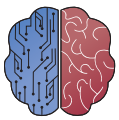
UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería de la Salud



INGENIERÍA
DE LA SALUD

TFG del Grado en Ingeniería de la Salud

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

Presentado por Lucía Vítores López
en Universidad de Burgos

19 de mayo de 2023

Tutor: Antonio Jesús Canepa Oneto

Índice general

Índice general i

Índice de figuras iii

Índice de tablas iv

Apéndice A Plan de Proyecto Software 1
A.1. Introducción. 1

Apéndice B Documentación de usuario 5
B.1. Requisitos software y hardware para ejecutar el proyecto. . . 5
B.2. Instalación / Puesta en marcha 5
B.3. Manuales y/o Demostraciones prácticas 6

Apéndice C Manual del programador. 7
C.1. Estructura de directorios 7
C.2. Compilación, instalación y ejecución del proyecto 7
C.3. Pruebas del sistema 10
C.4. Instrucciones para la modificación o mejora del proyecto. . . 10

Apéndice D Descripción de adquisición y tratamiento de datos 11
D.1. Descripción formal de los datos 13
D.2. Descripción clínica de los datos. 13

Apéndice E Manual de especificación de diseño 17
E.1. Planos 17
E.2. Diseño arquitectónico 17

Apéndice F Especificación de Requisitos **19**
F.1. Diagrama de casos de uso 19
F.2. Explicación casos de uso. 19
F.3. Prototipos de interfaz o interacción con el proyecto. 19

Apéndice G Estudio experimental **21**

Bibliografía **23**

Índice de figuras

Índice de tablas

A.1. Costes del personal 2
A.2. Costes del hardware/software 2
A.3. Costes totales 2

F.1. CU-1 Nombre del caso de uso. 20

Apéndice A

Plan de Proyecto Software

A.1. Introducción.

El proyecto se ha dividido en varias etapas, con el fin de que al unirlas, se obtenga un buen resultado final.

Planificación temporal.

El proyecto se ha organizado en distintos *Millestones*, cada uno de ellos enfocado a una parte del proyecto. En el primero de ellos se trabaja con información algo más general como la organización de la información a tratar en las tablas, los documentos necesarios para presentar al Comité de Bioética, el estudio de los datos necesarios para realizar el proyecto... En el segundo se tratan aspectos algo más específicos y se desarrolla el código necesario para el tratamiento de los datos.

Dentro de cada uno, se pueden encontrar varios *Issues* (uno por cada reunión con el tutor). En cada uno se explica el contenido general de cada reunión y las metas a conseguir antes de la siguiente. Generalmente, las reuniones son cada una o dos semanas, en función de la disponibilidad y la cantidad de trabajo.

También se realizan reuniones con el persona del HUBU encargado de dicho proyecto, con el fin de mejorar o responder a preguntas fundamentales para el desarrollo y el entendimiento de los principales objetivos que buscan. Sin embargo, estas no cuentan como *Issues*, pero parte del contenido es explicado en *Issues* de otras reuniones.

FALTA POR ACABAR!!!

CONCEPTO	COSTE(€)
Salario mensual neto	1225,7
Retención IRPF (15 porciento)	216,3
Seguridad Social (28,5 porciento)	569,16
Salario mensual bruto	2011,16
TOTAL 7 meses	14.078,12

Tabla A.1: Costes del personal

CONCEPTO	COSTE(€)	AMORTIZACIÓN
Ordenador portátil	400	
Licencia Microsoft	????	
TOTAL	14.078,12	

Tabla A.2: Costes del hardware/software

CONCEPTO	COSTE(€)
Costes del persona	14.078,12
Costes del hardware/software	??
TOTAL	??

Tabla A.3: Costes totales

Planificación económica.

Los costes se desglosarán en las siguientes categorías: costes del personal y costes de hardware/software de las herramientas.

Los costes del personal se puede ver en la tabla A1.

Para el desarrollo del proyecto no se ha adquirido ningún hardware nuevo, por lo que tan solo se incluirán en este apartado los costes del material con el que ya se contaba (asumiendo una amortización de aproximadamente 4 años) y calculando el coste de amortización correspondiente a la duración del proyecto (7 meses). No ha habido costes de software porque las herramientas usadas eran de código abierto. Se puede ver en la tabla A2.

El gasto total del proyecto se puede ver en la tabla A3.

Viabilidad legal.

El 14 de marzo de 2023 se llegó a un acuerdo de colaboración temporal para la gestión de datos entre la investigadora Patricia Saiz López con vincu-

lación laboral al Hospital Universitario de Burgos (HUBU) y los solicitantes Antonio Jesús Canepa Oneto y Lucía Vítores López como vinculantes a la Universidad de Burgos (UBU).

Dicho acuerdo se enmarca dentro de la solicitud de ampliación de gestión de datos del proyecto "Secuenciación múltiple de última generación en tumores sólidos: utilidad clínica en un Hospital de tercer nivel.ªprobado por el Comité de Ética de la Investigación con medicamentos de Áreas de Salud de Burgos y Soria, emitiendo favorable dicho dictamen. La colaboración se ha establecido hasta junio de 2023.

Con motivo de la colaboración para la realización de un Trabajo de Fin de Grado (TFG) del alumnado de la titulación Grado en Ingeniería de la Salud de la Universidad de Burgos durante el curso académico 2022-2023, la investigadora Patricia Saiz López propone el estudio de la generación de un algoritmo capaz de automatizar la extracción de datos de informes de secuenciación masiva.

El estudio se realizará cumpliendo los criterios éticos internacionales recogidos en la Declaración de Helsinki, garantizando no difundir el material a terceros y la eliminación de los materiales o sus copias en un plazo máximo de 15 días naturales.

Apéndice B

Documentación de usuario

B.1. Requisitos software y hardware para ejecutar el proyecto.

B.2. Instalación / Puesta en marcha

Para poder ejecutar correcatmente el código es necesario tener instalado Para poder trabajar correctamente en Python es necesario tenerlo instalado correctamente. Para ello podemos usar el comando py –version en nuestra consola para saber que versión estamos usando. En caso de no tenerlo instalado, es necesario instalar la última versión También es necesario saber si pip está disponible para ser usado, esto lo sabemos ejecutando py -m pip –version. Generalmente ya viene instalado, pero en el caso de usar Linux e instalarlo desde un administrados de paquetes del sistema operativo, es posible que se deba instalar por separado. Pip es un sistema de administración de paquetes que se usa para instalar y administrar los paquetes de software de Python antes de su importación y uso. Para poder instalar cualquier biblioteca, solo hay que poner python -m pip install biblioteca en la consola, siendo biblioteca el nombre de la biblioteca que nos interesa instalar.

Una vez tenemos instalados los paquetes de interés, los importamos en el fichero en el que vamos a trabajar. Las importaciones siguen una metodología:[5] import + nombre/biblioteca + as + nombre/abreviado.

- Import: importa la funcionalidad o bibliotecas en el script en el que se está trabajando.
- Nombre/biblioteca: nombre de la biblioteca que se quiere importar.

- As: .ªlias", es decir, permite tomar una palabra larga y hacer referencia a ella usando una palabra más corta.
- Nombre/abreviado: nombre abreviado estándar para hacer referencia al nombre de la biblioteca.

import pandas as pd import matplotlib.pyplot as plt from tabulate import tabulate import tabula

La biblioteca OS permite leer o escribir un archivo con open(), trabajar con rutas usando os.path o leer las dilas de todos los archivos de una línea de comandos usando fileinput.

La idea es desarrollar una máquina virtual con contraseña, de forma que al instalarla y ejecutarla en el ordenador del HUBU, sea capaz de instalar toda lo necesario para la correcta ejecución del programa y sea capaz de tratar con los datos reales de los pacientes. En este caso no se tendrían en cuenta los datos seudonimizados, ya que al hospital no le interesa tratar con identificadores falsos, sino que ya trabajan con los datos reales. Las máquinas virtuales están diseñadas para que no afecte al resto de los datos de la máquina anfitriona. .El funcionamiento de dichas máquinas se basa en mapear los dispositivos virtuales con los reales de la máquina física."

El uso de dicha máquina nos permite probar la aplicación en los ordenadores del HUBU, ya que el hospital está aislado para evitar la pérdida de información aislada. A DESARROLLAR MEJOR!!!!! También es una seguridad adicional sobre los datos, ya que, al estar aislada del resto del sistema, nos garantiza la confidencialidad de los datos. (((Maquinas virtuales: https://www.xataka.com/especiales/maquinas-virtuales-que-son-como-funcionan-y-como-utilizarlas)))

B.3. Manuales y/o Demostraciones prácticas

Explicar un poco la máquina virtual / repositorio

Apéndice C

Manual del programador.

C.1. Estructura de directorios

(Descripción de los directorios y ficheros entregados.)

C.2. Compilación, instalación y ejecución del proyecto

En el notebook de Python que se encuentra en el repositorio de GitHub con el nombre CodigoPython.ipynb.

Para poder ejecutar correctamente el código es necesario tener instalado Python. Siempre que lo tengamos instalado, nos saldrá un ejecutable python.exe que sera su intérprete. Para ver si tenemos Python instalado, podemos usar el comando py –version en nuestra consola para saber que versión estamos usando. En caso de no tenerlo instalado, es necesario instalar la última versión. [4] Para poder usarlo, nos descargamos Anaconda, que contiene aplicaciones, librerías y conceptos diseñados para el desarrollo de la ciencia de datos con Python.

Tras 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.

También es necesario saber si pip está disponible para ser usado, esto lo sabemos ejecutando py -m pip –version. Generalmente ya viene instalado, pero en el caso de usar Linux e instalarlo desde un administrados de paquetes del sistema operativo, es posible que se deba instalar por separado. [3]

Pip es un sistema de administración de paquetes que se usa para instalar y administrar los paquetes de software de Python antes de su importación y uso. Para poder instalar cualquier biblioteca, solo hay que poner python -m pip install biblioteca en la consola, siendo biblioteca el nombre de la biblioteca que nos interesa instalar.

En el caso de Anaconda, al instalarla incluye varias bibliotecas que son muy usadas como puede ser Pandas, Numpy o MatplotLib. Esto se debe a que está diseñada para ser usada en entornos de ciencias de datos y proporciona un entorno ya listo para usar con las bibliotecas más comunes. En nuestro caso habría que instalar Pandas (aunque como he comentado anteriormente, viene por defecto), Re, Os y Fitz.

Una vez tenemos instalados los paquetes de interés, los importamos en el fichero en el que vamos a trabajar.

Para poder ejecutar correctamente el fichero en Python es necesario instalar e importar las siguientes bibliotecas.

- **Import os**: [1] Este módulo proporciona una forma portátil de usar el sistema operativo. Se puede usar open() para leer o escribir un archivo, os.path() para modificar o manipular rutas, file.input() para leer todas las líneas de los archivos. Todas las funciones generan un OSError cuando las rutas o los nombres de los ficheros no son correctos o no existen.

- **Import fitz**: para esta importación es necesario tener instalada la biblioteca PyMuPDF, que actúa como enlace para la biblioteca MuPDF, la cual nos permite trabajar con ficheros PDF. El comando fitz.open() nos permite abrir el fichero en formato PDF con el que vamos a trabajar.

- **Import re**: este módulo nos permite trabajar con expresiones regulares. Las expresiones regulares son un pequeño lenguaje dentro del lenguaje de programación que especifica un conjunto de caracteres posibles que se desean hacer coincidir con 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.

- **Import pandas as pd**: [2] la comunidad de Python ha adoptado una serie de nomenclaturas convencionales para los módulos de uso más común. Entre ellos se encuentra import pandas as pd, junto con import numpy as np o import matplotlib.pyplot as plt. Pandas permite realizar muchas funciones dentro del análisis de datos, proporcionando estructuras de datos, Series y DataFrames. En nuestro caso se ha usado para la importación de los datos de dos hojas de cálculo de un fichero Excel para crear distintos DataFraes usando (pd.read-excel(io = ruta/fichero/Excel.xlsx"). Hay otros parámetros que se pueden añadir a la función como *sheet-name*: que permite escoger las hojas que se quieren usar y que en nuestro dejamos lo que viene por defecto (cero) porque nos permite obtener todas, *usecols*: que nos permite escoger con qué columnas queremos trabajar (como por ejemplo .ᴬ:E", que en este caso indica que queremos trabajar con las columnas que van desde la A hasta la E ambas incluidas) o *nrows* para determinar el número de columnas con el que vamos a trabajar, aunque en este caso también dejamos el valor que viene por defecto que es None.

Una vez que hemos ejecutado la celda donde se encuentran las importaciones sin ningún problema, podemos ejecutar el código que viene a continuación.

La manera más sencilla de desarrollar dicha máquina virtual ((((Desarrollo máquina virtual Windows: https://support.microsoft.com/es-es/windows/habilitar-la-virtualizaciDe esta forma podemos hacer que nuestro ordenador simule un sistema operativo al que de verdad es. En algunos dispositivos, generalmente aquellos que tratan con Windows 10 y 11, ya suelen tener habilitado la virtualización, de forma que ya no sería necesario realizar dicho proceso. En caso contrario, habría que acceder a la UEFI o BIOS y ejecutar: Inicio > Configuración > Sistema > Recuperación > Inicio avanzado y reiniciar el dispositivo. Al volver a encenderlo, aparecerá una pestaña que permite elegir distintas opciones: Solucionar problemas > Opciones avanzadas > Configuración de la UEFI > Reiniciar. Al volver a iniciarlo, ya estará el ordenador en BIOS.

En este punto se podrán activar o desactivar distintas opciones, por lo que es importante determinar bien el fabricante del PC, ya que al haber distintos fabricantes, hay distintas instrucciones y opciones.

Finalmente se activa la plataforma de máquina virtual: Inicio > Funciones de Windows > Activar o desactivar las funciones de Windows en la lista de resultados. Características de Windows que se acaba de abrir, se busca Plataforma de máquina virtual y se selecciona. Al volver a reiniciar el ordenador, el proceso ya debería haber terminado.

C.3. Pruebas del sistema

Esta sección puede ser opcional.

Puede tratarse de validación de la interfaz por parte de los usuarios, mediante escuestas o similar o validación del funcionamiento mediante pruebas unitarias.

C.4. Instrucciones para la modificación o mejora del proyecto.

Instrucciones y consejos para que el trabajo pueda ser mejorado en futuras ediciones.

BILIOGRAFÍA DATO: *EXPECIFICADA EN EL TEXTO (Desarrollo máquina virtual Windows: https://support.microsoft.com/es-es/windows/habilitar-la-virtualizaci

Apéndice D

Descripción de adquisición y tratamiento de datos

El HUBU cederá distintos informes en formato .pdf generados por el Software Oncomine Reporter. Dicha cesión será realizada por vía correo electrónico institucional durante el primer semestre de 2023, cumpliendo los requisitos.

Al trabajar con información sensible, se ha considerado la idea de anonimizar los datos (aunque los colaboradores de la Universidad de Burgos no tendrán acceso para asociarlo con otros datos personales de los pacientes.) El estudio se realiza siguiendo los criterios éticos internacionales recogidos en la Declaración de Helsinki.

La anonimización es un proceso en el que es imposible la vinculación de datos con la persona real a la que identifican. Un tipo es la seudonimización, cuyo objetivo se basa en limitar la trazabilidad entre el conjunto de datos tratados y la persona física a la que corresponden dichos datos, y al ser un proceso reversible, es posible identificar a la persona real.Es una de las técnicas de enmascaramiento que garantizan mayor seguridad a la hora de tratar los datos y una de las más usadas en el ámbito médico.

Sin embargo, hay otras técnicas de anonimización a parte de la desarrollada anteriormente disponibles en [7]

1. Enmascaramiento de datos: permite ocultar ciertos datos usando caracteres aleatorios en su lugar. Se sustituye la palabra por una clave.

2. Intercambio de datos: se basa en la variación del orden de los elementos de un conjunto ordenado, es decir, reordena valores de forma que sigan

estando presentes en el conjunto, pero no corresponden con el registro de datos originales.

3. Datos sintéticos: no son una técnica de anonimización real, más bien se usan para tratar con datos personales de forma que no interfiera con la ley. Un algoritmo crea un conjunto de datos sin ningún tipo de relación con los datos originales.

4. Perturbación de datos: agrega ruido a las bases de datos originales aportando confidencialidad a los registros. Puede sumar un valor a todos los valores numéricos del conjunto de datos que van a usarse para no trabajar directamente con los reales, pero hay que tener cuidado con las bases de datos iniciales porque si son demasiado grandes o demasiado pequeñas, es posible que los datos no se reconozcan bien y no se anonimicen.

5. Generalización: se basa en la eliminación de ciertos identificadores.

Tras el estudio y la comparación de los distintos tipos, se ha llegado a la conclusión de que la mejor para este proyecto es la seudonimización.

Podemos identificar cinco tipos distintos de seudonimización [8]:

1. Cifrado con clave secreta: se usa una clave capaz de generar un conjunto de datos que almacena dichos datos personales pero cifrados. En el momento en el que se conoce la clave de desencriptación, es posible revertir el proceso.

2. Función hash: se basa en el uso de un algoritmo, donde partiendo de uno o varios inputs, genera un output alfanumérico que resume la información obtenida. Solo es posible recuperar los valores originales si se conocen los valores de entrada iniciales que forman los inputs.

3. Cifrado determinista/función has con clave de borrado: se genera un número aleatorio por cada uno de los atributos/valores originales que se quieren sustituir, borrando la tabla que los relaciona, de modo que es irreversible.

4. Función clave almacenada: se asocia una clave secreta a cada valor original, de forma que, conociendo las claves, es posible identificar al sujeto original.

5. Descomposición en tokens: se basa en reemplazar los números de interés por otros valores usando tres métodos: mecanismos de cifrado unidireccional, números de secuencias mediante funciones de índice o números generados aleatoriamente.

La idea principal era anonimizar los datos, por lo que se buscó distinta información sobre el tema para elegir el mejor tipo y saber como llevarlo a cabo. Sin embargo, al pasar el código al ordenador del hospital, no va a ser necesario anonimizar los datos. La idea de anonimizarlo era para la hora de trabajar con datos reales, pero como los ficheros usados para el desarrollo del proyecto no contenían información real y el código se ejecutará directamente en el ordenador con los ficheros reales, no se ha visto necesario realizarlo.

D.1. Descripción formal de los datos

(Tablas, imágenes, señales, secuencias de ADN...)

D.2. Descripción clínica de los datos.

(Descripción y explicaciones clinicas del significado o interpretación de los datos.)

Los datos obtenidos por el Software Oncomire Reporter se obtienen en carpetas comprimidas y al abrirlas, habrá ocho PDF por cada una de ellas. El nombre de los PDF está formado por el número de paciente y el número de chip. Un ejemplo de esto sería Sample-1-v100 que es uno de los ejemplos que se va a usar para desarrollar el código. El uno indica el número de paciente (al haber ocho pacientes por carpeta, los números serán del uno al ocho) y el cien sería el número de chip. Todas las carpetas siguen el mismo formato, numeración del uno al ocho para el número de paciente y distintos número de chip.

Sin embargo, al abrir los PDF podemos ver que no todos siguen un formato estándar y esto se debe a las variaciones de cada persona. Vemos que hay tres tablas distintas, una de variantes de secuencia de ADN (que indica los cambios permanentes de la secuencia de ADN que forma un gen), fusiones de genes ARN (cambios en la secuencia de ARN) y variaciones en el número de copias (el número de copias de un segmento específico de ADN varía entre distintos genomas individuales).En función de los resultados obtenidos en estas tablas, se determinará el diagnóstico.

Sin embargo, todos ellos tienen información común como son:

- Número de historia clínica. (NHC)
- Fecha de informe.
- Número de biopsia.
- Biopsia sólida.
- Mutaciones detectadas (en las distintas tablas).
- Diagnóstico.
- Porcentaje de frecuencia alélica por cada mutación.
- Fármaco aprobado.
- Ensayos clínicos.

El número de historia clínica es un identificador único para cada persona y es asignado al paciente cuando se elabora su historia clínica manteniéndose de por vida. Este es uno de los valores de carácter sensible, por lo que habría se anonimizarlo.

La fecha de informe indica cuándo se ha llevado a cabo la extracción de la información de la muestra, es decir, cuando se ha creado el informe.

El número de biopsia es específico para cada biopsia tomada. En algunos casos es posible que el número venga acompañado de -A1 como en alguno de los ejemplos y esto se debe a que de una biopsia se pueden obtener distintos cortes y a cada uno se le asigna un sufijo distinto. De forma que varias muestras pueden tener el mismo nombre, pero distinto sufijo.

La biopsia sólida depende del valor de la tercera posición del número de biopsia, ya que este puede ser C (citología), P (punción) o B (biopsia). Nos interesa que sea B, ya que sino no entraría en nuestro algoritmo.

Las mutaciones detectadas vienen en las distintas tablas. En la tabla de variaciones de secuencias de ADN, nos interesan aquellas mutaciones que sean patogénicas, conflictivas o incluso vacías (una excepción en este caso sería el gen FGFR4 p.(P136L), que se ha determinado que no supone ningún cambio ni complicación para los distintos tipos de diagnósticos), pero en ninguno de los casos las benignas, ya que estas no supondrían ningún peligro. En el caso de fusiones de genes ARN y variaciones en el número de

copias, cogemos todas. Se ha creado un diccionario para poder relacionar las mutaciones con un número específico y enriquecer los resultados, haciéndolos más simples. De esta forma las tablas contienen dos columnas, una con el nombre del gen y otra con el número que le corresponde.

El diagnóstico viene determinado ya por el software. Sin embargo, nos interesa asignar a cada diagnóstico un número específico con el fin de obtener un estudio más organizado. Para ello, se ha realizado una búsqueda de todos los diagnósticos que es capaz de determinar y junto con la investigadora, se han determinado cuáles son los de mayor interés. Al ser tantos, el hospital no usa todos los tipos y es posible que se generalice (un ejemplo es el cáncer de colon o el cáncer rectal que se diagnosticarían con el nombre de cáncer colorrectal). Para asignar a cada diagnóstico un valor numérico, se ha creado un diccionario que almacene diagnóstico-valor para poder usarlo en el código y devolver dos columnas, una con la clave (diagnóstico) y otra con el valor (número del diagnóstico).

El porcentaje de frecuencia alélica es único de cada mutación en cada paciente y sirve para indicar el número de veces que aparece el alelo, dividiendo el número entre el número total de copias del gen. En el caso de tratarse de fusiones de ARN o variaciones en el número de copias no hay porcentaje de frecuencia alélica, en su lugar aparecen número de lecturas y número de copias respectivamente. Debemos tener en cuenta que en algunos casos no aparecen genes, sino fusiones. Para cada gen podemos encontrar fusiones, ya que un gen tiene varios posibles çompañeros de fusión". También hay que tener en cuenta la localización de la fusión. Generalmente suelen tener únicamente un nexo de unión, pero hay casos (como por ejemplo el primer fichero) donde están fusionados por distintos localizadores, por eso es posible que aparezcan dos veces. En estos casos, hay que indicar el ID de la variante, para poder identificar la variante. Estas fusione no vienen indicadas en la tabla de genes porque no se consideran genes, por lo que para el su obtención en el código habría que tener en cuenta esta excepción.

El fármaco aprobado indica los distintos tipos de fármacos que hay para cada diagnóstico. Podemos conocer esta información en el apartado de tratamientos relevantes en este tipo de cáncer. También ha sido interesante binarizar los resultados, siendo 0 si no hay ningún fármaco o 1 si hay uno o más.

Los ensayos clínicos aportan información sobre el número de ensayos que hay y en algunos casos, la fase en la que se encuentra cada uno. En este caso también se binarizan los resultados, siguiendo el mismo código que en el caso anterior.

Vemos que a parte de esta información, también podemos encontrar otros datos como el exón, el locus o el transcripto, pero estos valores no nos interesan para nuestro algoritmo, por eso no son tenidos en cuenta.

La idea final era ser capaz de obtener dos tablas: una con todas las variables de interés de todos los genes y otra teniendo en cuenta solo las mutaciones patogénicas.

Apéndice E

Manual de especificación de diseño

Si es necesario.

Planos (Si procede) Diseño arquitectonico (Si procede) Diagrama de clases, diagrama de despliegue

E.1. Planos

Si procede

E.2. Diseño arquitectónico

Si procede.

Diagramas de clases, diagramas de despliegue ...

Apéndice F

Especificación de Requisitos

Si procede.

F.1. Diagrama de casos de uso

F.2. Explicación casos de uso.

Se puede describir mediante el uso de tablas o mediante lenguaje natural.

Una muestra de cómo podría ser una tabla de casos de uso:

F.3. Prototipos de interfaz o interacción con el proyecto.

CU-1	**Ejemplo de caso de uso**
Versión	1.0
Autor	Alumno
Requisitos asociados	RF-xx, RF-xx
Descripción	La descripción del CU
Precondición	Precondiciones (podría haber más de una)
Acciones	1. Pasos del CU 2. Pasos del CU (añadir tantos como sean necesarios)
Postcondición	Postcondiciones (podría haber más de una)
Excepciones	Excepciones
Importancia	Alta o Media o Baja...

Tabla F.1: CU-1 Nombre del caso de uso.

Apéndice G

Estudio experimental

La definición más precisa para estudio experimental sería la siguiente.

> *Tipo de estudio en el que el investigador manipula deliberadamente algún factor o circunstancia, y así puede comprobar qué efecto produce esta modificación en otro fenómeno.* [6]

Como los datos con los que se ha realizado el proyecto no han sido manipulados o modificados de ninguna forma, no se llevará a cabo el desarrollo de este anexo. Tan solo se ha mejorado el resultado final con el objetivo de facilitar el estudio y manejo de los resultados, no se ha modificado ningún factor ni circunstancia ni se ha llevado a cabo un seguimiento de los pacientes.

Bibliografía

[1] os — miscellaneous operating system interfaces.

[2] pandas.

[3] Installing pip/setuptools/wheel with linux package managers, 2021.

[4] Properly installing python, 2022.

[5] Data independent. Import pandas as pd – bring pandas to python, 2022.

[6] Ignacio (dir.) Palacios Martínez. *Diccionario electrónico de enseñanza y aprendizaje de lenguas.* 2019.

[7] Pangeanic. 6 técnicas de anonimización de datos personales que debe conocer, 2023.

[8] Icaria technology. ¿qué es la seudonimización de los datos?, 2022.