

Universidad ORT Uruguay
Facultad de Ingeniería

Búsqueda eficiente de información en documentos PDF para auditorías de farmacovigilancia

Entregado como requisito para la obtención del título de Analista Programador/Analista en
Tecnologías de la Información

Gabriel Pérez Pedetti – 303361

Gonzalo Ulivi Vidal - 288901

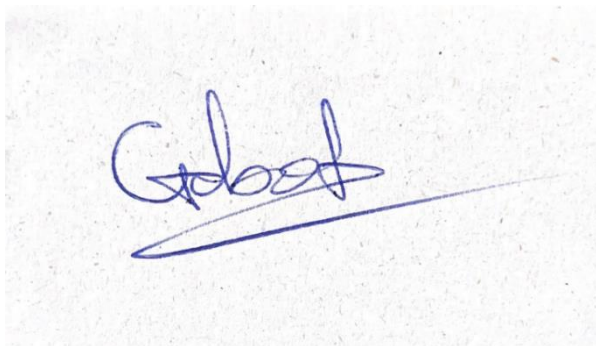
Tutor: Carlos Berruti

2024

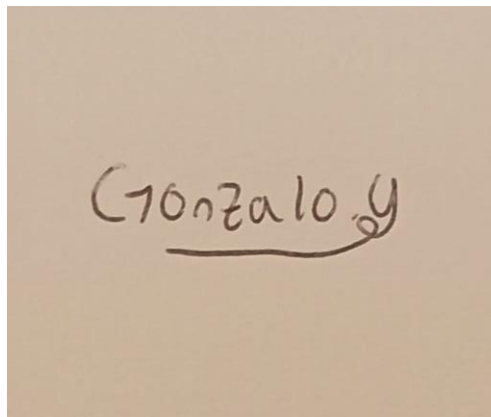
Declaración de autoría

Nosotros, Gabriel Pérez Pedetti y Gonzalo Ulivi Vidal, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos el Proyecto Integrador de la carrera Analista en Tecnologías de la Información;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.



Gabriel Pérez Pedetti
21 de octubre de 2024



Gonzalo Ulivi Vidal
21 de octubre de 2024

Abstract

En el presente documento se detalla el proyecto relacionado al software a realizar para hacer más eficiente uno de los procesos de negocio del departamento médico de MegaLabs Uruguay. Cuentan con una problemática que impide una revisión completa de cada documento necesario para las auditorías de farmacovigilancia correspondientes. Aquí presentamos una solución para evitar que esto siga sucediendo y garantizarles más seguridad y tranquilidad.

El producto para realizar se trata de un buscador de alrededor de mil de palabras clave en decenas de documentos PDF, rondando los cien. Esos documentos corresponden a cada artículo de cada revista científica nacional que es necesario leer para estas auditorías. Son diez revistas, y cada una cuenta con entre ocho y doce artículos aproximadamente.

Proponemos desarrollar un software con Python, utilizando una librería que permite el manejo de documentos PDF. Además de la búsqueda como funcionalidad principal, también tendrá un sistema de registro y autenticación de usuarios, de registro de búsquedas y de traducción de los artículos cuando sea necesario.

Elegimos utilizar una metodología de desarrollo iterativa donde estaremos mostrándole un mínimo producto viable al cliente con tal de saber su satisfacción y necesidades.

Luego de finalizado el proyecto, comentaremos nuestras conclusiones.

Palabras clave

Farmacovigilancia

Revista

Artículo

PDF

Python

SQL

Búsqueda

Palabras clave

Traducción

Web Scrapping

MegaLabs

Índice

Declaración de autoría.....	2
Abstract	3
Palabras clave.....	4
Índice.....	5
1. Introducción	7
2. Anteproyecto	8
2.1. Presentación del cliente	8
2.2. Presentación del problema.....	9
2.3. Alternativas y solución propuesta.....	10
2.3.1. Investigación y análisis previo	10
2.4. Lista de necesidades	11
2.5. Objetivos del proyecto.....	11
2.5.1. Objetivos generales y personales	11
2.5.2. Objetivos específicos del sistema.....	11
2.6. Actores involucrados	12
2.6.1. Personas.....	12
2.6.2. Entidades	12
2.7. Lista de requerimientos	12
2.7.1. Requerimientos funcionales	12
2.7.2. Requerimientos no funcionales	16
2.8. Alcances y limitaciones	16
2.9. Arquitectura	17
2.10. Plan de proyecto.....	17
2.10.1. Metodología / Marco de trabajo	17
2.10.2. Plan de gestión de riesgos.....	17
2.10.3. Plan de calidad (SQA)	21
2.10.4. Plan de configuración de software (SCM)	22
2.10.5. Plan de capacitación	22
2.10.6. Cronograma de trabajo y estimación de esfuerzo.....	22
2.10.7. Compromiso de proyecto.....	26
3. Proyecto.....	27
3.1. Introducción.....	27
3.2. Pila de producto final.....	27

3.3.	Diseño final de la solución	27
3.3.1.	Arquitectura final de la solución	27
3.4.	Desarrollo del proyecto	28
3.4.1.	Sprint 2 (dd-mm-yyyy / dd-mm-yyyy).....	28
3.4.2.	Sprint # (dd-mm-yyyy / dd-mm-yyyy).....	31
4.	Ejecución de los planes	32
4.1.	Introducción.....	32
4.2.	Plan de calidad.....	32
4.3.	Plan de configuración de software.....	32
4.4.	Plan de capacitación	32
4.5.	Plan de gestión de riesgos.....	32
5.	Conclusiones	33
5.1.	Introducción.....	33
5.2.	Grado de cumplimiento de objetivos.....	33
5.3.	Grado de satisfacción del cliente	33
5.4.	Lecciones aprendidas.....	33
5.5.	Posibles mejoras y desarrollos futuros	33
5.6.	Reflexiones finales	33
	Bibliografía.....	34
	Referencias bibliográficas	35
	Glosario	36
	Anexos.....	37
	Datos de acceso a la solución desplegada y repositorio GitHub.....	37
	Actas: Reuniones de validación de usuario.....	37
	Manual de deploy	37
	Manual de Usuario	37
	Inventario de recursos utilizados.....	37

1. Introducción

En el presente documento se detalla el proyecto relacionado al software a realizar para hacer más eficiente uno de los procesos de negocio del departamento médico de MegaLabs Uruguay. Cuentan con una problemática que impide una revisión completa de cada documento necesario para las auditorías de farmacovigilancia correspondientes. Aquí presentamos una solución para evitar que esto siga sucediendo y garantizarles más seguridad y tranquilidad.

El producto para realizar se trata de un buscador de cientos de palabras clave en un montón de documentos PDF. Esos documentos corresponden a cada artículo de cada revista científica nacional que se lee para estas auditorías. Son diez revistas, y cada una cuenta con entre ocho y doce artículos aproximadamente.

Proponemos desarrollar un software con Python, utilizando una librería que permite el manejo de documentos PDF. Además de la búsqueda como funcionalidad principal, también tendrá un sistema de registro y autenticación de usuarios, de registro de búsquedas y de traducción de los artículos cuando sea necesario.

Elegimos utilizar una metodología de desarrollo iterativa donde estaremos mostrándole un mínimo producto viable al cliente con tal de saber su satisfacción y necesidades.

2. Anteproyecto

2.1. Presentación del cliente

Megalabs es una empresa multinacional proveniente de capitales alemanes que tiene base en distintos países de Latinoamérica, Centroamérica y Estados Unidos. Megalabs Uruguay forma parte de ese grupo.

En Megalabs Uruguay hay una única gerencia encargada de los lineamientos de Mega, laboratorios Celsius, Haymann, Spedar y Servimedic. Funciona como un “hub”. Produce medicamentos, alimentos y cosméticos y son representantes de medicamentos que hacen otras fábricas de Mega, así también como de productos de otros laboratorios que no forman parte de Mega (empresas europeas o americanas). En estos productos se encuentran medicamentos-alimentos y dispositivos médicos (para salud y para cosmética).

Con quien nosotros tenemos contacto y será nuestro cliente y, a su vez, el usuario final del sistema resultante del proyecto es Mariana Pérez de Goldaracena, quien trabaja para Megalabs Uruguay. Es médica y se encuentra desarrollando su labor en la jefatura del departamento médico de la empresa. Llegamos a ella porque es familiar de Gabriel.

El departamento médico de Megalabs Uruguay está conformado por tres personas, entre ellas, Mariana. Algunas de sus responsabilidades son:

- Brindar apoyo desde el punto de vista de farmacovigilancia, donde se encuentra la problemática a resolver por nosotros en este proyecto.
- Brindar apoyo al sector regulatorio. Implica el registro de todos estos productos ante el Ministerio de Salud Pública, quien es su autoridad regulatoria, y la renovación de estos (cada tres años).
- Brindar apoyo a las líneas de marketing de cada laboratorio con la evidencia científica. Aportar estrategias para la promoción del producto, donde la promoción médica va a depender del tipo de medicamento desde el punto de vista regulatorio. Si se puede hacer promoción en la televisión o si se puede hablar del medicamento con el profesional de la salud.
- Presentar todos los años nuevas solicitudes de incorporación al FTM (Formulario Terapéutico Medicamento) o actualizaciones de información al FTM.
- Actualización anual de los productos de Mega y sus laboratorios asociados en el Farmanuario, libro donde se incluyen todos los medicamentos-alimentos y sus marcas.
- Consultas médicas. Por ejemplo, si un médico precisa información científica de algo, el departamento médico se la otorga.
- MegaLabs Uruguay tiene dos callcenters donde el paciente puede llamar. Un 0800 “diabetes” que es interno y un 0800 externo. En el departamento médico se encargan de entrenar con cierta periodicidad a las personas que atienden y ellos contestan según los prospectos, puesto que legalmente no pueden hacer una consulta médica al paciente que llama. Ante casos más complejos, derivan a quien corresponda.
- Si un producto muestra riesgos en los controles de calidad, evitar su comercialización.

Como mencionamos, nos estaremos enfocando en el primer punto. La farmacovigilancia implica la seguridad de los medicamentos. Minimizar los riesgos de los eventos que se saben que pueden suceder o de los que no se sabe que suceden y, efectivamente, suceden.

2.2. Presentación del problema

La farmacovigilancia es la ciencia que se encarga de la detección, evaluación, comprensión y prevención de los problemas relacionados con el empleo de medicamentos, incluyendo los efectos adversos. Los principales objetivos de la farmacovigilancia son, por un lado, prevenir el daño que podrían causar a los pacientes las eventuales reacciones adversas a los medicamentos comercializados y por otro, promover el uso seguro, efectivo y racional de los medicamentos. Esto se obtiene entregando información actualizada sobre el empleo, las indicaciones y la seguridad de los productos.

Dentro de las tareas periódicas que deben hacer en la empresa ante auditorías de farmacovigilancia, está la revisión de la bibliografía científica nacional. En cada revisión, deben controlar si de alguno de todos los productos que comercializan, es decir, de Megalabs o de alguno de los otros laboratorios que representan, se publicó algo referente a farmacovigilancia, como lo puede ser un evento reportable. Un evento reportable es un evento vinculado al uso de un producto, que ha sido transmitido a cualquier funcionario de la compañía por un profesional de la salud, por un paciente o cualquier persona que utilice ese producto. Pueden ser tales como una reacción adversa, uso en el embarazo, falta de eficacia, uso fuera de las indicaciones aprobadas, entre otros.

La bibliografía para revisar consta de diez revistas científicas. Cada una se actualiza con distinta periodicidad. Tienen varios artículos en formato PDF, que son los que se van a inspeccionar. Esto implica leer cada uno manualmente y la búsqueda se hace muy lenta, incluso a veces no se llegan a revisar todos los artículos por completo. Se le da prioridad a buscar sobre los productos más importantes. Cuentan con aproximadamente mil palabras clave a tener presente, entre nombres de los fármacos, nombres comerciales y otros términos para identificar determinado contenido.

En el caso de hallar información de farmacovigilancia sobre algún producto, tienen que ponerse en contacto con quien corresponda según el laboratorio que comercialice el mismo. Esto conlleva: tomar la parte del artículo que contenga la información, traducirla al inglés y enviársela a quien respecta. A su vez, también tiene que reportárselo al Ministerio de Salud Pública. Nuevamente, es un proceso que se hace de forma manual; usando las habilidades de cada uno para traducirlo, copiando y pegándolo en el traductor de Google o incluso usando IA como Chat-GPT.

Actualmente, no están usando ningún software específico para este proceso. Además, nuestro cliente también nos comentó un poco del entorno tecnológico de la empresa: cuentan con computadoras de escritorio para usar en el día a día en las oficinas de la empresa. Sin embargo, menciona que el área tecnológica está un poco precaria. También hizo alusión a un software de gestión que usan para registrar cierta información de la empresa, llamado Enaxis.

En todo este proceso, vemos una gran oportunidad de automatización. La idea es desarrollar un programa para acelerar esta búsqueda y que no sea tan manual, y asegurarse de revisar absolutamente todos los productos correspondientes, así como también facilitar la traducción al inglés en caso de que sea necesario.

2.3. Alternativas y solución propuesta

Existen herramientas que se podrían usar para hacer parte de lo que el cliente pide, como por ejemplo ChatPDF. Esta página permite cargar un archivo PDF y entablar una conversación con un “chatbot” al que se le pueden hacer preguntas sobre el contenido del documento. Sin embargo, su versión gratuita está limitada en cantidad de páginas y tamaño del archivo. Es aquí cuando el cliente tendría que pagar una suma de dinero. Además, tendría que cargar cada archivo PDF uno por uno y luego tendría que ir buscando cada palabra clave manualmente, entonces dejaría de ahorrarle trabajo.

No existe ningún producto que haga exactamente lo que precisa el cliente, ya que las herramientas que pueden leer un archivo PDF no tienen ninguna forma de añadir palabras clave específicas o de dejar una nota con la información que se requiere cuando una búsqueda es exitosa.

Tomamos la decisión de usar Python como lenguaje principal debido a que contiene una librería que va a servir para manejar documentos de tipo PDF y tendremos en cuenta un período de capacitación. Usaremos SQL como base de datos. Haremos un deploy en la web debido a que la clienta mencionó que le gustaría poder hacer este trabajo desde cualquier lugar y no depender de las computadoras de escritorio en la empresa. Por dar un ejemplo: si hubiese una pandemia y no se pudiera concurrir a las oficinas, se podría seguir realizando el proceso sin ningún impedimento, puesto que va a poder acceder al software desde su computadora personal.

2.3.1. Investigación y análisis previo

2.3.1.1. Automatización de la obtención de documentos

Lo ideal sería que el sistema se encargue de obtener los PDF a analizar de forma automática cuando el cliente se lo pide. Es por esto por lo que decidimos investigar las URLs de cada revista para intentar conseguirlos a través de ellas. Ahí nos percatamos de que las URLs eran muy volátiles; dentro de los parámetros de la URL habían IDs vacíos, algunos PDFs no se identificaban por un ID numérico sino por el nombre del artículo y eso significa que varía con cada uno de ellos, entre otras cosas. Llegamos a la conclusión de que no íbamos a lograr automatizarlo de esa manera.

Es entonces que decidimos hacer un web scrapping. Revisamos cada página web de cada revista, y nos dimos cuenta de que, a través de su HTML, podíamos encontrar los enlaces para acceder y/o descargar los archivos PDF correspondientes, entonces no habría ningún problema como IDs “salteados” o los mencionados anteriormente. Evitaríamos eso y podríamos obtener los archivos de forma automatizada.

2.4. Lista de necesidades

Identificamos las siguientes necesidades:

- Cargar un listado de palabras clave.
- Realizar una búsqueda de las palabras clave en un documento de forma rápida.
- Registrar las búsquedas para luego acceder a ellas con mayor rapidez y facilidad.
- La posibilidad de ponerle un nombre específico a la búsqueda.
- La opción de dejar un comentario más extenso, por ejemplo, sobre el análisis de los resultados.
- Que el sistema sea una aplicación web.
- Autenticación de usuarios.
- La posibilidad de traducir un artículo a elección.

2.5. Objetivos del proyecto

2.5.1. Objetivos generales y personales

El proyecto busca hacer más eficiente uno de los procesos de búsqueda de información que tiene el departamento médico de MegaLabs Uruguay, en este caso la indagación sobre información de farmacovigilancia de los productos representados y/o fabricados por la empresa en las revistas científicas nacionales. Con esto, también se asegurarían de revisar toda la documentación disponible para llegar a tiempo con las auditorías de farmacovigilancia y no tener que dejar algún artículo a “medio revisar”.

Queremos hacer el trabajo lo mejor posible, ya que sabemos que el departamento médico de la empresa tiene varias carencias en cuanto al software que se utiliza y, si logramos la satisfacción del cliente, podríamos continuar nuestra relación y desarrollar más aplicaciones para ellos luego.

Además, nos entusiasma el hecho de investigar y aprender una nueva tecnología como lo es Python, que sabemos que es muy usada en el área de TI.

2.5.2. Objetivos específicos del sistema

Implementar un sistema que:

- Permita la carga de cientos de palabras claves distintas, superando la barrera de las mil.
- Sea capaz de realizar una búsqueda de esas palabras clave anteriormente registradas por el usuario en un documento PDF en pocos minutos.
- Sea capaz de traducir un artículo y generarlo en otro PDF descargable para el usuario.

2.6. Actores involucrados

2.6.1. Personas

- Mariana Pérez, quien sería usuario final del sistema y además nuestra referencia para saber qué quiere en el mismo, cómo lo quiere y tomar decisiones al respecto.
- Leticia Carballal, quien sería otro usuario final del sistema.
- Iván Lago, quien sería otro usuario final del sistema.
- Gabriel Pérez y Gonzalo Ulivi, quienes seríamos los encargados de planificar y gestionar el proyecto y de desarrollar e implementar el software luego.
- El responsable local de farmacovigilancia de cada laboratorio dentro de MegaLabs Uruguay.

2.6.2. Entidades

- MegaLabs Uruguay
- Spefar
- Celsius
- Servimedic
- Haymann

2.7. Lista de requerimientos

2.7.1. Requerimientos funcionales

ID	RF01
Título	Registro de usuarios
Descripción	Permite dar de alta un nuevo usuario en el sistema, otorgando un nombre de usuario y una contraseña como credenciales.
Prioridad	Alta

ID	RF02
Título	Autenticación y autorización (login)
Descripción	Permite iniciar sesión usando las credenciales de un usuario existente en el sistema.
Prioridad	Alta

ID	RF03
Título	Cerrar sesión (logout)
Descripción	Permite cerrar la sesión del usuario.
Prioridad	Alta

ID	RF04
Título	ABM de palabras clave
Descripción	Permite al usuario logueado dar de alta, modificar y eliminar palabras clave dentro del listado de estas.
Prioridad	Alta

ID	RF05
Título	Visualizar listado de palabras clave
Descripción	Otorga al usuario logueado acceso a todo el listado de palabras clave, donde se permitirá seleccionar una de ellas para eliminarla o modificarla, así como también permitirá agregar una nueva.
Prioridad	Alta

ID	RF06
Título	Carga de documentos
Descripción	El sistema deberá permitir la carga de documentos con la extensión .pdf.
Prioridad	Alta

ID	RF07
Título	Búsqueda de palabras clave en un documento

Descripción	El sistema deberá ser capaz de realizar una búsqueda de las palabras claves registradas por el usuario en cada documento pdf cargado en el mismo.
Prioridad	Alta

ID	RF08
Título	Mostrar resultados de la búsqueda
Descripción	El sistema resalta las palabras clave que hayan sido encontradas en el documento donde se efectuó la búsqueda.
Prioridad	Alta

ID	RF09
Título	Asignar nombre a la búsqueda
Descripción	El sistema permitirá asignarle un nombre a la búsqueda realizada. Sería para que no quede identificado solo con el nombre del archivo original, dando la posibilidad de renombrarlo a preferencia del usuario para hacerlo más específico/entendible.
Prioridad	Media

ID	RF10
Título	Agregar comentario a la búsqueda
Descripción	El sistema dará la opción al usuario de escribir una reseña (ejemplo: para escribir un pequeño análisis sobre lo que encontró).
Prioridad	Media

ID	RF11
Título	Guardar búsqueda en la base de datos
Descripción	El sistema guardará la búsqueda realizada en la base de datos con los siguientes datos: fecha y hora de realización, usuario que la realizó,

	nombre, comentario/reseña, y una lista de las palabras claves encontradas.
Prioridad	Alta

ID	RF12
Título	Visualizar búsquedas anteriores
Descripción	Permite ver un listado de las búsquedas realizadas por el usuario ordenadas por fecha, mostrando el nombre. Puede seleccionar una para verla en detalle.
Prioridad	Media

ID	RF13
Título	Detalle de búsqueda anterior
Descripción	Permite, luego de seleccionar la búsqueda, ver el nombre, la fecha, la reseña y el listado de palabras encontradas. También permite la traducción de este.
Prioridad	Media

ID	RF14
Título	Traducir artículo
Descripción	Luego de una búsqueda, permite la opción de traducir el documento PDF (artículo). Esto devolvería un PDF que sería la traducción al inglés del archivo cargado en el que se efectuó la búsqueda. El sistema da la opción al usuario de descargarlo.
Prioridad	Media

ID	RF15
Título	Carga automatizada de documentos

Descripción	Ya que tendremos las URLs correspondientes a cada revista que el cliente necesita analizar, el sistema tendrá la posibilidad de obtener los PDFs necesarios de forma automatizada a través de un web scrapping.
Prioridad	Media

2.7.2. Requerimientos no funcionales

ID	RNF01
Título	Conexión con base de datos SQL
Descripción	Los datos deberán guardarse en una base de datos SQL.
Prioridad	Alta

ID	RNF02
Título	Aplicación web
Descripción	El sistema deberá tratarse de una aplicación web, con tal de poder acceder desde cualquier computadora.
Prioridad	Alta

2.8. Alcances y limitaciones

Decidimos desarrollar la aplicación cumpliendo con la totalidad de los requerimientos listados anteriormente. Creemos que el tiempo con el que contamos para la investigación y el desarrollo de este sistema es suficiente para realizarla.

En cuanto al requerimiento RF15, referido a la automatización de la obtención y carga de los PDFs de todos los artículos de todas las revistas: intentaremos realizarlo para todas las revistas posibles. Puede que alguna revista presente mayores dificultades o incluso un impedimento. En este caso, ya acordamos con el cliente que ellos no tendrían problema de descargar los documentos correspondientes y cargarlos al sistema manualmente.

Cualquier cambio en los requerimientos, que extienda o acorte el alcance definido, será debidamente documentado para una futura implementación durante el desarrollo del proyecto.

2.9. Arquitectura

Se tratará de una aplicación web que será desarrollada en Python y estará conectada a una base de datos SQL que será alojada en AWS o Azure. Esto último lo definiremos luego de tener los talleres de deploy.

2.10. Plan de proyecto

2.10.1. Metodología / Marco de trabajo

Aprovechando nuestra cercanía con el cliente y su compromiso con que salga adelante este proyecto, optamos por elegir una metodología iterativa donde podamos mostrarle los avances del proyecto periódicamente y obtener retroalimentación.

En cada sprint, iremos presentando un MVP (Producto Viable Mínimo) con alguna funcionalidad más que en el anterior y tomaremos en cuenta la opinión y la satisfacción o insatisfacción de nuestro cliente.

2.10.2. Plan de gestión de riesgos

Estos son los riesgos que hemos logrado identificar hasta la fecha. Más adelante, se encuentran detallados los planes de mitigación y/o de contingencia si lo hubiese para cada riesgo.

ID	R1
Título	Errores en la planificación
Descripción	Esta es nuestra primera experiencia realizando un proyecto de esta magnitud y planificándolo desde cero. Es probable que haya imprecisiones en las estimaciones de esfuerzo para la realización del producto. Esto puede acarrear demoras.
Probabilidad	Medio
Impacto	Medio
Nivel de exposición	Medio

ID	R2
Título	Cambios en los requerimientos

Descripción	Aunque ya los hemos definido para el proyecto, es factible que surjan nuevos requerimientos a partir de nuevas necesidades del cliente o de los desarrolladores.
Probabilidad	Medio
Impacto	Medio
Nivel de exposición	Medio

ID	R3
Título	Dificultad aprendiendo nueva tecnología
Descripción	Durante este proyecto estaremos aprendiendo a usar Python. Existe la posibilidad de que se nos complique su curva de aprendizaje y demore el proyecto.
Probabilidad	Medio
Impacto	Alto
Nivel de exposición	Alto

ID	R4
Título	Cambios en las páginas web de las revistas
Descripción	Si cambia el formato de las páginas web de las revistas científicas, la automatización de la obtención y carga de los PDFs dejaría de funcionar.
Probabilidad	Bajo
Impacto	Alto
Nivel de exposición	Medio

ID	R5
-----------	-----------

Título	Cambios en el formato de los documentos
Descripción	Puesto que vamos a usar una librería específica para tratar con documentos de tipo PDF, si el día de mañana los artículos de las revistas dejan de estar en este formato, habría mucho retrabajo en el proyecto.
Probabilidad	Bajo
Impacto	Alto
Nivel de exposición	Medio

ID	R6
Título	Barreras ante el web scrapping
Descripción	Si la página de una revista presenta un HTML dinámico para evitar el web scrapping, entonces ya no podríamos obtener sus PDFs de forma automatizada y habría que descargarlos y cargarlos manualmente en el sistema.
Probabilidad	Medio
Impacto	Medio
Nivel de exposición	Medio

2.10.2.1. Planes de mitigación y contingencia

ID	R1
Título	Errores en la planificación
Plan de mitigación	Estaremos constantemente monitoreando nuestro calendario de planificación y comparándolo con nuestro proceso de desarrollo con el objetivo de no atrasarnos.
Plan de contingencia	No aplica.

ID	R2
Título	Cambios en los requerimientos
Plan de mitigación	Definir los requerimientos de la mejor forma posible desde un principio e involucrar al cliente de forma temprana.
Plan de contingencia	Reestimar el requerimiento, evaluar su impacto y definir con el cliente si el cambio es viable o no.

ID	R3
Título	Dificultad aprendiendo nueva tecnología
Plan de mitigación	Estaremos en constante capacitación.
Plan de contingencia	No aplica

ID	R4
Título	Cambios en las páginas web de las revistas
Plan de mitigación	No aplica
Plan de contingencia	Evaluar si con el nuevo HTML de la página es viable continuar con una automatización y reestimar el esfuerzo del requerimiento.

ID	R5
Título	Cambios en el formato de los documentos
Plan de mitigación	No aplica.
Plan de contingencia	Investigar si se puede usar otra librería según el nuevo formato, o si se puede trabajar con texto plano. Cambiaría mucho la estructura de la aplicación.

ID	R6
Título	Barreras ante el web scrapping
Plan de mitigación	Ser cautelosos e intentar tomar patrones del HTML que no presenten dinamismo.
Plan de contingencia	Si no logramos obtener los PDFs de alguna revista, no se automatiza y el cliente se encargará de cargarlos manualmente al sistema.

2.10.3. Plan de calidad (SQA)

2.10.3.1. Estándares definidos

En los momentos en los que nos encontremos ambos desarrollando, estaremos en llamada con tal de evacuar dudas y ayudarnos entre nosotros de forma rápida. Definiremos desde un principio en qué se va a enfocar cada uno durante el día.

Al finalizar cada sprint, nos tomaremos un tiempo para detectar errores, posibles oportunidades de mejora y completar la documentación correspondiente.

Durante el transcurso del proyecto, utilizaremos los siguientes estándares:

- Documentación según la normativa que impone la facultad para la presentación de trabajos finales de carrera: Documento 302-ET, 303 y 306.
- UML para diagramas de clase.
- Diagramas de Casos de Uso.
- Camel case para la nomenclatura de clases y variables.

2.10.3.2. Testing

Dado que optamos por una metodología de tipo iterativa, nuestro objetivo está en presentar un MVP en cada sprint, por lo que realizaremos diferentes tipos de pruebas de todos los casos de uso implementados en el correspondiente sprint.

Ejecutaremos los siguientes tipos de pruebas:

- Pruebas unitarias: las realizaremos sobre el requerimiento que estemos desarrollando. Son realizadas por los desarrolladores.
- Pruebas de integración: las efectuaremos previo a liberar un nuevo incremento, es decir, antes de finalizar el sprint.
- Pruebas de aceptación: el cliente hará pruebas sobre el producto de la última iteración para validar y otorgar retroalimentación al equipo de desarrollo.

2.10.4. Plan de configuración de software (SCM)

Teniendo en cuenta nuestra experiencia, utilizaremos Git como sistema de control de versionado y Github para alojar los repositorios privados. Trabajaremos sobre dos ramas: una en la que iremos haciendo merge de las funcionalidades ya probadas y funcionando (lo que comúnmente se llamaría “producción”) y otra donde estaremos constantemente desarrollando funcionalidades nuevas y testeándolas.

Con respecto a la documentación, la estaremos alojando en Google Drive, con tal de ambos tener la última versión de esta.

Por último, para llevar rastro de las horas trabajadas en el proyecto, usaremos Toggl. Esta herramienta nos permite prender un cronómetro cada vez que nos sentemos a dedicarle horas al proyecto y especificar para qué son. Nos permite “trackear” nuestro tiempo de trabajo.

2.10.5. Plan de capacitación

2.10.5.1. Capacitación del equipo de desarrollo

Nos dedicaremos a investigar por nuestra cuenta, sea viendo videos, cursos, leyendo foros o incluso preguntándole a gente con experiencia en el lenguaje.

2.10.5.2. Capacitación del cliente

Nuestra idea es que el producto final sea lo más intuitivo y sencillo posible, a pedido del cliente. Él mismo irá viendo en cada sprint cómo va evolucionando el producto y ya lo irá conociendo. Igualmente, cuando la aplicación esté terminada, le explicaremos cómo utilizarla y le evacuaremos sus dudas. Nos reuniremos presencialmente.

2.10.6. Cronograma de trabajo y estimación de esfuerzo

Definimos el cronograma de trabajo en base a 7 sprints. El primero involucra todo el anteproyecto y tuvo una duración de 44 días. Del segundo al sexto sprint, tienen una duración de 14 días aproximadamente, exceptuando el quinto que abarca 22 días. Por último, el séptimo y último sprint conlleva 25 días e implica la recta final del proyecto. En su totalidad, estimamos 532 horas de proyecto.

Sprint 1 – Anteproyecto (62hs en total)	
Inicio: 07/09/2024	Fin: 21/10/2024
Descripción de la tarea	Horas/hombre efectuadas
Reuniones con el cliente	3hs
Tutorías	2hs

Análisis de la solución	5hs
Documentación y gestión	9hs
Clases guía	12hs
Total	31hs/hombre

Sprint 2 – Investigación sobre las tecnologías (86hs)	
Inicio: 22/10/2024	Fin: 04/11/2024
Descripción de la tarea	Horas/hombre estimadas
Clases sobre deploy	21hs
Tutorías	2hs
Capacitación sobre Python	15hs
Documentación y gestión	5hs
Total	43hs/hombre

Sprint 3 – Creación del sistema y primeras funcionalidades (68hs)	
Inicio: 05/11/2024	Fin: 19/11/2024
Descripción de la tarea	Horas/hombre estimadas
Tutorías	2hs
Creación y conexión de la base de datos	4hs
Creación del repositorio	2hs
ABM de usuarios y login (RF01, RF02, RF03)	5hs
Carga de documentos PDF (RF06)	10hs
Pruebas	3hs
Reuniones con el cliente	1hs

Documentación y gestión	7hs
Total	34hs/hombre

Sprint 4 – Búsqueda de palabras clave (72hs)	
Inicio: 20/11/2024	Fin: 04/12/2024
Descripción de la tarea	Horas/hombre estimadas
Tutorías	3hs
ABM de palabras clave (RF04)	4hs
Visualizar listado de palabras clave (RF05)	2hs
Búsqueda de palabras clave en un documento (RF07)	12hs
Pruebas	7hs
Reuniones con el cliente	1hs
Documentación y gestión	7hs
Total	36hs/hombre

Sprint 5 – Resultados de búsqueda (78hs)	
Inicio: 05/12/2024	Fin: 27/12/2024
Descripción de la tarea	Horas/hombre estimadas
Tutorías	4hs
Mostrar resultados de la búsqueda (RF08)	9hs
Asignar nombre (RF09)	2hs
Agregar comentario (RF10)	2hs
Guardar búsqueda en la base de datos (RF11)	3hs
Visualizar búsquedas anteriores (RF12)	3hs

Detalle de búsqueda anterior (RF13)	2hs
Pruebas	7hs
Reuniones con el cliente	1hs
Documentación y gestión	6hs
Total	39hs/hombre

Sprint 6 – Traducción de un artículo (70hs)	
Inicio: 03/01/2025	Fin: 16/01/2025
Descripción de la tarea	Horas/hombre estimadas
Tutorías	3hs
Investigar cómo y traducir un artículo (RF14)	12hs
Comenzar a investigar sobre web scrapping	7hs
Pruebas	5hs
Reuniones con el cliente	1hs
Documentación y gestión	7hs
Total	35hs/hombre

Sprint 7 – Carga automatizada de documentos y finalización (96hs)	
Inicio: 17/01/2025	Fin: 11/02/2025
Descripción de la tarea	Horas/hombre estimadas
Tutorías	3hs
Cargar los PDF de forma automatizada (RF15)	20hs
Pruebas	8hs
Reuniones con el cliente	2hs

Documentación y gestión	15hs
Total	48hs/hombre

2.10.7. Compromiso de proyecto

Nosotros los desarrolladores, Gonzalo y Gabriel, nos comprometemos a dedicar al menos 30 horas semanales (15 horas cada uno) durante la realización del proyecto. No podemos especificar cuántas horas vamos a dedicar cada día de la semana puesto que depende de otros factores como si alguno comienza a trabajar, que tengamos otros compromisos, festividades, entre otros.

En estas horas están incluidas la capacitación e investigación para lo que corresponda, el diseño, el desarrollo efectivo, el testing, las reuniones con el tutor y con el cliente, así como la capacitación, la gestión del proyecto y la documentación.

Nuestro cliente ya ha expresado una gran disponibilidad para contactarse con nosotros. Tenemos una comunicación rápida por mensaje directo vía Whatsapp, por lo que es fácil organizar las reuniones necesarias.

3. Proyecto

3.1. Introducción

Que se va a ver en esta sección

3.2. Pila de producto final

De lo planificado originalmente como quedo la pila de producto final (salió algo entro algo, sufrió cambios el alcance).

3.3. Diseño final de la solución

3.3.1. Arquitectura final de la solución

Diagrama final de la solución y breve explicación de sus componentes y servicios. Si hay otros diagramas (deploy, componentes, MER o tablas, etc.).

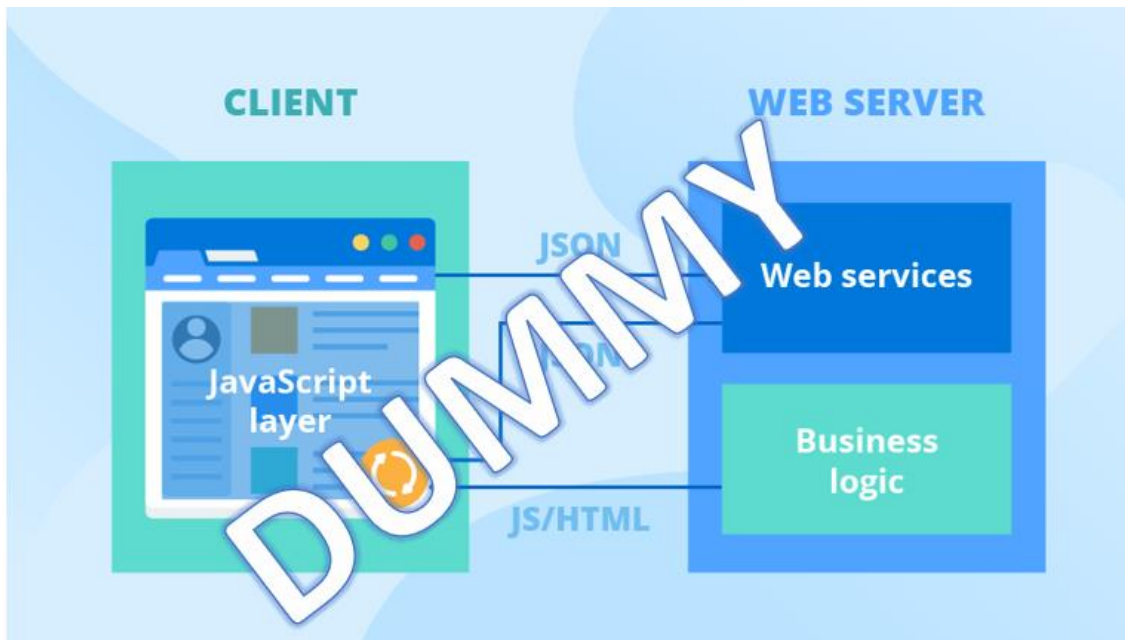


Ilustración 2. Diagrama final de la solución

3.4. Desarrollo del proyecto

3.4.1. Sprint 2 (dd-mm-yyyy / dd-mm-yyyy)

3.4.1.1. Planificación del sprint

El objetivo principal de este sprint

3.4.1.2. Tareas completadas/realizadas

Nombre de la tarea	Horas estimadas	Horas totales	Estado
Tarea 1			Finalizado
Tarea 2			En Proceso
Tarea 3			Replanificada para SP#
Tarea 4			Eliminada del alcance
Tarea 5			
Tarea 6			
Tarea #			
Total ### ###			

Comentar cualquier cambio o ajuste a las tareas que se planifico originalmente, si surgieron nuevos requerimientos comentar, pero detallar en apartado nuevos requerimientos.

3.4.1.3. Tareas no completadas/realizadas

Que tareas de las planificadas o replanificadas para este sprint no se pudieron terminar o realizar. En caso de que no ocurra se elimina titulo o se coloca No Aplica en la descripción.

3.4.1.4. Entrega del trabajo a los interesados

¿Hubo entrega o muestra al cliente? ¿Se realizaron UATs? ¿Cuál fue el feedback? ¿Cuál fue el resultado de la muestra o UAT? Agregar registros de pruebas en un Anexo y referenciarlo desde aquí. En caso de que no ocurra se elimina titulo o se coloca No Aplica en la descripción.

3.4.1.5. Cierre/revisión del sprint

¿Se logro los objetivos del sprint? ¿En qué grado? Comentario general del desempeño.

3.4.1.5.1. Desvíos y gestión de riesgos

¿Se materializaron riesgos? ¿Como los gestionaron? ¿Qué impactos genero? En caso de que no ocurra se elimina titulo o se coloca No Aplica en la descripción.

3.4.1.5.2. Nuevos requerimientos surgidos

- Descripción de los nuevos requerimientos que surgieron y se agregaron a este sprint
- En caso de que no ocurra se elimina título o se coloca No Aplica en la descripción.

3.4.1.5.3. Ajuste del plan de trabajo del próximo sprint

En función a como se desarrolló este sprint si surgieron cambios en la planificación para el próximo sprint detallar los cambios (tanto si por ejemplo se adelantó trabajo y por ende una tarea que estaba planificada para el próximo sprint se hizo en este como si surgieron requerimientos nuevos o se postergaron tareas de este para el próximo). En caso de que no ocurra se elimina título o se coloca No Aplica en la descripción.

Tareas que se quitan del próximo sprint:

- Tarea # - (##hrs)

Tareas que se agregan al próximo sprint:

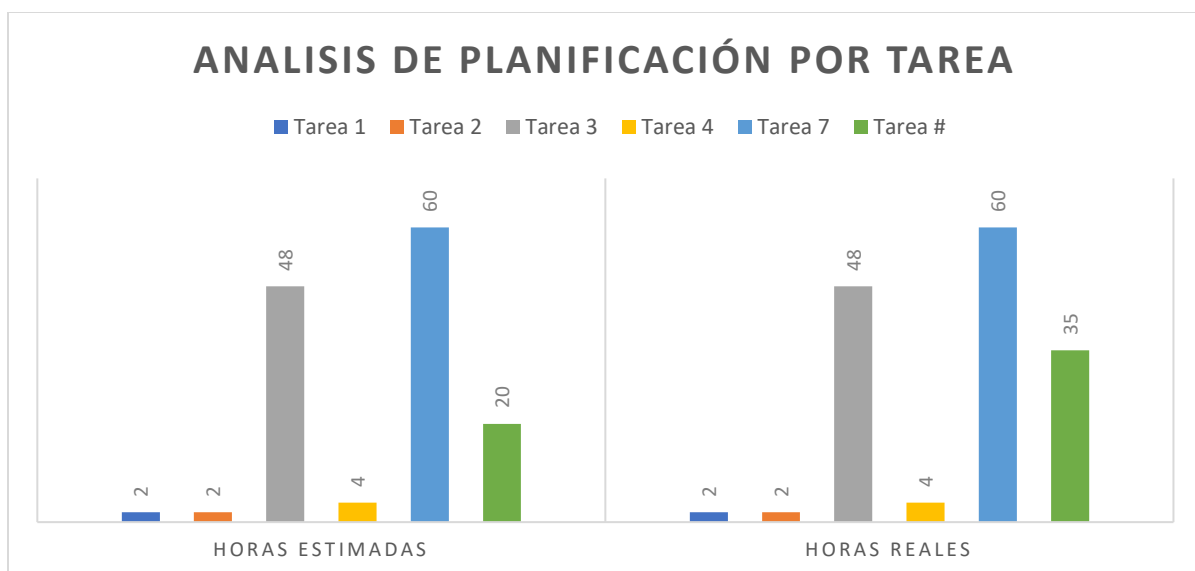
- Tarea ## (#hrs)
- Tarea ### (##hrs)

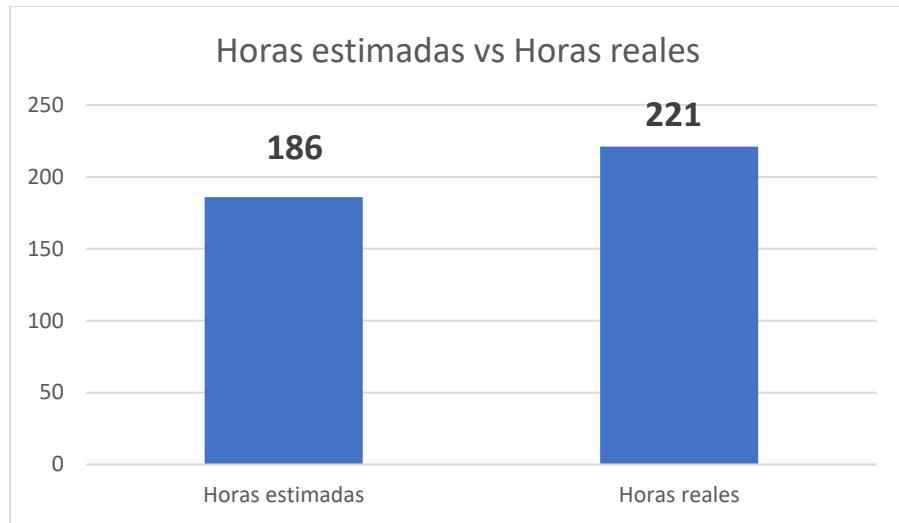
Tareas reestimadas:

- Tarea #####, explicar razón (#hrs).

Si tenía horas de reserva y se utilizaran comentarlo.

3.4.1.5.4. Resumen del esfuerzo dedicado en el sprint





Incluir comentarios/observaciones sobre el desempeño del sprint.

3.4.1.5.5. Retrabajo derivado de los problemas no resueltos del sprint anterior

Si por ejemplo se encuentra problemas derivados de un error y cambio explicar.

3.4.1.5.6. Gráfica de avance

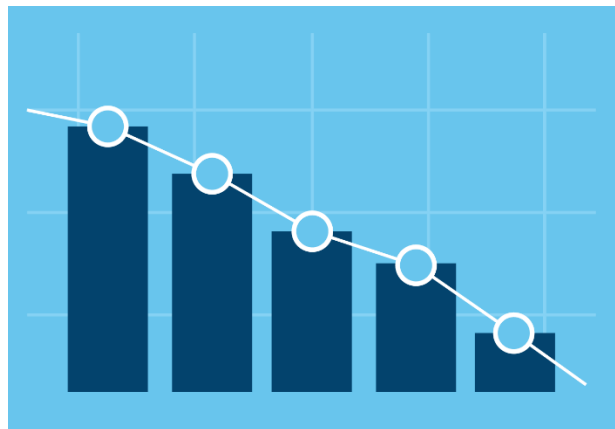


Ilustración 3. Burndown - Sprint 2

3.4.2. Sprint # (dd-mm-yyyy / dd-mm-yyyy)

3.4.2.1. Planificación del sprint

El objetivo principal de este sprint

3.4.2.2. Tareas completadas/realizadas

Nombre de la tarea	Horas estimadas	Horas totales	Estado
Total	##	##	

3.4.2.3. Tareas no completadas/realizadas

3.4.2.4. Entrega del trabajo a los interesados

3.4.2.5. Cierre/revisión del sprint

3.4.2.5.1. Desvíos y gestión de riesgos

3.4.2.5.2. Nuevos requerimientos surgidos

3.4.2.5.3. Ajustes del plan de trabajo del próximo sprint

3.4.2.5.4. Resumen del esfuerzo dedicado en el sprint

3.4.2.5.5. Retrabajo derivado de los problemas no resueltos en el sprint anterior

3.4.2.5.6. Gráfica de avance

4. Ejecución de los planes

4.1. Introducción

Explicar cómo se llevaron a cabo los planes y si sufrieron modificaciones.

4.2. Plan de calidad

4.3. Plan de configuración de software

4.4. Plan de capacitación

4.5. Plan de gestión de riesgos

5. Conclusiones

5.1. Introducción

Explicar que van a describir en las conclusiones el camino recorrido, cumplimiento de objetivos, calidad del producto obtenido, aprendizajes (gestión de proyecto y cliente, desarrollo de habilidades, etc.), experiencias ganadas, etc.

5.2. Grado de cumplimiento de objetivos

5.3. Grado de satisfacción del cliente

5.4. Lecciones aprendidas

5.5. Posibles mejoras y desarrollos futuros

5.6. Reflexiones finales

Bibliografía

Ver formato y ejemplos en 302

Referencias bibliográficas

Ver formato y ejemplos en 302

Glosario

Anexos

Datos de acceso a la solución desplegada y repositorio GitHub

Actas: Reuniones de validación de usuario

Manual de deploy

Manual de Usuario

Inventario de recursos utilizados

Aplicación / Herramienta /Framework	Versión

Tabla 7. Versión de recursos utilizados