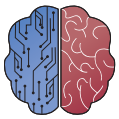
UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería de la Salud



INGENIERÍA
DE LA SALUD

TFG del Grado en Ingeniería de la Salud

Automatización de extracción de datos de informes de secuenciación masiva y análisis.

Presentado por Lucía Vítores López
en Universidad de Burgos

19 de mayo 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 el alumno 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 y análisis.

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, 19 de mayo 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 en el 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 procesamiento.

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 escogiendo los datos.

Descriptores

Anatomía patológica, cáncer, genes, automatizar, extracción de datos, Python, GitHub ...

Abstract

This final degree project is about data extraction automation in the Pathological Anatomy Service at Burgos Hospital.

Manual extraction is a difficult job and often prone to mistakes. However, the use of automation can improve efficiency and accuracy of the process. The objective of this project is to implement an automatic system for retrieve information from various files using different processing techniques.

Results show that implementing automation has a lot of benefits and improvements when working with results. It provides specific information about cancer to study different types and genes and it's an easy way to not waste time.

Keywords

Pathological anatomy, cancer, genes, automate, data extraction, Python, GitHub ...

Índice general

Índice de figuras

Índice de tablas

Objetivos

En este apartado aclararemos los objetivos generales del proyecto, así como las posibles dudas sobre su finalidad.

1.1. Objetivos marcados por software, hardware o análisis.

Los principales objetivos que abarcan este apartado son:

- Desarrollo de un algoritmo capaz de facilitar la extracción de datos en el proyecto Secuenciación múltiple de última generación en tumores sólidos: utilidad clínica en un Hospital de tercer nivel del Hospital Universitario de Burgos.
- Programación del algoritmo.
- 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.
- Mejora de la eficiencia del resultado final.

1.2. Objetivos técnicos.

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

- Lectura de todos los PDF proporcionados para la obtención de la información.
- Creación de código sencillo que seleccione las variables de interés de un PDF usando técnicas de PDFscraping.
- Separación de la información en tablas específicas.
- Obtención de varios archivos .CSV con los resultados.
- Implementar dicho algoritmo en otros ordenadores.

1.3. Objetivos de aprendizaje

Los principales objetivos con los que se pretende finalizar el código son:

- Adquisición de nuevos conocimientos en el ámbito anatomopatológico.
- Aumento de conocimiento en el ámbito de programación.
- Uso de distintas herramientas para un resultado final adecuado.
- Comprensión de la información biológica para su aplicación en el desarrollo del código.

Introducción.

Se trata de un proyecto que estudia la forma de solucionar el problema de obtención manual de los datos en la máquina Oncomine Reporter del servicio de Anatomía Patológica en el Hospital Universitario de Burgos (HUBU) mediante la automatización de la extracción de los datos, partiendo de los campos más relevantes para el manejo de datos de secuenciación masiva generados directamente por el software comercial en un informe en formato PDF.

De forma que lo que se intenta es que la extracción sea mucho más rápida y eficaz obteniendo solo los datos de interés, para generar nuevo conocimiento en la investigación de cáncer y solucionar problemas tanto prácticos como teóricos que ayuden en el diagnóstico y tratamiento de la enfermedad del paciente o de los futuros pacientes.

Es un estudio no experimental, ya que únicamente se trabaja usando los datos empleados por el HUBU, en ningún caso se modifican dichos datos ni existe repercusión clínica de ningún tipo.

A lo largo del trabajo se podrá encontrar todo lo necesario para entender las bases del proyecto (tanto información biológica como informática), cómo se ha organizado el tiempo e incluso las posibles mejoras. En los distintos apéndices se encontrará información mucho más específica y desarrollada de algunos de los apartados que necesitaban más explicación y el código junto con una explicación de su funcionamiento.

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. [24]

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.

Debido a la gran cantidad de pruebas que se realizan diariamente y los datos que se obtienen en este sector, se vio la necesidad de automatizar la extracción de datos para hacerlo de una forma mucho más rápida y organizada, con el fin de facilitar la tarea a los profesionales y la comprensión de los resultados obtenidos.

Por ello, en este proyecto nos vamos a centrar en trabajar con grandes cantidades de informes para obtener datos específicos considerados de mayor importancia y que puedan ser usados posteriormente en análisis.

Automatización de extracción de datos.

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

La finalidad de automatizar este proceso es ahorrar tiempo y recursos a la hora de almacenar la información. Es un procedimiento generalmente útil y que muchas empresas están empezando a usar, ya que estas herramientas permiten 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. [29] Las empresas lo usan principalmente para trabajar con datos de ventas, financieros, de marketing...

Aunque también es posible enfocarlo al ámbito médico [4] 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 se pretende anticiparse a los hechos y 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.

Las herramientas encargadas obtienen los datos de una fuente y lo transforman en un formato más útil y fácil de entender.

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.

Para ello se sigue el proceso ETL:[8]

- Extraer datos de sus respectivas fuentes.
- Transformar el formato en el necesario para trabajar con la herramienta final.
- Cargar el resultado final en el sistema destino.

Cáncer.

El cáncer es la principal causa de muerte en el mundo según la Organización Mundial de la Salud. En 2020 se atribuyeron casi 10 millones de muertes. Puede afectar de igual forma a hombres que mujeres, aunque hay estudios que determina que los hombres tienen una mayor tasa. Es una enfermedad bastante difícil de detectar, ya que hay muchas variables y factores que deben tenerse en cuenta y son específicos en cada persona. En las primeras etapas los síntomas son poco notables o incluso pueden llegar a no detectarse, también es posible que sean confundidos con los síntomas de otras enfermedades.[23] Los tipos más comunes son el cáncer de mama, de próstata, de pulmón y colorrectal. En muchos casos se pueden curar siempre y cuando se detecten a tiempo y se traten de la forma más eficaz posible. Por ello cuanto antes se diagnostique esta enfermedad y cuanto más específico sea el tratamiento, más probabilidades hay de vencerla.

Çáncer"(o sus sinónimos "tumores malignos.º "neoplasias malignas") es un término que designa a un gran número de enfermedades que afecta a cualquier parte del organismo.

El cáncer puede comenzar en cualquier parte del cuerpo, ya que este está formado por billones de células. En condiciones normales, las células humanas se forma y se multiplican mediante división celular para formar nuevas células. Cuando éstas envejecen o se dañan, mueren y son reemplazadas por las nuevas. Esta información está muy bien explicada en [18]. En algunos casos

no ocurre este proceso. Las células dañadas se multiplican en lugar de desaparecer, llegando a formar tumores, ya sean benignos o malignos. Los tumores malignos son capaces de invadir tejidos cercanos o incluso tejidos lejanos, proceso conocido como metástasis. El cáncer metastásico tiene el mismo nombre y el mismo tipo de células que el cáncer primario del que proceden. Un ejemplo que explica muy bien este proceso se encuentra en NIH y dice: .[El] cáncer de seno (mama) que forma un tumor metastásico en el pulmón es cáncer de seno metastásico, no es cáncer de pulmón."

A diferencia de los tumores benignos, que no son capaces de desplazarse a otro tejido. En algunas ocasiones son muy grandes, pero una vez que se extirpan, no vuelen a aparecer. Cosa que no se garantiza con los tumores malignos o cancerígenos, ya que es posible que estos tumores remita, es decir, que los signos y síntomas de la enfermedad hayan desaparecido por completo pero sigan existiendo células malignas capaces de volver a causar la enfermedad. Se puede encontrar más información relacionada con el tema en [17]. La mayoría de los tipos de cáncer, generalmente vuelven a aparecer en los primeros 5 años tras el tratamiento, aunque también es posible que lo hagan una vez pasado ese periodo. Debido a esto, es de gran importancia realizar revisiones una vez pasada la enfermedad para ver si los posibles signos o síntomas vuelven a aparecer.

Las células cancerosas son muy distintas a las normales. Un ejemplo de ello es que las cancerosas se originan sin ningún tipo de señal, mientras que las normales solo se replican cuando existe una señal que lo indica, además, las células cancerosas no hacen caso a las señales que indican muerte produciendo la conocida muerte celular programada o apoptosis.

Las posibles causas de sufrir esta enfermedad son muy variadas, abarcan genética (hay ciertas mutaciones genéticas que pueden aumentar el riesgo de desarrollo de distintos tipos), edad (el riesgo de cáncer aumenta con la edad, puede deberse a que se van acumulando factores de riesgo), estilo de vida (hábitos poco saludables como el tabaco, el alcohol o una vida sedentaria aumenta el riesgo), la exposición a sustancias tóxicas (como productos químicos o radiactivos) o infecciones (virus del papiloma humano VPH o hepatitis B y C). Los principales cambios genéticos que causan el cáncer se deben a errores en la multiplicación de las células, daños en el ADN (ácido desoxirribonucleico) o por herencia.

El cuerpo humano es capaz de eliminar aquellas células que están dañadas antes de que se vuelvan cancerosas. Sin embargo, con el paso de los años, la capacidad del cuerpo de realizar este proceso va desapareciendo, lo que explica el aumento de riesgo.

Los cambios génicos que contribuyen a esta enfermedad suelen afectar a tres genes:

- **Protooncogenes**: genes encargados del crecimiento normal de las células. Sus mutaciones pueden conseguir que se conviertan en genes causantes del cáncer o también conocidos como oncogenes. [15]
- **Genes supresores de tumores**: genes encargados de la codificación de proteínas para controlar la división celular. Cuando estos genes se inactiva, la proteína deja de funcionar correctamente, provocando una división celular descontrolada. [16]
- **Genes de reparación de ADN**: genes encargados de la reparación de los errores cometidos durante la transcripción. [10]

Al tener tanta información sobre estos genes, los científicos han sido capaces de determinar las mutaciones más comunes en los distintos tipos de cáncer. Lo que conlleva un mayor número de tratamientos dirigidos a esos genes específicos. Esto hace posible aplicar un tratamiento determinado en cualquier tipo de cáncer siempre que contenga la mutación específica.

La Organización Mundial de la Salud (OMS) ha determinado que alrededor del 30 - 50 porciento de los casos de cáncer pueden ser evitados reduciendo los factores de riesgo y aplicando estrategias preventivas. A esto hay que añadirle la detección precoz de la enfermedad, que aumentaría la probabilidad de disminución.

La detección precoz tiene dos componentes principales:

1. **Diagnóstico precoz**: cuando el cáncer se detecta pronto, es probable que responda mejor al tratamiento. Consta de tres componentes:
   - Conocimiento de la sintomatología y la importancia de acudir al médico cuando se ven anomalías.
   - Acceso a los servicios clínicos de evaluación y diagnóstico.
   - Derivación del paciente a los servicios de tratamiento más adecuados.
2. **Tamizaje** o cribado, que consiste en detectar indicios de un cáncer concreto. Cuando se encuentran anomalías en este proceso, habría que llevar a cabo las pruebas pertinentes para confirmar o descartar el diagnóstico y derivar al paciente a los servicios necesarios en caso

de que sea necesario. Sin embargo, el principal inconveniente de este proceso es que no es igual de eficaz para los distintos tipos, ya que algunos son mucho más complejos y requieren estudios más específicos.

Para su tratamiento, es imprescindible un buen diagnóstico, ya que cada tipo de cáncer requiere un tratamiento concreto. Algunos tratamientos son intervenciones quirúrgicas, quimioterapia, radioterapia o terapia sistémica.

Hay tipos que son más frecuentes que el resto, como por ejemplo el cáncer de mama o el de cuello uterino, que tienen una mayor tasa de curación cuando se detectan de forma temprana debido a los grandes conocimientos que se tienen sobre el tema.

El principal objetivo a la hora de aplicar el tratamiento es su cura o la prolongación de la vida del paciente. Sin embargo, hay otros casos en los que el objetivo fundamental es mejorar la calidad de vida del paciente mediante cuidados paliativos.

Oncomine Reporter.

Oncomine Reporter es una herramienta de software de análisis génico desarrollada específicamente para un examen más profundo de la secuenciación masiva, lo que permite un informe final en pasos muy sencillos. Es el software usado por el servicio de Anatomía Patológica del Hospital Universitario de Burgos y del cuál obtendremos los resultados.

Permite una investigación contextual de variantes específicas de la muestra para comprender su uso con respecto a ensayos clínicos globales actuales. 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. También es muy útil a la hora de desarrollar nuevos fármacos.

Partiendo de una muestra del paciente, el software es capaz de cargar los datos en una plataforma y usar distintas herramientas o algoritmos para identificar los patrones de expresión génica relevantes para el cáncer. 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.

Sin embargo, uno de los principales inconvenientes es que devuelve falsos casos. Es decir, algunos polimorfismos que son benignos vienen como patológicos, por lo que es importante revisarlo de forma manual para evitar estos fallos.

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. Cada vez se están usando un mayor número de técnicas de aprendizaje automático, procesamiento de lenguaje natural y minería de datos para conseguir mejoras en este campo.

Algunas de las herramientas más destacados son:

1. **Astera ReportMiner**: es un software de extracción de datos automatizado a nivel empresaria 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. La interfaz de usuario visual simplifica mucho la extracción, mientras que reduce el esfuerzo manual y tiempo que esto conllevaría. 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 [13] y [9]

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. [6]

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. [22]

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. Tiene dos entornos gráficos para su uso, aunque también se puede usar por comandos. Sin embargo, no es muy útil cuando personas no técnicas necesitan usarlo y los resultados obtenidos no sirven de gran utilidad de forma directa (es decir, habría que trabajar con dichos datos para poder usarlos una vez se hayan obtenido). [12]

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 a cada uno y en su conjunto, unos mejores resultados. Actualmente se esta usando la información que recoge para mejorar algunas áreas.

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. Por ello es importante que existan distintas herramientas especializadas en los distintos campos, ya que se entiende que una herramienta desarrollada principalmente para la extracción de datos en una empresa de marketing no es capaz de satisfacer la extracción de información necesaria en un hospital.

***** También puede ser necesario utilizar notas al pie [1], para aclarar algunos conceptos.

[1] como por ejemplo esta

Metodología

3.1. Descripción de los datos.

(Breve descripción de los datos. En caso de tratarse de un trabajo donde los datos son muy importantes, puede haber explicaciones extra en el anexo correspondiente.)

Los datos necesarios para llevar a cabo este trabajo de fin de grado han sido proporcionados por el Hospital Universitario de Burgos, provenientes de una fuente primaria de datos como es Oncomine Reporter. Se trata de distintas carpetas en las que hay PDF con la información resultante de los análisis de los pacientes.

Al tratarse de datos de pacientes reales, se ha firmado un acuerdo de confidencialidad en conjunto con el HUBU para no mostrar los datos reales. De este modo, a la hora de presentar el código frente al tribunal, se han usado unos PDF inventados (que no contienen ni nombres de personas ni del hospital ni datos reales de pacientes) aportados por la investigadora del Hospital Universitario con el fin de tener unos modelos similares a los resultantes de .ºncomine Reporterçon los que poder trabajar y enseñar el resultado. Esto nos permite trabajar de una manera más segura, reduciendo los riesgos de los interesados y 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 el 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 suficiente avanzado como para ser capaz de programar dicho algoritmo.

SQL se ha usado para crear las primeras tablas, debido a su gran capacidad de manejo de tablas.

Tanto Python como R han sido lenguajes usados en clase (aunque principalmente Python). Se ha usado Python porque tiene una gran biblioteca de estándares, herramientas e integración con otros lenguajes de programación. El uso de R destaca por ser un lenguaje extensible y que se ejecuta en muchos sistemas operativos, además de permitir trabajar con grandes conjuntos de datos. Sin embargo, no se ha usado para este código.

También se ha tenido en cuenta el ranking de los lenguajes de programación más usados en 2022, podemos verlo en [19]

3.3. Técnicas y herramientas.

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

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, los posibles problemas o dudas que han surgido.

GitHub [28]se trata de una de las principales plataformas usadas para la creación de distintos repositorios abiertos que puede ser colaborativa (donde

todos los usuarios pueden aportar algo de ayuda para la mejora del código) o privada (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.[14] 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.

Sin embargo, también ofrece unas herramientas propias que permiten complementar el programa. Permite dejar anotaciones en el código y optimizarlo, gráficos para informar sobre el tiempo que pasa cada usuario, descargar el trabajo...

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, así como de los comandos necesarios para su uso.

El repositorio creado para este trabajo se ha denominado Automatizacion-PDF-scraping (siendo los guiones en realidad barras bajas, pero por un fallo de LaTeX se determinan de esta manera) y al tratarse de datos reales, se ha mantenido en todo momento privado, de forma que tan solo el tutor y la alumna encargada de su realización podían tener acceso.

Sin embargo, cuando el repositorio pasa de privado a público, no existirá ningún problema, ya que los ficheros con los que se trabajan no contienen ningún nombre capaz de relacionar dichos resultados con el paciente real.

Una vez creado el repositorio y descargado GitHub Desktop, podemos añadir, eliminar, modificar distintos ficheros y que queden almacenados los cambios en el propio repositorio. A la hora de subir cualquier fichero desde Desktop se le asigna un nombre y una descripción (el nombre es obligatorio pero la descripción no) y al subirlo se debe hacer un push para indicar que se quieren llevar a cabo las modificaciones que se han realizado. Una vez hecho el push, el documento ya aparece en el repositorio de GitHub con la descripción dada y la información sobre cuándo se ha subido.

En el caso de que haya un segundo colaborador, el trabajo es algo más complejo, ya que se debería de realizar previamente un pull para importar al repositorio local los cambios que se han realizado en el repositorio antes

de hacer el push. De esta forma se garantiza que no se pierda ningún tipo de información y que se trabaje de manera ordenada.

Se ha creado un branch para trabajar en una rama nueva paralela a la principal, es decir, permite hacer cambios en los archivos sin modificar la rama principal (también conocida como main). [5]

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

Para la organización del contenido de cada semana, se han ido creando distintos Milestones donde se almacenan los distintos Issues, que es donde se explica que objetivos se pretende finalizar en cada periodo de tiempo entre reuniones.

El principal problema que ha surgido a la hora de usarlo ha sido la recuperación de TAC o Personal Access Token que nos permite acceder a los recursos en GitHub con nuestro nombre. También se ha intentado unir la memoria de LaTeX con el propio repositorio, pero no ha sido posible porque no existía una versión gratuita para su uso.

EXPLICAR UN POCO EL FICHERO FUNCIONESGIT!!!!!!

Anaconda se ha usado para el desarrollo del código en Python. [1] Es una de las plataformas más populares para la implementación de código en este lenguaje.

Se ha usado Python [20] porque es el lenguaje de programación que más se ha usado en la carrera y por tanto, del que más conocimiento se tiene. Además, cuenta con una gran cantidad de bibliotecas que ya vienen instaladas, es un lenguaje fácil de usar y de alto nivel (tiene en cuenta las capacidades cognitivas de los humanos y no las capacidades de las máquinas para llevar a cabo órdenes).

Para ello ha sido necesario descargar Anaconda Navigator. Aparecen una serie de aplicaciones que ya vienen por defecto, como por ejemplo Jupyter Notebook.En Anaconda podemos instalar las distintas aplicaciones, y como a nosotros nos interesa trabajar con Jupyter, lo instalamos. Una vez instalado, podemos acceder siguiendo la ruta de las carpetas a la carpeta que tiene toda la información del proyecto. Ahí podemos crear un nuevo fichero para el desarrollo del código necesario.

Se había creado inicialmente un notebook llamado PruebaLucia.ipynb para crear el código, aunque posteriormente se ha modificado el nombre

a CodigoPython.ipynb para que esté más acuerdo con el proyecto y cuyo contenido se puede encontrar en el anexo de manual del programador.

LaTeX es un sistema para crear textos estructurados o con fórmulas matemáticas. 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 en parte de [27]

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... 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. Lo mismo ocurre con las enumeraciones (begin - end enumerate, begin - end itemize...), con las tablas (begin - end tabular), con las imágenes (begin - end figure)...

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.

También es posible añadir imágenes o tablas para aclarar el contenido del texto, hacer enumeraciones o introducir fórmulas matemáticas.

El paquete BibTeX nos permite almacenar los distintos libros o páginas web que se han utilizado y hacer referencias en el texto mediante identificadores. Para ello se crea un archivo .bib nuevo 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 como puede ser PDF.

Excel es una hoja de cálculo que nos permite trabajar tanto con datos numéricos como texto en distintas tablas formadas por líneas y columnas. Estas hojas de cálculo nos permite analizar o realizar distintas acciones mediante gráficos o tablas. [3]

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 fila siguiente al último elemento junto con 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.

PgAdmin 4 es una herramienta usada a lo largo del curso que permite gestionar y administrar PostgreSQL. Es la principal base de datos de código abierto. Se ha usado principalmente para la creación de las tablas que se van a tener en cuenta a lo largo del proyecto.[21] Se trata de una herramienta de administración y gestión de bases de datos de código abierto diseñada para PostgreSQL.

Es muy intuitiva y fácil de usar, contiene iconos muy visuales, permite monitorizar el estado del servidor y de las bases de datos usadas, alta velocidad, una interfaz flexible... lo que hace que sea muy cómoda y simple para su uso.

DRAW.IO PARA DIAGRAMAS??

Conclusiones.

4.1. Resumen de resultados.

Como se ha comentado anteriormente, el cáncer es una enfermedad bastante complicada de detectar y cuanto antes se detecte, más probabilidades hay de superarla usando el tratamiento más adecuado.

La creación de este algoritmo permitirá tener una información más detallada sobre los distintos tipos de cáncer, los genes implicados... en distintas personas para poder detectar antes el cáncer y ser capaces de crear tratamientos enfocados a pacientes individuales basándose en sus genes (farmacogenética).

Se han detallado algoritmos o aplicaciones capaces de extraer datos de PDF en distintas empresas, pero ninguno capaz de satisfacer las necesidades del Hospital Universitario de Burgos. Por lo que la creación desde cero de un proyecto 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.

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.

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 se secuenciación múltiple de última generación en tumores sólidos.

Podemos distinguir dos tablas finales, una con la información de todos los genes (Número de chip, Número de paciente, NHC, Número de biopsia, Biopsia sólida, Fecha de informe, Diagnóstico, Número del diagnóstico, Mutaciones detectadas, Número de la mutación específica, Total del número de mutaciones, Porcentaje de frecuencia alélica (ADN), Fusiones ID, Ensayos clínicos, SI/NO ensayo, Fármaco aprobado, SI/NO fármacos) y otra con la información de los genes patogénicos (Número de chip, Número de paciente, NHC, Número de biopsia, Biopsia sólida, Fecha de informe, Diagnóstico, Número del diagnóstico, Genes patológicos, Número de la mutación específica, Porcentaje de frecuencia alélica, Número de mutaciones patológicas, Ensayos clínicos, SI/NO ensayo, Fármaco aprobado, SI/NO fármacos). Estas son las columnas de cada tabla obtenidas en el resultado final. De esta forma, cada una de las filas de resultados definirían los datos de un fichero.

4.2. Discusión.

(Discusión y análisis de los resultados obtenidos.) Los resultados obtenidos son una nueva forma de ordenar y entender los datos, en ningún momento se han creado nuevos datos. Lo único que ha cambiado ha sido el formato de los datos, que ha pasado de un formado PDF inicial a un formato Excel final. [7]

La idea de esto es obtener un archivo Excel que almacene todos los resultados de interés para que luego sea más fácil tanto editarlos como trabajar sobre ellos. Al convertir los datos de PDF a Excel se ha conseguido un resultado en estructura tabular mucho más legible y mostrando solo los resultados de interés de una forma mucho más clara sin tener que buscar cada valor en todo el PDF, ya que estos últimos no tienen una estructura tabular definida.

Además, los archivos PDF son bastante más difíciles de manipular, editar o escribir sobre ellos, son principalmente usados para la visualización. De esta forma se garantiza que cualquier cambio que se desee realizar pueda hacerse con la mayor brevedad y facilidad posible. Los PDF suelen mantener el formato original, en Excel es posible modificar los valores de las celdas, letras, bordes, estilo de la hoja...

El principal inconveniente es que los PDF son ampliamente compatibles y pueden ser abiertos en casi cualquier dispositivo sin necesidad de tener descargado algún programa específico. Esto no ocurre con Excel, ya que es necesario tener Microsoft Excel o un programa compatible para poder abrir este tipo de ficheros. Nos hemos asegurado de que el ordenador principal

con el que se va a trabajar es capaz de abrir ficheros Excel sin ningún inconveniente.

4.3. Aspectos relevantes.

((Este apartado pretende recoger los aspectos más interesantes del **desarrollo del proyecto**, comentados por los autores del mismo. Debe incluir los detalles más relevantes en cada fase del desarrollo, justificando los caminos tomados, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del proyecto y también los resultados negativos obtenidos por soluciones previas a la solución entregada. Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.))

Como se ha comentado en apartados anteriores, el uso de Python como lenguaje de programación para el desarrollo del código se debe a que tiene una gran cantidad de bibliotecas que facilitan el código. Se ha usado el libro *Python for Data Analisis* para la ayuda del código de Python [20]

Para la realización del código se comenzó con la búsqueda de información sobre PDF-scraping y las distintas bibliotecas que podían usarse. Para ello se buscaron ventajas e inconvenientes sobre cada una de las bibliotecas.

Una vez buscada esta información, nos centramos en el entendimiento tanto de los datos como del resultado final ideal para poder crear las tablas de una forma organizada. Se pretendía poder responder a preguntas básicas como ¿cuántas tablas necesito?, ¿cómo las voy a unir?, ¿tengo en los datos toda la información para rellenar las tablas? Se ha usado PgAdmin 4 para el trabajo con tablas, por las ventajas comentadas en el apartado anterior. Al entrar daba un problema, por lo que la mejor solución fue eliminar la aplicación y volver a descargarla. Aún así da varios errores al volver a descargar y ejecutar, esto se debe principalmente a un error en el intento de acceso a un socket no permitido por el permiso de acceso, lo que conlleva fallos de conexión en el servidor localhost y en la autenticación para el usuario postgres. Para solucionarlo se ha introducido la ruta del ejecutable en el path del sistema.

Al principio había 6 tablas. La primera de ellas llamada **1-Paciente** contenía el número de chip, el número de paciente, el NHC, el número de biopsia y la fecha de informe. La segunda tabla **2-Biopsia** contenía el apartado de

biopsia sólida, el número de biopsia, el texto diagnóstico y las mutaciones detectadas. La tercera tabla **3-Mutaciones** contenía las mutaciones detectadas, porcentaje de frecuencia alélica, fármacos aprobados, ensayos clínicos y patogénico-benigno. La cuarta **4-Subtipo mutación** estaba formada por las columnas mutaciones detectadas, tipo de mutación específico y número de mutación. La penúltima tabla **5-Diagnóstico** contenía número de diagnóstico y texto diagnóstico. La última tabla **6-Especificación de mutaciones** formada por el nombre de la mutación y el número correspondiente. No se hizo así porque las columnas no correspondían con el resultado final y las columnas no estaban bien separadas en columnas. Un ejemplo de ello es que las columnas número de biopsia y biopsia sólida deberían ir en la misma tabla, ya que una depende de la otra y no en tablas diferentes. También se había creado una columna extra llamada Patogénico - Benigno para que almacenara cómo era el gen, sin embargo, no se tenía en cuenta si no era ninguno de los dos valores.

Las siguientes tablas que se crearon fueron las siguientes. **1-Pacientes** formada por número de chip, número de paciente, NHC, número de biopsia, biopsia sólida, fecha de informe y diagnóstico, **2-Diagnóstico** que contenía las columnas diagnóstico y número del diagnóstico,la tercera **3-Mutaciones** posee mutaciones detectadas de ADN, porcentaje de frecuencia alélica, fusiones de genes (ARN), número de lecturas, variaciones del número de copias, número de copias y número total de mutaciones. La tabla **4-Mutaciones específicas** formada por las mutaciones de ADN, fusiones de genes (ARN), variaciones del número de lecturas y el número correspondiente de cada uno de ellos. La última tabla contenía información sobre los fármacos y ensayos, siendo **5-Información** una tabla con columnas diagnóstico, ensayos clínicos y fármaco aprobado. Estas tablas tampoco han sido las definitivas, ya que no contenían toda la información necesaria para el resultado final. Además, contenía información no necesaria. Un ejemplo de esto último era la separación de mutaciones en función de si era ADN, ARN o variaciones del número de copias, ya que esta información no interesaba, lo importante era ser capaz de obtener todas las mutaciones en una variable.

Finalmente se volvieron a actualizar las tablas hasta obtener las actuales, las cuales se pueden ver en el fichero tablas.sql que se encuentra subido en el repositorio. La diferencia de las tablas actuales frente a las otras es la buena organización de las columnas realmente necesarias. En la primera tabla se almacena información específica de cada paciente. La segunda se basa en el diagnóstico. La tercera sobre las mutaciones y como podemos ver, no separa las mutaciones por si es ADN, ARN o variaciones en el número de copias, sino que hay una variable que almacena todas ellas. Esta

tabla tiene en cuenta todos los genes, excepto los benignos, ya que estos se ha demostrado que no tienen ningún peligro en la salud del paciente y, por tanto, no son de interés. También se ha creado una columna nueva (Patogénicas) que almacena información sobre las mutaciones patogénicas específicas. La última tabla está formada por los valores de ensayos clínicos, fármacos aprobados y dos columnas nuevas creadas para la binarización por individual de las columnas anteriores.

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 NHC es un número único e intransferible para cada persona, se ha determinado como primer atributo. En el caso de que una persona pudiera realizarse más pruebas, se ha usado el número de chip como segundo atributo. Esto nos asegura que aunque un paciente se realice dos pruebas en dos días seguidos, no se crearán filas/espurias, ya que como en cada chip hay tan solo ocho pacientes, es imposible que sus dos muestras se estudien en el mismo chip.

Para la realización del código, inicialmente se intentó leer el texto usando la biblioteca tabula-py, cuya finalidad era leer los archivos PDF. Sin embargo, se dejó de implementar el código porque esto requería que los datos estuvieran ordenados, cosa que no ocurre con estos ficheros. Se intentó usar otro método, que se basaba en el uso de la biblioteca PDFquery, pero tras el aumento de conocimiento de dicha biblioteca, se decidió buscar otra alternativa, ya que esta biblioteca requería la modificación del PDF a un XML que incluyera todos los datos. Para la obtención de los distintos datos se requería unos valores que determinaran las coordenadas (izquierda, abajo, derecha, arriba) en las que se encontraba cada palabra en el texto. Se determinó que había formas mucho más sencillas de obtener la información de los PDF que buscando las coordenadas.

Finalmente se ha optado por crear un código capaz de cargar el fichero PDF como un documento formado por diferentes páginas, donde cada una contiene distintos datos. Para ello se ha usado la biblioteca fitz. La función **LeerDocumento** recibe el nombre de un fichero y devuelve una lista de lineas que componen cada uno de los PDF.

Una vez leídos los PDF, se pretendía obtener el valor de las variables de interés. Para ello se creó una función llamada **BuscarValor**, donde al buscar una palabra, devolvía el resultado que había tras encontrarla. Como no funcionaba en todos los casos (por ejemplo en tratamiento disponible o ensayo clínico, el número estaba delante) se han tenido que buscar distintos

métodos para ser capaces de obtener todos los valores. En algunos casos se han usado expresiones regulares que servían de patrones para buscar dicho patrón en el text, uso de posiciones...

Las expresiones regulares son un pequeño apartado de gran interés dentro del lenguaje de programación que especifica un conjunto de caracteres posibles que se desean encontrar en el texto sobre el que se está trabajando. No todas las búsquedas se pueden realizar siguiendo este método, en algunos casos es mejor realizar un código de búsqueda que usar dichas expresiones, aunque sea un proceso más laborioso, puede llegar a ser más comprensible.

Hay patrones simples muy básicos como una sola letra, donde el código encontrará todas las letras determinadas. Sin embargo, hay un conjunto de caracteres conocidos como metacarcteres que tienen un significado especial y no pueden ser buscados a lo largo del texto. Un ejemplo de ellos son los corchetes [], que permiten identificar una clase de carácter o un guión (-) que nos permitirá indicar el rango de valores entre los dos caracteres. Por ejemplo, [abc] nos indica que quiere buscar cualquiera de las tres letras que aparece ente los corchetes. El mismo resultado pero usando un guión sería [a-c], que busca tanto las letras indicadas como los posibles valores que podría tomar -. Al igual que estos ejemplos, hay muchas más excepciones.

También hay expresiones mucho más complejas donde se repiten valores de búsqueda o se usan valores con referencias propias. Esto quiere decir que existen valores como D que coincide con cualquier valor que no sea numérico o w que coincide con cualquier carácter alfanumérico.

El resultado final ha sido la obtención de una variable por cada campo de interés, para poder crear posteriormente las tablas.

Una vez obtenidas todas las variables, se debían unir las tablas. Pero el principal problema era cómo realizar la unión. Para ello se investigó un poco a cerca de todas las posibles opciones [25] [2]

Para ello inicialmente se pensó en usar el método concat para concatenar las diferentes tablas o DataFrames creados. No generaba por sí solo columnas espurias, pero si no se trabajaba correctamente con las columnas, podía darse el caso de que sí aparecieran. Si las columnas tienen nombres duplicados, se generarán columnas espurias lo que llevaría a problemas a la hora del resultado final. Permite unir en el eje de filas (axis = 0) o en el de columnas (axis = 1) uniendo simplemente los valores del eje especificado, sin realizar coincidencia entre los valores.

Sin embargo, se vio que la función merge de la biblioteca pandas era una mejor opción si se querían combinar DataFrames basándose en los valores

de las columnas. Además, suele ser más rápido y permite usar una columna como clave (similar a join en SQL). Pero cuando los datos tienen valores duplicados o vaciçíos (un ejemplo de esto es la columna de diagnóstico, que puede haber siempre algún diagnóstico que se repita dentro de los distintos casos de un chip) o si los nombres de las columnas no coinciden en ambas tablas, se pueden producir resultados espurios.

Aunque se han estudiado ambos, la solución final escogida ha sido join (también perteneciente a la biblioteca de pandas), ya que es especialmente útil cuando se quieren unir DataFrames basándose en columnas comunes. Proporciona opciones flexibles a la hora de elegir el tipo de unión (inner, left, right o outer). Además, nos garantiza que no se crearán columnas espurias al unir los PDF, porque usa los índices en lugar de las columnas (merge une por columnas, creando columnas espurias en algunos casos).Los parámetros lsuffix y rsuffix permiten agregar sufijos a los nombres de las columnas para evitar conflictos en los nombres evitando la generación de estas columnas, sin embargo, en nuestro caso no ha sido necesario usarlas. También permite unir varios DataFrames en una única operación, pero en nuestro caso lo hemos hecho por separado para ir viendo el resultado de cada una de las tablas que se creaban al unir los DataFrames de uno en uno.

Una vez obtenidas las tablas, el siguiente paso era estudiar la forma de exportarlas. Se ha probado con dos ficheros distintos, uno de ellos CSV y otro Excel. [26] 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. Esto se debe a que los datos en CSV se almacenan en texto plano y se separan por un delimitador que suele ser una coma (","), un punto (".") o un punto y coma (";"). En Excel los datos se organizan 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, cosa que CSV no permite. Esto ofrece una gran ventaja al hospital, por si en un futuro se quiere modificar o trabajar sobre los datos que aparecen en este fichero resultante.

Uno de los principales inconvenientes de usar Excel y no CSV es que este último ofrece unos ficheros mucho más simples y ligeros, ocupando mucho menos espacio y tiene gran compatibilidad. También es importante saber que los ficheros CSV pueden ser abiertos desde casi cualquier dispositivo, pero los archivos Excel están asociados con Microsoft Excel, lo que conlleva a tener la aplicación o un programa similar que permita abrir dichos ficheros.

Para crear la máquina virtual, inicialmente se tuvo en cuenta Hyper-V

Aspectos relevantes del desarrollo del proyecto

Lineas de trabajo futuras

(Este capítulo debería ser informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.)

El proyecto desarrollado en este trabajo tiene un gran potencial para ser utilizado en el campo de la farmacogenética.

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

El proyecto cumple con todos los puntos especificados, a parte de algunos puntos adicionales que no constaban al principio como por ejemplo la binarización del fármaco o de los ensayos clínicos o la generación de imágenes partiendo de los PDF. La idea principal era poder almacenar la información más relevante de toda la obtenida en los PDF del software Oncomine Reporter para aumentar el conocimiento sobre el cáncer.

Sin embargo, una de las posibles mejoras que se ha visto ha sido la posibilidad de añadir una nueva columna a la tabla final en la que se indicara el cambio de aminoácido de cada uno de lo genes mutados. En este caso si se ha tenido en cuenta el cambio de aminoácido P.(p136l) para el gen FGFR4, ya que este gen aparece bastantes veces y se ha demostrado que no es influyente en ningún caso. Pero sería de ayuda si apareciera una nueva variable al lado de los genes para saber donde se ha producido exactamente

el cambio del aminoácido para poder estudiar las consecuencias a distintos niveles.

También sería conveniente crear dos columnas nuevas. Una para determinar el número de lecturas cada fusión que se encuentre en la tablas fusiones de genes (ARN) y otra en la que se determinen el número de copias de cada gen perteneciente a la tabla de variaciones del número de copias.

Otra posible mejora, basada en la estética del resultado, sería transformar los valores [] obtenidos en algunos resultados de las tablas por 0, - o la palabra vacío. La aparición de este símbolo indica que en ese apartado no se han encontrado resultados, sin embargo, hay otras formas un poco más claras o no tan cargadas de indicarlo.

Bibliografía

[1] Anaconda, 2023.

[2] Pandas, 2023.

[3] ¿qué es excel y para qué sirve?, 2023. blog.

[4] ADEA. La automatización de procesos en el sector salud, 2022.

[5] ALTASSIAN. Git branch.

[6] Apify. Build reliable web scrapers. fast. Uso de la herramienta.

[7] APPSTATE. Pros and cons of pdfs., 2021.

[8] ASTERA. Automatización de datos: cómo transforma el panorama empresarial, 2022.

[9] ASTERA. Astera: Reporminer, 2023.

[10] Cancer.net. Genes and cancer. Definición.

[11] Clinithink. The key to saving human lives is understanding human words. Uso de la herramienta.

[12] CTakes. Apache ctakes. Uso de la herramienta.

[13] G2. Reporminer, 2023. Página de reseñas.

[14] Red Hat. ¿qué es el open source?, 2023.

[15] National Cancer Institute. Proto-oncogenes. Definición.

[16] National Cancer Institute. Tumor suppressor gene. Definición.

[17] National Cancer Institute. Understanding cancer prognosis.

[18] National Cancer Institute. What is cancer?

[19] IRONHACK. Los 10 lenguajes de programación más demandados en 2022, 2023. Ranking.

[20] Wes McKinney. *Python for Data Analisis*. O'RIELLY, 2022.

[21] Aurelio Morales. Descubre el nuevo pgadmin 4 para trabajar con postgis., 2023.

[22] Nanonets. Automate manual data entry using ai. Uso de la herramienta.

[23] OMS. Cáncer, 2022.

[24] Sonia Ordoñez. ¿qué es anatomía patológica? Página usada en la memoria de Anatomía Patológica 2021-2022 durante prácticas.

[25] Kyle Stratis. Combining data in pandas with merge(), .join(), and concat().

[26] Toggl track. Difference between csv and xls, 2023.

[27] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].

[28] XATAKA. Qué es github y qué es lo que le ofrece a los desarrolladores, 2019.

[29] ZAKHAR YUNG. 10 best data extraction tools in 2023 for your business, 2023.