así hasta llegar a un análisis donde todos los valores fueran satisfactorios.

Esta primera homologación se realizó en septiembre de 2018 y de los 340 usuarios, se tomaron al azar 34 y se los comparó en todos sus parámetros. Solo se detectó un error (solucionado) en el caso que los registros en el documento original tuvieran dos títulos de tablas en hojas seguidas para un mismo registro.

[10] http://sicytar.mincyt.gob.ar/micvimpreso/#/ es un sitio que brinda un SIGEVA resumido. En este lugar también se puede seleccionar ítems de interés para el usuario.

6. Resultados

La investigación se encuentra hoy en día finalizada desde lo programático restando solo las tareas del desarrollo del informe final y la realización de las bases de datos comprometidas en la propuesta original.

Durante estos dos años se han desarrollado script que trabajan tanto sobre *.pdf como documentos originados en procesadores de texto tanto para CVAR, SIGEVA CONICET y SIGEVA(UNIV). Los mismos estarán disponibles para el acceso libre a partir del año 2020 fecha de caducidad de la investigación. Todas las distribuciones contarán con las libertades de uso y cambio del usuario según se establece en GLP3 con la única obligación -ante cualquier publicación abierta de mejoras- de la cita de todos los autores de esta investigación y del origen del software.

Referencias

[1] Challenger Perez, I., Diaz Ricardo, Y., & Becerra Garcia, R. (2014). El lenguaje de programación Python. Ciencias Holguín, volumen 20 - Nº 2. Recuperado el 10/11/2018, http://www.ciencias.holguin.cu/index.php/cienciasholguin/article/view/826/887

[2] Martinez, A., Turczak, P., Fillet, F., Cassino, J., & Faraldi, R. (2017). Administración operativa. Version 2.0. San Justo, Buenos Aires: Universidad Nacional de La Matanza.

[3] PDF2XL. (s.f.). Recuperado el 30 de 09 de 2018, de https://pdf2xl.com/ https://www.cogniview.com/es/pdf-to-excel/pdf2xl-basic

[4] Plaza, J. (2015). Administración y Gestión. Buenos Aires: Ediciones Plaza.

[5] Python, M. D. docs.pytho.org.ar/pdfs/TutorialPython2.pdf. (s.f.). Recuperado el 31 de 10 de 2018, de Comunidad Python Argentina: http://www.python.org.ar/

[6] Saunders, S. (2013). Sistemas y procedimientos administrativos. Cordoba: Asociacion Cooperadora de la Facultad de Ciencias Economicas de la UNC.

[7] Stair, R. M., & Reynolds, G. W. (2010). Principios de sistemas de información. Un enfoque administrativo. México: Cengage Learning Editores, SA de CV.

[8] Volpentesta, J. (2015). Organizaciones, procedimientos y estructuras. Buenos Aires: Osmar D. Buyatti.

[9] Waldbott de Bassenheim, C., Freijedo, C. F., Tricoci, G., Briano, J. C., & Rota, P. (2011). Sistemas de Información Gerencial. Tecnología para agregar valor a las Organizaciones. Buenos Aires: Pearson.