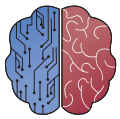
UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería de la Salud



TFG del Grado en Ingeniería de la Salud

Automatización de extracción de datos de informes de secuenciación masiva del software *Oncomine Reporter*

Presentado por Lucía Vítores López
en Universidad de Burgos

27 de junio de 2023

Tutor: Antonio Jesús Canepa Oneto

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería de la Salud

D. Antonio Jesús Canepa Oneto, Departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que la alumna D. Lucía Vítores López, con DNI 71970531N, ha realizado el Trabajo final de Grado en Ingeniería de la Salud titulado: Automatización de extracción de datos de informes de secuenciación masiva del software *Oncomine Reporter*.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 27 de junio de 2023

Vº. Bº. del Tutor:

D. Antonio Jesús Canepa Oneto

Resumen

Este Trabajo de Fin de Grado se centra en la automatización de la extracción de datos provenientes del software Oncomine Reporter, del servicio de anatomía patológica del Hospital universitario de Burgos (HUBU).

La extracción manual de datos es un proceso muy costoso donde es muy probable que se cometan errores. Sin embargo, el uso de automatización puede mejorar tanto la eficiencia como la precisión del proceso. El estudio se basa en implementar un sistema automático para la extracción de una serie de datos de especial interés de distintos archivos, utilizando técnicas de PDFscraping.

Los resultados muestran que la automatización proporciona muchas ventajas y mejoras a la hora de trabajar con los resultados, aportando información más precisa, concreta y exacta sobre el cáncer para estudiar los distintos tipos y genes y no perder tiempo pasando los datos.

Descriptores

Anatomía patológica, Oncomine Reporter, automatizar, extracción de datos, PDFscraping, secuenciación masiva, Python.

Abstract

This Final Degree Project focuses on the automation of data extraction from Oncomine Reporter software in the pathological anatomy service of the University Hospital of Burgos (HUBU).

Manual data extraction is a very costly process where errors are very likely to be made. However, the use of automation can improve both the efficiency and accuracy of the process. The study is based on implementing an automatic system for the extraction of a series of data of special interest from different files, using PDFscraping techniques.

The results show that automation provides many advantages and improvements when working with the results, and more precise, concrete and accurate information on cancer, to study the different types and genes and not wasting time in choosing the data.

Keywords

Pathological anatomy, Oncomine Reporter, automate, data extraction, PDFscraping, massive sequencing, Python.

Índice general

Índice de figuras

Índice de tablas

Objetivos

El objetivo general del presente proyecto es solucionar el problema de obtención manual de los datos del software Oncomine Reporter del servicio de Anatomía Patológica en el Hospital Universitario de Burgos (HUBU) mediante la automatización de la extracción de los datos.

Para conseguir este objetivo general, se ha dividido en distintos objetivos más específicos que al unirlos permiten la creación final del proyecto.

1.1. Objetivos marcados por software, hardware o análisis.

Los principales objetivos que abarcan este apartado son:

- Desarrollo de un código capaz de facilitar la extracción de datos de ficheros PDF.
- Lograr una interfaz sencilla para que cualquier persona sea capaz de usarla correctamente y obtener los resultados esperados, con el fin de facilitar el estudio.

1.2. Objetivos técnicos.

El proyecto fue fragmentado en pequeños bloques de trabajo, que al unirlos han creado el proyecto final completo.

- Revisión bibliográfica sobre PDFscraping.

- Determinar y localizar las variables necesarias.
- Creación de código para la lectura de todos los PDF proporcionados.
- Código para seleccionar las variables de interés de un PDF usando técnicas de PDFscraping.
- Separación de la información en tablas específicas.
- Creación de varias tablas exportables a archivos .xlsx con los resultados.
- Creación de material complementario para ayudar a su implementación en otros dispositivos.

1.3. Objetivos de aprendizaje.

Los principales objetivos basados en la adquisición de nuevos conocimientos son:

- Mejora del entendimiento de los análisis más comunes en anatomopatología.
- Aumento del conocimiento en el ámbito de programación, enfocado en la extracción de datos mediante la técnica PDFscraping.
- Comprensión de la información biológica para su aplicación en el desarrollo del código.
- Selección de las herramientas informáticas más aptas para la realización del proyecto.

Introducción

2.1. Conceptos teóricos básicos.

Vamos a proceder a explicar los conceptos teóricos básicos para el correcto entendimiento del trabajo.

Anatomía patológica.

La anatomía patológica es una rama de la Medicina que se encarga principalmente del estudio de los efectos que produce una enfermedad en los distintos órganos del cuerpo, tanto en aspectos macroscópicos como microscópicos, con el fin de diagnosticar y tratar las distintas enfermedades [1].

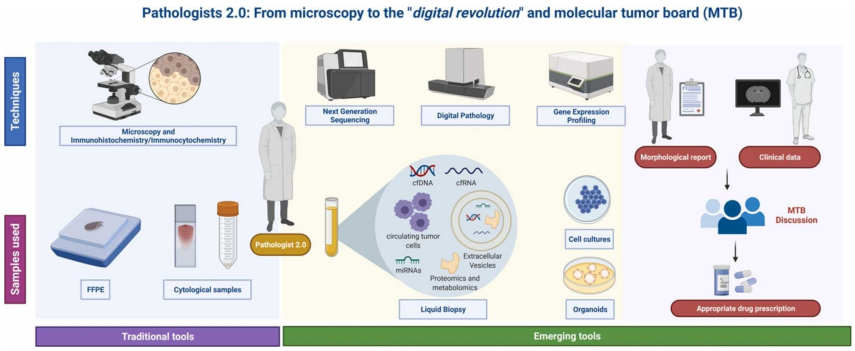


Figura 2.1: Resumen de las principales tareas realizadas en el servicio de anatomía patológica. Imagen obtenida directamente de Science Direct.

Todas las especialidades médicas necesitan la información de este servicio para llevar a cabo sus respectivos procedimientos, por lo que el informe

anatomopatológico final que se genera con toda la información obtenida a través de los exámenes de las muestras del tejido o de las células mediante distintos procedimientos es de gran importancia.

En este servicio se realizan técnicas de secuenciación masiva, es decir, determinar la secuencia del ADN de un gran número de genes relacionados con la causa de una enfermedad o de un trastorno genético.

Es por ello que la anatomía patológica desempeña un papel fundamental en el estudio del cáncer, una de las principales causa de muerte en el mundo según la Organización Mundial de la Salud [2].

Defunciones según la causa de muerte más frecuente - Año 2021

	Total	Hombres	Mujeres	Variación anual. Total	Variación anual. Hombres	Variación anual. Mujeres
Total de defunciones	450.744	231.410	219.334	-8,7	-7,3	-10,2
Covid-19 virus identificado	39.444	22.449	16.995	-34,6	-30,9	-39,0
Enfermedades isquémicas del corazón	28.852	17.747	11.105	-2,7	-2,1	-3,7
Enfermedades cerebrovasculares	24.858	11.004	13.854	-3,7	-2,3	-4,8
Cáncer de bronquios y pulmón	22.413	16.754	5.659	2,4	0,9	6,9
Insuficiencia cardiaca	20.173	7.739	12.434	4,2	1,9	5,7

Variación anual: diferencia respecto al año anterior

Figura 2.2: Causas de muerte mundiales. Numeración de las principales causas de muerte en 2021 respecto a los valores del año 2020. Obtenida de la fuente oficial de INE.

Se encuentra disponible en el repositorio un archivo Excel con un histórico de las muertes causadas por los distintos tipos de cáncer conocidos en un periodo de 23 años (desde 1999 hasta 2021).

El análisis de las características microscópicas de las células y de los tejidos tumorales permiten diagnosticar el cáncer, determinando su etapa y guiando las decisiones para su tratamiento. La experiencia y el análisis preciso que ofrece este campo, son aspectos fundamentales para proporcionar información clínica y mejorar la atención al paciente en el campo de la oncología.

Al tener tanta información sobre algunos genes, los científicos han sido capaces de determinar las mutaciones más comunes en los distintos tipos de cáncer. Esto permite saber que, por ejemplo, el gen FGFR4 con un cambio en el aminoácido p.(P136L) no tiene gran importancia en su diagnóstico. Sin embargo, otros genes como KRAS, HRAS o CTNNB1 son considerados patogénicos por estar asociados a un aumento de riesgo de sufrir cáncer.

Una imagen que determina muy bien los genes que producen los distintos tipos de cáncer es la figura 2.3:

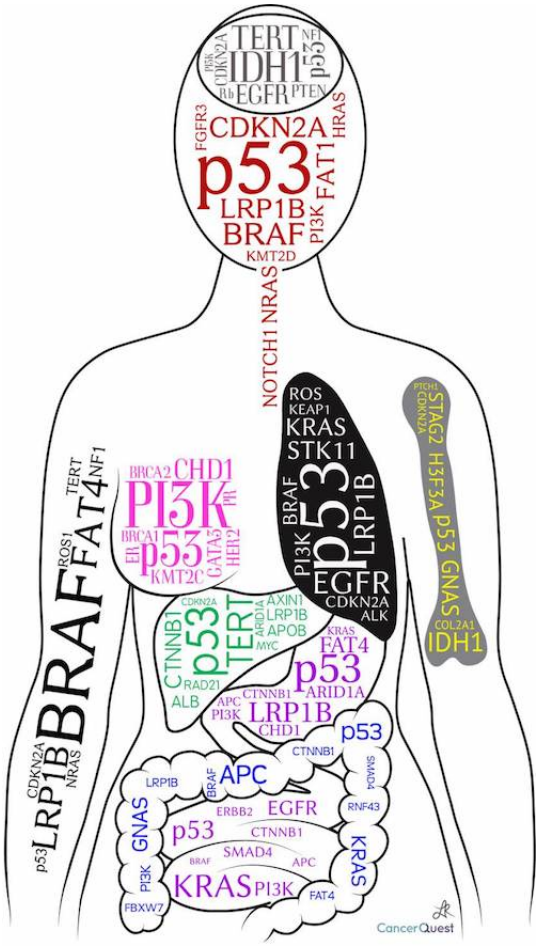


Figura 2.3: Genes causantes de cáncer. Representación de los principales genes que causan los distintos tipos de cáncer. Imagen obtenida de CancerQuest.

El conocimiento sobre los genes conlleva un mayor número de tratamientos dirigidos a esos genes específicos, haciendo posible la aplicación de un tratamiento determinado para cualquier tipo de cáncer siempre que contenga la mutación específica.

El servicio de Anatomía Patológica del Hospital Universitario de Burgos obtiene sus resultados en formato PDF, impidiendo la manipulación de los datos de forma directa y dificultando el trabajo del investigador. Para ello,

su solución actual es crear un archivo Excel en el que se van introduciendo los datos de interés de forma manual para así poder trabajar sobre ellos. Sin embargo, es un trabajo muy laborioso porque se trabajan con grandes cantidades de datos al día y es propenso a fallos. La automatización de la extracción de estos datos ahorraría mucho tiempo y evitaría errores humanos a la hora de la creación de este archivo.

Herramientas usadas en anatomía patológica.

Existen una gran variedad de herramientas centradas en el análisis y la interpretación de datos de secuenciación masiva en el contexto de la investigación, las cuales son de vital importancia para comprender y estudiar las bases genéticas y moleculares del cáncer, con el fin de detectar variantes estructurales, caracterizar los tumores o descubrir nuevas terapias u objetivos terapéuticos.

Algunos de los softwares más empleados en los servicios de anatomía patológica son: [3]

- **PacBio** de Pacific Biosciences: permiten secuenciar moléculas largas, de hasta 30kb.
- **GATK** por Broad Institut: este marco de programación permite desarrollar herramientas eficaces y robustas para secuenciadores de ADN de nueva generación aplicando la estrategia MapReduce.
- **NovaSeq** de Illumina: se usa para generar datos de secuenciación masiva a gran escala y es conocida por su alta capacidad de producción y flexibilidad.
- **Oncomine Reporter** de Thermo Fisher: proporciona opciones y soluciones completas para el perfilado rápido de biomarcadores clave, así como el perfilado genómico integral u otras aplicaciones.

Pero solo nos vamos a centrar en Oncomine Reporter, por ser el software usado por el hospital.

Se trata de un software de análisis terciario de los datos de secuenciación masiva que usa como entrada datos informáticos almacenados en Variant Call Format (vcf) obtenidos previamente de los ácidos nucleicos de las muestras de los pacientes. Se ha desarrollado específicamente para un examen más profundo de la secuenciación masiva, lo que permite un informe final en pasos muy sencillos.

Incluye también una gran variedad de flujos de trabajo de análisis reconstruidos y firmas génicas para distintos tipos de cáncer, lo que permite a los investigadores usar dicha información para avanzar en investigación.

Una vez realizados los análisis, el software devuelve los resultados en una carpeta con distintos ficheros (un fichero por cada muestra de paciente analizada). Estos ficheros son los que nos interesan para llevar a cabo este proyecto.

El archivo que devuelve se trata de un Portable Document Format con varias páginas donde se encuentra toda la información que necesitamos. La mayoría de ellos sigue el mismo formato, aunque pueden variar en función de las mutaciones de cada paciente. El inicio es común en todos los casos y es donde se encuentran los datos comunes, como son el número de historia clínica (NHC), el número de biopsia, la fecha, el tipo de cáncer, los tratamientos disponibles y ensayos clínicos.

El apartado de biomarcadores relevantes no es de interés, ya que esa información va a ser desarrollada posteriormente.

En el apartado de detalles de la variante podemos encontrar tres tablas distintas, variantes de la secuencia de ADN, fusiones de genes (ARN) o variaciones del número de copias. En estas tablas podremos encontrar los genes que han sido mutados, indicando si son o no peligrosos para el individuo.

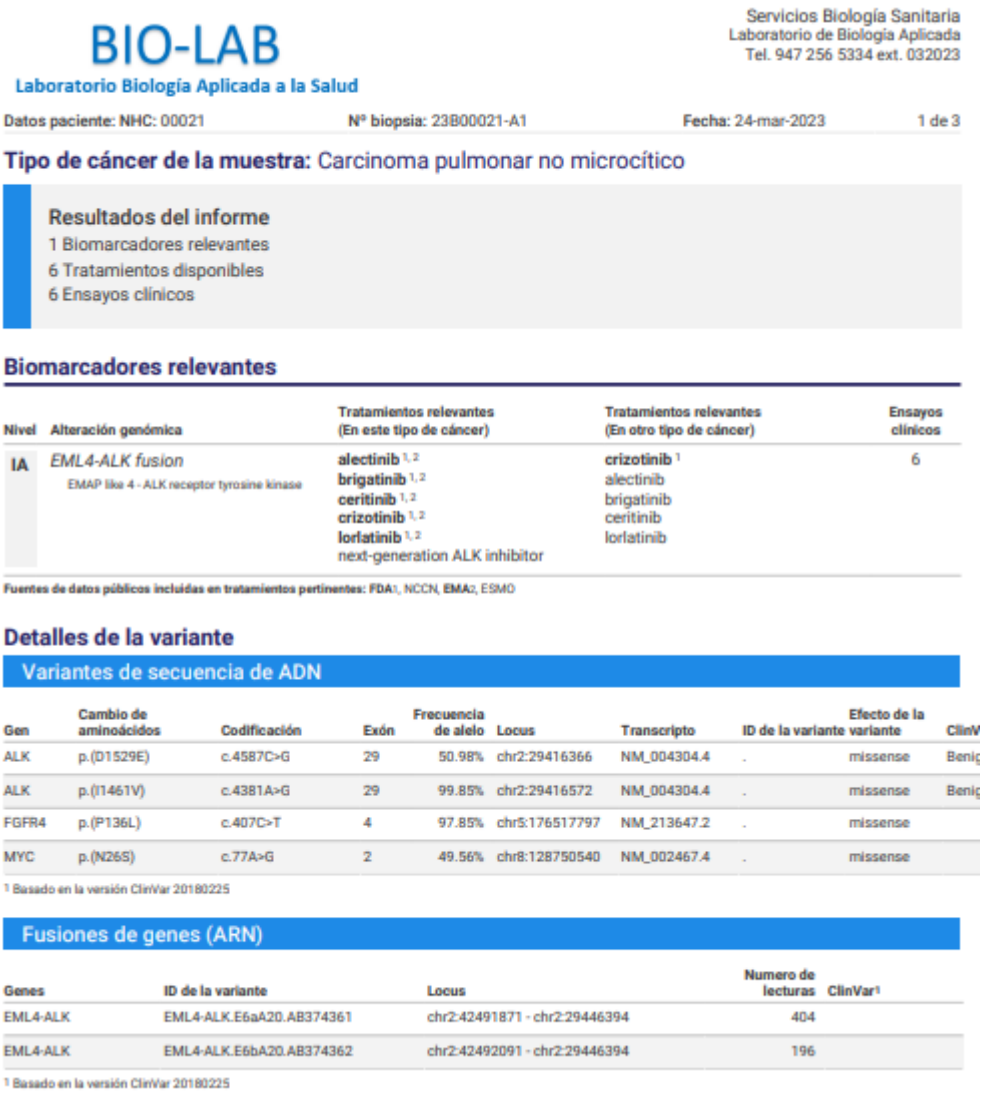
Existen dos tipos de informes distintos, pero en la figura 2.4 solo se muestra uno de ellos, ya que es el formato con el que se ha trabajado a lo largo del proyecto. El código también es capaz de automatizar la extracción de datos del segundo tipo de informe, pero al ser bastante reciente, no se ha tenido en cuenta porque no se ha trabajado con él, solo se ha probado.

Automatización de extracción de datos.

La extracción de datos se basa en la obtención de información de distintas fuentes para posteriormente usarla con otros fines.

La finalidad de automatizar este proceso es ahorrar tiempo y recursos a la hora de almacenar la información, así como minimizar errores y facilitar el trabajo.

La automatización de la extracción de los datos es un procedimiento generalmente útil y que muchas empresas están empezando a usar, ya que esto permite recopilar y trabajar con grandes cantidades de datos pertenecientes a varias fuentes distintas para contrastarlas en el menor tiempo posible y filtrar aquellas de mayor interés. [4]



BIO-LAB
Laboratorio Biología Aplicada a la Salud

Servicios Biología Sanitaria
Laboratorio de Biología Aplicada
Tel. 947 256 5334 ext. 032023

Datos paciente: NHC: 00021 | Nº biopsia: 23B00021-A1 | Fecha: 24-mar-2023 | 1 de 3

Tipo de cáncer de la muestra: Carcinoma pulmonar no microcítico

Resultados del informe
1 Biomarcadores relevantes
6 Tratamientos disponibles
6 Ensayos clínicos

Biomarcadores relevantes

Nivel	Alteración genómica	Tratamientos relevantes (En este tipo de cáncer)	Tratamientos relevantes (En otro tipo de cáncer)	Ensayos clínicos
IA	*EML4-ALK fusion* EMAP like 4 - ALK receptor tyrosine kinase	**alectinib** [1,2] **brigatinib** [1,2] **ceritinib** [1,2] **crizotinib** [1,2] **lorlatinib** [1,2] next-generation ALK inhibitor	**crizotinib** [1] alectinib brigatinib ceritinib lorlatinib	6

Fuentes de datos públicos incluidas en tratamientos pertinentes: FDA1, NCCN, EMA2, ESMO

Detalles de la variante

Variantes de secuencia de ADN

Gen	Cambio de aminoácidos	Codificación	Exón	Frecuencia de alelo	Locus	Transcripto	ID de la variante	Efecto de la variante	ClinV
ALK	p.(D1529E)	c.4587C>G	29	50.98%	chr2:29416366	NM_004304.4	.	missense	Benig
ALK	p.(I1461V)	c.4381A>G	29	99.85%	chr2:29416572	NM_004304.4	.	missense	Benig
FGFR4	p.(P136L)	c.407C>T	4	97.85%	chr5:176517797	NM_213647.2	.	missense	
MYC	p.(N26S)	c.77A>G	2	49.56%	chr8:128750540	NM_002467.4	.	missense	

[1] Basado en la versión ClinVar 20180225

Fusiones de genes (ARN)

Genes	ID de la variante	Locus	Numero de lecturas	ClinVar[1]
EML4-ALK	EML4-ALK.E6aA20.AB374361	chr2:42491871 - chr2:29446394	404	
EML4-ALK	EML4-ALK.E6bA20.AB374362	chr2:42492091 - chr2:29446394	196	

[1] Basado en la versión ClinVar 20180225

Figura 2.4: Ejemplo de informe anonimizado exportado del software Oncomine Reporter.

Aunque también es posible enfocarlo al ámbito médico [5] para trabajar con datos reales de pacientes, ya que la gran cantidad de datos con los que se trabaja habitualmente hace necesario sacarles el mayor provecho posible para mejorar la salud pública. Con este procedimiento de automatización de la extracción, se pretende tener unos resultados lo más seguros y precisos posibles. Algunas de las ventajas que presentaría son la mejora de la calidad

de la atención, el aumento de la productividad o la satisfacción de un mayor número de pacientes entre otros.

Por esto y muchas más ventajas es necesario automatizar los datos, es decir, ser capaz de trabajar con dichos datos usando herramientas en lugar de hacerlo de forma manual, como actualmente ocurre en el HUBU.

Introducción a PDFscraping.

El scraping es una técnica de extracción de datos de manera automática de páginas web o archivos en distintos formatos.

Si nos centramos en PDFscraping, este se enfoca en extraer los datos o el contenido almacenado en archivos PDF de manera automática. Para ello, una vez obtenido el fichero, es de vital importancia conocer la estructura interna, lo que implica un gran desafío, ya que no todos siguen la misma estructura.

Actualmente existen una gran cantidad de herramientas y programas de PDFscraping específicos para extraer los datos deseados.

Los estudiados para la realización de este proyecto se encuentran disponibles en el apartado 4.3.

2.2. Estado del arte y trabajos relacionados.

La automatización de la extracción de datos es un tema que está adquiriendo cada vez más importancia tanto en la academia como en la industria, ya que permite automatizar tareas repetitivas. Esto ahorra mucho tiempo, trabajo, disminuye los errores humanos y mejorar la eficiencia.

Vemos que cada vez se requiere más el uso de herramientas que permitan extraer datos de forma automática, pero no siempre existen las mismas necesidades.

En el campo de análisis de datos del sector de seguros, las compañías pueden utilizar esta técnica para extraer datos relevantes de pólizas, formularios de reclamación y otros documentos relacionados con el sector, con el objetivo de facilitar el análisis de riesgos y la toma de decisiones. Algunas de las herramientas que destacan en este sector son UiPath o Kofax Capture. También está teniendo gran importancia en el sector de procesos administrativos, ya que usan esta técnica en entornos empresariales para automatizar la extracción de datos de facturas, contratos, formularios, etc. En este ámbito destacan herramientas como UiPath o Kapow.

Una herramienta común es UiPath, ya que es la plataforma líder para la automatización robótica de procesos que permite la extracción de datos de forma automática de un gran conjunto de sectores. Sin embargo, no todas las herramientas son comunes, cada una se centra en unas características distintas y son específicas para cada sector. Por ello es importante que existan distintas herramientas especializadas en los distintos campos.

Algunas de las más destacados son:

1. **Astera ReportMiner**: es un software de extracción de datos automatizado a nivel empresarial que extrae datos no estructurados de archivos PDF a una base de datos con funciones de limpieza y programación integradas. Las herramientas también pueden automatizar el proceso de extracción de los datos y cargarlos o almacenarlos en una base de datos o en un archivo de Excel sin el uso de código. Su edición Enterprise Edition también proporciona un programa integrado con funciones en tiempo real capaces de realizar procesos de programación y mantenimiento, mejorando el resultado de la extracción. La información de este apartado hace referencia a la información encontrada en [6] y [7]

2. **Apify**: es una plataforma para automatización y extracción de datos de distintos sitios web, haciendo posible que cualquier persona independiente o empresa, sea capaz de automatizar cualquier flujo de trabajo. También permite a los programadores implementar y monitorizar distintas herramientas de automatización. Proporciona plantillas de código con las que se puede comenzar a desarrollar dichas herramientas siendo posible trabajar en varios lenguajes de programación entre los que se encuentran Python y Java. [8]

3. **Nano**: se trata de una inteligencia artificial capaz de leer documentos semiestructurados o que no siguen una plantilla estándar para extraer sus datos. Aprende y mejora a medida que su uso va aumentando, acortando el tiempo de respuesta. Además, es posible extraer solo aquellos campos de interés y no todo el documento, añadiendo nuevos campos en caso de que sea necesario. Está desarrollado principalmente para empresas, donde los clientes envían los documentos por correo electrónicos y Nanonets es capaz de exportarlo/importarlo a su flujo de trabajo sin interrumpir su sistema. [9]

Como se trabaja con un gran número de datos diarios, es importante almacenarlos y tratarlos para su posterior uso.

La automatización de la extracción de datos de historias clínicas, informes o distintos registros de los pacientes permiten una mejora de la eficiencia en la gestión de la información médica, en los procesos de análisis de datos o en la propia investigación.

También existen algunas herramientas más enfocadas al tratamiento de datos médicos.

1. **Apache cTAKES**: es una herramienta de código abierto que usa procesamiento de lenguaje natural (NLP) para extraer la información de textos clínicos no estructurados, siendo capaz de identificar conceptos médicos y relaciones semánticas y así conseguir un resultado más rápido. La información se devuelve en un archivo y en forma de código. Sin embargo, no es muy útil cuando personas sin conocimientos informáticos necesitan usarlo y habría que procesar dichos datos antes de su uso. [10]

2. **Clinithink**: es una empresa de tecnología creada en torno a CLiX, la primera inteligencia artificial de atención médica capaz de comprender los datos médicos no estructurados. CLiX es capaz de generar conocimiento sobre el ser humano, ahorrando tiempo y esfuerzo a las personas a la hora de realizar estos procedimientos. [11] Se puede usar para cualquier tipo de consulta médica, lo que permite un mejor tratamiento individual y en su conjunto, unos mejores resultados. Actualmente se esta usando la información que recoge para mejorar algunas áreas.

El proyecto OMIM [12] (Online Mendelian Inheritance in Man) es una base de datos que destaca por incluir todas las enfermedades de base genética y que ofrece, siempre que exista evidencia científica, información sobre su manifestación fenotípica.

En esta base de datos se pueden ver los principales genes humanos y los distintos trastornos génicos que causan.

Se ha realizado una búsqueda para tener conocimiento sobre la cantidad de información que hay sobre el PDFscraping en cáncer usando el lenguaje de programación Python. Los resultados obtenidos nos permiten ver que no existe mucha información que abarque esta búsqueda. 2.5

Como el software que se ha usado en el hospital es Oncomine Reporter, se ha decidio realizar una segunda búsqueda. 2.6

Sin embargo, vemos que no existe ningún resultado para la búsqueda de PDFscraping en este software para Python.

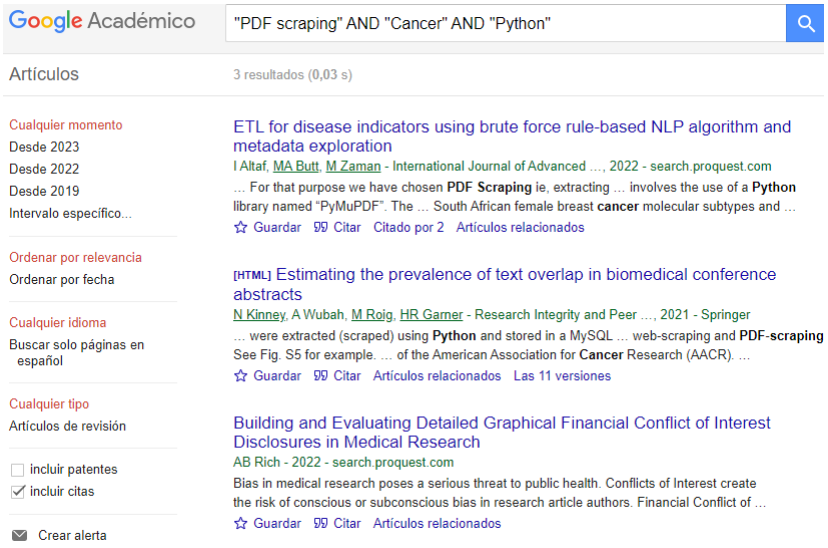


Figura 2.5: Resultados sobre la búsqueda PDFscraping en cáncer usando Python.

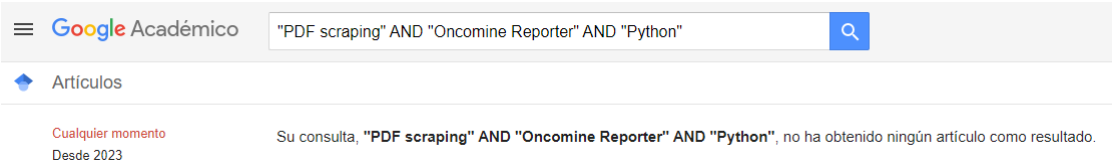


Figura 2.6: Resultados sobre la búsqueda PDFscraping en el software Oncomine Reporter usando Python.

Metodología

3.1. Descripción de los datos.

Los datos necesarios para llevar a cabo este trabajo de fin de grado han sido proporcionados por el Hospital Universitario de Burgos.

En el servicio de anatomía patológica del HUBU se usa el software Oncomine Reporter para el análisis de las muestras. Este software solo es capaz de trabajar con 8 muestras a la vez, de forma que el resultado es una única carpeta con 8 archivos Portable Document Format o PDF (cada uno de ellos representa los resultados de una de las muestras analizadas). Cada vez que se ejecute, devolverá una carpeta distinta con 8 archivos en su interior.

Los informes con los que se ha trabajado están disponibles aquí.

Al tratarse de datos de pacientes reales, se ha firmado un acuerdo de confidencialidad, **ACUERDO DE CONFIDENCIALIDAD** entre el Hospital Universitario de Burgos y la Universidad de Burgos para no mostrar los datos reales. Dicho acuerdo se encuentra disponible en el anexo **Planificación**.

De este modo, a la hora de presentar el código frente al tribunal, se han usado unos archivos sintéticos a los que se les ha eliminado todo tipo de información, tanto personal como institucional, para cumplir dicho acuerdo. Estos archivos han sido aportados por la investigadora del Hospital Universitario con el fin de tener unos modelos similares a los resultantes de Oncomine Reporter con los que poder trabajar y enseñar el resultado. Esto nos permite trabajar de una manera más segura, cumpliendo el acuerdo de confidencialidad de datos.

Sin embargo, al implementar dicho código en el hospital, los técnicos e investigadores podrán trabajar con los datos reales de los pacientes sin ningún problema.

Se realizará una descripción más intensa de los datos en su anexo extra correspondiente.

3.2. Lenguajes de programación.

A lo largo del grado se han usado distintos lenguajes de programación, entre los cuales se encuentran Python, R, Java y SQL.

Java es un lenguaje sobre el que se adquieren conocimientos básicos en la carrera, pero no lo suficientemente avanzados como para ser capaz de programar dicho algoritmo. Lo mismo ocurre con SQL.

Como Python y R son lenguajes cuya forma de trabajar es muy similar y ambos poseen gran cantidad de bibliotecas enfocadas al análisis y tratamiento de grandes cantidades de datos, inicialmente se estudió la posibilidad de realizar el código en ambos lenguajes.

Sin embargo, como Python ha sido el lenguaje más usado en la carrera y del que más conocimientos se tienen, se eligió esa opción.

3.3. Técnicas y herramientas.

Se han usado distintas técnicas metodológicas así como herramientas para el proyecto. Este apartado se centra en explicar las posibles alternativas que han surgido, por qué se han usado y los posibles problemas o dudas.

Oncomine Reporter

Se trata de una herramienta de software de análisis genómico desarrollada específicamente para un examen más profundo de la secuenciación masiva, lo que permite un informe final en pasos muy sencillos. Permite una investigación contextual de variantes específicas de la muestra para comprender su uso con respecto a ensayos clínicos globales actuales. [13]

Python

Es un lenguaje de programación de alto nivel, orientado a objetos y con semántica dinámica.[14] Fue creado por Guido van Rossum y lanzado por

primera vez el 20 de febrero de 1991. [15] La sintaxis se considera simple y fácil de aprender, reduciendo el costo de mantenimiento del programa. Python admite módulos y paquetes, lo que fomenta la modularidad del programa y la reutilización del código. Su interprete y su extensa biblioteca estándar se encuentran disponibles en formato fuente o binario sin ningún tipo de coste para todas las plataformas principales y se pueden distribuir gratuitamente.

Librerías.

Pandas. La librería Pandas [16] fue desarrollada por Wes McKinney en 2008. Se usa principalmente para el tratamiento de big data. Permite leer los datos de los ficheros .xlsx en DataFrame de una forma directa, lo que es de gran interés al tener dos archivos .xlsx con los que se tienen que trabajar.

Numpy. El precursor de NumPy, Numeric, fue creado por Jim Hugunin con contribuciones de varios otros desarrolladores, hasta que en 2005, Travis Oliphant creó NumPy incorporando nuevas características. [17]

El uso de Numpy [16] para el tratamiento y análisis de los datos es de gran ayuda. Esta librería se encarga del tratamiento, procesamiento y cálculo de los datos que internamente realizan el programa.

Re. Este módulo proporciona operaciones de coincidencia de expresiones regulares. [18]

Os. Proporciona una forma portátil de usar la funcionalidad dependiente del sistema operativo. [19]

Anaconda

Se trata de una plataforma de distribución y gestión de paquetes tanto para Python como para R, usada principalmente en ciencia de datos, inteligencia artificial o desarrollo de aplicaciones. [20] Fue fundada en 2012 por Peter Wang y Travis Oliphant a partir de la necesidad de llevar Python al análisis de datos comerciales.[21] Proporciona un entorno de trabajo completo que incluye una amplia colección de paquetes, bibliotecas y herramientas con el fin de facilitar el proceso de desarrollo, análisis y visualización de datos.

Uno de sus paquetes más interesantes es Conda, ya que nos permite instalar, actualizar y administrar fácilmente paquetes y dependencias de software. Además, Anaconda incluye un gestor de entornos virtuales para crear y gestionar entornos de desarrollo aislados. Esto es especialmente útil

cuando se trabaja en proyectos con diferentes versiones de paquetes o cuando se desea mantener un entorno limpio y consistente.

La plataforma Anaconda es compatible con múltiples sistemas operativos, como son Windows, macOS o Linux y ofrece una interfaz gráfica de usuario llamada Anaconda Navigator, que facilita la gestión de paquetes y entornos de desarrollo de forma visual e intuitiva.

Anaconda Navigator y Jupyter Notebook son dos herramientas que forman parte de la plataforma Anaconda. Anaconda Navigator es una interfaz gráfica de usuario (GUI) que permite trabajar con entornos virtuales, paquetes y proyectos en Anaconda, así como crear y gestionar entornos, instalar paquetes, y administrar aplicaciones y extensiones. A través de Anaconda Navigator, se accede a Jupyter Notebook.

Un environment o entorno se refiere a un espacio de trabajo aislado que contiene una instalación específica de Python y las bibliotecas asociadas. Esto permite tener distintos entornos independientes donde en cada uno haya instalado un conjunto específico de paquetes.

Cada environment en Anaconda puede tener su propia versión de Python y una selección de paquetes. Esto permite que diferentes proyectos utilicen diferentes versiones de Python o conjuntos de bibliotecas sin interferir entre sí.

También es posible acceder a entornos de desarrollo integrados (IDE) que permiten a los programadores consolidar los distintos aspectos de la escritura de un programa, aumentando la productividad combinando distintas actividades dentro de un software [22] como en este caso sería Python. Permiten depurar y editar el código, realizar pruebas, etc. como por ejemplo Spyder o PyCharm.

Jupyter Notebook es una aplicación web interactiva para crear y compartir documentos, también conocidos como "notebooks", pero no es considerada una IDE completa. [23] Una vez instalado y abierto Anaconda Navigator, se puede seleccionar Jupyter Notebook para lanzarlo y poder trabajar directamente en este sin tener que realizar ninguna otra instalación mediante la creación de un fichero .ipynb.

El notebook que contiene el código se encuentra en el repositorio, con el nombre de CodigoPython.ipynb

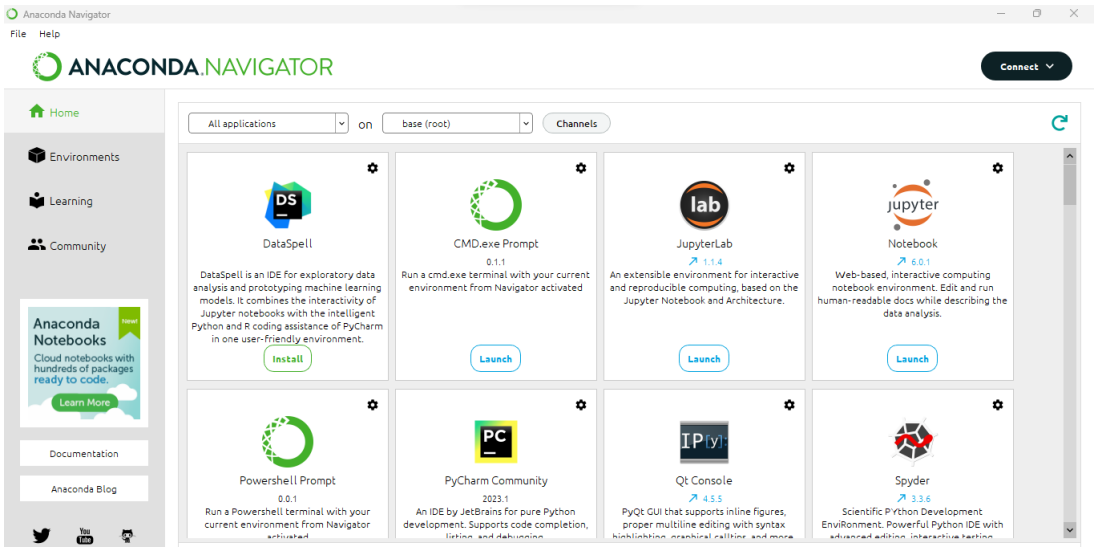


Figura 3.1: Apertura de Anaconda

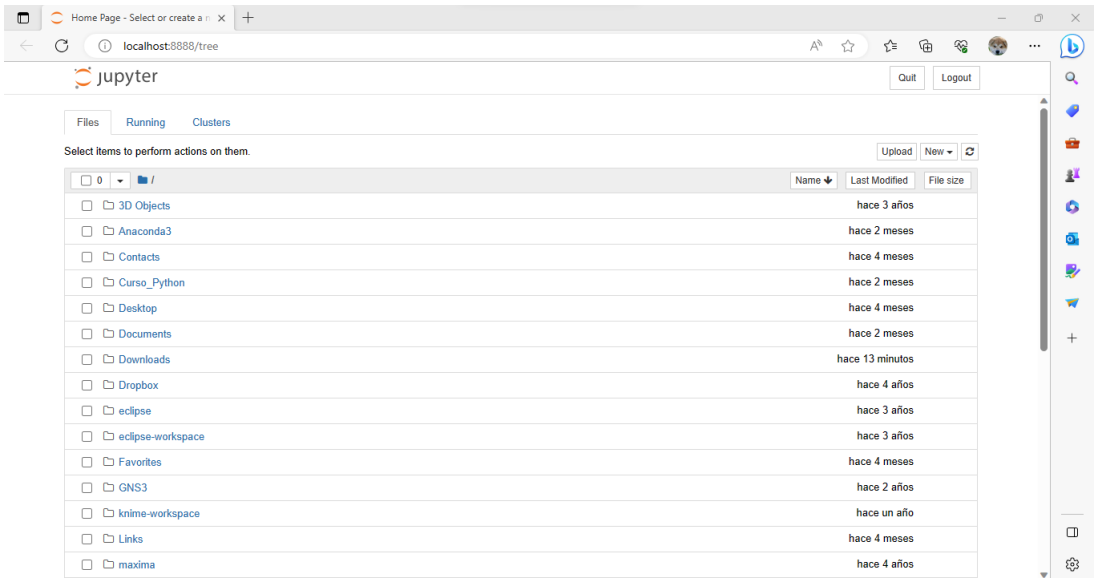


Figura 3.2: Imagen de los posibles archivos a encontrar al entrar a Jupyter.

Git

Es un sistema de control de versiones o VCS (Version Control System) que rastrea el historial de cambios conforme las personas y los equipos colaboran juntos en los proyectos. [24] Conforme los desarrolladores hacen

cambios al proyecto, cualquier versión anterior de este puede recuperarse en cualquier momento.

Los VCS proporcionan a cada contribuyente una vista consistente y unificada de un proyecto, mostrando el trabajo que está en progreso. Permite ver quién ha realizado los cambios, qué modificaciones ha realizado, etc. ayudando a los miembros a cooperar entre sí.

En un sistema de control de versiones distribuido, cada desarrollador tiene una copia integral del proyecto y del historial. Git es el sistema de control de versiones distribuido más popular.

GitHub

Se trata de una de las principales plataformas usadas para la creación de repositorios abiertos. [25] Comenzó a desarrollarse el 19 de octubre de 2007 pero no fue lanzado hasta abril de 2008 por Tom Preston-Werner , Chris Wanstrath , PJ Hyett y Scott Chacon, tras haber estado disponible para unos meses como una versión beta. [26]

Los repositorios pueden ser colaborativos (donde todos los usuarios pueden aportar algo de ayuda para la mejora del código) o privados (donde solo tiene acceso al repositorio el propio colaborador y los invitados que este elija).

Permite a todos los usuarios desarrollar proyectos creando repositorios de forma gratuita, por lo que se trata de proyectos de código abierto. El código abierto hace referencia a que es accesible a todo el público donde pueden ver, modificar, actualizar y distribuir el código de la forma que ellos consideren oportuna.[27] Por lo que el código depende tanto del propio creador como de los distintos usuarios que forman la comunidad que se encargan de modificarlo o revisarlo.

El repositorio creado para este trabajo se ha denominado Automatizacion_PDF_scraping y actualmente se encuentra disponible de manera pública.

A su vez, se ha trabajado con GitHub Desktop como aplicación de interfaz gráfica de usuario (GUI) que proporciona una interfaz para trabajar con Git dentro de GitHub.

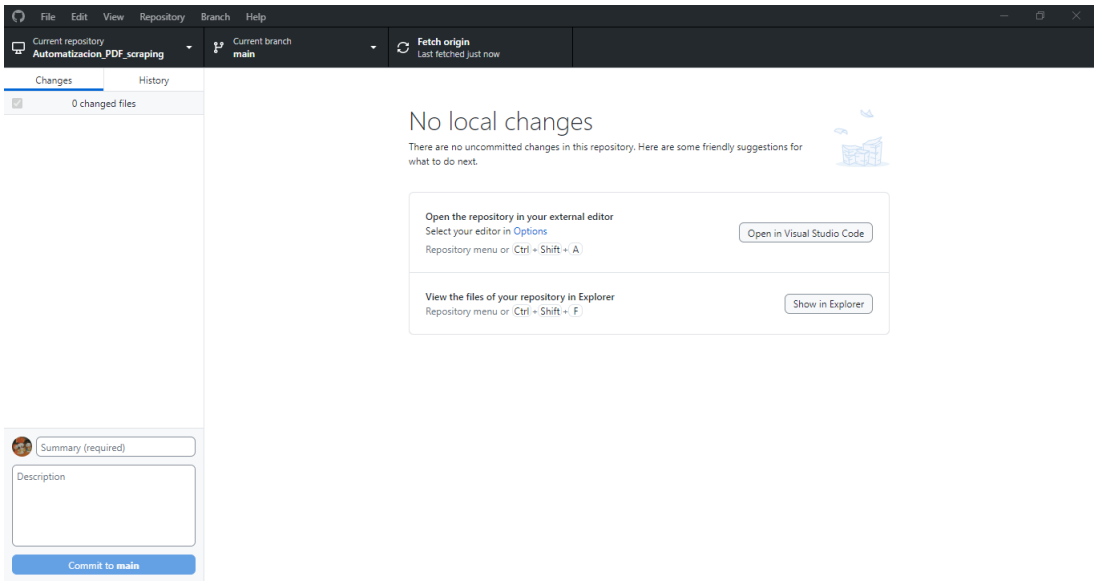


Figura 3.3: GitHub Desktop.

Se ha seguido la estrategia de ramas paralelas para trabajar en varias líneas de desarrollo simultáneamente. Esta estrategia permite trabajar de forma paralela realizando funciones sin modificar la rama principal. Por lo que se ha creado un branch cada vez que se ha visto la necesidad de realizar cambios sin modificar la rama main. [28]

En este caso, la nueva rama ha sido creada para subir los PDF sintéticos, pero como inicialmente estos contenían un nombre real del personal del HUBU, se eliminaron y se volvieron a subir posteriormente los archivos correctos.

Posteriormente se ha creado otra rama para la modificación del archivo README.

Para la organización del contenido de cada semana, se han ido creando distintos Millestones donde se almacenan los Issues, que es donde se explica que objetivos se pretende finalizar en cada periodo de tiempo entre reuniones. Esto se verá correctamente explicado en el anexo **Planificación**.

Se ha intentado unir la memoria de Overleaf con el propio repositorio, pero no ha sido posible porque no existía una versión gratuita para su uso. Por ello, se han ido subiendo distintos ficheros Portable Document Format con las modificaciones que se han ido realizando para llegar a tener la memoria final.

En algunas de las asignaturas de la carrera ya se había usado dicha plataforma, por lo que ya se tenían unos conocimientos básicos sobre su funcionamiento, pero ha sido necesario la búsqueda de contenido para ser capaz de crear un repositorio totalmente claro y entendible.

LaTeX

Es un sistema de composición de documentos basado en TeX, que usa Overleaf como plataforma en línea, ya que ofrece un entorno colaborativo para editar y compilar documentos LaTeX sin requerir instalaciones locales. LaTeX fue desarrollado Leslie Lamport en 1980, siendo esta una extensión del sistema de composición de documentos TeX, desarrollado por Donald Knuth en 1970. Aunque desde entonces ha experimentado grandes mejoras. Overleaf fue desarrollado por John Hammersley y John Lees-Miller y lanzada públicamente en 2012, con el objetivo de proporcionar una solución fácil y colaborativa para la edición de documentos LaTeX en la nube. [29]

Es un sistema para crear textos estructurados o con fórmulas matemáticas, siendo Overleaf su entorno de edición en línea. Se usa principalmente en textos donde lo importante es el texto y su estructura, no el tipo de letra o el salto de página. Esta y parte de la información encontrada posteriormente ha sido obtenida de [30]

Los documentos se escriben en texto plano, por lo que existen distintos parámetros o caracteres especiales para usar distintos comandos, como barras bajas (_)para subíndices o almohadilla (%) para comentarios. El encabezado debe tener unas instrucciones claras para determinar cosas como el idioma, tipo y tamaño de letra, etc. Además debe incluir comando para inicio y fin del documento (begin document/ end document). Entre estos dos comandos se debe desarrollar el informe, usando las secciones y subsecciones necesarias para su organización. También es posible añadir imágenes o tablas para aclarar el contenido del texto, hacer enumeraciones o introducir fórmulas matemáticas.

Un comportamiento que tiene por defecto LaTeX es que prefiere que una palabra sobrepase el margen derecho a tener que pasarlo a la siguiente línea y dejar un hueco demasiado grande. Para ello existe un comando especial denominado *sloppy* que permite el salto de línea de dicha palabra.

El paquete BibTeX nos permite almacenar las URLs los distintos libros o páginas web que se han utilizado para buscar la información del proyecto y hacer referencias en el texto mediante identificadores. Para ello se crea un archivo .bib nuevo (en este caso llamado bibliografia.bib para la bibliografía

de la memoria y otro llamado bibliografiaAnexos.bib para la bibliografía de los anexos) donde se almacenan todos los datos para tenerlos registrados y poder crear la bibliografía final.

Cuando recompilamos, podemos observar como quedaría toda la documentación organizada en el archivo. En el caso de que haya algún error, LaTeX es capaz de indicarlo. Es posible imprimir el documento final e incluso pasarlo a otro tipo de archivo.

Excel

Es una hoja de cálculo de Microsoft que nos permite trabajar tanto con datos numéricos como con texto en distintas tablas formadas por líneas y columnas.[31] No fue desarrollado por Bill Gates, sino por un programador llamado Charles Simonyi, al que también se le atribuye el desarrollo de otros productos del software de Microsoft. [32]

Estas hojas de cálculo nos permite analizar o realizar distintas acciones mediante gráficos o tablas.

Un archivo .xls es un archivo de hoja de cálculo que puede ser creado con Excel o con otras hojas de cálculo. Sin embargo, en versiones posteriores a 2007 se usan archivos .xlsx, ya que permiten un formato más abierto y estructurado, aunque también son capaces de devolver y leer su versión anterior. Los archivos habilitados para almacenar instrucciones para la automatización de los procesos de Excel usan una extensión .xlsm. [33]

En este trabajo se han usado dos hojas de cálculo distintas. Una para crear una tabla que determine los genes de interés junto con un valor numérico que se ha asignado a cada uno de ellos para tener un formato distinto con el que poder buscar estos valores (Genes.xlsx). La otra hoja ha sido creada para determinar los diagnósticos que usa el hospital junto con un valor numérico que identifica a cada uno (Diagnostico.xlsx). Ambos ficheros están disponibles dentro de la subcarpeta Datos que se encuentra en la carpeta INPUT.

Se ha hecho de esta manera por si en algún momento es necesario insertar un nuevo gen o diagnóstico. En este caso se añadiría el nuevo gen o diagnóstico en la última fila, añadiendo en la columna continua su valor numérico determinado (siguiendo la numeración de la hoja). Como el código realizado es capaz de leer los Excel, al guardar el cambio y ejecutar el código, ya aparecería el nuevo gen/diagnóstico y será tenido en cuenta desde el momento que se añada.

Draw.io

Se trata de una herramienta gratuita con la que se puede dibujar cualquier tipo de mapa, esquema o representaciones gráficas, como diagramas de jerarquía o de flujo. [34] Se puede usar en versión online o de escritorio. En versiones online se puede vincular a diferentes cuentas y guardar los trabajos en sistemas de almacenamiento en la nube (Google Drive, OneDrive, Dropbox) o descargarlos, incluso es posible importar diagramas de otras aplicaciones para trabajar sobre ellos. Consta de una serie de librerías con una gran cantidad de formas, líneas, dibujos, ect. que se pueden ir ampliando. También es posible crear o importar otras en el caso de que sea necesario.

Dos de las principales ventajas que presenta esta herramienta es que el guardado de cualquier trabajo realizado se hace de manera automática y es muy fácil de exportar.

Conclusiones

Las conclusiones derivadas del desarrollo del proyecto han sido las siguientes:

- Extracción de datos de archivos PDF de forma anonimizada.
- Código capaz de generar una tabla con los datos estructurados obtenidos del software usado en el sector de anatomía patológica.
- Entrega de resultados de una forma sencilla y sin errores.
- Búsqueda de herramientas informáticas óptimas para el desarrollo del código y su ejecución.
- Entendimiento de los resultados exportados por el software, posibles excepciones de los genes y conocimientos anatomopatológicos.
- Creación de un manual de usuario necesario para su ejecución en otros dispositivos.

4.1. Resumen de resultados.

Los ficheros exportados por el software Oncomine Reporter se encuentran en formato PDF, lo que permite compartir documentos de una forma muy segura, garantizando que no se realizará ningún cambio. Estos formatos no son nada fáciles de editar, por lo que el proceso fundamental para comenzar con el proyecto ha sido la extracción de los datos anonimizados que se encuentran en estos ficheros, para su posterior tratamiento y estudio.

Para ello se ha desarrollado un código capaz de leer la información que hay en cada uno de los PDF, escogiendo los datos que eran de real interés.

Los resultados obtenidos en este proyecto han sido distintas tablas con diferente información para mejorar el estudio y la comprensión de los resultados de las pruebas de secuenciación múltiple de última generación en tumores sólidos, así como facilitar la búsqueda de la información.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		Número de	o de p	NHC	úmero de biopsi	sia sc	echa de inform	Diagnóstico	o del diagi	utaciones detectad	la mutaci	úmero de	de frecuencia al	Fusiones ID	sayos clíni	/NO ensay	naco apro	NO fármac
2	0	100	1	21	23B00021-A1	1	24-mar-2023	Carcinoma pulmona	15,1	['MYC']	[40]	1	['49.56']	['EML4-ALK.E6aA	6	1	6	1
3	1	100	2	22	22B00022-A2	1	24-mar-2023	Carcinoma pulmona	15,1	['KRAS']	[35]	1	['66.50']	[]	5	1	1	1
4	2	100	3	23	21B00023-A1	1	24-mar-2023	Carcinoma pulmona	15,1	[]	[]	0	[]	[]	0	0	0	0
5	3	100	4	24	23B00024-A1	1	24-mar-2023	Carcinoma pulmona	15,1	['FGFR1', 'PIK3CA']	[22, 47]	2	[]	[]	0	0	0	0
6	4	100	5	25	23C00025-A1	3	24-mar-2023	Carcinoma pulmona	15,1	['CDK6', 'MET']	[10, 38]	2	['21.75', '21.75']	[]	3	1	3	1
7	5	100	6	26	23P00026	2	24-mar-2023	Carcinoma pulmona	15,1	['CTNNB1', 'EGFR']	[11, 13]	2	['27.51', '35.40']	[]	5	1	0	0
8	6	100	7	27	22B00027-C4	1	24-mar-2023	Cáncer tiroideo	30	['HRAS']	[28]	1	['46.17']	[]	1	1	0	0
9	7	100	8	28	23B00028-A2	1	24-mar-2023	Cáncer gástrico	11	['KIT', 'MYC']	[34, 40]	2	['53.12', '60.64']	[]	0	0	1	1
10	8	100	1	1234567	23B000000-A1/CH	1	25-may-2023	Carcinoma pulmona	15,1	['MYC']	[40]	1	['49.56']	['EML4-ALK.E6aA	6	1	6	1
11	9	100	2	1234567	23B00000-A1/CHI	1	25-may-2023	Carcinoma pulmona	15,1	['KRAS']	[35]	1	['66.50']	[]	4	1	1	1
12																		

Figura 4.1: Tabla resultante final capaz de almacenar toda la información obtenida de los informes. Resultados de elaboración propia.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Número de	o de p	NHC	úmero de biopsi	opsia sólid	Fecha de informe		Diagnóstico	o del diagnes patogén		la mutaci	cuencia al	utaciones	sayos clíni	/NO ensay	naco apro	NO fármacos
2	0	100	1	21	23B00021-A1	1	24-mar-2023	Carcinoma pulmonar no microcítico	15,1	[]	[]	[]	0	6	1	6	1
3	1	100	2	22	22B00022-A2	1	24-mar-2023	Carcinoma pulmonar no microcítico	15,1	['KRAS']	[35]	['66.50']	1	5	1	1	1
4	2	100	3	23	21B00023-A1	1	24-mar-2023	Carcinoma pulmonar no microcítico	15,1	[]	[]	[]	0	0	0	0	0
5	3	100	4	24	23B00024-A1	1	24-mar-2023	Carcinoma pulmonar no microcítico	15,1	[]	[]	[]	0	0	0	0	0
6	4	100	5	25	23C00025-A1	3	24-mar-2023	Carcinoma pulmonar no microcítico	15,1	[]	[]	[]	0	3	1	3	1
7	5	100	6	26	23P00026	2	24-mar-2023	Carcinoma pulmonar no microcítico	15,1	['CTNNB1']	[11]	['27.51']	1	5	1	0	0
8	6	100	7	27	22B00027-C4	1	24-mar-2023	Cáncer tiroideo	30	[]	[]	[]	0	1	1	0	0
9	7	100	8	28	23B00028-A2	1	24-mar-2023	Cáncer gástrico	11	[]	[]	[]	0	0	0	1	1
10	8	100	1	1234567	23B000000-A1/CH	1	25-may-2023	Carcinoma pulmonar no microcítico	15,1	[]	[]	[]	0	6	1	6	1
11	9	100	2	1234567	23B00000-A1/CH	1	25-may-2023	Carcinoma pulmonar no microcítico	15,1	['KRAS']	[35]	['66.50']	1	4	1	1	1
12																	

Figura 4.2: Tabla resultante final capaz de almacenar toda la información sobre los genes patogénicos obtenida de los informes. Resultados de elaboración propia.

Se ha estudiado el funcionamiento de distintas herramientas informáticas, así como librerías del propio lenguaje de programación para ver cuál eran las más adecuadas para la realización del proyecto. De todas las estudiadas, se han descartado varias, por el hecho de existir otras más simples o adaptarse mejor a los objetivos del propio proyecto.

No solo se ha buscado herramientas y librerías, la búsqueda de información en el ámbito anatomopatológico ha sido realmente necesario para poder entender los datos con los que se partían, el tipo de resultados que devolvía el software y los resultados finales que el hospital esperaba obtener.

El proceso de investigación ha sido un trabajo fácil de llevar a cabo gracias a la ayuda de Patricia, investigadora principal del hospital, ya que la forma de exponer sus conocimientos era muy clara y concisa, haciendo

que los datos fueran más fáciles entender y, por lo tanto, consiguiendo una mayor facilidad a la hora de trabajar con ellos.

La combinación de la información obtenida en ambos ámbitos (anatomopatológico y programación enfocado en PDFscraping) ha hecho posible un mayor entendimiento del producto final, complementándose de manera que haya sido posible la realización del proyecto de una manera sencilla.

Los resultados finales han sido 4 tablas, aunque este número puede variar en función de los ficheros iniciales que se usen. Podemos ver un resumen de los resultados en la tabla 4.1.

Tabla	**Tabla resumen de resultados obtenidos.**	**Exportados**
Histórico general	Recoge la información relevante de los ficheros que se encuentran dentro de la ruta especificada. Se va actualizando según se vayan añadiendo más informes a la ruta.	TablaGeneral.xlsx
Histórico patogénico	Se seleccionan solo aquellas filas que contengan genes patogénicos. Si no son patogénicos los genes, no se almacenan en este fichero.	TablaPato.xlsx
Parcial general	En lugar de tener un histórico general que contenga toda la información porque sería muy largo, interesa crear ficheros de menor longitud, para que su estudio e interpretación sea más sencilla. Estos se van creando en función de los informes existentes, creando uno nuevo cada 80 líneas (8 chips).	tabla_finali.xlsx
Parcial patogénico	Al igual que ocurre con el histórico general, también es interesante poder crear ficheros más pequeños para el histórico patogénico.	patogenicos_i.xlsx

Tabla 4.1: Detalles de los ficheros creados a partir del código.

Como ninguno de los algoritmos estudiados es capaz de satisfacer las necesidades del Hospital Universitario de Burgos, se ha decidido crear desde cero un proyecto nuevo. Un trabajo de esta importancia ha sido un gran

reto, principalmente porque va a ser una aplicación usada para la mejora de la salud de la población.

Debido a que este proyecto se encuentra disponible en GitHub y cualquier persona interesada puede acceder al código creado para su estudio, implementación o modificación, se ha desarrollado un anexo denominado **Manual del usuario**, donde se explican las distintas funciones del código, qué hace cada línea, que resultados devuelve, etc.

4.2. Discusión.

Desde la creación de los ficheros Portable Document Format en 1993 [35], estos se han convertido en uno de los métodos más usados para compartir datos, ya que está disponible en todos los sistemas operativos, garantizando que no se modifique el aspecto actual ni la estructura.

El software Oncomine Reporter usado por el servicio de anatomía patológica usa dichos ficheros para exportar los datos que obtiene tras el análisis de las muestras.

Como actualmente no existe ninguna publicación sobre la relación de PDFscraping en dicho software usando Python 2.6, se ha extendido la búsqueda a únicamente el software y el lenguaje de programación usado, pero los resultados seguían siendo nulos. 4.3

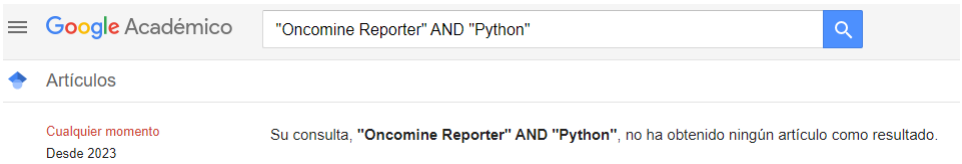


Figura 4.3: Resultados sobre la búsqueda Oncomine Reporter en PDF

Se cambió de estrategia y se decidió usar el propio manual publicado por IonTorrent denominado **Oncomine™ Reporter USER GUIDE** para la versión 5.6, que es la que se usa en el hospital. Este manual ha sido proporcionado por la investigadora principal de este servicio.

Se realizó una búsqueda sobre los algoritmos existentes que pudieran ser compatibles o útiles para este fin y como ninguno de ellos era capaz de satisfacer las necesidades del Hospital Universitario de Burgos, se decidió crear un código propio desde cero.

El código es capaz de leer la información que hay en cada uno de los PDF, escogiendo los datos que eran de real interés usando la técnica PDFscraping. Esta búsqueda ya tenía muchos más resultados que las anteriores, con distintas publicaciones de interés y utilidad.

Los resultados obtenidos en este proyecto han sido distintas tablas en formato Excel. Esta conversión de archivos ha proporcionado un resultado en estructura tabular mucho más legible y mostrando solo los resultados de interés de una forma mucho más clara, sencilla y organizada. De esta forma, es posible la modificación de los datos de una forma directa.

Para la realización de este proyecto ha sido necesario saber qué procesos había que realizar y cuales eran las mejores herramientas y librerías con las que se podía trabajar para garantizar el correcto funcionamiento.

La búsqueda de información en el ámbito anatomopatológico ha sido de gran interés para poder entender los datos con los que se trabajaba durante todo el periodo de tiempo que ha durado.

El desarrollo del anexo **Manual de usuario** tiene como objetivo explicar correctamente las distintas funciones del código y conseguir que sea fácil de entender. De esta forma, cualquier persona podrá ser capaz de ejecutar el código sin problema y ser capaz de pone a prueba por sí mismo su funcionamiento.

4.3. Aspectos relevantes.

Elección del lenguaje.

Inicialmente se pretendía usar tanto R como Python. A la hora de la búsqueda de información sobre PDFscraping se tuvieron en cuenta los posibles paquetes de ambos lenguajes, su funcionamiento, las instalaciones necesarias, etc. Una vez se tenía en código en Python, se comenzó con R, pero debido a una falta de tiempo se decidió que era mejor centrarse en pulir el archivo en Python que ya estaba completo, antes que seguir con el código en R.

El uso de Python como lenguaje de programación principal se debe a que tiene una gran cantidad de bibliotecas y herramientas diseñadas para el análisis de datos (al igual que R). Además, al ser el lenguaje de programación usado por excelencia en la carrera, se tienen más conocimientos acerca de este.

R también es un lenguaje que se usa, pero no considero que a lo largo de la carrera se obtengan los conocimientos necesarios para la realización de un código usando dicho lenguaje.

Se ha usado el libro *Python for Data Analysis* para la ayuda del código de Python [16]

PDFscraping en Python.

Se comenzó con la búsqueda de información sobre PDFscraping y las distintas bibliotecas que podían usarse en Python.

Entre ellas destacaban:

- **Tabula-py**: biblioteca que permite extraer información de un formato estructurado introduciendo la ubicación de los datos tabulares dentro del PDF. Las coordenadas siguen el formato (arriba, izquierda, abajo, derecha). Esto se obtendrá principalmente por prueba y error. Si el fichero solo contiene la tabla en la que se quiere buscar la información, no es necesario especificar el área porque las filas y las columnas deberían detectarse automáticamente. [36] También se puede extraer información de formatos no estructurados, pero se debería de seguir una serie determinada de pasos mucho más complejos.

  Los datos de los informes con los que trabajamos están desordenados, por lo que esta biblioteca no sería capaz de trabajar correctamente. El método de prueba-error sería buena elección en caso de que todos los ficheros siguieran una misma estructura, no cuando se quiere trabajar con grandes conjuntos de datos donde los informes ni siguen la misma estructura ni los datos se encuentran en la misma posición siempre. Además, esto conllevaría una gran cantidad de tiempo para encontrar las posicione de todos los valores que se piden.

- **PDFQuery**: para poder trabajar con esta biblioteca es necesario convertir los ficheros PDF en formatos XML, ya que este formato permite definir un conjunto de reglas para codificar PDF en un formato que es tanto legible por humanos como por máquinas. [37] Para obtener los valores hay que buscar las coordenadas de la palabra de interés (izquierda, abajo, derecha, arriba) dentro del cuadro de texto, siendo el eje X el ancho de la página y el eje Y el alto. Otro método para obtener información es usando palabras clave vecinas, es decir, buscar las palabras clave que nos interesan para extraer los datos asociados a esa palabra. Una vez tenemos detectada la palabra clave, podemos usar

el método keywords.get para extraer las coordenadas de la palabra clave y desplazarlas sumando o restando valores para obtener las verdaderas coordenadas de la palabra de interés.

Esta biblioteca tampoco se ha usado debido a que los informes exportados por el software están en formato PDF y no nos interesa pasarlo a XML habiendo bibliotecas que nos permitan no modificar su extensión. Se tardaría más tiempo en encontrar las coordenadas que en realizar el código, ya que el texto de nuestro PDF no contiene tablas donde se encuentre toda la información que necesitamos, sino que los datos de interés se encuentran como texto a lo largo del PDF.

- **PDFMiner**: es una herramienta de extracción de información de ficheros PDF. Permite obtener una ubicación real y exacta de texto de una página. Viene con dos herramientas de interés como son pdf2txt.py que permite extraer el contenido del PDF representando el texto en ASCII o Unicode y dumppdf.py que vuelva el contenido del PDF en un formato xml. [38]

  Viendo algunos ejemplos se concluyó que es una biblioteca bastante compleja de usar y requiere un conocimiento profundo de la estructura interna de Portable Document Format. No tiene actualizaciones recientes, por lo que puede haberse quedado algo obsoleto frente a otras bibliotecas y cuenta con una menor documentación.

- **PyPDF2**: esta biblioteca puede usarse como herramienta de línea de comandos para crear o modificar archivos de formato PDF, también se puede ejecutar la biblioteca dentro de los scripts de Python importándolo como un módulo y llamando a sus funciones (esto es especialmente útil si se quiere automatizar tareas), se puede leer, analizar y escribir archivos PDF y permite trabajar con cadenas Unicode para poder manejar distintos caracteres. [39] Se suele utilizar cuando se trabaja con formatos PDF no estructurados.

  Fue una de las últimas en descartar, ya que nos permitía trabajar con datos no estructurados. Pero se vio que para este proyecto era mejor el uso de fitz, ya que se adaptaba mejor a las necesidades que se pretendían solventar.

- **PyMuPDF o Fitz**: permite trabajar con datos no estructurados y contiene una gran cantidad de funciones avanzadas para el procesamiento de PDF. Puede combinarse con otras bibliotecas y tiene una gran documentación de interés. Debido a su alto rendimiento y sólido soporte, se ha determinado como la mejor opción para este proyecto.

Tablas.

Una vez buscada esta información, nos centramos en el entendimiento tanto de los datos que tenemos como del resultado final ideal. Como el objetivo es obtener una tabla que contenga toda la información de una manera ordenada y estructurada, se plantean las posibles tablas específicas. Hay que poder responder a preguntas básicas como ¿cuántas tablas necesito?, ¿cómo las voy a unir?, ¿tengo toda la información para rellenar las tablas?, ¿necesito información adicional?, ¿para qué es cada tabla?

Formación de tablas.

Se ha optado por crear cinco tablas distintas que al unirlas, permitan obtener las tablas finales.

Se ha decidido que la mejor forma de crear las tablas es:

1. **Tabla 1: Pacientes** es la tabla más importante porque nos permite identificar a cada uno de los pacientes por separado.

	Número de chip	Número de paciente	NHC	Número de biopsia	Biopsia sólida	Fecha de informe
0	100	1	00021	23B00021-A1	1	24-mar-2023
1	100	2	00022	22B00022-A2	1	24-mar-2023
2	100	3	00023	21B00023-A1	1	24-mar-2023
3	100	4	00024	23B00024-A1	1	24-mar-2023
4	100	5	00025	23C00025-A1	3	24-mar-2023
5	100	6	00026	23P00026	2	24-mar-2023
6	100	7	00027	22B00027-C4	1	24-mar-2023
7	100	8	00028	23B00028-A2	1	24-mar-2023
8	100	1	1234567	23B000000-A1/CHIP100.1	1	25-may-2023
9	100	2	1234567	23B00000-A1/CHIP100.2	1	25-may-2023

Figura 4.4: Almacenamiento de la información sobre pacientes. Elaboración propia.

2. **Tabla 2: Diagnóstico** es necesaria ya que es capaz de indicar qué diagnóstico tiene cada persona, junto con el número de diagnóstico para que sea más fácil de buscar en el caso de futuros usos.

	Número de chip	Número de biopsia	Diagnóstico	Número del diagnóstico
0	100	23B00021-A1	Carcinoma pulmonar no microcítico	15.1
1	100	22B00022-A2	Carcinoma pulmonar no microcítico	15.1
2	100	21B00023-A1	Carcinoma pulmonar no microcítico	15.1
3	100	23B00024-A1	Carcinoma pulmonar no microcítico	15.1
4	100	23C00025-A1	Carcinoma pulmonar no microcítico	15.1
5	100	23P00026	Carcinoma pulmonar no microcítico	15.1
6	100	22B00027-C4	Cáncer tiroideo	30.0
7	100	23B00028-A2	Cáncer gástrico	11.0
8	100	23B000000-A1/CHIP100.1	Carcinoma pulmonar no microcítico	15.1
9	100	23B00000-A1/CHIP100.2	Carcinoma pulmonar no microcítico	15.1

Figura 4.5: Almacenamiento de la información sobre el diagnóstico de los distintos pacientes. Elaboración propia.

3. **Tabla 3: Mutaciones** recoge toda la información relativa a las mutaciones para cada uno de los pacientes. Es de gran importancia, ya que es la que determina los genes mutados que tiene una persona (tanto benignos como patogénicos) y por tanto, define el diagnóstico.

	Número de chip	Número de biopsia	Mutaciones detectadas	Número de la mutación específica	Total del número de mutaciones	Porcentaje de frecuencia alélica (ADN)	Fusiones ID
0	100	23B00021-A1	[MYC]	[40]	1	[49.56]	[EML4-ALK.E6aA20.AB374361, EML4-ALK.E6bA20.AB3...
1	100	22B00022-A2	[KRAS]	[35]	1	[66.50]	[]
2	100	21B00023-A1	[]	[]	0	[]	[]
3	100	23B00024-A1	[FGFR1, PIK3CA]	[22, 47]	2	[]	[]
4	100	23C00025-A1	[CDK6, MET]	[10, 38]	2	[21.75, 21.75]	[]
5	100	23P00026	[CTNNB1, EGFR]	[11, 13]	2	[27.51, 35.40]	[]
6	100	22B00027-C4	[HRAS]	[28]	1	[46.17]	[]
7	100	23B00028-A2	[KIT, MYC]	[34, 40]	2	[53.12, 60.64]	[]
8	100	23B000000-A1/CHIP100.1	[MYC]	[40]	1	[49.56]	[EML4-ALK.E6aA20.AB374361, EML4-ALK.E6bA20.AB3...
9	100	23B00000-A1/CHIP100.2	[KRAS]	[35]	1	[66.50]	[]

Figura 4.6: Almacenamiento de la información sobre las mutaciones de los pacientes. Elaboración propia.

4. **Tabla 4: Patogénicas** permite almacenar toda la información sobre los genes patogénicos que aparecen en cada uno de los pacientes. Es una tabla independiente a la de mutaciones debido a que son los genes patológicos los que predisponen al paciente a tener ciertas enfermedades o trastornos.

	Número de chip	Número de biopsia	Genes patogénicos	Número de la mutación específica	% frecuencia alélica	Total de mutaciones patogénicas
0	100	23B00021-A1	[]	[]	[]	0
1	100	22B00022-A2	[KRAS]	[35]	[66.50]	1
2	100	21B00023-A1	[]	[]	[]	0
3	100	23B00024-A1	[]	[]	[]	0
4	100	23C00025-A1	[]	[]	[]	0
5	100	23P00026	[CTNNB1]	[11]	[27.51]	1
6	100	22B00027-C4	[]	[]	[]	0
7	100	23B00028-A2	[]	[]	[]	0
8	100	23B000000-A1/CHIP100.1	[]	[]	[]	0
9	100	23B00000-A1/CHIP100.2	[KRAS]	[35]	[66.50]	1

Figura 4.7: Almacenamiento de la información sobre los genes patogénicos. Elaboración propia.

5. **Tabla 5: Información** recoge la información sobre los tratamientos disponibles y los ensayos clínicos para cada diagnóstico. Estos resultados se han binarizado para complementar los resultados.

	Número de chip	Número de biopsia	Ensayos clínicos	SI/NO ensayo	Fármaco aprobado	SI/NO fármacos
0	100	23B00021-A1	6	1	6	1
1	100	22B00022-A2	5	1	1	1
2	100	21B00023-A1	0	0	0	0
3	100	23B00024-A1	0	0	0	0
4	100	23C00025-A1	3	1	3	1
5	100	23P00026	5	1	0	0
6	100	22B00027-C4	1	1	0	0
7	100	23B00028-A2	0	0	1	1
8	100	23B000000-A1/CHIP100.1	6	1	6	1
9	100	23B00000-A1/CHIP100.2	4	1	1	1

Figura 4.8: Almacenamiento de la información sobre tratamientos y ensayos clínicos. Elaboración propia.

Estas tablas se unen mediante una única clave primaria compuesta, formada por dos atributos. Para elegir estos atributos, nos aseguramos de que sean valores específicos y que no se repitan para evitar la formación de filas/columnas espurias. Como el número de biopsia es un número único para cada muestra, se ha determinado como primer atributo. En el caso de que una muestra necesite otro análisis, se ha usado el número de chip como segundo atributo. Esto nos asegura que aunque una muestra necesite dos pruebas, la probabilidad de que se analice la misma prueba dos veces en un chip es nula, lo que nos garantiza que no se crearán filas/espurias.

Unión de tablas.

Se compararon varios métodos para ver con cuál se obtenía el mejor resultado.

- **Concat** concatena tablas o DataFrames. Con este método pueden aparecer columnas espurias si estas tienen nombres duplicados o si no se usa correctamente este método. La unión se realizaba sin tener en cuenta las coincidencias de los valores, por lo que este método no unía realmente los DataFrames, sino que posponía uno tras otro.
- **Merge** es una biblioteca de Pandas bastante rápida que realiza la unión basándose en los valores de las columnas. Cuando aparecen valores duplicados o vacíos o si los nombres de las columnas no coinciden en ambas tablas, aparecen resultados espurios.
- **Join** también pertenece a Pandas. Proporciona opciones flexibles a la hora de realizar las uniones (join, left, rigth o outer). No se crearán columnas espurias al unir los DataFrames porque une usando índices en lugar de columnas. Existen métodos como **lsuffix** o **rsuffix** que permiten agregar sufijos a los nombres de las columnas para evitar resultados irrelevantes.

La mejor opción en este caso es join por no crear resultados espurios tras las uniones, facilitando el proceso de la creación de DataFrames finales.

Tablas resultantes.

La combinación de las cinco tablas iniciales nos ha permitido obtener dos hojas de cálculo, una donde se almacena la información completa y otras donde se almacenan solo 80 líneas de resultado.

Lo más lógico en este caso sería usar una base de datos porque permitiría almacenar los datos de una manera ordenada manteniendo la integridad y consistencia de los datos. Pero como el usuario necesitaba una forma sencilla de almacenar y modificar posteriormente los resultados, se decidió usar Excel.

Antes de la selección final, se estudió la diferencia entre dos ficheros distintos, CSV y Excel. [40] Se ha determinado que la mejor forma de exportación era Excel, ya que nos devolvía los resultados de una forma mucho más ordenada que CSV, por filas y columnas sin necesidad de usar separadores. Además, Excel proporciona una gran cantidad de fórmulas y

funciones para trabajar con los datos y facilita la formación de gráficos, ofreciendo una gran ventaja al hospital, por si en un futuro se quiere modificar o trabajar sobre los datos que aparecen en este fichero resultante.

Líneas de trabajo futuras

Este proyecto tiene un gran potencial para ser usado en el campo de farmacogenética. Se espera que en un futuro, pueda ser mejorado para adaptarse a las necesidades de los distintos hospitales y así, permitir el análisis e interpretación de un mayor número de datos.

También se espera que este proyecto sea capaz de formar parte de empresas dedicadas a la farmacogenética, para analizar como reaccionarían distintos genes a fármacos determinados para conseguir que el cáncer sea una enfermedad más fácil de tratar y que no se encuentre ente las primeras causas de muerte mundiales.

Partiendo del punto en el que se ha dejado y enfocado a su uso como herramienta futura en el área de salud, se propone configurarlo para que sea capaz de ejecutarse de forma automática en intervalos regulares de tiempo determinados por el investigador principal y una vez finalizada la ejecución, enviar una notificación, tanto al ordenador desde el que se ejecuta como a un dispositivo electrónico de interés, para indicar que los resultados están listos.

También se deja pendiente el crear una base de datos en SQL para que almacenar toda la información obtenida y realizar posteriores estudios partiendo de ella.

Sería interesante si se lograra implementar dicho código (con mejoras en el caso de que se viera necesario) en otros lenguajes de programación como R, por ser uno de los lenguajes enfocado especialmente en análisis de datos y muy utilizado en campos de las ciencias de la vida y la salud. Una vez obtenido este código, se pueda implementar una aplicación alojada en la web como servicio usando por ejemplo el framework Shiny. Shiny es una herramienta de desarrollo web de código abierto creada por *Posit, Inc.* que

permite construir aplicaciones interactivas basadas en R y desplegarlas en la web, por lo que para llegar a este punto se debería tener el código en R, aunque también da la opción de usar Python.

Hay que tener en cuenta que si cambia el formato de los archivos exportados por el software, es posible que el código no funcione correctamente, por lo que haría que realizar ciertas modificaciones para actualizarlo y que fuera capaz de devolver la misma información en el mismo formato.

Bibliografía

[1] Sonia Ordoñez. *¿Qué es anatomía patológica?* Página usada en la memoria de Anatomía Patológica 2021-2022 durante prácticas. URL: https://medac.es/blogs/sanidad/que-es-anatomia-patologica.

[2] OMS. *Cáncer*. (2022). URL: https://www.who.int/es/news-room/fact-sheets/detail/cancer.

[3] David Rodríguez-Lázarob y José María Eirosc Marta Hernández Narciso Quijada. *Aplicación de la secuenciación masiva y la bioinformática al diagnóstico microbiológico clínico*. URL: https://www.elsevier.es/es-revista-revista-argentina-microbiologia-372-articulo-aplicacion-secuenciacion-masiva-bioinformatica-al-S0325754119300811.

[4] Zakhar Yung. *10 Best Data Extraction Tools in 2023 For Your Business*. (2023). URL: https://blog.coupler.io/data-extraction-tools/.

[5] ADEA. *La automatización de procesos en el sector Salud*. (2022). URL: https://www.adea.es/blog/automatizacion-salud/.

[6] G2. *ReporMiner*. Página de reseñas. (2023). URL: https://www.g2.com/products/reportminer/reviews.

[7] ASTERA. *ASTERA: ReporMiner*. (2023). URL: https://www.astera.com/es/products/report-miner/.

[8] Apify. *Build reliable web scrapers. Fast.* Uso de la herramienta. URL: https://apify.com/.

[9] Nanonets. *Automate Manual Data Entry Using AI*. Uso de la herramienta. URL: https://nanonets.com/.

[10] CTakes. *Apache cTAKES*. Uso de la herramienta. URL: https://ctakes.apache.org/.

[11] Clinithink. *The key to saving human lives is understanding human words*. Uso de la herramienta. URL: https://www.clinithink.com/.

[12] Universidad Johns Hopkins. *OMIM*. URL: omim.org.

[13] ThermoFischer SCIENTIFIC. *Oncomine Reporter*. URL: https://www.thermofisher.com/order/catalog/product/es/es/A34298.

[14] Python TM. *What is Python? Executive Summary*. (2023). URL: https://www.python.org/doc/essays/blurb/.

[15] Python Institute. *Python® – the language of today and tomorrow*. (2023). URL: https://pythoninstitute.org/about-python.

[16] Wes McKinney. *Python for Data Analisis*. O'RIELLY, (2022).

[17] Wikipedia. *Numpy*. WIKIPEDIA, (2023).

[18] Python. *re — Regular expression operations*. URL: https://docs.python.org/3/library/re.html.

[19] Python. *os — Miscellaneous operating system interfaces*. URL: https://docs.python.org/3/library/os.html.

[20] Anaconda. *ANACONDA*. (2023). URL: https://www.anaconda.com/.

[21] Anaconda. *ANACONDA*. (2023). URL: https://www.anaconda.com/about-us.

[22] Code Academy. *What Is an IDE?* (2023). URL: https://www.codecademy.com/article/what-is-an-ide.

[23] Dimitris Poulopoulos. *Jupyter Is Now a Full-fledged IDE: Annual Review*. (2022). URL: https://towardsdatascience.com/jupyter-is-now-a-full-fledged-ide-annual-review-751675634493.

[24] GitHub. *Aprende sobre el sistema de control de versiones, Git y cómo funciona con GitHub*. (2023). URL: https://docs.github.com/es/get-started/using-git/about-git.

[25] XATAKA. *Qué es Github y qué es lo que le ofrece a los desarrolladores*. (2019). URL: https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores.

[26] Wikipedia. ***GitHub***. (2023). URL: https://en.wikipedia.org/wiki/GitHub.

[27] Red Hat. *¿Qué es el open source?* (2023). URL: https://www.redhat.com/es/topics/open-source/what-is-open-source.

[28] ALTASSIAN. *Git Branch*. URL: atlassian.com/git/tutorials/using-branches.

[29] Wikipedia. *LaTeX*. (2023). URL: https://es.wikipedia.org/wiki/LaTeX.

[30] *LaTeX General Help*. (2016). URL: http://www.personal.ceu.hu/tex/latex.htm.

[31] *¿Qué es Excel y para qué sirve?* blog. (2023). URL: https://excelparatodos.com/que-es-excel/?utm_content=cmp-true.

[32] *Britannica*. (2023). URL: https://www.britannica.com/technology/Microsoft-Excel.

[33] Alexander Gillis. *Excel*. URL: https://www.techtarget.com/searchenterprisedesktdefinition/Excel#:~:text=Excel%20is%20a%20spreadsheet%20program,calculate%20data%20in%20a%20spreadsheet.

[34] draw.io. *The easiest way for Confluence teams to collaborate using diagrams*. URL: https://drawio-app.com/.

[35] Wikipedia. *PDF*. (2023). URL: https://euske.github.io/pdfminer/.

[36] Aaron Zhu. *How to Scrape and Extract Data from PDFs Using Python and tabula-py*. (2021). URL: https://towardsdatascience.com/scrape-data-from-pdf-files-using-python-fe2dc96b1e68.

[37] Aaron Zhu. *How to Scrape and Extract Data from PDFs Using Python and PDFQuery*. (2022). URL: https://towardsdatascience.com/scrape-data-from-pdf-files-using-python-and-pdfquery-d033721c3b28.

[38] *PDFMiner*. (2016). URL: https://euske.github.io/pdfminer/.

[39] Dhana Shree. *PYPDF2 Library: How Can You Work With PDF Files in Python?* (2022). URL: https://nanonets.com/blog/pypdf2-library-working-with-pdf-files-in-python/.

[40] Toggl track. *Difference Between CSV and XLS*. (2023). URL: https://toggl.com/track/difference-between-csv-xls/#:~:text=The%20difference%20between%20CSV%20and,including%20both%20content%20and%20formatting.