



Escuela Técnica Superior de
Ingeniería Informática

TRABAJO FIN DE GRADO

Aplicación web para el análisis de datos del Atlas del Genoma del Cáncer

Realizado por
Rafael Estrada Ornedo

Para la obtención del título de
Grado en Ingeniería Informática - Ingeniería del Software

Dirigido por
María del Mar Martínez Ballesteros
Manuel Jesús Jiménez Navarro

Realizado en el departamento de
Lenguajes y Sistemas Informáticos

3ª Convocatoria, curso 2023/24

Agradecimientos

La realización de este TFG ha sido un viaje en el que no he estado solo, y quiero expresar mi gratitud a todas las personas que han sido parte fundamental de este camino.

En primer lugar, mi familia ha sido el pilar que ha sostenido cada paso de este proceso. A mis padres, les agradezco no solo por su incansable esfuerzo para asegurar un futuro laboral digno para mí, sino también por haberme inculcado valores que han forjado mi carácter. A mi hermana melliza, mi compañera de vida, le agradezco por ser una constante fuente de apoyo, compartiendo risas y desafíos a lo largo de nuestra travesía juntos.

Mis amigos han sido mi ancla en medio de las aguas turbulentas de los estudios. Agradezco sus risas contagiosas, su compañía en los momentos difíciles y la distracción valiosa que han aportado a mi vida.

Mi profunda gratitud se dirige a mi tutora, María del Mar Martínez Ballesteros. Su paciencia infinita y dedicación incansable fueron fundamentales para el éxito de este proyecto. Siempre dispuesta a responder mis correos y consultas, su orientación y motivación fueron el impulso final que necesitaba para concluir este trabajo. También quiero agradecer a Manuel Jesús Jiménez Navarro, que como cotutor del TFG, trajo nuevas ideas y opiniones que contribuyeron significativamente a mejorar el proyecto. Las tutorías conjuntas con ambos no solo fueron inmensamente útiles, sino también increíblemente divertidas.

En resumen, agradezco a cada persona que ha creído en mí y ha sido parte de este logro. De nuevo, muchas gracias por su apoyo inquebrantable.

Resumen

El Programa Atlas del Genoma del Cáncer (TCGA) representa un ambicioso proyecto de investigación destinado a desentrañar las complejas alteraciones genómicas presentes en diversos tipos de cáncer, generando así una rica fuente de datos de gran valor para investigadores, biólogos y profesionales de la salud.

Este proyecto tiene como objetivo central el desarrollo de una aplicación web que pueda extraer los datos generados por el TCGA, llevar a cabo diversos análisis y presentar los resultados de manera intuitiva a través de una interfaz gráfica interactiva. La premisa es permitir a los usuarios obtener información significativa sin requerir conocimientos informáticos especializados.

La aplicación deberá adquirir las secuencias de expresión genética tanto de ARN como de miARN procedentes de diversas muestras. Posteriormente, se llevarán a cabo análisis de expresión diferencial, análisis de enriquecimiento y análisis de supervivencia. Estos análisis se centrarán en la comparación de las muestras extraídas de tejido sano con aquellas provenientes de tejido cancerígeno.

La implementación de este objetivo involucra la integración de diversas tecnologías, destacando GDCRNATools, un paquete de R/Bioconductor diseñado para análisis y acceso a datos desde el portal Genomic Data Common (GDC), que alberga, entre otros, los proyectos de TCGA. En el desarrollo de la aplicación web, se recurre a Django como framework, respaldado por Celery para gestionar tareas asíncronas, incluyendo análisis complejos. La presentación de los resultados de los análisis se llevará a cabo mediante tablas y gráficos, complementados con la disponibilidad de una API que permitirá realizar consultas para acceder a la información correspondiente. El desarrollo concluye con la obtención de la aplicación llamada “GenCancer Analyzer”.

A lo largo del documento se describen minuciosamente los pasos seguidos en la construcción del software. Esto abarca desde el análisis de requisitos y el diseño, seguido por la planificación y el seguimiento del desarrollo, hasta culminar con una fase de cierre que comprende la explicación detallada del producto y la presentación de las conclusiones obtenidas a lo largo del proyecto.

Palabras clave: TCGA, Cáncer, Expresión génica, Aplicación web, Análisis de expresión diferencial, Análisis de enriquecimiento, Análisis de supervivencia.

Abstract

The Cancer Genome Atlas Program (TCGA) represents an ambitious research project aimed at unraveling the complex genomic alterations present in various types of cancer, thereby generating a rich source of valuable data for researchers, biologists, and healthcare professionals.

The central objective of this project is the development of a web application capable of extracting data generated by TCGA, conducting diverse analyses, and presenting results intuitively through an interactive graphical interface. The premise is to enable users to obtain meaningful information without requiring specialized computer knowledge.

The application is designed to acquire gene expression sequences, both RNA and miRNA, from various samples. Subsequently, it performs analyses, including differential expression, enrichment, and survival analysis, focusing on comparing samples extracted from healthy tissue with those from cancerous tissue.

The implementation of this objective involves the integration of various technologies, with a notable mention of GDCRNATools, an R/Bioconductor package designed for analysis and data access from the Genomic Data Common (GDC) portal, which hosts TCGA projects. In the development of the web application, Django is used as the framework, supported by Celery for handling asynchronous tasks, including complex analyses. Additionally, an API is implemented to facilitate access to results through different queries. The results of the analyses will be displayed through tables and graphs, along with the provision of an API featuring queries to access the data. The development concludes with the obtaining of the application called 'GenCancer Analyzer.'

Throughout the document, the steps taken in the software construction are meticulously described. This encompasses requirements analysis and design, followed by planning and development monitoring, culminating in a closing phase that includes a detailed explanation of the product and the presentation of conclusions drawn throughout the project.

Keywords: TCGA, Cancer, Gene expression, Web Application, Differential Expression Analysis, Enrichment Analysis, Survival Analysis.

Índice general

I	INTRODUCCIÓN	1
1.	Introducción	2
1.1.	Contexto	2
1.2.	Motivación	2
1.3.	Definición de objetivos	3
1.4.	Estructura del trabajo	3
II	ESTUDIO PREVIO	5
2.	Conceptos teóricos	6
2.1.	Datos RNA	6
2.1.1.	Definiciones	6
2.1.2.	Estructura de datos	7
2.2.	GDC Data Portal	9
2.3.	Técnicas	10
2.3.1.	Recuento de datos	10
2.3.2.	Normalización	11
2.3.3.	Análisis de expresión diferencial	11
2.3.4.	Análisis de enriquecimiento	12
2.3.5.	Análisis de supervivencia univariado	13
III	SISTEMA DESARROLLADO	15
3.	Objetivos	16
4.	Análisis	17
4.1.	Requisitos	17
4.1.1.	Requisitos de información	17
4.1.2.	Requisitos funcionales	19
4.1.3.	Reglas de negocio	21
4.1.4.	Requisitos no funcionales	21
4.1.5.	Matriz de trazabilidad	23
4.2.	Casos de uso	24
5.	Diseño	26
5.1.	Diagrama de clases	26
5.2.	Arquitectura del sistema	30
5.3.	Técnicas de diseño aplicadas	34
5.4.	Tecnologías	34

5.5. Diseño API	36
IV PLANIFICACIÓN	39
6. Riesgos	40
6.1. Análisis probabilidad e impacto	40
6.2. Registro de riesgos	40
7. Planificación	42
7.1. Metodología de desarrollo	42
7.2. Listado de hitos	43
7.3. Estructura de descomposición del trabajo	43
7.4. Diccionario de la EDT	44
7.5. Cronograma	57
8. Costes	59
8.1. Coste del personal	59
8.2. Coste material	59
8.3. Coste operacional	60
8.4. Presupuesto final	60
V SEGUIMIENTO	61
9. Introducción del seguimiento	62
10.Sprint 1	63
10.1. Incidencias del desarrollo	63
10.2. Desviaciones	63
10.3. Riesgos	64
10.4. Retrospectiva	64
10.5. Replanificación	65
11.Sprint 2	66
11.1. Incidencias del desarrollo	66
11.2. Desviaciones	66
11.3. Riesgos	67
11.4. Retrospectiva	68
11.5. Replanificación	69
12.Sprint 3	70
12.1. Incidencias del desarrollo	70
12.2. Desviaciones	70
12.3. Riesgos	71
12.4. Retrospectiva	72
12.5. Replanificación	72
13.Sprint 4	74

13.1. Incidencias del desarrollo	74
13.2. Desviaciones	74
13.2.1. Fecha de fin	75
13.3. Riesgos	75
13.4. Retrospectiva	76
13.5. Replanificación	76
 VI CIERRE	 77
14. Aplicación final	78
14.1. Manual de instalación	78
14.2. Manual de usuario	84
14.3. Pruebas	92
15. Cumplimiento de objetivos	94
16. Coste final	95
17. Esfuerzo empleado	97
18. Conclusiones	100
18.1. Lecciones aprendidas	100
18.2. Trabajo futuro	101
 VII APÉNDICE	 102
19. Glosario	103
20. Bibliografía	104

Índice de figuras

2.1. Archivo RNA-seq Star	7
2.2. Archivo miRNA-seq	8
2.3. Inicio de GDC Portal	9
2.4. Estructura RNA counts	10
2.5. Ejemplo volcano plot [8]	12
2.6. Ejemplo gráfica de supervivencia [24]	14
4.1. Diagrama casos de uso	25
5.1. Diagrama de clases	26
5.2. Diagrama de arquitectura	30
5.3. Diagrama de arquitectura Celery [1]	31
5.4. Diagrama de flujo - Análisis [1]	33
7.1. Diagrama EDT general	44
7.2. Diagrama EDT Sprint 1	45
7.3. Diagrama EDT Sprint 2	45
7.4. Diagrama EDT Sprint 3	45
7.5. Diagrama EDT Sprint 4	46
7.6. Diagrama Gantt - parte 1	57
7.7. Diagrama Gantt - parte 2	57
7.8. Diagrama Gantt - parte 3	58
7.9. Diagrama Gantt - parte 4	58
7.10. Diagrama Gantt - parte 5	58
14.1. Configuración pgAdmin - 1	79
14.2. Configuración pgAdmin - 2	80
14.3. Configuración pgAdmin - 3	80
14.4. Configuración pgAdmin - 4	81
14.5. Configuración pgAdmin - 5	81
14.6. Configuración pgAdmin - 6	82
14.7. Redis respuesta	83
14.8. App - Vista análisis	84
14.9. App - Vista proyectos analizados	85
14.10App - Vista información API	85
14.11App - Vista Swagger	86
14.12App - Vista información web	86
14.13App - Vista resultados - Información	87
14.14App - Vista resultados - Datatables	88
14.15App - Vista resultados - Metadata resumen	88
14.16App - Vista resultados - Volcano plot	89
14.17App - Vista resultados - Bar plot	90
14.18App - Vista resultados - Correlation plot	91
14.19App - Vista resultados - Enrichment Bar plot	91

14.20App - Vista resultados - Bubble plot	92
14.21App - Vista resultados - Survival plot	93
16.1. Desviación de costes	95
16.2. Evolución coste acumulado	96

Índice de cuadros

4.1. Requisitos de información	17
4.2. Requisitos funcionales	19
4.3. Reglas de negocio	21
4.4. Requisitos no funcionales	22
4.5. Matriz de trazabilidad de requisitos / objetivos	23
6.1. Riesgos - Análisis de probabilidad	40
6.2. Riesgos - Análisis de impacto	40
6.3. Registro de riesgos	41
7.1. Hitos	43
7.2. Diccionario de la EDT - Planificación	47
7.3. Diccionario de la EDT - Sprint 1	48
7.4. Diccionario de la EDT - Sprint 2	51
7.5. Diccionario de la EDT - Sprint 3	53
7.6. Diccionario de la EDT - Sprint 4	55
7.7. Diccionario de la EDT - Cierre	56
8.1. Coste del personal	59
8.2. Coste material	60
8.3. Coste operacional	60
8.4. Presupuesto final	60
10.1. Incidencias - Sprint 1	63
11.1. Incidencias - Sprint 2	66
11.2. Replanificación Sprint 3	69
12.1. Incidencias - Sprint 3	70
12.2. Replanificación Sprint 4	73
13.1. Incidencias - Sprint 4	74
17.1. Horas de trabajo reales	97

Índice de extractos de código

14.1. Comandos entorno virtual	78
14.2. Comando instalar requirements	78
14.3. Comandos descarga GDCRNATools	79
14.4. Archivo .env	82
14.5. Comandos manage.py	83
14.6. Celery despliegue - 1	83
14.7. Celery despliegue - 2	83
14.8. Comandos pruebas	92

Parte I

INTRODUCCIÓN

1. Introducción

En este capítulo se introduce el tema del trabajo describiendo el contexto, la motivación y los principales objetivos que lo definen. También se detalla la estructura del presente documento.

1.1. Contexto

El proyecto “The Cancer Genome Atlas” (TCGA) [29], que forma parte del título del proyecto, se trata de un proyecto de investigación que tenía como objetivo recopilar y analizar datos genómicos, transcriptómicos y epigenómicos de miles de muestras de cáncer. Esto incluía información sobre la secuencia del ADN, la expresión génica, las modificaciones epigenéticas y otros aspectos relacionados con la genética del cáncer.

La información generada por el TCGA se encuentra de manera pública [26], a través del portal de datos “Genomic Data Commons” (GDC) [14], constituyendo una fuente invaluable para científicos e investigadores. Esto posibilita la realización de diversos análisis y la obtención de resultados significativos.

1.2. Motivación

Entre distintas propuestas para la realización del trabajo de fin de grado, el estudiante se decantó por esta idea. Desarrollar una aplicación útil para investigadores suponía un reto, debido al entendimiento del dominio del problema y el aprendizaje de nuevas tecnologías. Sin embargo, aprender es la principal motivación del estudiante por lo que se trataba del trabajo perfecto.

“En 2020, se calcula que en los Estados Unidos se diagnosticarán 1 806 590 casos nuevos de cáncer y que 606 520 personas morirán por la enfermedad”, “La tasa de casos nuevos de cáncer (incidencia de cáncer) es de 442,4 por 100 000 hombres y mujeres por año (según los casos de 2013 a 2017)” [15]. Estos son algunos datos estadísticos de Estados Unidos del gran impacto que tiene esta enfermedad sobre la sociedad. A lo largo de los años surgen numerosos proyectos de investigación respectivos a este campo y el software forma una parte fundamental en estos proyectos.

Al proporcionar una herramienta que simplifica el proceso de análisis, se busca potenciar la investigación en el campo del cáncer, permitiendo a los científicos y profesionales de la salud obtener insights más rápidos y precisos en sus investigaciones capaz de mostrar correlaciones y patrones entre los datos de entrada.

1.3. Definición de objetivos

El objetivo principal de este Trabajo de Fin de Grado es concebir, diseñar y desarrollar una aplicación dedicada a la obtención y análisis de datos del proyecto TCGA (The Cancer Genome Atlas). Se busca no solo realizar análisis exhaustivos de estos datos, sino también proporcionar una interfaz intuitiva y eficiente que permita a los usuarios interactuar con los resultados de manera significativa.

En la fase de análisis, la aplicación se centrará en diferentes aspectos, incluyendo la expresión génica, análisis diferencial, y análisis de supervivencia. La aplicación no solo deberá ejecutar estos análisis de manera eficiente, sino también presentar los resultados de manera visual y comprensible a través de diversos gráficos y representaciones gráficas. Se priorizará la accesibilidad y la usabilidad para que incluso aquellos usuarios sin un profundo conocimiento técnico puedan interpretar los hallazgos.

Además, se considera crucial implementar una funcionalidad que permita a los usuarios descargar los resultados de los análisis realizados. Esto no solo fomentará la transparencia y la reproducibilidad de los resultados, sino que también empoderará a los usuarios para realizar análisis adicionales por fuera de la aplicación.

Dado que este proyecto se enmarca en el ámbito del departamento de Lenguajes y Sistemas Informáticos, se prestará especial atención a la planificación y al seguimiento del desarrollo. Se aplicarán las metodologías y mejores prácticas aprendidas a lo largo del grado para asegurar una ejecución eficaz y una entrega exitosa.

Es importante destacar que estos objetivos podrían evolucionar durante la fase de planificación del proyecto, permitiendo ajustes en su alcance o enfoque según las necesidades específicas y las discusiones con el tutor responsable.

1.4. Estructura del trabajo

A continuación se describe cómo está estructurado el proyecto describiendo cada una de sus partes.

Parte I: Introducción. En la primera parte del presente documento se describe el tema del trabajo dando contexto, motivación y los principales objetivos de este.

Parte II: Estudio Previo. En esta parte se da una base teórica sobre conceptos de gran importancia en el proyecto y que serán necesarios para definir los requisitos de la aplicación.

Parte III: Desarrollo del Sistema. Aquí se delinea la aplicación a desarrollar, proporcionando detalles sobre los objetivos (Capítulo 3), análisis (Capítulo 4) y diseño (Capítulo 5) del sistema.

Parte IV: Planificación. En esta sección, se aborda la planificación integral del proyecto, describiendo las tareas que constituyen su ejecución completa. En el Capítulo 6 se describen los riesgos identificados. El Capítulo 7 es donde se define la metodología,

se establecen hitos, y se elabora la EDT y el cronograma. Por último, en el Capítulo 8 se detallan los costes de los recursos.

Parte V: Seguimiento: En esta parte se describe el desarrollo del proyecto a lo largo de los sprints. Se da información acerca de las desviaciones, incidencias y riesgos (Capítulos 9-13).

Parte VI: Cierre. Es la parte donde se exponen las conclusiones del trabajo. El Capítulo 14 describe mediante un manual de usuario la aplicación desarrollada. Se describen cómo se han completado los objetivos (Capítulo 3), la variación final de los costes (Capítulo 16) así como las horas de esfuerzo reales del proyecto (Capítulo 17). Finaliza con la conclusión del trabajo en el Capítulo 18.

Parte VII: Apéndice. Contiene un glosario (Capítulo 19) con términos recurrentes a lo largo del proyecto y la bibliografía consultada.

Parte II

ESTUDIO PREVIO

2. Conceptos teóricos

Para llevar a cabo el proyecto, es fundamental adquirir un conocimiento básico sobre el dominio del problema: datos genómicos y las técnicas empleadas en su estudio. Esta comprensión inicial resulta crucial para la identificación y análisis de los requisitos necesarios en el desarrollo de la aplicación. Recomendamos consultar el manual de “Análisis estadístico de datos ómicos” de Manuel Ayala [2] donde se describen muchos de los puntos que veremos a continuación.

2.1. Datos RNA

En esta sección se definirán los tipos de datos que tienen relevancia en el estudio del cáncer y un breve contexto teórico.

2.1.1. Definiciones

A continuación se presentan definiciones fundamentales de conceptos teóricos.

- **RNA:** El RNA [17] es una molécula esencial en la síntesis de proteínas y en la regulación de la expresión génica. En el contexto del cáncer, la expresión de ciertos genes puede estar desregulada, lo que lleva a una producción anormal de proteínas que contribuyen al crecimiento y la propagación de células cancerosas. La secuenciación de RNA (RNA-Seq) se utiliza para analizar el transcriptoma, que es el conjunto de todos los transcritos (ARN mensajero o mRNA) presentes en una muestra. Esto permite identificar genes que están sobreexpresados o subexpresados en el cáncer en comparación con las células normales. El estudio del RNA también ha llevado al desarrollo de terapias dirigidas, como la terapia de interferencia de RNA (RNAi), que apunta a genes específicos relacionados con el cáncer para reducir su expresión.
- **miRNA:** Los microARN (miARN o miRNA, por sus siglas en inglés MicroRNA) son pequeñas moléculas de ácido ribonucleico (ARN) no codificante que desempeñan un papel crucial en la regulación de la expresión génica en organismos eucariotas.

“La expresión de los miARN se encuentra alterada en el proceso de carcinogénesis y puede ser inhibida o estimulada dependiendo del papel de cada miARN en particular. Por lo tanto, se pueden diseñar oncomiARN para inhibir a los miARN involucrados en el silenciamiento de genes supresores tumorales y miARN para silenciar oncogenes y de algún modo tratar el cáncer. Asimismo, algunos miARN podrían predecir el desarrollo de algunos tipos de cáncer y, en un futuro cercano, esta tecnología podría explotarse como herramienta para superar los retos diagnósticos [27].”

De esta cita podemos concluir que los miRNA son una importante fuente de estudio para detectar el cáncer y su tratamiento.

2.1.2. Estructura de datos

Los datos de entrada que veremos a lo largo del proyecto son productos a su vez de una alineación de la secuencia de ARN o miRNA. Esto implica el proceso de mapear las secuencias de datos de secuenciación obtenidas a partir de una muestra a un formato que puede ser representado en un archivo CSV u otro formato tabular.

Existen diversos métodos de alineamiento, sin embargo, el propósito de esta sección no es estudiar cómo se realizan esos métodos sino entender los resultados. Los ejemplos de datos mostrados a continuación fueron obtenidos desde GDC Portal.

# gene-model: GENCODE v36									
gene_id	gene_name	gene_type	unstranded			stranded_first	stranded_second	tpm_unstranded	fpkm_unstranded fpkm_uq_unstranded
N_unmapped			1851688	1851688	1851688				
N_multimapping			4760971	4760971	4760971				
N_noFeature			4081061	27335289		27451805			
N_ambiguous			4932941	1346739	1335365				
ENSG00000000003.15	TSPAN6	protein_coding	5109	2591	2518	68.4976	27.3173	26.9289	
ENSG00000000005.6	TNMD	protein_coding	0	0	0	0.0000	0.0000	0.0000	
ENSG000000000419.13	DPM1	protein_coding	1637	831	806	82.4810	32.8939	32.4263	
ENSG000000000457.14	SCYL3	protein_coding	2256	1854	1756	19.9331	7.9494	7.8364	
ENSG000000000460.17	C1orf112	protein_coding	914	1150	1195	9.3107	3.7132	3.6604	
ENSG000000000938.13	FGR	protein_coding	375	193	182	6.7433	2.6893	2.6510	
ENSG000000000971.16	CFH	protein_coding	3334	1675	1659	25.4083	10.1330	9.9889	
ENSG000000001036.14	FUCA2	protein_coding	2149	1597	1528	46.3118	18.4694	18.2069	
ENSG000000001084.13	GCLC	protein_coding	2812	1574	1667	19.8436	7.9138	7.8013	
ENSG000000001167.14	NFYA	protein_coding	1755	965	950	28.0060	11.1690	11.0102	
ENSG000000001460.18	STPG1	protein_coding	374	195	195	2.6724	1.0658	1.0506	
ENSG000000001461.17	NIPAL3	protein_coding	1133	580	569	7.3333	2.9246	2.8830	
ENSG000000001497.18	LAS1L	protein_coding	2168	1066	1120	10.5016	4.1881	4.1286	

Figura 2.1: Archivo RNA-seq Star

En la Figura 2.1 podemos ver un tipo de archivo producto del proceso de secuenciación de RNA según el método STAR [7]. Los campos principales del archivo son:

- **gene-model:** Esta línea indica la versión del modelo de genes utilizado para la alineación y cuantificación. En este caso, se menciona que se utilizó el modelo GENCODE v36.
- **gene-id:** Este campo contiene un identificador único para cada gen en el conjunto de datos. Los identificadores suelen seguir una convención estándar, como el formato ENSG00000000003.15.
- **gene-name:** Aquí se encuentra el nombre del gen correspondiente al identificador proporcionado en el campo anterior. Por ejemplo, “TSPAN6” es el nombre del gen correspondiente a “ENSG00000000003.15”.
- **gene-type:** Indica el tipo de gen. En este caso, todos los genes tienen el tipo “protein-coding”, lo que significa que estos genes codifican proteínas.
- **unstranded, stranded-first, stranded-second:** Estos campos representan la cuantificación de la expresión del gen en diferentes condiciones o réplicas del experimento. Pueden indicar el nivel de expresión del gen en diferentes muestras o condiciones experimentales.

- tpm-unstranded: Representa la cantidad de transcripciones por millón (TPM, Transcripts Per Million) de moléculas de ARN del gen en la muestra no estratificada.
- fpkm-unstranded: Indica el valor de expresión del gen en FPKM (Fragments Per Kilobase of transcript per Million mapped reads) en la muestra no estratificada.
- fpkm-ug-unstranded: Es el valor de expresión del gen en FPKM con una normalización adicional llamada Upper Quartile (UQ) en la muestra no estratificada.

miRNA_ID	isoform_coords	read_count	reads_per_million_miRNA_mapped	cross-mapped	miRNA_region
hsa-let-7a-1	hg38:chr9:94175961-94175982:+	2	0.313153	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175961-94175983:+	2	0.313153	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175961-94175984:+	10	1.565763	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175981:+	292	45.720270	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175982:+	12836	2009.812948	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175983:+	16646	2606.368520	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175984:+	30314	4746.452921	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175985:+	833	130.428029	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175962-94175986:+	19	2.974949	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175982:+	1	0.156576	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175984:+	2	0.313153	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175963-94175985:+	1	0.156576	N	mature,MIMAT0000062
hsa-let-7a-1	hg38:chr9:94175965-94175983:+	7	1.096034	N	mature,MIMAT0000062

Figura 2.2: Archivo miRNA-seq

La Figura 2.2 representa un archivo que contiene datos de cuantificación de expresión de miARNs [14] utilizando la base de datos miRBase21. Estos campos describen la expresión de diferentes isoformas del miARN hsa-let-7a-1 en una muestra de secuenciación de ARN.

- miRNA-ID: Este campo contiene el identificador único para cada miARN en el conjunto de datos. Por ejemplo, “hsa-let-7a-1” es un miARN específico.
- isoform-coords: Indica las coordenadas de la isoforma del miARN en el genoma de referencia. Esto incluye la información sobre el cromosoma (hg38:chr9), la posición inicial (94175961) y final (94175982), así como la dirección del miARN (+ indica que está en el sentido positivo).
- read-count: Representa el recuento de lecturas o secuencias de ARN que se asignaron a este miARN específico en la muestra analizada.
- reads-per-million-miRNA-mapped: Es el número de lecturas por millón de lecturas de ARN mapeadas a miARN. Esta métrica es útil para normalizar la expresión del miARN y compararla entre muestras.
- cross-mapped: Indica si las lecturas se han mapeado cruzadamente, es decir, si se han asignado a múltiples ubicaciones en el genoma. En este caso, “N” sugiere que no se han mapeado cruzadamente.
- miRNA-region: Describe la región del miARN a la que pertenecen las lecturas mapeadas. En este caso, todas las lecturas pertenecen a la región “mature” del miARN MIMAT0000062.

2.2. GDC Data Portal

El GDC Data Portal (Genomic Data Commons Data Portal) es una plataforma en línea desarrollada y gestionada por el National Cancer Institute (NCI) de los Estados Unidos. Su principal objetivo es proporcionar acceso público y centralizado a una amplia variedad de datos genómicos y clínicos relacionados con el cáncer, incluyendo datos de proyectos de investigación genómica como TCGA (The Cancer Genome Atlas) y otros proyectos similares.

Este portal permite a los investigadores, científicos y profesionales de la salud acceder a una gran cantidad de información genómica, como datos de secuenciación del ADN, datos de expresión génica, datos de metilación del ADN y datos clínicos de pacientes con cáncer. Estos datos son esenciales para la comprensión de la genética subyacente de diferentes tipos de cáncer y para el desarrollo de investigaciones y tratamientos más efectivos.

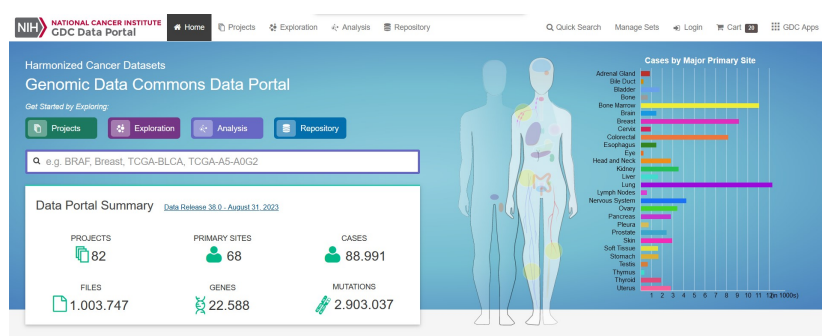


Figura 2.3: Inicio de GDC Portal

En la Figura 2.3 se muestra la pantalla de inicio al acceder al portal GDC. En él ya se puede obtener una idea de la información a la que se puede acceder, así como de la funcionalidad que ofrece. Si nos fijamos en la barra de navegación, distinguimos:

- **Projects:** “project” se refieren a proyectos de investigación genómica específicos que han generado datos genómicos y clínicos relacionados con el cáncer u otras áreas de investigación biomédica. Estos proyectos suelen involucrar la recopilación y el análisis de datos de secuenciación del ADN, expresión génica, metilación del ADN y datos clínicos de pacientes. Al acceder al apartado projects podemos ver información acerca de todos los proyectos que hay almacenados, así como la información acerca de en qué tipo de cáncer se centran.
- **Exploration:** De forma similar al apartado projects, en este caso los usuarios pueden filtrar los casos, genes y mutaciones especificando las características deseadas.
- **Analysis:** En esta sección los usuarios pueden realizar comparaciones entre conjuntos de datos previamente filtrados.
- **Repository:** En esta última sección los usuarios pueden filtrar los archivos almacenados y añadirlos al carrito desde donde podrán descargarlos.

El GDC Data Portal no solo almacena y proporciona acceso a los datos, sino que también ofrece herramientas y recursos para la búsqueda, visualización y análisis de

los mismos. Además, promueve la colaboración entre investigadores y la comunidad científica al facilitar la compartición de datos genómicos y clínicos, lo que contribuye al avance de la investigación sobre el cáncer y otras enfermedades relacionadas con la genética.

Uno de los recursos más valiosos que ofrece esta plataforma es una API que permite a los usuarios acceder a la información almacenada mediante consultas. Esta API puede resultar de gran utilidad para la aplicación que estamos desarrollando, ya que nos permitirá obtener datos de manera eficiente.

Entre la distinta información que almacena destacamos los **metadatos** [16] por su importancia a lo largo del proyecto. Los metadatos son “datos que describen otros datos”, en otras palabras, son datos que describen características, propiedades y contextos relacionados con los datos principales. En el ámbito de la genómica y la bioinformática, los metadatos son esenciales para entender y contextualizar la información genómica. Los metadatos pueden abarcar detalles sobre la muestra, como el tipo de tejido, así como información relacionada con el paciente, como un identificador, entre otros.

2.3. Técnicas

A continuación se describen algunas de las principales técnicas que se realizan en el estudio de los datos genómicos del cáncer.

2.3.1. Recuento de datos

Partiendo de que se tiene un conjunto de muestras, y para cada muestra su correspondiente archivo de expresión genética como el mostrado en las Figuras 2.1 y 2.2, el objetivo de hacer el recuento implica ir iterando por todos los archivos de expresión para sumar los valores de cada gen de tal forma que queda un archivo resultante donde las columnas se refieren a las muestras y las filas a los genes específicos.

	sample1,	sample2,	sample3,	sample4,	...
ENSG00000000003,	8293,	2838,	789,	8972,	
ENSG00000000005,	892,	1345,	1211,	6237,	
ENSG000000000419,	234.	2032,	2321,	2312,	
ENSG000000000460,	2341,	2311,	4544,	456,	
...					

Figura 2.4: Estructura RNA counts

La Figura 2.4 representa la estructura resultante después de realizar el conteo descrito. Es importante destacar que este proceso también cumple la función de fusionar o combinar todas las muestras en un solo conjunto de datos consolidado.

2.3.2. Normalización

Suele ser el primer paso en la mayoría de análisis. objetivo principal de la normalización [20] es permitir comparaciones significativas entre muestras y genes, asegurando que las diferencias observadas en la expresión sean más representativas de las diferencias biológicas que de las variaciones técnicas.

Las técnicas de normalización comunes incluyen:

- Normalización por tamaño de la biblioteca: Se ajusta la cantidad de lecturas en cada muestra para que sean comparables. Por ejemplo, utilizando el número total de lecturas mapeadas o TPM (Transcripts Per Million).
- Normalización por longitud del gen: Se ajusta la expresión del gen según su longitud para comparar genes de diferentes tamaños.
- Normalización por eficiencia de secuenciación: Se corrigen los sesgos de secuenciación utilizando métodos como DESeq2 o edgeR.
- Normalización de la profundidad de secuenciación: Se ajusta la profundidad de secuenciación para comparar entre muestras con diferentes niveles de secuenciación.

La elección del método de normalización depende fundamentalmente del tipo de análisis a realizar.

2.3.3. Análisis de expresión diferencial

El objetivo de este análisis es identificar genes cuya expresión varía significativamente entre dos o más condiciones o grupos diferentes, como muestras de tejido de pacientes con enfermedades y muestras de tejido de individuos sanos.

Los resultados a tener más en cuenta del análisis de expresión diferencial son:

- **Fold change:** El fold change se utiliza para cuantificar la magnitud del cambio en la expresión de un gen entre dos condiciones o grupos diferentes, como un grupo de control y un grupo de tratamiento. Se calcula como la razón entre la expresión del gen en una condición (por ejemplo, tratamiento) y la expresión en otra condición (por ejemplo, control). Generalmente, se toma el logaritmo en base 2 del fold change (\log_2FC o $\log FC$) para representar de manera más clara la magnitud del cambio. Un valor positivo de $\log FC$ indica un aumento en la expresión (sobreexpresión) en la condición de tratamiento en comparación con el control, mientras que un valor negativo indica una disminución en la expresión (subexpresión). Por ejemplo, si el $\log FC$ de un gen es 1, significa que la expresión en el grupo de tratamiento es aproximadamente el doble de la expresión en el grupo de control. Si es -1, la expresión en el grupo de tratamiento es aproximadamente la mitad de la expresión en el grupo de control.
- **P-value:** El valor p es una medida estadística que se utiliza para evaluar la significación de una diferencia observada en un análisis de expresión génica diferencial.

Representa la probabilidad de observar una diferencia tan extrema o más extrema que la observada, asumiendo que no hay diferencia real entre las condiciones comparadas. Un valor p bajo (generalmente < 0.05) indica que la diferencia en la expresión del gen es estadísticamente significativa y que es poco probable que se deba al azar.

Teniendo en cuenta estas dos propiedades se puede concluir que los genes más significativos son aquellos cuyo fold change es o muy pequeño o muy grande mientras que el p -value es también un valor muy pequeño. La representación gráfica se puede realizar con un “volcano plot” (ver Figura 2.5).

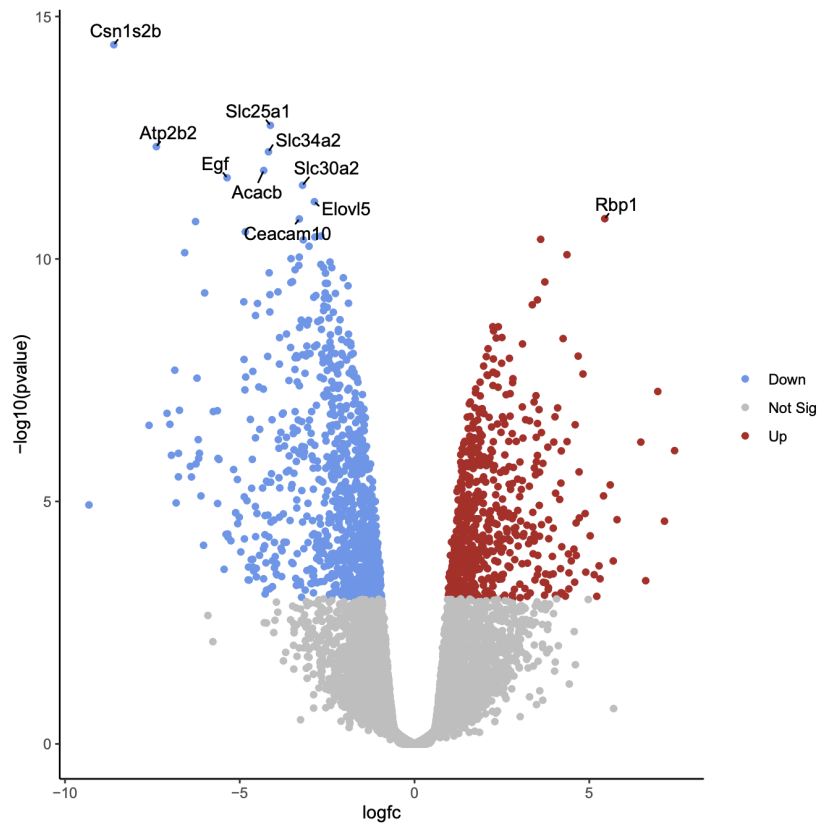


Figura 2.5: Ejemplo volcano plot [8]

2.3.4. Análisis de enriquecimiento

El análisis de enriquecimiento [5] tiene como objetivo descubrir anotaciones biológicas que están sobre-representadas en una lista de genes en comparación con un fondo de referencia. Estas anotaciones se utilizan para interpretar los mecanismos moleculares y los procesos biológicos que están asociados con la condición experimental en estudio.

1. **Selección de genes de interés:** Se comienza con un conjunto de genes de interés, que a menudo se obtiene a partir de un experimento o análisis previo. Estos genes pueden ser, por ejemplo, los genes diferencialmente expresados en una condición específica o un conjunto de genes relacionados con una enfermedad.

2. **Base de datos de anotación:** Se utilizan bases de datos de anotación genómica, como Gene Ontology (GO), Kyoto Encyclopedia of Genes and Genomes (KEGG), Reactome u otras, que contienen información sobre las funciones biológicas, vías metabólicas y procesos celulares asociados con los genes.
3. **Comparación con la base de datos de anotación:** Se compara el conjunto de genes de interés con la base de datos de anotación. Se calcula si hay una sobre-representación significativa de genes relacionados con funciones o vías específicas en el conjunto de genes de interés en comparación con lo que se esperaría al azar.
4. **Cálculo de estadísticas de enriquecimiento:** Se utilizan pruebas estadísticas, como la prueba de hipótesis de Fisher o el método de hipergeométrica, para calcular la significación estadística del enriquecimiento. Esto determina si la asociación observada entre los genes de interés y las funciones o vías es estadísticamente significativa.
5. **Interpretación de resultados:** Los resultados del análisis de enriquecimiento se presentan en forma de listas ordenadas de funciones o vías en función de su significación estadística. Esto permite identificar las funciones o vías más relevantes y relacionadas con el conjunto de genes de interés.

De forma similar al análisis diferencial, al hacer el análisis de enriquecimiento obtendremos distintos atributos, entre ellos el “p-value” que indica la significancia estadística.

2.3.5. Análisis de supervivencia univariado

El análisis de supervivencia univariado [12] se trata de un método que sirve para evaluar el tiempo hasta que ocurre un evento de interés, como la supervivencia de pacientes después de un diagnóstico o el tiempo hasta que se presenta una complicación médica.

Los datos de entrada para realizar el análisis son, para un conjunto de muestras: un atributo referente al evento que queremos estudiar como podría ser el estado vital del paciente (vivo o muerto) y otro atributo que indique el número de días hasta la muerte o número de días de seguimiento del caso. El análisis nos proporciona una función de la probabilidad (Figura 2.6) de que ocurra el evento según el tiempo. En el caso descrito definiría la probabilidad de muerte a lo largo de los días.

Antes de realizar el análisis es ideal dividir las muestras según algún atributo para comparar las líneas de supervivencia de cada grupo.

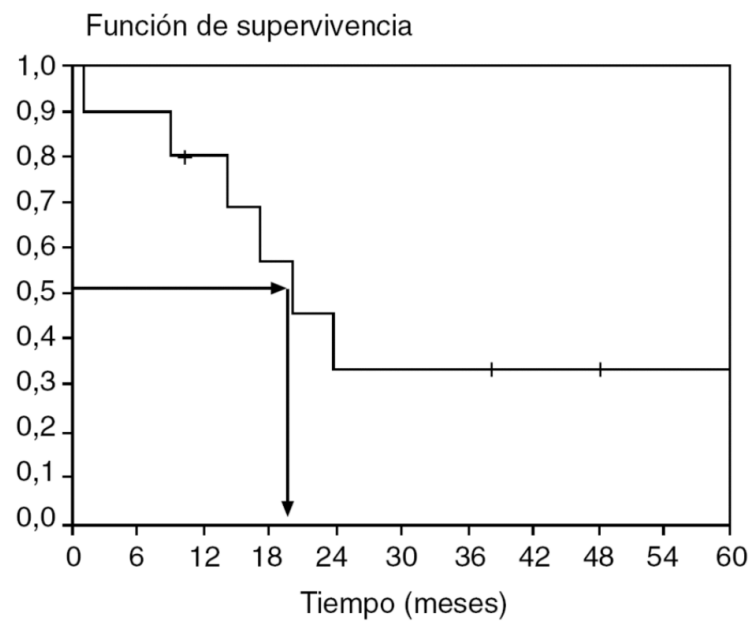


Figura 2.6: Ejemplo gráfica de supervivencia [24]

Parte III

SISTEMA DESARROLLADO

3. Objetivos

Es esencial establecer claramente los objetivos de este proyecto. Esto implica identificar qué queremos ofrecer a los usuarios con la aplicación software.

- **Obj. 1 - Aplicación Web.** El software a desarrollar debe ser una aplicación Web.
- **Obj. 2 - Obtención de datos desde GDC portal.** Los datos obtenidos sobre las expresiones genéticas deben ser desde el portal de datos GDC (Genomic Data Commons Data Portal)
- **Obj. 3 - Realizar análisis.** Con los datos obtenidos, la aplicación debe poder ejecutar los análisis.
- **Obj. 4 - Resultados de análisis.** Los usuarios deben poder consultar los datos resultado de los análisis.
- **Obj. 5 - Gráficas resultado.** La aplicación debe mostrar gráficas que interpreten los resultados de los análisis de forma visual.

A partir de estos objetivos se empezará a realizar el proceso de elicitación y análisis de requisitos.

4. Análisis

A continuación se analiza el sistema a desarrollar a través de la definición de los requisitos y los casos de uso.

4.1. Requisitos

Los requisitos del producto son aquellos que definen las limitaciones, el comportamiento y las funcionalidades del producto software a desarrollar.

Diferenciamos distintos tipos de requisitos según lo que describan.

4.1.1. Requisitos de información

Los requisitos de información definen la información que debe almacenar el sistema. Nuestro producto se centra en realizar análisis sobre un conjunto de datos, por lo que la información que debe almacenar serán los datos de los conjuntos de análisis y los resultados obtenidos.

Los requisitos de información se encuentran definidos en el Cuadro 4.1.

En la Sección 5.1 se proporcionan definiciones de los campos que componen los requisitos de información.

Cuadro 4.1: Requisitos de información

ID	Título	Descripción
RI-001	StudyCase	StudyCase se refiere al caso de estudio a analizar. Los campos principales a almacenar serán: <ul style="list-style-type: none">■ project■ data_type■ state
Continúa en la siguiente página		

Cuadro 4.1 – Continuación desde la página anterior

ID	Título	Descripción
RI-002	Metadata	<p>Son los metadatos del project. Estos datos se deben obtener a través del GDC Portal. Los campos que lo definen son:</p> <ul style="list-style-type: none"> ▪ case_id ▪ file_name ▪ file_id ▪ patient ▪ sample ▪ submitter_id ▪ sample_type ▪ gender ▪ age_at_diagnosis ▪ tumor_stage ▪ tumor_grade ▪ days_to_death ▪ days_to_last_follow_up ▪ vital_status ▪ project_id
RI-003	DiffExprAnalysisData	<p>Se deben almacenar los resultados del análisis de expresión diferencial. Los campos que contiene son:</p> <ul style="list-style-type: none"> ▪ gene id ▪ symbol ▪ group ▪ base_mean ▪ log_fc ▪ lfc_se ▪ stat ▪ pvalue ▪ fdr
Continúa en la siguiente página		

Cuadro 4.1 – Continuación desde la página anterior

ID	Título	Descripción
RI-004	EnrichData	Se debe almacenar los resultados del análisis de enriquecimiento. Los campos que continen son: <ul style="list-style-type: none"> ■ terms ■ counts ■ gene_ratio ■ bg_ratio ■ pvalue ■ fdr ■ fold_enrichment ■ gene_ids ■ gene_symbols ■ category
RI-005	RNAExpression	Se debe almacenar el resultado de la normalización del conteo de las expresiones genéticas. Los campos a contener son: <ul style="list-style-type: none"> ■ gene_id ■ data
RI-006	SurvivalAnalysisResults	Se debe almacenar el resultado del análisis de supervivencia univariada. Los campos que lo componen son: <ul style="list-style-type: none"> ■ gene id ■ symbol ■ hr (Hazard Ratio) ■ lower95 ■ upper95 ■ pvalue

4.1.2. Requisitos funcionales

Los requisitos funcionales describen como se tiene que comportar el sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. En el Cuadro 4.2 se describen estos requisitos.

Cuadro 4.2: Requisitos funcionales

ID	Título	Descripción
RF-001	Descarga datos	La aplicación debe poder descargar los datos desde GDC portal según los campos project y datatype (RNA o miRNA).
Continúa en la siguiente página		

Cuadro 4.2 – Continuación desde la página anterior

ID	Título	Descripción
RF-002	Obtener meta-datos	La aplicación debe obtener los metadatos según el project y datatype seleccionados.
RF-003	Análisis de expresión diferencial	La aplicación debe poder realizar un análisis de expresión diferencial sobre los datos obtenidos.
RF-004	Análisis de enriquecimiento	La aplicación debe poder realizar un análisis de enriquecimiento sobre los datos obtenidos.
RF-005	Análisis de supervivencia univariado	La aplicación debe poder realizar un análisis de supervivencia univariable.
RF-006	Datatable meta-data	Los usuarios deben poder consultar los metadatos en una tabla.
RF-007	Datatable DE	Los usuarios deben poder consultar los resultados del análisis diferencial en una tabla.
RF-008	Datatable EA	Los usuarios deben poder consultar los resultados del análisis de enriquecimiento en una tabla.
RF-009	Datatable SA	Los usuarios deben poder consultar los resultados del análisis de supervivencia univariable en una tabla.
RF-010	Filtrar en datatables	Los usuarios deben poder filtrar en las datatables que muestran los resultados.
RF-011	Descargar datatables	Los usuarios deben poder descargar los resultados de las datatables en formato CSV.
RF-012	Volcano Plot	Mostrar un gráfico de volcano según “pvalue” y “fold_change”.
RF-013	Project Información	Los usuarios deben ver información del proyecto seleccionado: name, project-id, disease-type, primary-site, db-gap-accession-number.
RF-014	Bar Plot DE	El usuario debe poder ver un gráfico de barras del resultado del análisis diferencial que distinga entre upregulated y downregulated genes.
RF-015	Significant genes VP	El usuario debe poder ajustar los umbrales del volcano Plot que distingue los genes significativos (por defecto: P-value: 0.05, logFC: 2).
RF-016	Correlation genes Plot	El usuario debe poder ver un plot de correlación entre dos genes del análisis de expresión diferencial.
RF-017	Bar Plot EA	El usuario debe poder ver un gráfico de barras del resultado del análisis de enriquecimiento filtrando por categorías y el número de términos.
RF-018	Bubble Plot EA	El usuario debe poder ver un gráfico de burbujas filtrado por categorías y número de términos.
RF-019	Survival Plot	El usuario debe poder ver un gráfico de supervivencia de un gen seleccionado a partir del resultado del análisis de supervivencia.
Continúa en la siguiente página		

Cuadro 4.2 – Continuación desde la página anterior

ID	Título	Descripción
RF-020	PNG Plots	El usuario debe poder descargar como PNG los distintos plots.
RF-021	Resumen estadístico metadata	La aplicación debe mostrar un resumen estadístico de las variables numéricas de metadata.
RF-022	Resumen categorías metadata	La aplicación debe mostrar un conteo de los casos agrupados
RF-023	Listado de StudyCase	Listado de StudyCases almacenados pudiendo filtrar por project y datatype.
RF-024	Ocultar resultados de enriquecimiento para miRNAs	Como para el tipo de datos miRNAs no se hará el análisis de enriquecimiento se le debe ocultar esta vista a los resultados.

4.1.3. Reglas de negocio

Las reglas de negocio describen limitaciones de las funcionalidades que el sistema debe respetar. Están definidas en el Cuadro 4.3.

Cuadro 4.3: Reglas de negocio

ID	Título	Descripción
RN-001	Error de selección de gen	En las gráficas donde se pida seleccionar un gen en concreto, aparecerá un error si el gen especificado no se encuentra en los datos.
RN-002	Error API GDC	El análisis no se puede llevar a cabo cuando la API de GDC se encuentre inoperativa. En ese caso se mostrará una pantalla de error 500.
RN-003	Análisis único	En el caso de seleccionar unos datos de entrada ya analizados se consultará la información ya guardada.
RN-004	API solo consulta	La API solo debe tener métodos de consulta, no de escritura ni borrado. Cualquier otra operación debe realizarse mediante la aplicación.
RN-005	StudyCase analizándose	La aplicación no debe realizar un análisis para un StudyCase que está en ejecución.

4.1.4. Requisitos no funcionales

Los requisitos no funcionales son especificaciones que describen características y atributos del sistema de software que no se refieren a funciones específicas, sino que se centran en aspectos de calidad, rendimiento, seguridad, usabilidad y otros criterios que afectan la eficacia y la experiencia del usuario. Estos requisitos definen cómo debe comportarse el sistema en términos de velocidad, confiabilidad, capacidad de respuesta,

escalabilidad, eficiencia energética y otros atributos que no están relacionados con las funciones principales del software, pero que son críticos para su éxito y aceptación.

Las categorías que diferenciamos son:

- Rendimiento: Estos requisitos se refieren a cómo debe responder el sistema en términos de velocidad, tiempo de respuesta y capacidad de procesamiento. Ejemplos incluyen el tiempo de carga de una página web, la capacidad de manejar un cierto número de usuarios concurrentes o la eficiencia en el uso de recursos de hardware.
- Usabilidad: Estos requisitos describen la facilidad de uso y la experiencia del usuario. Pueden incluir pautas de diseño de interfaz de usuario, accesibilidad y pruebas de usabilidad.
- Portabilidad: Estos requisitos establecen la capacidad del software para ejecutarse en diferentes plataformas y sistemas operativos. Pueden incluir la compatibilidad con navegadores web, sistemas operativos específicos o dispositivos móviles.
- Arquitectura: Estos requisitos definen la arquitectura y el diseño de la aplicación a desarrollar.

Los requisitos no funcionales comprenden más categorías como disponibilidad, seguridad o mantenibilidad pero no se definieron de ese tipo.

El Cuadro 4.4 define los requisitos no funcionales.

Cuadro 4.4: Requisitos no funcionales

ID	Título	Descripción	Categoría
RNF-001	Error 404	La aplicación debe redireccionar a los usuarios a una vista 404 cuando no encuentre la url solicitada.	Usabilidad
RNF-002	Error 500	La aplicación debe redireccionar a los usuarios a una vista 500 cuando surga algún error del sistema.	Usabilidad
RNF-003	Consultas asíncronas	Debido a la gran cantidad de información a mostrar en los resultados, las consultas se deben realizar de forma asíncrona.	Usabilidad
RNF-004	Web Responsive	La interfaz debe ser web responsive, es decir que se pueda visualizar correctamente con cualquier resolución de dispositivo.	Portabilidad
RNF-005	Inglés	El idioma de la aplicación será el inglés.	Usabilidad
RNF-006	Paginación API	Las consultas que devuelvan una gran cantidad de información deben poder usar parámetros de paginación.	Rendimiento

Continúa en la siguiente página

Cuadro 4.4 – Continuación desde la página anterior

ID	Título	Descripción	Categoría
RNF-007	Análisis asíncrono	El análisis se debe hacer de forma asíncrona para no bloquear la aplicación a los usuarios.	Usabilidad
RNF-008	Select con search	Se debe poder buscar con facilidad entre las opciones de los selects que tengan muchas.	Usabilidad
RNF-009	API Backend	El sistema debe proporcionar acceso a los datos a través de una API que al menos permita obtener los resultados de los distintos análisis según el Study-Case.	Arquitectura
RNF-010	Acceso público	El acceso a la información almacenada de los resultados de análisis será pública de forma que no hace falta autenticarse en el sistema para consultarla.	Accesibilidad

4.1.5. Matriz de trazabilidad

La matriz de trazabilidad indica las dependencias que existen entre los requisitos y los objetivos del proyecto. Ayuda a tener una visión clara de como alcanzar dichos objetivos en específico.

El Cuadro 4.5 representa la matriz de trazabilidad de requisitos de proyecto, funcionales y no funcionales con respecto a los objetivos definidos en la sección 3.

Cuadro 4.5: Matriz de trazabilidad de requisitos / objetivos

	Obj-1	Obj-2	Obj-3	Obj-4	Obj-5
RI-001		x	x		
RI-002		x	x		
RI-003				x	
RI-004				x	
RI-005				x	
RI-006				x	
RF-001		x			
RF-002		x			
RF-003			x		
RF-004			x		
RF-005			x		
RF-006				x	
RF-007				x	
RF-008				x	
Continúa en la siguiente página					

Cuadro 4.5 – Continuación desde la página anterior

	Obj-1	Obj-2	Obj-3	Obj-4	Obj-5
RF-009				x	
RF-010				x	
RF-011				x	
RF-012					x
RF-013				x	
RF-014					x
RF-015					x
RF-016					x
RF-017					x
RF-018					x
RF-019					x
RF-020				x	x
RF-021				x	
RF-022				x	
RF-023			x		
RF-024				x	
RNF-001	x				
RNF-002	x				
RNF-003	x		x	x	
RNF-004	x				
RNF-005	x				
RNF-006				x	
RNF-007		x	x		
RNF-008	x	x			
RNF-009	x				
RNF-010				x	
RN-001					x
RN-001		x	x		
RN-001			x		
RN-001	x				
RN-001		x	x		

4.2. Casos de uso

Los casos de uso describen la interacción entre los usuarios y el sistema.

Para los casos de uso solo definimos el actor “usuario” que se refiere al usuario de la aplicación. Este actor tiene acceso a todas las funcionalidades que se ofrecen.

La Figura 4.1 representa el diagrama de los casos de uso del sistema. Estos casos de uso son:

- **Análisis StudyCase:** EL usuario debe poder analizar un caso de estudio. Este caso de uso incluye:

- **Obtener datos:** Para empezar el análisis primero se debe obtener los datos de expresión genética.
 - **Análisis de expresión diferencial:** El análisis incluye un análisis de expresión diferencial comparando las muestras de tejido sano con las de tejido cancerígeno.
 - **Análisis de enriquecimiento:** El análisis incluye un análisis de enriquecimiento de los genes más significativos resultado del análisis de expresión diferencial.
 - **Análisis de supervivencia:** El análisis incluye un análisis de supervivencia Kanplan-Meier.
- **Consultar resultados:** Los usuarios deben poder consultar los resultados de los distintos análisis.
 - **Consultar gráficos resultados:** Los usuarios deben poder ver interpretaciones gráficas de los resultados de los análisis.
 - **Listar StudyCases:** Los usuarios deben poder listar los casos de estudio, viendo cuales de ellos han sido ya analizados.
 - **Descargar resultados:** Los usuarios deben poder descargar los resultados mostrados por la aplicación.

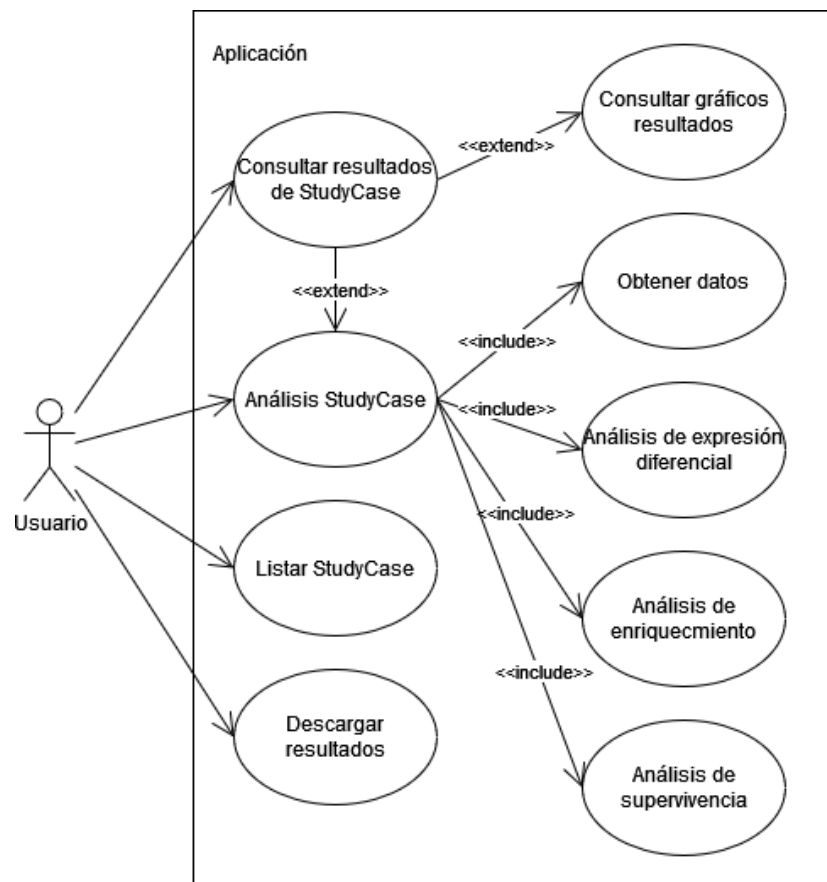


Figura 4.1: Diagrama casos de uso

5. Diseño

En esta sección, nuestro enfoque principal es delinear con precisión el sistema que planeamos desarrollar concretando los técnicas de diseño, la estructura de las clases del modelo, las tecnologías que se usaran y definir las consultas de la API.

5.1. Diagrama de clases

El diagrama de clases define el modelo de las entidades que van a ser almacenadas en la base de datos y que va a gestionar el software. Las clases viene definidas por los requisitos de información anteriormente descritos (ver Cuadro 4.1).

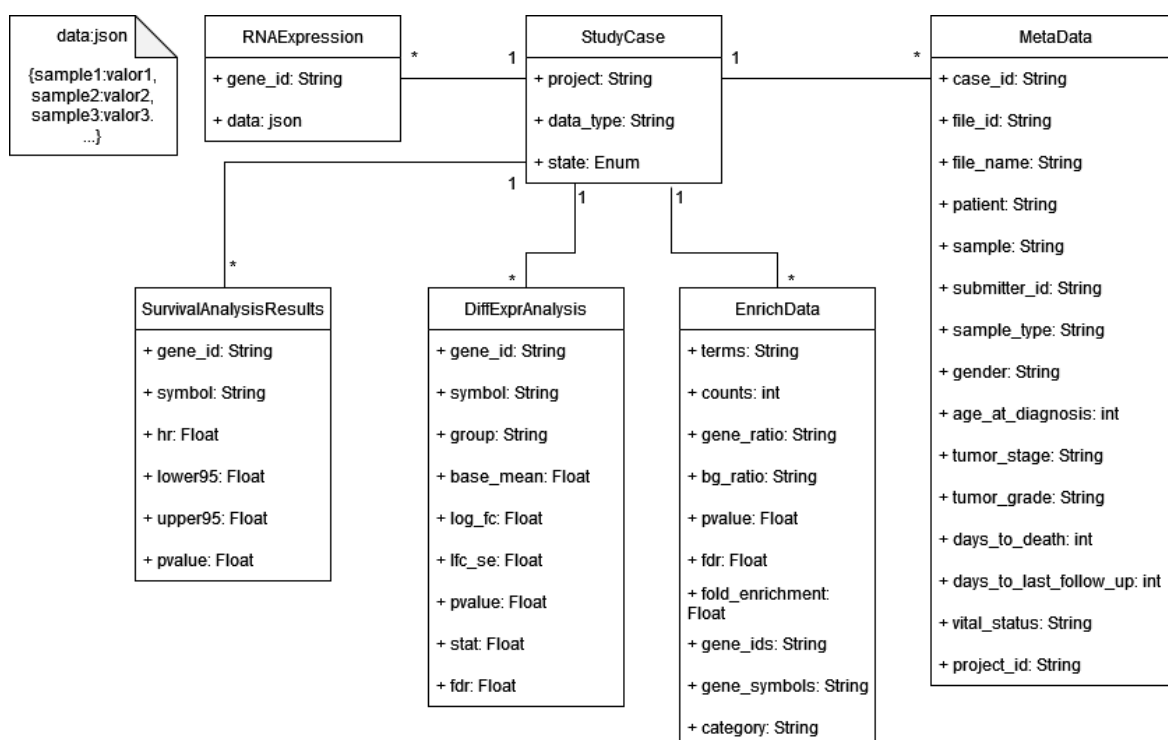


Figura 5.1: Diagrama de clases

La Figura 5.1 representa el diagrama de clases del sistema. A continuación se procede a explicar en detalle que representan dichas clases y algunas restricciones:

- **StudyCase**: Se trata de caso de estudio a analizar. Está formado por los campos project y datatype que son el proyecto de investigación y el tipo de datos a analizar respectivamente. Existe una restricción de unicidad de la tupla formada por esos dos campos. Adicionalmente, tiene el campo state que será “ANALISIS” cuando el caso de estudio se encuentre en pleno análisis y en “DONE” cuando el análisis haya terminado.

- **MetaData:** Representan los metadatos de los proyectos. Los campos están definidos analizando los datos obtenidos tras la descarga de dichos metadatos.
 - “case_id”: Contiene un identificador de la muestra y el paciente.
 - “file_name”: Contiene el nombre asociado a cada entrada de datos, que incluyen información detallada sobre pacientes o muestras.
 - “file_id”: Se trata de un código único que identifica cada archivo mencionado en “file_name”. Estos códigos son usados para hacer referencia única a los archivos.
 - “patient”: Representa un identificador único asociado a cada paciente en el conjunto de datos.
 - “sample”: Similar a “patient”, este campo proporciona un identificador único para cada muestra relacionada con un paciente específico.
 - “submitter_id”: Es un identificador proporcionado por la entidad que envía los datos.
 - “sample_type”: Indica el tipo de muestra recopilada, pueden ser “Primary-Tumor”, lo que sugiere tejido tumoral primario, o “SolidTissueNormal” que indica tejido sano.
 - “gender”: Indica el género del paciente, que puede ser “male” (hombre) o “female” (mujer).
 - “age_at_diagnosis”: Representa la edad del paciente en el momento del diagnóstico, expresada en días.
 - “tumor_stage”: Describe la etapa del cáncer en el momento del diagnóstico. Los valores pueden variar según el sistema de clasificación. Un posible valor es “Stage II”.
 - “tumor_grade”: Describe el grado de malignidad del tumor. Ejemplos de posibles valores son “High Grade” o “Intermediate Grade”.
 - “days_to_death”: Indica el número de días desde el diagnóstico hasta la fecha de fallecimiento del paciente.
 - “days_to_last_follow_up”: Representa el número de días desde el diagnóstico hasta la última fecha de seguimiento para pacientes que no fallecieron. Complementario con el campo “days_to_death”.
 - “vital_status”: Indica el estado del paciente estaba “Alive” (vivo) o “Dead” (fallecido).
 - “project_id”: Proporciona un identificador para el proyecto o estudio al que pertenecen estos datos.
- **RNAExpr:** Almacenan los resultados del conteo y normalización de las expresiones genéticas de los proyectos. Los campos son:
 - “gene_id”: Identificador del gen.

- “data”: Estructura json que contiene un diccionario con los identificadores de las muestras del proyecto y el valor normalizado para dicha muestra.
- **DiffExprAnalysis:** Se trata de los resultados del análisis de expresión diferencial. Los campos que lo describen son:
- “gene_id”: Este campo es el identificador del gen.
 - “symbol”: Este campo representa el símbolo del gen, que es una abreviatura única para identificar un gen específico.
 - “group”: Indica a qué grupo o categoría pertenece el gen.
 - “base_mean”: Este valor representa la media de lecturas o conteos de expresión del gen en todas las muestras o grupos del estudio. Es una medida de la expresión promedio del gen en el conjunto de datos.
 - “log_fc”: Es el logaritmo en base 2 del cambio en la expresión del gen entre dos grupos o condiciones. Indica la magnitud de la diferencia en la expresión del gen. Un valor negativo indica que el gen está menos expresado en el segundo grupo en comparación con el primero.
 - “lfc_sE”: Es el error estándar del logaritmo en base 2 del cambio en la expresión (logFC). Proporciona información sobre la precisión de la estimación del cambio en la expresión.
 - “stat”: Es el valor de estadística de prueba asociado con el logFC y su error estándar. Se utiliza para calcular el valor P y determinar si el cambio en la expresión es estadísticamente significativo.
 - “pvalue”: Es el valor P , que indica la probabilidad de observar un cambio en la expresión del gen tan grande como el observado, si no hubiera diferencia real entre los grupos. Valores bajos de P generalmente indican diferencias significativas.
 - “fdr”: El valor de “FDR” (tasa de descubrimiento falso) ajusta el valor P para controlar los errores de tipo I (falso positivo). Los valores bajos de FDR indican que las diferencias son menos propensas a ser falsos positivos.
- **EnrichData:** Se trata de los resultados del análisis de enriquecimiento. Para realizar este análisis se pasa de entrada una lista de genes. Los campos que lo forman son:
- “terms”: Representa los términos de la ontología génica asociados con un enriquecimiento específico. Los términos describen funciones biológicas, procesos celulares o componentes celulares asociados con conjuntos de genes.
 - “counts”: Indica el número de genes en tu lista que están asociados con el término.
 - “gene_ratio”: Proporciona la proporción de genes en la lista que están asociados con el término. Por ejemplo, “228/426” indica que 228 genes de un total de 4262 en la lista están asociados con ese término.

- “bg_ratio”: Representa la proporción de genes en tu conjunto de fondo (o conjunto de genes de referencia) que están asociados con el término. Ayuda a entender la distribución de ese término en tu conjunto de referencia más grande.
 - “p_value”: Es el valor P asociado con el enriquecimiento del término ontológico. Indica la significancia estadística del enriquecimiento. Valores bajos de p_value sugieren que el enriquecimiento es más significativo.
 - “fdr” (Tasa de Descubrimiento Falso): Ajusta el valor P para controlar los errores de tipo I. Los valores bajos indican que el enriquecimiento es menos probable que sea un falso positivo.
 - “fold_enrichment”: Indica cuántas veces los genes asociados con el término están enriquecidos en tu lista en comparación con el fondo. Un valor mayor que 1 sugiere enriquecimiento.
 - “gene_ids”: Proporciona los ID de los genes que contribuyen al enriquecimiento del término de la ontología. Es una cadena de identificadores de genes separados por comas.
 - “gene_symbol”: Muestra los símbolos de los genes correspondientes a los IDs mencionados anteriormente. Al igual que el campo “gene_ids”, se trata de una cada con identificadores separados por comas.
 - “category”: Indica la categoría de los términos, como “GO_BP” para procesos biológicos.
- **SurvivalAnalysisResults:** Se trata de los resultados del análisis de supervivencia univariada. Para este análisis las muestras se dividen según el valor de la mediana de los valores de RNAExpression para cada gen. Los campos son:
- “gene_id”: Identificador del gen.
 - “symbol”: Representa el símbolo del gen correspondiente.
 - “hr” (Hazard Ratio): Es una medida de la razón instantánea de riesgo entre dos grupos. En este caso compara el riesgo de fallecimiento entre dos grupos.
 - Si $hr > 1$: Indica un aumento en el riesgo del evento para el grupo mencionado.
 - Si $hr < 1$: Indica una disminución en el riesgo del evento para el grupo mencionado.
 - Si $hr = 1$: Indica que no hay diferencia en el riesgo entre los dos grupos.
 - “lower95” y “upper95”: Representan los límites inferior y superior de un intervalo de confianza al 95 % para el Hazard Ratio. Este intervalo proporciona una estimación de la variabilidad del Hazard Ratio y ayuda a determinar su precisión.
 - “pvalue”: Es el valor P asociado con el Hazard Ratio. Indica la significancia estadística de la asociación entre el gen y el evento de interés (supervivencia). Un valor P más bajo sugiere una mayor significancia estadística.

Al ser las relaciones manyToOne provoca que exista una foreign key en las entidades llamada “studyCase” apuntando al id de StudyCase.

5.2. Arquitectura del sistema

La arquitectura de la aplicación sigue el patrón Model-View-Template que viene dado por el framework Django. Sin embargo, se propone la construcción de una API a través de la extensión Django Rest Framework. Dicha API posibilita la opción de que a través de scripts de javascript, en las plantillas, se pueda acceder de forma dinámica a los datos de la base de datos con peticiones Axios. Se puede de todas formas acceder a la base de datos a través de Django ORM (Object-Relational Mapping) para consultas rápidas como obtener un identificador específico.

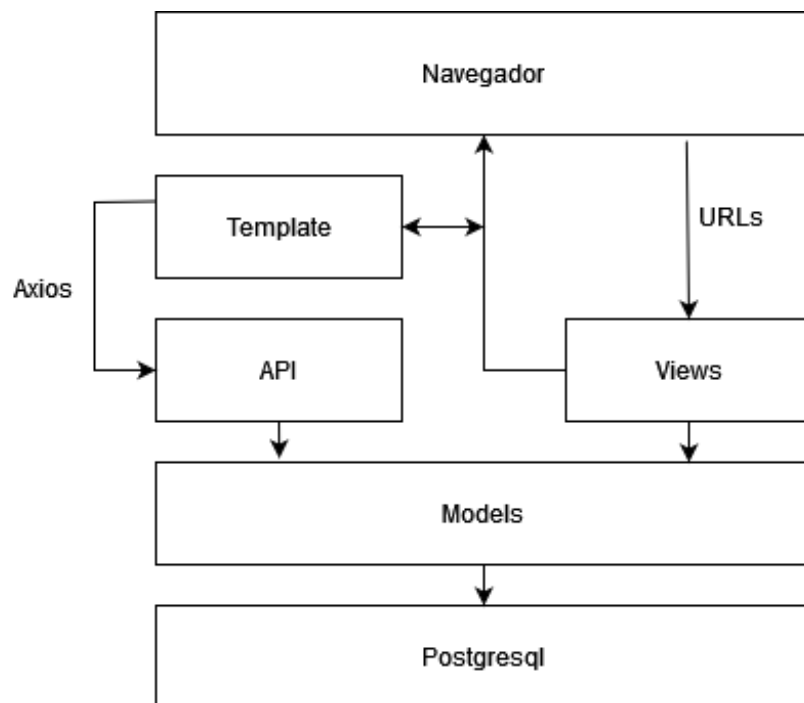


Figura 5.2: Diagrama de arquitectura

La Figura 5.2 representa el diagrama de arquitectura del sistema. Entrando más en detalle, el usuario introduce una url a través de un navegador como puede ser Firefox o Chrome, la URL es procesada por la view que busca la plantilla correspondiente. La view puede acceder al modelo para cargar la plantilla requerida con datos específicos almacenados en la base de datos. Devuelve la plantilla cargada con los datos al navegador para que sea mostrada al usuario. Por otra parte, las plantillas pueden tener scripts de javascript con peticiones axios a la API de la aplicación para acceder a la base de datos y que las vistas sean dinámicas. La base de datos es Postgresql.

La API no solo es útil para dinamizar las vistas, sino que también puede ser de utilidad a otros desarrolladores que quieran acceder a los datos de la aplicación.

Al utilizar Django, nos enfrentamos al desafío de la sincronidad en el proceso de análisis. Se necesita que la función de análisis sea asíncrona para evitar bloquear la navegación de los usuarios en la web. Al hacer clic en el botón de análisis, es crucial que la vista no quede en un estado de carga indefinida mientras se descargan y analizan una gran cantidad de información. Es por ello que nos valemos del uso de Celery y Redis para gestionar la asincronicidad de la tarea.

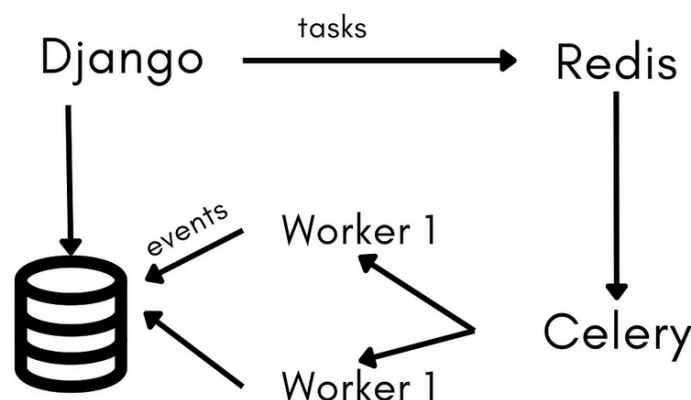


Figura 5.3: Diagrama de arquitectura Celery [1]

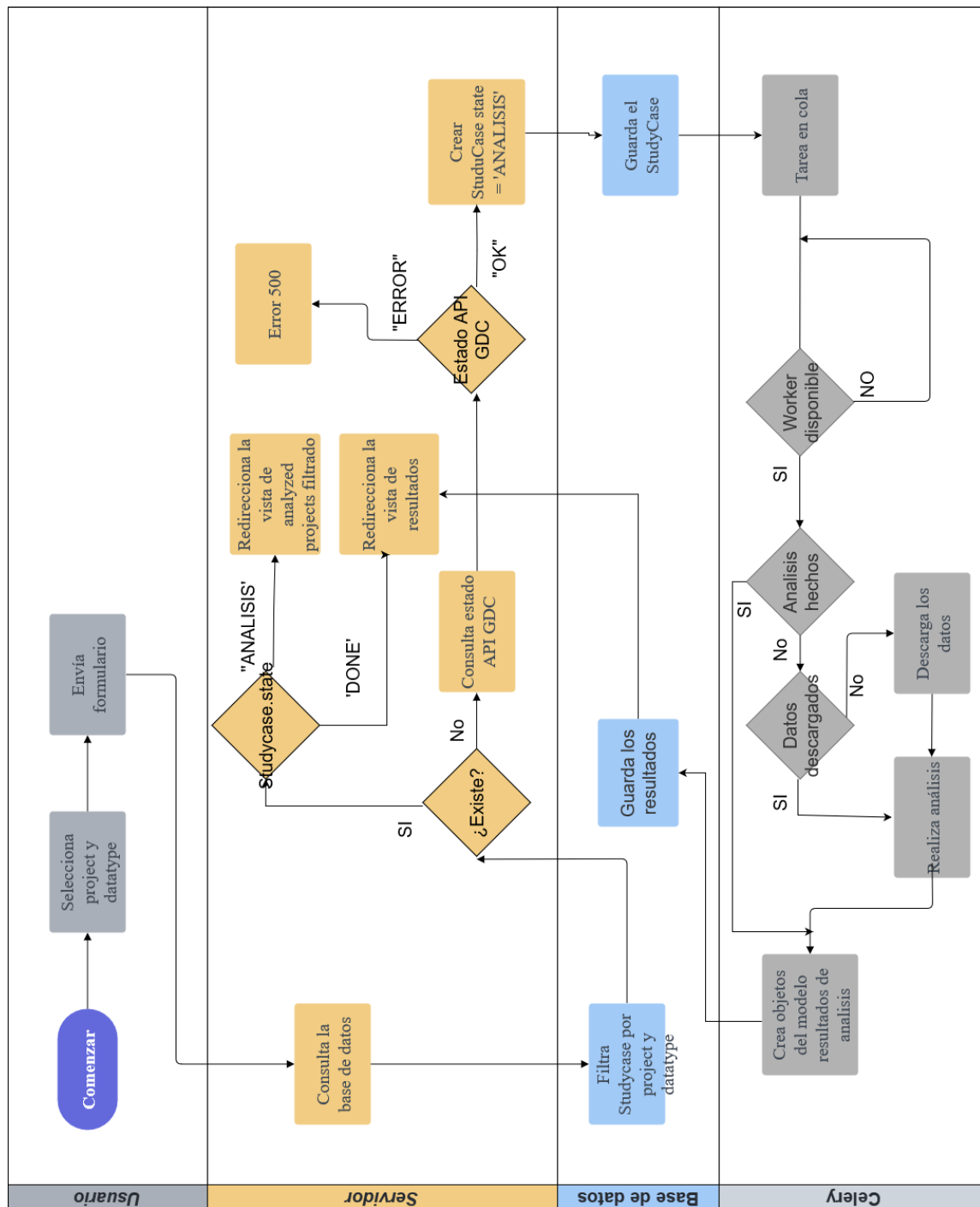
En la Figura 5.3 vemos como se relaciona Celery y Redis con Django y la base de datos. Cuando un usuario realiza una acción en la aplicación Django que desencadena una tarea intensiva, como un análisis, en lugar de procesar la tarea de inmediato, dicha tarea se queda en una cola en Redis. Los trabajadores de Celery, que son procesos por separado, toman tareas de la cola de Redis y las procesan de manera asíncrona. Mientras tanto, Django no está bloqueado y puede seguir manejando otras solicitudes. Una vez que la tarea está completa, Celery podría almacenar los resultados en algún lugar (por ejemplo, en una base de datos) o notificar a Django de alguna manera (por ejemplo, a través de websockets o un sistema de eventos).

La Figura 5.4 ilustra el flujo del proceso de análisis. Inicialmente, el usuario completa y envía el formulario con la información del caso de estudio para su análisis. La vista de Django maneja este formulario, primero verificando si el caso de estudio ya existe en la base de datos. Si existe, redirige al usuario a la vista de resultados correspondiente; de lo contrario, verifica el estado de la API GDC. Si la API está activa, se crea un objeto de caso de estudio con el estado “ANÁLISIS” y se almacena en la base de datos. Si la API no está operativa, se devuelve un error 500 al usuario.

El análisis en sí se gestiona mediante Celery como una tarea asíncrona. Inicialmente, la tarea se coloca en la cola. Una vez que un trabajador está disponible, comienza el análisis. Se verifica la existencia de resultados de análisis; si no están presentes, se verifica la existencia de los datos. Si tampoco están los datos, se descargan y se realiza el análisis. Con los resultados obtenidos, se crean objetos del modelo que representan los resultados y se almacenan en la base de datos. Además, el estado del caso de estudio se actualiza a “DONE”. Finalmente, el usuario es redirigido a la vista de resultados.

Este flujo asegura un manejo eficiente de tareas intensivas, permitiendo que la

aplicación continúe siendo receptiva mientras se realizan procesos de análisis en segundo plano.



SI

Figura 5.4: Diagrama de flujo - Análisis [1]

5.3. Técnicas de diseño aplicadas

A continuación se describen algunos estándares y técnicas a seguir en el diseño de la aplicación:

- **Patrón Modelo-Vista-Plantilla (MVT):** Django sigue una variante del patrón MVC llamada MVT (Modelo-Vista-Plantilla). Esto ayuda a separar la lógica de negocio (Modelo), la presentación (Vista) y el manejo de solicitudes (Controlador) en capas claramente definidas.
- **API RESTful:** DRF se basa en los principios de REST (Representational State Transfer) y facilita la creación de una API RESTful. Esto incluye el uso de URIs (Uniform Resource Identifiers) para identificar recursos, los métodos HTTP para operaciones CRUD y la representación de recursos en formatos como JSON.
- **Serialización:** DRF utiliza el patrón de serialización para convertir los modelos de Django en representaciones legibles por máquinas, como JSON. Esto facilita la comunicación entre el cliente y el servidor.
- **Acceso a Datos Object-Relational Mapping (ORM):** Django utiliza su ORM para abstraer la interacción con la base de datos PostgreSQL, permitiendo a los desarrolladores trabajar con modelos de objetos en lugar de SQL directo.
- **Patrón de Cliente-Servidor:** Al utilizar Axios para realizar solicitudes HTTP desde el lado del cliente a través de la API, se sigue el patrón cliente-servidor. El servidor (Django) proporciona recursos a los cuales el cliente (JavaScript con Axios) accede.

5.4. Tecnologías

En esta sección se explican y enumeran las distintas tecnologías y herramientas de las que nos serviremos no solo para desarrollar el producto sino también para el desarrollo de la documentación.

Herramientas de documentación

Las herramientas de documentación son software que facilita la gestión y el desarrollo de la documentación del proyecto.

- **Google drive:** Repositorio para guardar documentación y compartirla con los tutores del proyecto. Ofrece compatibilidad con otras herramientas de google como draw.io
- **Draw.io:** Plataforma en línea que proporciona herramientas para la creación de diagramas de todo tipo.
- **MSPProject:** Herramienta de gestión de proyecto en la cual se definirá el cronograma del proyecto y ayudará a realizar el seguimiento del desarrollo.

- **Overleaf**: Plataforma en línea que ofrece una interfaz visual para crear documentación con LaTeX. Es ideal para generar la memoria final del proyecto.
- **PowerPoint**: Herramienta para la creación de presentaciones.

Herramientas de desarrollo

Son aquellas tecnologías que facilitan el desarrollo y añaden valor a la aplicación.

- **Visual Studio Code**: Entorno de desarrollo integrado (IDE) de código abierto desarrollado por Microsoft. Será el entorno donde se desarrollará el código del producto debido la preferencia del desarrollador.
- **GitHub**: Nos servirá como repositorio del código del producto desarrollado.
- **Python 3.10**: Lenguaje de programación principal para el desarrollo de la aplicación con django.
- **R 4.3.2**: Lenguaje de programación necesario para ejecutar los métodos proporcionados por la librería GDCRNATools.
- **JavaScript**: Lenguaje para realizar scripts en las plantillas de la aplicación.
- **JQuery**: Librería de javascript que facilita la manipulación del DOM. También es una dependencia de otras herramientas como Datatables.
- **Rpy2** [10]: Librería de python que sirve par ejecutar trozos de código en R. Nos sirve para integrar GDCRNATools con python.
- **Django** [9]: Se trata del framework que facilitará el desarrollo de la aplicación web.
- **Django-rest-framework** [4]: Extensión de django que facilita la construcción de una API.
- **drf-yasg** [28]: Extensión de django-rest-framework que autogenera la documentación en swagger para la API.
- **GDCRNATools** [14]: Es una de las dependencias más importantes del proyecto ya que proporciona los métodos de análisis principales de la aplicación. Es un paquete de R/Bioconductor [21].
- **Lifelines** [6]: La biblioteca lifelines es una herramienta en Python para el análisis de datos de supervivencia y la estimación de funciones de supervivencia. En la aplicación se usará para obtener la gráfica de supervivencia de los datos.
- **Celery** [22] **con Redis**: Sistema de gestión de tareas distribuido y asíncrono que integrado junto a django permitirá realiza el análisis de forma asíncrona.
- **Bootstrap** [19]: Paquete para el front-end de la aplicación que proporciona clases web-responsive sencillas de usar.
- **Datatables JS** [18]: DataTables es un complemento de jQuery para la manipulación de tablas HTML. Proporciona funcionalidades avanzadas para ordenar,

buscar, paginar y filtrar datos en tablas HTML. DataTables facilita la creación de tablas interactivas y dinámicas en una página web. Nos permite mostrar datos de los resultados de una manera organizada en la vista.

- **Plotly.js** [13]: Es una biblioteca JavaScript de código abierto que se utiliza para crear visualizaciones interactivas de datos en la web. Con esta biblioteca podremos representar gráficos de los resultados en la vista. Además plotly ofrece una interfaz en los gráficos que proporciona funcionalidades adicionales como descargar el gráfico como png.
- **Axios** [30]: Biblioteca JavaScript basada en promesas que se utiliza para realizar solicitudes HTTP desde el navegador o desde Node.js. En otras palabras, proporciona una interfaz fácil de usar para interactuar con servicios web y realizar operaciones HTTP, como obtener datos de un servidor o enviar datos a un servidor. El uso en la aplicación es necesario para cargar los resultados en la vista de forma asíncrona reduciendo los tiempos de espera del usuario.

5.5. Diseño API

A continuación, se detallan las consultas que pueden llevarse a cabo mediante la API. Dicha API será de acceso público, por lo que no será necesario realizar ninguna autenticación para acceder. En consecuencia, las solicitudes se limitarán al método “GET”, ya que no se permitirá que los usuarios realicen modificaciones. La única forma de actualizar la base de datos será a través de la función de análisis que proporcionará la aplicación.

Para cada petición se dará una descripción y una definición de sus parámetros de ruta (path param) y parámetros de consulta (query param).

Las diferentes consultas son:

- **GET: /api/StudyCase**
 - **Descripción:** Obtiene una lista de StudyCase.
 - **Parámetros de consulta:**
 - **project:** Proyecto por el que filtrar.
 - **data_type:** Datatype por el que filtrar (Puede ser RNAseq o miRNAs).
 - **page:** Número de página.
 - **maxItems:** Limite de resultados por página.
- **GET: /api/StudyCase/id**
 - **Descripción:** Obtiene el StudyCase por su id.
 - **Parámetros de ruta:**
 - **id:** id del StudyCase.
- **GET: /api/StudyCase/id/metadata**

- **Descripción:** Obtener los metadatos de un StudyCase.
- **Parámetros de ruta:**
 - **id:** id del StudyCase.
- **Parámetros de consulta:**
 - **page:** Número de página.
 - **maxItems:** Limite de resultados por página.
- **GET: /api/StudyCase/id/rnaExpression**
 - **Descripción:** Obtiene los valores normalizados de los genes del StudyCase.
 - **Parámetros de ruta:**
 - **id:** id del StudyCase.
 - **Parámetros de consulta:**
 - **genes_ids:** Lista de ids de genes separados por comas para filtrar.
 - **page:** Número de página.
 - **maxItems:** Limite de resultados por página.
- **GET: /api/StudyCase/id/rnaExpression/gene_id**
 - **Descripción:** Obtiene los valores del normalizados del gene_id de un StudyCase.
 - **Parámetros de ruta:**
 - **id:** id del StudyCase.
 - **gene_id:** Identificador de un gen.
 - **Parámetros de consulta:**
 - **sep:** Separa las muestras en dos grupos dado una fórmula del umbral que puede ser: “mean” o ”median”.
 - **show_metadata:** Con “true” se expande la información resultado con los metadatos de cada muestra.
 - **calculate_kmsurvival_function:** Con “true” devuelve la función de supervivencia KM.
 - **sample_type:** Filtra las muestras por el sample_type.
- **GET: /api/StudyCase/id/differentialExpression**
 - **Descripción:** Obtiene los resultados del análisis de expresión diferencial del StudyCase especificado.
 - **Parámetros de ruta:**
 - **id:** id del StudyCase.

- **Parámetros de consulta:**
 - **order_by:** Campo por el cual ordenar.
 - **group:** Filtro por grupo.
 - **page:** Número de página.
 - **maxItems:** Limite de resultados por página.
- **GET: /api/StudyCase/id/enrichAnalysis**
 - **Descripción:** Obtiene los resultados del análisis de enriquecimiento del StudyCase especificado.
 - **Parámetros de ruta:**
 - **id:** id del StudyCase.
 - **Parámetros de consulta:**
 - **order_by:** Campo por el cual ordenar.
 - **category:** Filtro por categoría
 - **page:** Número de página.
 - **maxItems:** Limite de resultados por página.
- **GET: /api/StudyCase/id/survivalAnalysis**
 - **Descripción:** Obtiene los resultados del analisis de supervivencia univariado del StudyCase especificado.
 - **Parámetros de ruta:**
 - **id:** id del StudyCase.
 - **Parámetros de consulta:**
 - **order_by:** Campo por el cual ordenar.
 - **page:** Número de página.
 - **maxItems:** Número de resultados por página.

Parte IV

PLANIFICACIÓN

6. Riesgos

Reconocer los riesgos que podrían impactar en el progreso del proyecto es una tarea fundamental para garantizar su éxito. No solo implica la identificación inicial de los riesgos, sino también el seguimiento continuo, aplicando planes de contingencia para abordarlos de manera efectiva.

6.1. Análisis probabilidad e impacto

Se pretende definir una escala de análisis de riesgos y evaluar el impacto. Para la probabilidad se describe el Cuadro 6.1, mientras que para medir el impacto se define el Cuadro 6.2.

Cuadro 6.1: Riesgos - Análisis de probabilidad

Muy bajo	Bajo	Moderado	Alto	Muy alto
10 %	30 %	50 %	70 %	90 %
Casi imposible	Poco probable	Ocurre de vez en cuando	Con frecuencia	Casi seguro

Cuadro 6.2: Riesgos - Análisis de impacto

Dimensión	Muy bajo	Bajo	Moderado	Alto	Muy alto
Alcance	Afecta al menos 5 % paquetes de trabajo	5 % - 10 % paquetes de trabajo	10 % - 20 % paquetes de trabajo	20 % - 30 % paquetes de trabajo	Más 30 % paquetes de trabajo
Tiempo	Supone un retraso menor a 2h	Retraso entre 2-5 h	Retraso entre 5-10 h	Retraso entre 10-24 h	Retraso mayor 24 h
Coste	Aumenta coste en menos 1 %	Aumenta costes en menos del 3 %	Aumenta costes en menos 5 %	Aumenta costes en menos del 7 %	Aumenta costes más del 7 %

6.2. Registro de riesgos

En el Cuadro 6.3 se describen los riesgos identificados y la estrategia a seguir para mitigar o evitar dicho riesgo.

A lo largo de los sprints se controlará los riesgos que surgen y se propondrán estrategias a seguir según el impacto que supongan.

Cuadro 6.3: Registro de riesgos

ID	Descripción	Estrategia	Probabilidad	Impacto Coste	Impacto Tiempo	Impacto Alcance
RIE-001	Desviación en el alcance	Analizar desviación en el tiempo y concretar reunión si supone un gran retraso.	Alto	Alto	Alto	Muy alto
RIE-002	Estimación de trabajo incorrecta	Reserva de contingencia y holgura en sprints.	Alto	Alto	Alto	Muy bajo
RIE-003	Requisito no identificado	Menos carga de trabajo en últimos sprints para asignar tareas nuevas.	Muy alto	bajo	bajo	bajo
RIE-004	Problemas con el portafolio	Usar el pc de sobremesa.	Medio	bajo	bajo	bajo
RIE-005	Tareas incompletas	Holgura en sprints para mitigar el riesgo.	Alto	Bajo	Alto	Muy bajo
RIE-006	Desconocimiento de alguna tecnología	Intentar escoger tecnología con experiencia.	Muy Alto	Medio	Medio	Muy Bajo
RIE-007	Retraso en fases del proyecto	Se analizará el impacto sobre la siguiente fase y si es necesario modificar hitos.	Medio	Alto	Muy Alto	Muy bajo
RIE-008	Enfermedad del equipo	La holgura en los sprints puede mitigar desviaciones	Bajo	Bajo	Medio	Muy bajo
RIE-009	Dependencia de la API GDC	Un cambio en la disponibilidad de la API GDC supondría un gran impacto en el producto.	Muy bajo	Muy Alto	Muy Alto	Muy Alto.

7. Planificación

7.1. Metodología de desarrollo

La metodología de desarrollo se refiere a un conjunto de principios, prácticas y procesos estructurados que se siguen durante el ciclo de vida de desarrollo de un proyecto o producto.

En este caso la metodología será estar basada en iteraciones (sprints). Distribuiremos las tareas planificadas a lo largo de cuatro sprints. Algunos puntos a tener en cuenta respecto a la metodología son:

- Se hará un seguimiento a lo largo de los sucesivos sprints produciendo:
 - Incidencias: Se identificarán incidencias en el desarrollo, para poder entender los principales problemas.
 - Desviaciones: Identificar las desviaciones respecto al trabajo, al coste y a las fechas.
 - Retrospectiva: Realizar una retrospectiva para cada sprint destacando los puntos positivos negativos e ideas de mejora para el siguiente sprint.
 - Identificación de riesgos: Identificar que riesgos han ocurrido durante el sprint, analizando su impacto y posibles soluciones.
- Para gestionar el código nos serviremos de un repositorio de GitHub sobre el cual iremos subiendo commits para cada avance en el proyecto. Los commits seguirán la estructura:

Categoría: Nombre

Descripción

Donde “Categoría” puede ser:

- Config: Si el objetivo de los cambios es la configuración del entorno.
- Feat: Si el cambio produce una nueva funcionalidad en la aplicación.
- Fix: Si el objetivo del cambio era arreglar algún error.
- Test: Si el cambio proporciona nuevos tests.
- Refactor: Si el objetivo del cambio es refactorizar el código.

“Nombre” sería el nombre del commit y “Descripción” (opcional) una descripción de los cambios que ha producido el commit.

7.2. Listado de hitos

Con el fin de establecer de manera precisa los plazos de las tareas, se han identificado y definido una serie de hitos que corresponden a las fases clave del proyecto. Estos hitos actúan como puntos de referencia significativos a lo largo del ciclo de vida del proyecto, marcando el inicio o la finalización de etapas importantes.

La identificación y definición de estos hitos permite una planificación más efectiva y un seguimiento más preciso del progreso del proyecto, ya que brindan una visión clara de los momentos críticos en los que se deben alcanzar objetivos específicos. Además, los hitos facilitan la comunicación y la coordinación entre los miembros del equipo y las partes interesadas, ya que proporcionan puntos de control claros para evaluar el avance del proyecto y asegurarse de que se cumplan los plazos establecidos.

En el Cuadro 7.1 se establecen los hitos del proyecto.

Cuadro 7.1: Hitos

Inicio del proyecto	19/06/2023
Planificación	15/07/2023
Sprint 1	01/8/2023
Sprint 2	17/09/2023
Sprint 3	02/09/2023
Sprint 4	19/09/2023
Cierre	30/09/2023

7.3. Estructura de descomposición del trabajo

La estructura de descomposición del trabajo (EDT) se trata de una técnica de gestión de proyectos que se utiliza para descomponer o dividir un proyecto en tareas más pequeñas y manejables. Esta descomposición jerárquica ayuda a comprender y organizar mejor las partes individuales que componen un proyecto completo.

Se diferencian los siguiente niveles jerárquicos dentro de la EDT:

- **Proyecto:** Se trata del nivel máximo de jerarquía de la EDT y se refiere al proyecto en su completitud.
- **Cuenta de control:** Una cuenta de control es un elemento de nivel superior en la EDT que representa una fase, componente o área de responsabilidad importante del proyecto. Agrupan paquetes de planificación y de trabajo. En el proyecto representan las fases de este.
- **Paquetes de planificación:** A un nivel inferior que la cuenta de control, un paquete de planificación se usa para agrupar paquetes de trabajo con una responsabilidad similar y están relacionados.
- **Paquetes de trabajo:** Representan el nivel inferior de la EDT y están asociados a una actividad a realizar en el proyecto.

La EDT del presente proyecto está basada en fases. La Figura 7.1 representa el diagrama general de la EDT. Las siguientes Figuras: 7.2, 7.3, 7.4, 7.5; corresponden a una vista de la EDT más detallada con respecto a los sucesivos sprints.

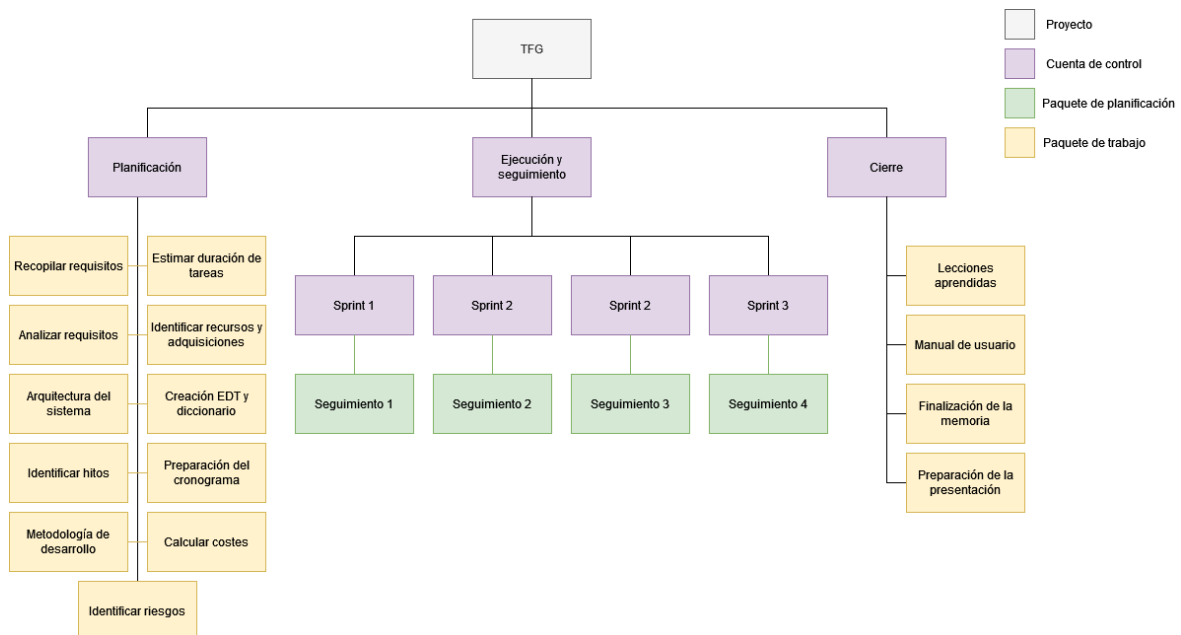


Figura 7.1: Diagrama EDT general

7.4. Diccionario de la EDT

En esta subsección detallaremos los elementos de la EDT proporcionando la siguiente información:

- Id: Identificador único que se utiliza para referirse a cada elemento de la EDT.
- Recursos: Aquellos recursos asignados al elemento de la EDT.
- Inicio: Fecha en el que está previsto que comience la actividad.
- Fin: Fecha en la que se espera que finalice la actividad.
- Trabajo: Se trata de una estimación del número de horas que tardará en realizar el elemento. Calculada según la fórmula de estimación de PERT [3].

El diccionario de la EDT para las tareas de la fase de planificación corresponde al Cuadro 7.2. Los siguientes Cuadros 7.3, 7.4, 7.5, 7.5, 7.6 son los diccionarios para los sucesivos Sprints 1, 2, 3 y 4. El Cuadro 7.7 es el diccionario para la fase de Cierre.

El proyecto está programado para comenzar el 19 de junio de 2023, y se espera que concluya alrededor del 30 de septiembre de 2023. En total, se estima que se dedicarán 300,6 horas de trabajo al proyecto.

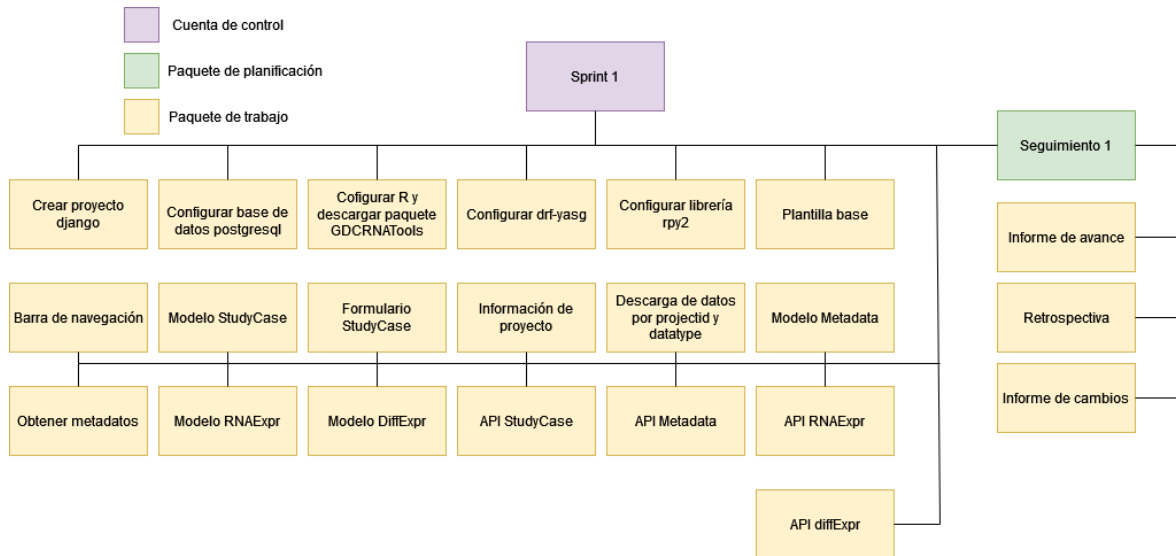


Figura 7.2: Diagrama EDT Sprint 1

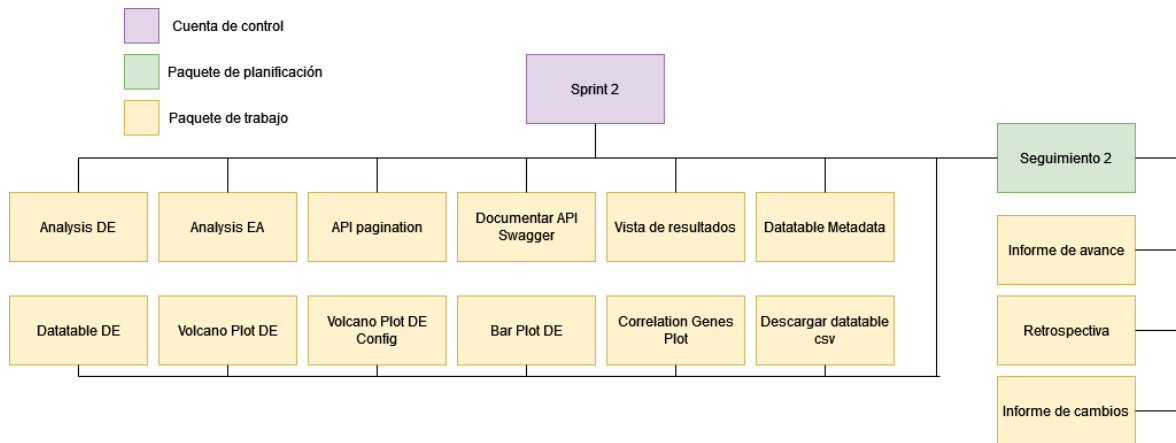


Figura 7.3: Diagrama EDT Sprint 2

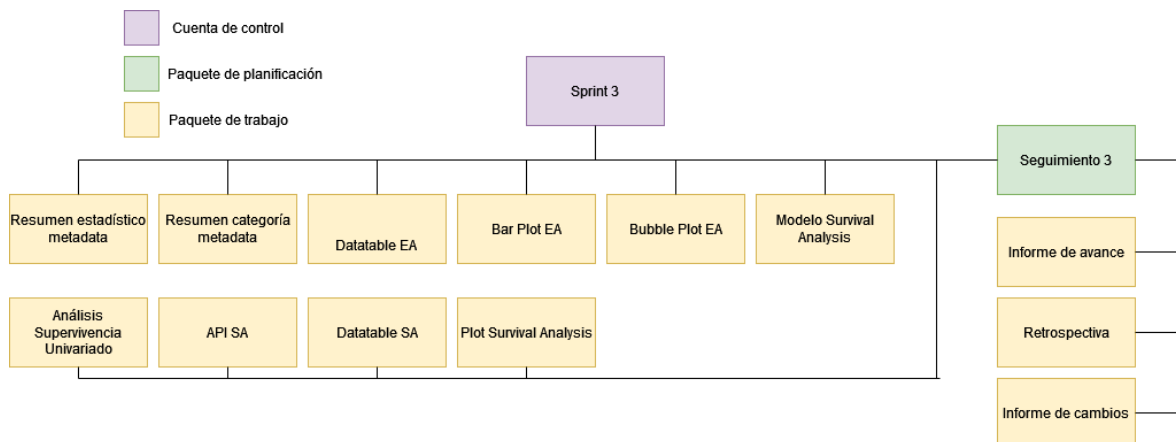


Figura 7.4: Diagrama EDT Sprint 3

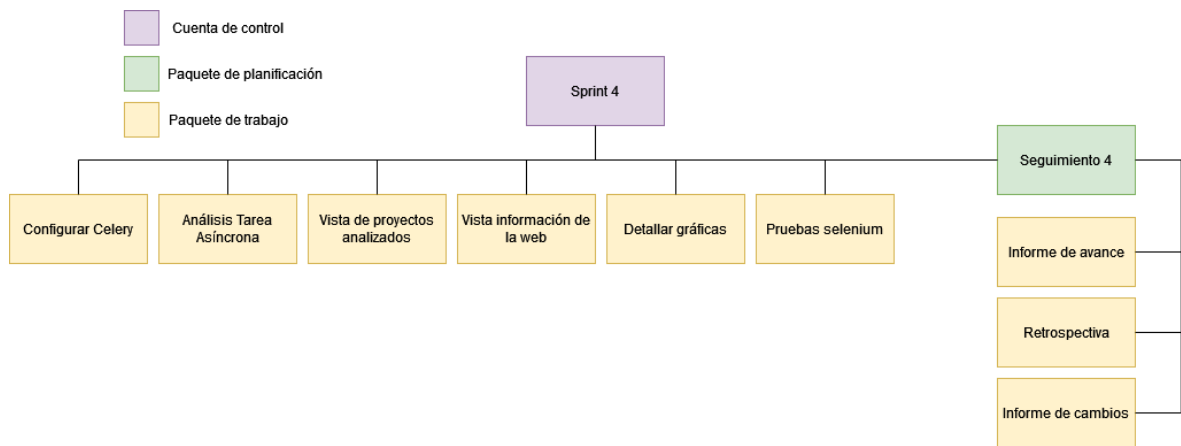


Figura 7.5: Diagrama EDT Sprint 4

Cuadro 7.2: Diccionario de la EDT - Planificación

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.1	Planificación	Fase de planificación del proyecto	lun 19/06/23	sáb 15/07/23	103 horas		3915,25 €
1.1.1	Recopilar requisitos	Estudiar el dominio del problema y elicitar los requisitos	lun 19/06/23	vie 23/06/23	24 horas	Analista	822,816 €
1.1.2	Analizar requisitos	Analizar los requisitos y como pueden ser cumplidos, así como la trazabilidad.	sáb 24/06/23	mié 28/06/23	8 horas	Analista	274,272 €
1.1.3	Arquitectura del sistema	Estudiar el tipo de arquitectura del sistema a desarrollar,	jue 29/06/23	sáb 01/07/23	7 horas	Analista	239,988 €
1.1.4	Crear EDT y diccionario	Definir las actividades y la EDT	lun 03/07/23	mié 05/07/23	15 horas	Jefe de proyecto	604,26 €
1.1.5	Identificar hitos	Identificar los hitos del trabajo.	jue 06/07/23	jue 06/07/23	5 horas	Jefe de proyecto	201,42 €
1.1.6	Estimar duración de cada tarea	Estimar los tiempos de cada tarea así como el trabajo.	vie 07/07/23	vie 07/07/23	7 horas	Jefe de proyecto	281,988 €
1.1.7	Identificar recursos y adquisiciones	Buscar herramientas y tecnologías, así como identificar los recursos del proyecto	sáb 08/07/23	lun 10/07/23	10 horas	Jefe de proyecto	402,84 €
1.1.8	Preparación del cronograma	Crear el cronograma del proyecto y tenerlo listo para el seguimiento	mar 11/07/23	mié 12/07/23	8 horas	Jefe de proyecto	322,272 €
1.1.9	Calcular costes	Estimar los costes del proyecto de los recursos.	jue 13/07/23	jue 13/07/23	8 horas	Jefe de proyecto	322,272 €
1.1.10	Identificar riesgos	Identificar los riesgos del proyecto.	vie 14/07/23	vie 14/07/23	6 horas	Jefe de proyecto	241,704 €
1.1.11	Metodología de desarrollo	Definir la metodología de desarrollo	lun 03/07/23	mie 05/07/23	5 horas	Jefe de proyecto	201,42 €

Cuadro 7.3: Diccionario de la EDT - Sprint 1

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.1	Sprint 1	Fase de la primera iteración.	lun 17/07/23	mar 01/08/23	45,68 horas	838,68 €	1244,97312 €
1.2.1.1	Crear proyecto Django	Crear el proyecto django base para el proyecto.	lun 17/07/23	lun 17/07/23	4 horas	Programador	101,136 €
1.2.1.2	Configurar base de datos PostgreSQL	Crear la base de datos para el proyecto y configurar django para su uso.	mar 18/07/23	mar 18/07/23	3,17 horas	Programador	80,15028 €
1.2.1.3	Configurar R y descargar paquete GDCRNATools	Descargar R y el paquete de GDCRNATools probando su funcionamiento.	mar 18/07/23	mar 18/07/23	1,92 horas	Programador	48,54528 €
1.2.1.4	Configurar drf-yasg	Instalar la librería de drf-yasg.	mié 19/07/23	mié 19/07/23	1,92 horas	Programador	48,54528 €
1.2.1.5	Configurar librería rpy2	Instalar la librería rpy2 probando su funcionamiento.	mié 19/07/23	jue 20/07/23	1,92 horas	Programador	48,54528 €
1.2.1.6	Plantilla base	Crear una plantilla base que será extendida con en las demás plantillas.	jue 20/07/23	jue 20/07/23	1,92 horas	Programador	48,54528 €
1.2.1.7	Barra de navegación	Crear una barra de navegación para la plantilla base.	vie 21/07/23	vie 21/07/23	1,67 horas	Programador	42,22428 €
1.2.1.8	Modelo StudyCase	Crear el modelo de StudyCase del Backend.	vie 21/07/23	sáb 22/07/23	1,08 horas	Programador	27,30672 €
1.2.1.9	Formulario StudyCase	Crear un formulario de StudyCase con los campos de datatype y project. Project debe ser un select con todos los projects disponibles de GDC.	sáb 22/07/23	lun 24/07/23	1,67 horas	Programador	42,22428 €
Continúa en la siguiente página							

Cuadro 7.3 – Continuación desde la página anterior

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.1.10	Información de proyecto	Añadir un bloque a la vista de formulario que muestre información del proyecto seleccionado.	lun 24/07/23	mar 25/07/23	2,17 horas	Programador	54,86628 €
1.2.1.11	Descarga de datos por proyecto y datatype	Implementar la función para descargar los datos según el study-Case.	mar 25/07/23	mié 26/07/23	2,58 horas	Programador	65,23272 €
1.2.1.12	Modelo Metadata	Crear el modelo de Metadata en el backend.	mié 26/07/23	mié 26/07/23	1,58 horas	Programador	39,94872 €
1.2.1.13	Obtener metadatos	Implementar la función de descarga de metadatos según el study-Case.	jue 27/07/23	jue 27/07/23	1,5 horas	Programador	37,926 €
1.2.1.14	Modelo RNAExpr	Crear el modelo RNAExpr en el backend.	jue 27/07/23	jue 27/07/23	3 horas	Programador	75,852 €
1.2.1.15	Modelo DiffExpr	Crear el modelo DiffExpr en el backend.	vie 28/07/23	vie 28/07/23	1,58 horas	Programador	39,94872 €
1.2.1.16	API StudyCase	Crear las consultas de la API para StudyCase	vie 28/07/23	vie 28/07/23	2 horas	Programador	36,72 €
1.2.1.17	API Metadata	Crear las consultas de la API para Metadata	sáb 29/07/23	sáb 29/07/23	2 horas	Programador	36,72 €
1.2.1.18	API RNAExpr	Crear las consultas de la API para RNAExpr	sáb 29/07/23	sáb 29/07/23	2 horas	Programador	36,72 €
1.2.1.19	API diffExpr	Crear las consultas de la API para diffExpr	lun 31/07/23	lun 31/07/23	2 horas	Programador	36,72 €
1.2.1.20	Seguimiento 1	Se documenta el desarrollo del sprint	lun 31/07/23	mar 01/08/23	6 horas	Jefe de proyecto	241,704 €
Continúa en la siguiente página							

Cuadro 7.3 – Continuación desde la página anterior

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.1.20.1	Informe de avance e incidencias	Se identifican las incidencias del sprint indicando las soluciones propuestas	lun 31/07/23	mar 01/08/23	2 horas	Jefe de proyecto	80,568 €
1.2.1.20.2	Retrospectiva	Puntos positivos, negativos e ideas de mejora.	lun 31/07/23	mar 01/08/23	2 horas	Jefe de proyecto	80,568 €
1.2.1.20.3	Informe de cambios	Desviaciones del desarrollo y posible replanificación	lun 31/07/23	mar 01/08/23	2 horas	Jefe de proyecto	80,568 €

Cuadro 7.4: Diccionario de la EDT - Sprint 2

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.2	Sprint 2	Fase de la segunda iteración.	mié 02/08/23	jue 17/08/23	40,34 horas		1109,96 €
1.2.2.1	Analysis DE	Implementar función de análisis de expresión diferencial.	mié 02/08/23	mié 02/08/23	2,25 horas	Programador	56,889 €
1.2.2.2	Analysis EA	Implementar función de análisis de enriquecimiento.	jue 03/08/23	jue 03/08/23	3 horas	Programador	75,852 €
1.2.2.3	API pagination	Implementar paginación en las consultas de la API que devuelven listas con mucha información.	vie 04/08/23	vie 04/08/23	3,17 horas	Programador	80,15028 €
1.2.2.4	Documentar API Swagger	Documentar las consultas de la API indicando los parámetros y posibles errores en swagger.	sáb 05/08/23	sáb 05/08/23	5,17 horas	Programador	130,71828 €
1.2.2.5	Vista de resultados	Crear un template para todos los tipos de resultados que sea fácil de navegar entre ellas.	lun 07/08/23	lun 07/08/23	3,17 horas	Programador	80,15028 €
1.2.2.6	Datatable Metadata	Mostrar los metadatos de un studyCase con una datatable.	mar 08/08/23	mar 08/08/23	2,58 horas	Programador	65,23272 €
1.2.2.7	Datatable DE	Mostrar los resultados de los análisis de expresión diferencial mediante una datatable.	mié 09/08/23	mié 09/08/23	2,25 horas	Programador	56,889 €
1.2.2.8	Volcano Plot DE	Representar un volcano plot con los resultados del análisis de expresión diferencial.	jue 10/08/23	jue 10/08/23	3,17 horas	Programador	80,15028 €
1.2.2.9	Volcano Plot DE Config	Crear menú con formulario para poder modificar los umbrales del volcano plot.	vie 11/08/23	vie 11/08/23	2,08 horas	Programador	52,59072 €
Continúa en la siguiente página							

Cuadro 7.4 – Continuación desde la página anterior

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.2.10	Bar Plot DE	Crear un gráfico de barras para los resultados del análisis de expresión diferencial	sáb 12/08/23	sáb 12/08/23	2,08 horas	Programador	52,59072 €
1.2.2.11	Correlation Genes Plot	Crear un gráfico de correlación de los valores de RNAExpr para dos genes especificados.	lun 14/08/23	lun 14/08/23	3,17 horas	Programador	80,15028 €
1.2.2.12	Descargar data-table csv	Crear una opción para poder descargar los datatables como csv.	mar 15/08/23	mar 15/08/23	2,25 horas	Programador	56,889 €
1.2.2.13	Seguimiento 2	Documentar el desarrollo del sprint	mié 16/08/23	mié 16/08/23	6 horas	Jefe de proyecto	241,704 €
1.2.2.13.1	Informe de avance e incidencias	Se identifican las incidencias del sprint indicando las soluciones propuestas	mié 16/08/23	mié 16/08/23	2 horas	Jefe de proyecto	80,568 €
1.2.2.13.2	Retrospectiva	Puntos positivos, negativos e ideas de mejora.	mié 16/08/23	mié 16/08/23	2 horas	Jefe de proyecto	80,568 €
1.2.2.13.3	Informe de cambios	Desviaciones del desarrollo y posible replanificación.	mié 16/08/23	mié 16/08/23	2 horas	Jefe de proyecto	80,568 €

Cuadro 7.5: Diccionario de la EDT - Sprint 3

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.3	Sprint 3	Fase de la tercera iteración.	vie 18/08/23	sáb 02/09/23	36,42 horas		1010,84328 €
1.2.3.1	Resumen estadístico metadata	Mostrar un resumen de los campos de metadatos que son numéricos con medias, max, min y mediana por lo menos.	vie 18/08/23	vie 18/08/23	2,17 horas	Programador	54,86628 €
1.2.3.2	Resumen categorías metadata	Hacer conteos por los valores de los campos de metadatos que son categóricos.	sáb 19/08/23	sáb 19/08/23	2,17 horas	Programador	54,86628 €
1.2.3.3	Datatable EA	Mostrar la información del análisis de enriquecimiento en una datatable.	lun 21/08/23	lun 21/08/23	2,08 horas	Programador	52,59072 €
1.2.3.4	Bar Plot EA	Representar un gráfico de barras para los resultados del análisis de enriquecimiento.	mar 22/08/23	mar 22/08/23	3,17 horas	Programador	80,15028 €
1.2.3.5	Bubble Plot EA	Representar un gráfico de burbujas para los resultados del análisis de enriquecimiento.	mié 23/08/23	mié 23/08/23	3,17 horas	Programador	80,15028 €
1.2.3.6	Modelo SurvivalAnalysis	Crear el modelo de SurvivalAnalysis para el backend.	jue 24/08/23	jue 24/08/23	3,17 horas	Programador	80,15028 €
1.2.3.7	Análisis Supervivencia univariado	Implementar la función de análisis de supervivencia univariada.	vie 25/08/23	vie 25/08/23	4,33 horas	Programador	109,47972 €
1.2.3.8	API SA	Crear las consultas de la API para los resultados del survival análisis.	sáb 26/08/23	sáb 26/08/23	3,83 horas	Programador	96,83772 €
Continúa en la siguiente página							

Cuadro 7.5 – Continuación desde la página anterior

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.3.9	Datatable SA	Mostrar los resultados del análisis de supervivencia en una data-table.	lun 28/08/23	lun 28/08/23	2,08 horas	Programador	52,59072 €
1.2.3.10	Plot Survival Analysis	Representar el análisis de supervivencia KM para un gen específico.	mar 29/08/23	mar 29/08/23	4,25 horas	Programador	107,457 €
1.2.3.11	Seguimiento 3	Documentar el desarrollo.	mié 30/08/23	mié 30/08/23	6 horas	Jefe de proyecto	241,704 €
1.2.3.11.1	Informe de avance e incidencias	Se identifican las incidencias del sprint indicando las soluciones propuestas	mié 30/08/23	mié 30/08/23	2 horas	Jefe de proyecto	80,568 €
1.2.3.11.2	Retrospectiva	Puntos positivos, negativos e ideas de mejora.	mié 30/08/23	mié 30/08/23	2 horas	Jefe de proyecto	80,568 €
1.2.3.11.3	Informe de cambios	Desviaciones del desarrollo y posible replanificación.	mié 30/08/23	mié 30/08/23	2 horas	Jefe de proyecto	80,568 €

Cuadro 7.6: Diccionario de la EDT - Sprint 4

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.4	Sprint 4	Desarrollo de la cuarta iteración.	lun 04/09/23	mar 19/09/23	28,5 horas		810,594 €
1.2.4.1	Configurar Celery	Configuración de Celery y Redis con Django.	lun 04/09/23	lun 04/09/23	5 horas	Programador	126,42 €
1.2.4.2	Análisis tarea asíncrona	Declarar la función de análisis y descarga como asíncrona con Celery	mar 05/09/23	mar 05/09/23	4 horas	Programador	101,136 €
1.2.4.3	Vista de proyectos analizados	Implementar una vista con los proyectos analizados con un filtro.	mié 06/09/23	mié 06/09/23	2,5 horas	Programador	63,21 €
1.2.4.4	Vista web information	Vista de información sobre la aplicación.	lun 11/09/23	lun 11/09/23	3 horas	Programador	75,852 €
1.2.4.5	Detallar gráficas resultado	Definir todos los resultados en la aplicación y menús.	mar 12/09/23	mar 12/09/23	3 horas	Programador	75,852 €
1.2.4.6	Pruebas Selenium	Hacer pruebas Selenium de la aplicación.	mié 13/09/23	mié 13/09/23	5 horas	Programador	126,42 €
1.2.4.7	Seguimiento 4	Documentar el desarrollo	jue 14/09/23	jue 14/09/23	6 horas	Jefe de proyecto	241,704 €
1.2.4.7.1	Informe de avance e incidencias	Se identifican las incidencias del sprint indicando las soluciones propuestas	jue 14/09/23	jue 14/09/23	2 horas	Jefe de proyecto	80,568 €
1.2.4.7.2	Retrospectiva	Puntos positivos, negativos e ideas de mejora.	jue 14/09/23	jue 14/09/23	2 horas	Jefe de proyecto	80,568 €
1.2.4.7.3	Informe de cambios	Desviaciones del desarrollo y posible replanificación.	jue 14/09/23	jue 14/09/23	2 horas	Jefe de proyecto	80,568 €

Cuadro 7.7: Diccionario de la EDT - Cierre

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.3	Cierre	Fase final del proyecto	mié 20/09/23	sáb 30/09/23	46,66 horas		1879,65144 €
1.3.1	Lecciones aprendidas	Documentar las lecciones aprendidas durante el desarrollo del proyecto.	mié 20/09/23	mié 20/09/23	6 horas	Jefe de proyecto	241,704 €
1.3.2	Manual de usuario	Hacer un manual de instalación y de usuario.	vie 22/09/23	lun 25/09/23	10 horas	Jefe de proyecto	402,84 €
1.3.3	Finalización de la memoria	Completar la memoria y revisarla para su finalización.	sáb 23/09/23	mié 27/09/23	15,33 horas	Jefe de proyecto	617,55372 €
1.3.4	Preparación de la presentación	Preparar las diapositivas y la presentación oral de la memoria.	jue 28/09/23	sáb 30/09/23	15,33 horas	Jefe de proyecto	617,55372 €

7.5. Cronograma

Se utilizará la información proporcionada en el Diccionario de la Estructura de Desglose del Trabajo (EDT) para crear un cronograma utilizando la herramienta Microsoft Project. Este cronograma nos permitirá visualizar un diagrama de Gantt y establecer una línea de base que será utilizada para comparar el progreso del proyecto durante su ejecución.

En las Figuras 7.6, 7.7, 7.8, 7.9, 7.10 podéis ver el diagrama gantt generado por el cronograma con la herramienta MSPProject.

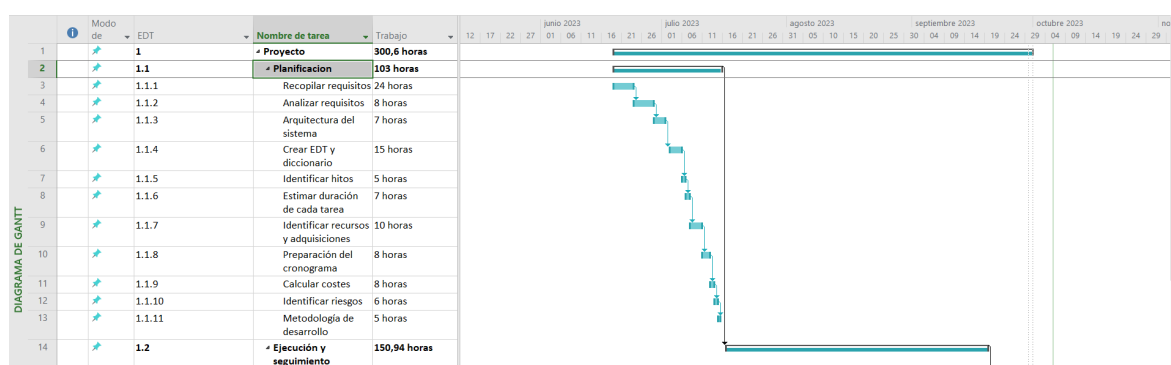


Figura 7.6: Diagrama Gantt - parte 1

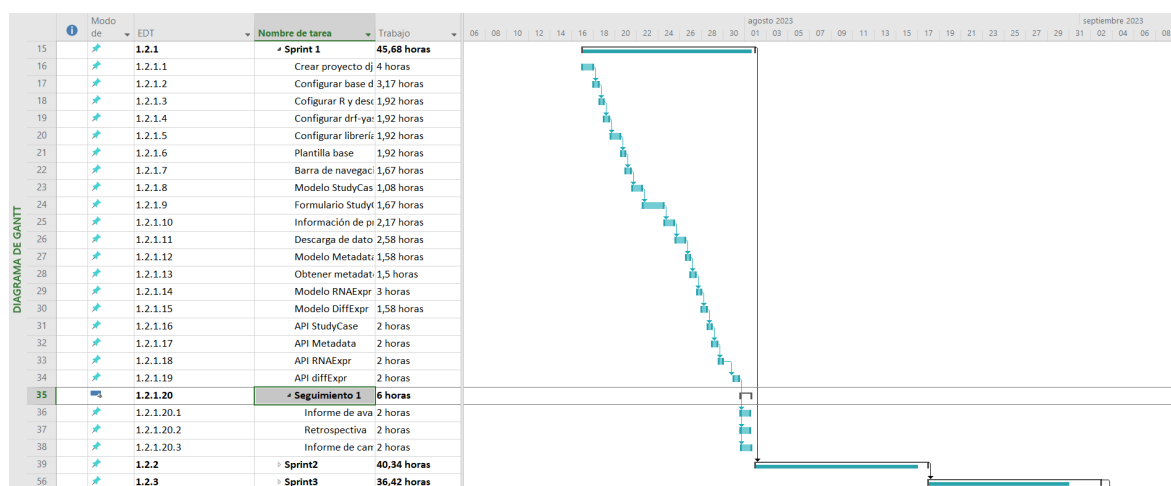


Figura 7.7: Diagrama Gantt - parte 2

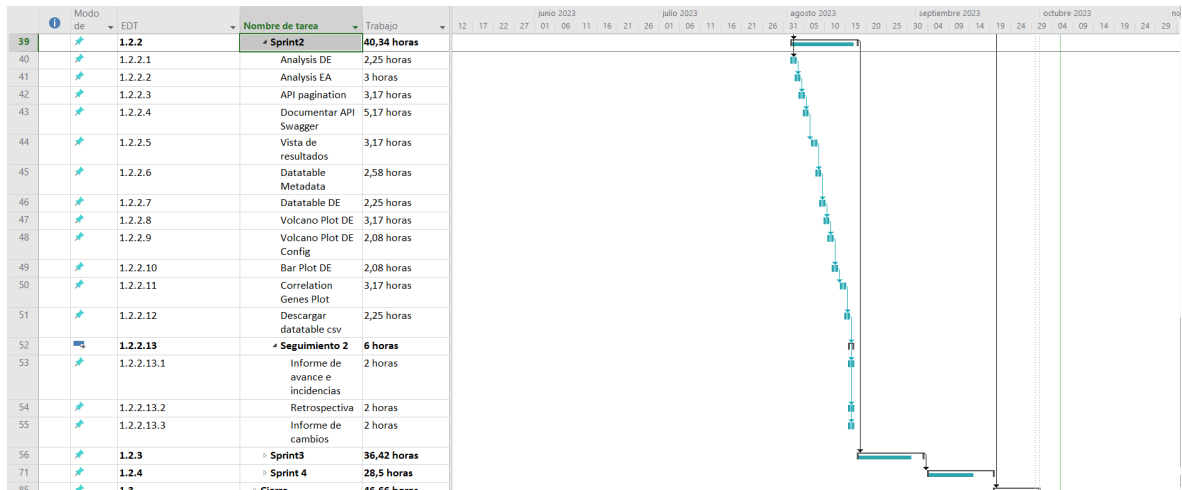


Figura 7.8: Diagrama Gantt - parte 3

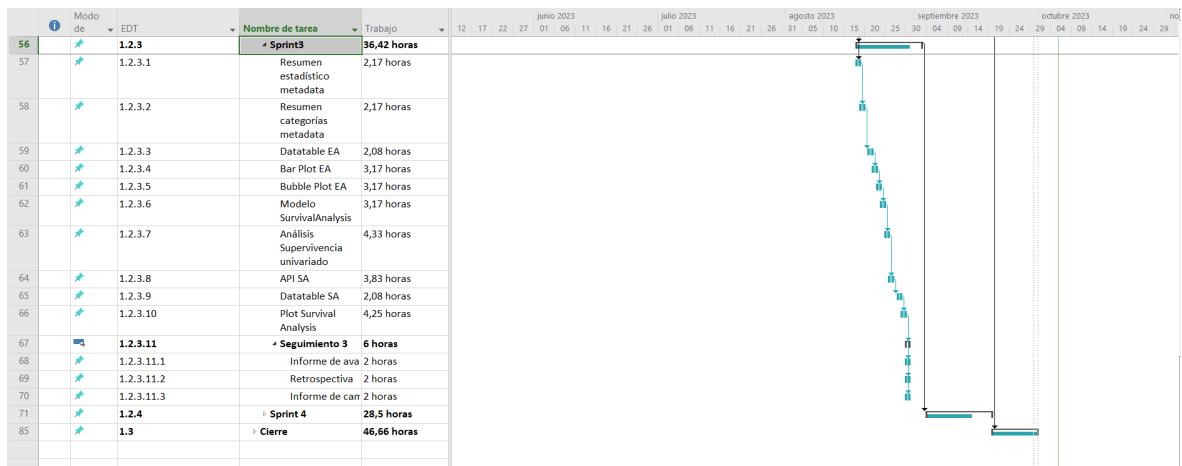


Figura 7.9: Diagrama Gantt - parte 4

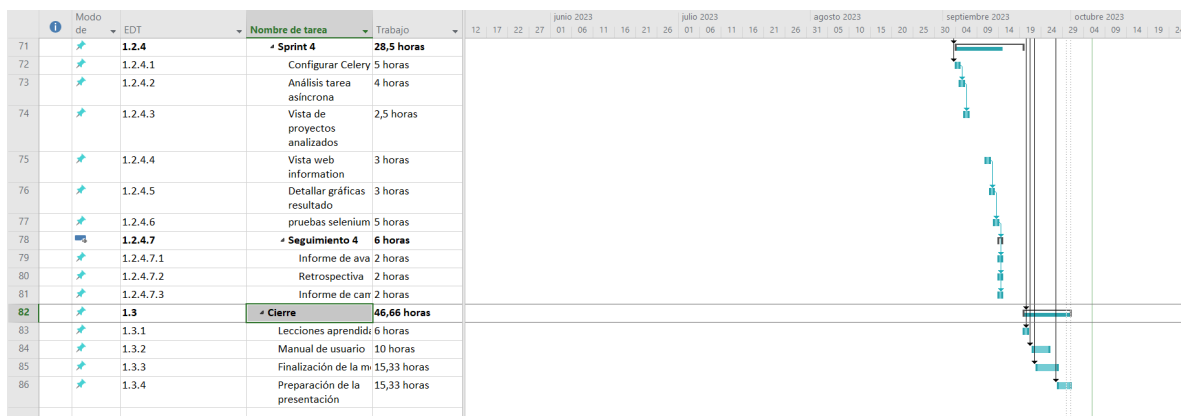


Figura 7.10: Diagrama Gantt - parte 5

8. Costes

En esta sección se desglosan los costes estimados del proyecto dando un presupuesto.

8.1. Coste del personal

Son los costes relacionados con salarios de los empleados y colaboradores que participaran en el proyecto.

En el Cuadro 8.1 se especifican los costes del personal del proyecto. Se definen diferentes roles dentro del proyecto que tendrán responsabilidad en distintas tareas. Estos roles son:

- Jefe de proyecto: Es el responsable de gestionar el progreso del proyecto y se encargará de las tareas tales como planificación, gestión de costes, de riesgos, documentar el seguimiento y la fase de cierre.
- Analista: Será responsable de analizar los requisitos, así como del diseño del producto a desarrollar.
- Programador: Su principal responsabilidad será el desarrollo del software a lo largo de los sprints.

Los costes por hora de los distintos roles se han obtenido de un documento almacenado en la web de Trabajo de Fin de Grado de la E.T.S.I.I. llamado “informe-precios_perfiles_informáticos.pdf”.

Cuadro 8.1: Coste del personal

Rol	Coste/Hora	Horas totales	Coste total
Jefe de proyecto	40 €	134,66	5386,4 €
Analista	34 €	39	1326 €
Programador	25 €	126,94	3173,5 €
		300.6	9885,9 €

8.2. Coste material

El costo material en un proyecto de software se refiere a los gastos relacionados con los recursos físicos y tangibles necesarios. Dentro de estos costos se puede diferenciar: hardware, software, impresoras, mobiliario de oficina entre otros.

En el Cuadro 8.2 se define el coste material del proyecto. El coste por hora del portátil es resultado de amortizarlo a 4 años con 8 horas de uso de lunes a viernes.

Cuadro 8.2: Coste material

Nombre	Coste total	Coste/Hora	Horas totales	Coste total
Portatil	700	0,084 €	300,66	25,56 €

8.3. Coste operacional

Los costes de operacion se definen como aquellos costes necesarios para mantener y operar un negocio en su funcionamiento diario. El único gasto que definimos (ver Cuadro 8.3) de este tipo para este proyecto se trata del consumo de electricidad del portatil. El precio de Kwh es una aproximación de lo que cuesta en España[25].

Cuadro 8.3: Coste operacional

Nombre	Unidad	Coste/Unidad	Horas totales	Coste total
Electricidad	Kwh	0,2 €	300.66	60.13 €

8.4. Presupuesto final

En conclusión el coste estimado final viene dado por la suma de los costes totales de los Cuadros 8.1, 8.2 y 8.3 más una reserva de contingencia del 10 %. En el Cuadro 8.4 se resume el presupuesto.

Cuadro 8.4: Presupuesto final

	Coste (€)
Personal	9885,9
Material	25,56
Operacional	60,13
Suma Costes	9971,59
Contingencia (10 %)	997,16
Presupuesto	10968,75

El presupuesto final sería de 10968,75 euros.

Parte V

SEGUIMIENTO

9. Introducción del seguimiento

En esta parte del documento se describe el desarrollo del proyecto a lo largo de los sprints. Para cada sprint se describirá las incidencias que ha sufrido el desarrollo indicando las soluciones aportadas, las desviaciones respecto a la planificación inicial descrita en el diccionario de la EDT (ver Capítulo [7.4](#)), los riesgos que se han identificado durante el desarrollo y las estrategias a seguir según su impacto, la retrospectiva del sprint y por último una replanificación del sprint siguiente.

10. Sprint 1

Vamos a describir a continuación como ha sido el desarrollo durante el Sprint 1.

10.1. Incidencias del desarrollo

A continuación se describen las incidencias que se ha encontrado durante el desarrollo y las soluciones implantadas (ver Cuadro 10.1).

Cuadro 10.1: Incidencias - Sprint 1

EDT	Nombre	Descripción del problema	Solución
1.2.1.5	Configurar librería rpy2	A la hora de probar la librería no encontraba salía un error en la importación.	Se encuentra el mismo problema por internet y una solución de un usuario de github[11].
1.2.1.7	Barra de navegación	Se implementó primero una barra de navegación “sidenav” pero no era responsive	Cambio del diseño por una barra de navegación clásica con bootstrap.

10.2. Desviaciones

En esta subsección se describen las desviaciones que ha sufrido el proyecto, tanto positivas como las negativas, dando una explicación.

Variación del coste

- Variación: 14,87 %
- Coste previsto: 1244,97 €
- Coste real: 1430,05 €
- Coste acumulado previsto: 5160,22 €
- Coste acumulado real: 5345,30 €
- Justificación: El coste ha aumentado debido a que las horas de trabajo también han aumentado y con ello el uso de los recursos.

Variación del trabajo

- Variación: 16,02 %

- Trabajo real: 53 h
- Trabajo previsto: 45,68 h
- Justificación: Las tareas han supuesto más trabajo del previsto debido principalmente al desconocimiento del uso de algunas tecnologías y buscar documentación constantemente, así como los errores provocados por el desconocimiento.

Completitud

- Porcentaje de completitud: 100 %
- Tareas pendientes: No procede.
- Justificación: Se han completado todas las tareas propuestas para el sprint.

10.3. Riesgos

Aquí se especifican los riesgos que han surgido durante el sprint. Los riesgos son:

- RIE-002
 - Descripción: Estimación del trabajo incorrecta.
 - Análisis: Según lo visto en el apartado de variaciones, el trabajo real ha sido mayor en un 16 % del trabajo previsto provocando un aumento del coste de 185,08 €, sin embargo, no ha habido variación en la fecha de finalización.
 - Estrategia: El impacto solo ha sido sobre el coste pero no supone una variación respecto al presupuesto gracias a la reserva de contingencia de 997,16 € que se estimó. Teniendo esto en cuenta para los siguientes sprints, la variación de coste aceptada para mantenernos en el presupuesto se quedaría en 812 € aproximadamente.

10.4. Retrospectiva

En la retrospectiva se describen los puntos positivos, negativos y se proponen ideas de mejora para el siguiente sprint.

Positivo

- Las fechas de inicio y fin de las tareas tenían cierta holgura lo que ha permitido más flexibilidad en el trabajo.
- Se ha conseguido realizar todas las tareas previstas.
- Buena elección de herramientas destacando GDRCRNATools que ha facilitado la descarga y el análisis de los datos.

- Las incidencias y problemas se han resuelto de forma ágil.

Negativo

- Se han subestimado las tareas provocando un mayor trabajo del previsto.
- El horario de trabajo no ha sido constante. Hay días en los que se ha trabajado mucho debido a que otros no se ha avanzado.

Ideas de mejora

- Proponer un horario de trabajo constante. De 11:00 a 13:00 y de 16:00 a 19:30. Esto es una primera idea que puede cambiar si no es del todo cómodo.

10.5. Replanificación

El sprint 2 sigue como lo planeado ya que no se han quedado tareas pendientes del sprint 1.

11. Sprint 2

Vamos a describir a continuación como ha sido el desarrollo durante el Sprint 2.

11.1. Incidencias del desarrollo

A continuación se describen las incidencias que se ha encontrado durante el desarrollo y las soluciones implantadas (ver Cuadro 11.1).

Cuadro 11.1: Incidencias - Sprint 2

ID	EDT	Nombre	Descripción del problema	Solución
IN-2.1	1.2.2.2	Analysis EA	El método para realizar el análisis de enriquecimiento no es funcional para los datos miRNAs.	Se ha omitido dicho análisis si el datatype es miRNAs.
IN-2.2	1.2.2.2	Analysis EA	El análisis tarda bastante y es ineficiente realizarlo para todos los genes del studyCase	Se ha prefiltrado los genes a actualizar tras el análisis DE para tomar los genes más significativos.
IN-2.3	1.2.2.8	Volcano Plot	Problemas con responsiveness de los gráficos	Se activa la opción de responsive y se actualiza las dimensiones del gráfico al mostrar los resultados de la clase de bootstrap “accordion”.

11.2. Desviaciones

En esta subsección se describen las desviaciones que ha sufrido el proyecto, tanto positivas como las negativas, dando una explicación.

Variación del coste

- Variación: 2,6 %
- Coste previsto: 1109,96 €
- Coste real: 1139,29 €
- Coste acumulado previsto: 6270,18 €

- Coste acumulado real: 6484,59 €
- Justificación: El coste ha aumentado debido a que las horas de trabajo también han aumentado y con ello el uso de los recursos.

Variación del trabajo

- Variación: 2,8
- Trabajo previsto: 40,34 h
- Trabajo real: 41,5 h
- Justificación: La variación es muy poca lo cual es positivo, sin embargo, esta pequeña variación se debe también a que se han dejado incompleta una tarea..

Compleitud

- Porcentaje de completitud: 93,4
- Tareas pendientes: 1.2.2.12 Descargar Datatable CSV.
- Justificación: Se ha decidido aplazar una tarea con el fin de no retrasar el fin del sprint 2. Aun así el porcentaje de completitud es mayor al 90 %.

11.3. Riesgos

Aquí se especifican los riesgos que han surgido durante el sprint. Los riesgos son:

- **RIE-003**
 - Descripción: Requisito no identificado.
 - Análisis: Debido al incidente IN-2.1 se ve necesario crear un requisito no funcional de usabilidad. (RF-024)
 - Estrategia: Para el requisito se ha creado un paquete de trabajo para el sprint 3 “Ocultar resultados EA para miRNAs” con una estimación de 1 h.
- **RIE-005**
 - Descripción: Tareas incompletas
 - Análisis: Ha quedado sin completar una tarea en este sprint. La tarea pospuesta es la 1.2.2.12 que estaba estimada con 2,25 horas.
 - Estrategia: La tarea se realizará durante el sprint 3. No hay ninguna tarea que dependa de esta por lo que se podrá hacer al principio o al final según se vea conveniente.
- **RIE-006**

- Descripción: Desconocimiento de la tecnología.
- Análisis: Plotly js es una tecnología con la que no tenía experiencia previa el desarrollador. Sin embargo, la estimación de las tareas preveyendo esta situación ha mitigado los retrasos.
- Estrategia: Con los gráficos realizados en este sprint ya se ha adquirido cierta soltura y no es necesario tomar medidas para los siguientes sprints.

11.4. Retrospectiva

En la retrospectiva se describen los puntos positivos, negativos y se proponen ideas de mejora para el siguiente sprint.

Positivo

- Aunque no se ha completado el 100 %, solo queda una tarea incompleta lo cual se considera positivo.
- La desviación en el coste y en el trabajo no es significativa.
- La tecnología escogida para la realización de gráficos, no solo es bastante sencilla, sino que además aporta valor a la aplicación dando la funcionalidad de descargar como png.
- Se sobreestimaron las tareas que usaban tecnologías nuevas teniendo en cuenta la inexperiencia lo cual ha sido acertado.
- El desarrollador se está acostumbrando al horario de trabajo propuesto en la anterior retrospectiva y está siendo eficiente.

Negativo

- No se consideró utilizar framework de frontend como react lo cual empeora la mantenibilidad del código javascript.
- Las estimaciones de tareas de análisis son bastante tardías sobre todo para hacer comprobaciones y pueden llevar retrasos.

Ideas de mejora

- A mitad del desarrollo no se considera implementar un framework de frontend como react aunque se propone que se vaya comentando el código javascript con el fin de mejor mantenibilidad.

11.5. Replanificación

Para el sprint 3 se añaden los siguientes paquetes de trabajo (ver Cuadro 11.2) aparte de los ya asignados en el Cuadro 7.5.

Cuadro 11.2: Replanificación Sprint 3

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.2.12	Descargar Datatable CSV	Descargar la información de los Datatables como CSV	vie 18/08/23	mar 29/08/23	2,25 h	Programador	56,889 €
1.2.3.12	Ocultar resultados EA para miRNAs	Hacer que los botones para acceder a la vista de resultados de enriquecimiento para los proyectos de tipo miRNAs se oculte a los usuarios.	vie 18/08/23	mar 29/08/23	1 h	Programador	25,286 €

12. Sprint 3

Vamos a describir a continuación como ha sido el desarrollo durante el Sprint 3.

12.1. Incidencias del desarrollo

A continuación se describen las incidencias que se ha encontrado durante el desarrollo y las soluciones implantadas (ver Cuadro [12.1](#)).

Cuadro 12.1: Incidencias - Sprint 3

EDT	Nombre	Descripción del problema	Solución
1.2.3.3	Datatable EA	El campo de listado de genes y simbolos del análisis de enriquecimiento es demasiado largo para mostrarlo.	Ocultar la columna de dicho campo en el datatable de forma que no se vea en la vista pero al descargarlo si esté.
1.2.3.7	Análisis de supervivencia2	El análisis proporcionado por GDCRNATools no da los resultados de la función de supervivencia necesaria para el plot de supervivencia.	Se implementa el método de cálculo de función de supervivencia con la librería lifelines de python.
1.2.3.5	Bubble plot	El tamaño de las burbujas no escala correctamente.	Se ha hecho una función que devuelve el tamaño de las burbujas bien diferenciado sin tener en cuenta los genes que no aparecen en el plot.

12.2. Desviaciones

En esta subsección se describen las desviaciones que ha sufrido el proyecto, tanto positivas como las negativas, dando una explicación.

Variación del coste

- Variación: -10,88 %
- Coste previsto: 1010,84 €
- Coste real: 911,73 €
- Coste acumulado previsto: 7281,02 €
- Coste acumulado real: 7396,32 €

- Justificación: El coste en esta ocasión ha disminuido debido al menor uso de los recursos por el trabajo sobreestimado.

Variación del trabajo

- Variación: -10,61 %
- Trabajo real: 32,5 h
- Trabajo previsto: 36,42 h
- Justificación: En este caso el sprint fue un poco sobreestimado ya que las tareas se han conseguido completar, en general, con menos tiempo del estimado. Esta variación negativa supone un alivio para el presupuesto ya que los dos anteriores sprints tuvieron mayor costo del estimado.

Completitud

- Porcentaje de completitud: 93 %
- Tareas pendientes: 1.2.3.10 - Plot Survival Analysis.
- Justificación: Se han completado todas las tareas propuestas para el sprint.

12.3. Riesgos

Aquí se especifican los riesgos que han surgido durante el sprint. Los riesgos son:

- RIE-005
 - Descripción: Tareas incompletas
 - Análisis: Se ha quedado una tarea sin completar la cual supone un retraso estimado de 4,25.
 - Estrategia: La tarea será pospuesta para el inicio del sprint 4. No se vé necesario aumentar la duración del sprint.
- RIE-003
 - Descripción: Requisito no identificado.
 - Análisis: Requisitos añadidos.
 - RN-007 — StudyCase analizándose
 - RNF-008 — Select con search
 - Requisito modificado: RI-001 — Se añade atributo state.
 - Estrategia: Se ha creado los siguientes paquetes de trabajo:

- 1.2.4.8 — Detalle de proyecto: estado — Actualizar la vista de listado de proyectos para mostrar el estado.
- 1.2.4.9 — Actualizar estado — Modificar la función de análisis y para que se vaya actualizando el estado de los StudyCases.
- 1.2.4.10 — Select2 Implementar — Implementar el select2 que es una librería de js para hacer selects con search.

El trabajo estimado de estos paquetes de trabajo en total suman 4,5 horas.

12.4. Retrospectiva

En la retrospectiva se describen los puntos positivos, negativos y se proponen ideas de mejora para el siguiente sprint.

Positivo

- Agilidad a la hora de resolver errores que suponían un posible cambio en los requisitos funcionales.
- Compromiso del desarrollador a la completitud de las tareas.

Negativo

- El horario propuesto en el sprint 1 no ha sido seguido con regularidad. Provocando que días fueran de más trabajo y días con menos.
- El desarrollador ha sufrido más estrés a lo largo de este sprint que comparado a los anteriores.

Ideas de mejora

- Se propone acudir a la biblioteca del CRAI para concentrarse más en el trabajo.

12.5. Replanificación

Para el sprint 4 se añaden los siguientes paquetes de trabajo (ver Cuadro [12.2](#)) aparte de los ya asignados en la Cuadro [7.6](#).

Cuadro 12.2: Replanificación Sprint 4

EDT	Título	Descripción	Inicio	Fin	Trabajo	Responsable	Coste
1.2.3.10	Plot Survival Analysis	Representar el gráfico de supervivencia KM para un gen específico.	lun 04/09/23	mar 05/10/23	4,25	Programador	107,157 €
1.2.4.8	Detalle de proyectos: Estado	Actualizar la vista de detalles de proyectos con el estado de los proyectos.	jue 14/09/23	jue 14/09/23	1,5	Programador	37,926 €
1.2.4.9	Actualizar estado	Modificar la función de análisis y para que se vaya actualizando el estado de los StudyCases	vie 15/09/23	sab 16/09/23	2	Programador	50,568 €
1.2.4.10	Select2 Implementar	Implementar el select2 que es una librería de js para hacer selects con search.	sab 15/09/23	sab 16/09/23	1	Programador	25,286 €

13. Sprint 4

Vamos a describir a continuación como ha sido el desarrollo durante el Sprint 4.

13.1. Incidencias del desarrollo

A continuación se describen las incidencias que se ha encontrado durante el desarrollo y las soluciones implantadas (ver Cuadro 13.1).

Cuadro 13.1: Incidencias - Sprint 4

EDT	Nombre	Descripción del problema	Solución
1.2.4.9	Pruebas selenium	Realizar las pruebas selenium han presentado complejidad debido al dinamismo de la páginas con js.	Se ha usado la extensión del navegador Selenium IDE para la realización de los tests.
1.2.4.10	Select2 Implementer	El desplegable del select y el botón de envío de formulario se superponían, así que al clickar en la opción se clickaba sin querer en el botón de análisis.	Se cambia la estructura del diseño del formulario de análisis para que no se superponga.
1.2.4.8	Detalle de proyectos: Estado	Al añadir el campo state a Study-Case hubo problemas de migración ya que se borraron los archivos de makemigrations.	Se borra y crea de nuevo la base de datos.

13.2. Desviaciones

En esta subsección se describen las desviaciones que ha sufrido el proyecto, tanto positivas como las negativas, dando una explicación.

Variación del coste

- Variación: 46,78 %
- Coste previsto: 810,594 €
- Coste real: 1189,854 €
- Coste acumulado previsto: 8091,616 €
- Coste acumulado real: 8586,17 €
- Justificación: El coste en esta ocasión ha sido bastante superior al estimado. Esto se debe al aumento de horas de trabajo de los recursos.

Variación del trabajo

- Variación: 31,57 %
- Trabajo real: 37,5 h
- Trabajo previsto: 28,5 h
- Justificación: Ha habido una variación significativa del trabajo causada principalmente por tareas retrasadas y nuevos requisitos que surgieron del sprint anterior.

Compleitud

- Porcentaje de completitud: 100 %
- Tareas pendientes: No procede
- Justificación: Se han completado todas las tareas propuestas para el sprint.

13.2.1. Fecha de fin

- Fecha de fin prevista: mar 19/09/23
- Fecha de fin real: vie 22/09/23
- Justificación: El sprint ha supuesto más trabajo del considerado y como era el último se ha creído oportuno aumentar la duración de este hasta el viernes 22.

13.3. Riesgos

Aquí se especifican los riesgos que han surgido durante el sprint. Los riesgos son:

- RIE-002
 - Descripción: Estimación de trabajo incorrecta.
 - Análisis: Se han subestimado las tareas. El trabajo realizado ha sido mayor al previsto provocando un mayor coste. El aumento del coste ha sido de 379,26 €. La diferencia del coste acumulado es de 494,56 €.
 - Estrategia: El coste adicional está suficientemente cubierto por la reserva de contingencia por lo cual no es necesario tomar medidas adicionales.
- RIE-007
 - Descripción: Retraso en fases del proyecto.
 - Análisis: La fecha de fin del sprint se ha extendido hasta el viernes 22 de septiembre, dejando menos tiempo para la fase de cierre.

- Estrategia: Aún se considera que hay tiempo suficiente para la fase de cierre por el hito de está no será modificado.

13.4. Retrospectiva

En la retrospectiva se describen los puntos positivos, negativos y se proponen ideas de mejora para el siguiente sprint.

Positivo

- El retraso del sprint es de poco tiempo, por lo que no se cree que retrase la entrega.
- El plan sugerido en el sprint 3 de ir a la biblioteca ha sido favorable para la concentración y motivación del desarrollador.
- Se han completado todas las tareas propuestas.

Negativo

- Aunque el plan de ir a la biblioteca aumenta la motivación, también puede suponer más cansancio en el desarrollador ya que tiene que levantarse más temprano e ir en autobús.
- Otras actividades se han visto afectada por el exceso de trabajo, como por ejemplo ir al gimnasio o salir con amigos.

Ideas de mejora

- No procede.

13.5. Replanificación

El desarrollo ya ha acabado. Lo único restante es la fase de cierre donde se completará la documentación a entregar. La planificación sigue igual para el resto del proyecto.

Parte VI

CIERRE

14. Aplicación final

La aplicación web desarrollada toma el nombre de “**GenCancer Analyzer**”.

A continuación se detallará se detallará la aplicación mediante el manual de instalación, el manual de usuario y las pruebas.

14.1. Manual de instalación

El manual de instalación describe como instalar la aplicación y las dependencias de esta para poder realizar el despliegue en local.

1. **Descargar aplicación:** Accede al [repositorio de github](#) del proyecto.
2. **Descargar python 3.10.11** desde la [página oficial](#).
3. **Descargar R 4.3.1** desde el [sitio oficial de R](#) .
4. Se recomienda crear un **entorno virtual** de Python para no tener conflictos con las dependencias. Para crear el entorno virtual desde la consola:

Extracto de código 14.1: Comandos entorno virtual

```
# Crea un entorno virtual
python3.10 -m venv myenv

# Activa el entorno virtual (en sistemas Unix)
source myenv/bin/activate

# Activa el entorno virtual (en sistemas windows)
myenv\Scripts\activate

#Para desactivarlo basta con poner
deactivate
```

5. Instalar **dependencias de pip**: En la carpeta del proyecto se encuentra un archivo llamado “requirements.txt” son las dependencias que se deben instalar. Para hacerlo abriendo una ventana de comandos desde el directorio donde se encuentra ese archivo:

Extracto de código 14.2: Comando instalar requirements

```
pip install -r requirements.txt
```

6. Instalación de **paquete GDCRNATools**: Es un paquete de R que proporciona los principales métodos de análisis de la aplicación. Para instalarlo abre R y pon el comando:

```

Extracto de código 14.3: Comandos descarga GDCRNATools
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("GDCRNATools")

```

7. **Instalar PostgreSQL:** La base de datos usada es postgresql. Deberás descargarla desde el [sitio oficial](#) A lo largo de la instalación os pedirá una contraseña para el super-usuario. Es importante que la recordéis.
8. **Configurar base de datos;** Para crear la base de datos utilizaremos pgAdmin. Quizás con el paso anterior ya se haya descargado, en caso contrario podéis hacerlo desde el [sitio oficial](#).

Ahora empezaremos a configurar la base de datos. Durante el proceso es probable que os pida la contraseña que escribisteis al instalar PostgreSQL.

- a) Una vez instalado pgAdmin lo abrimos y empezamos creando un usuario. Desplegando el menú de la izquierda y haciendo clic derecho en “login/group roles” llegaremos a la información de la Figura 14.1.

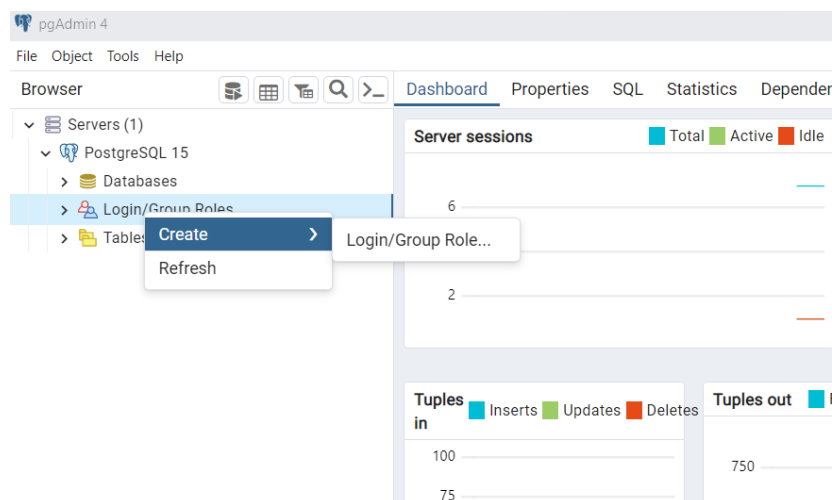


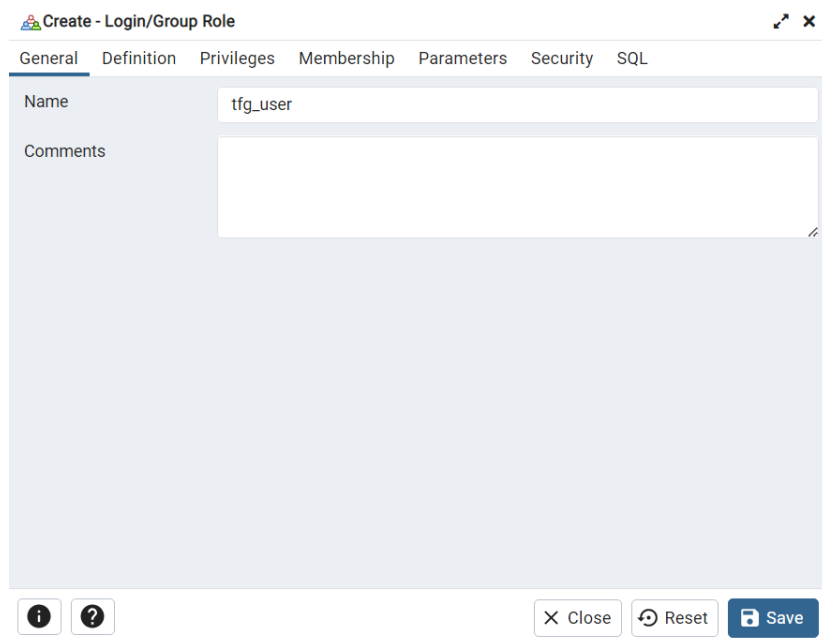
Figura 14.1: Configuración pgAdmin - 1

Haciendo click en “create>Login/Group Role” llegaremos a la Figura 14.2. Deberemos ponerle un nombre al usuario, por ejemplo “tfg_user”.

En la pestaña de “Definition” deberemos poner una contraseña del usuario, por ejemplo “tfg_password” (ver Figura 14.3).

Por último, daremos privilegios a dicho usuario (ver Figura 14.4) y haremos clic en “Save” para guardarlo.

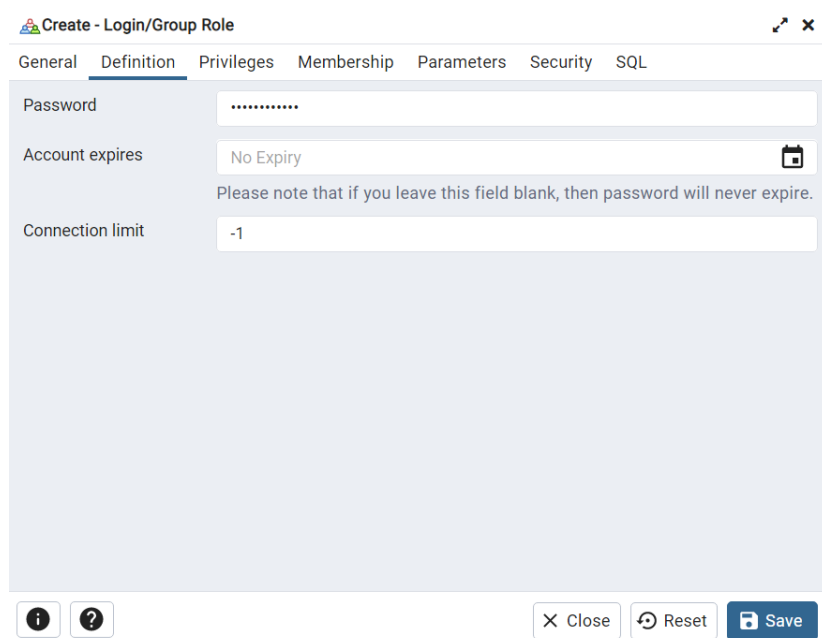
- b) Crear la base de datos: Parecido al paso anterior para crear la base de datos haremos clic derecho en “Databases”, clic izquierdo en “create>Database...” (ver Figura 14.5).



The image shows the 'Create - Login/Group Role' dialog box in pgAdmin, with the 'General' tab selected. The 'Name' field contains 'tfg_user'. The 'Comments' field is empty. At the bottom, there are buttons for 'Close', 'Reset', and 'Save', along with information and help icons.

Field	Value
Name	tfg_user
Comments	

Figura 14.2: Configuración pgAdmin - 2



The image shows the 'Create - Login/Group Role' dialog box in pgAdmin, with the 'Definition' tab selected. The 'Password' field is masked with dots. The 'Account expires' field is set to 'No Expiry'. The 'Connection limit' field is set to '-1'. A note states: 'Please note that if you leave this field blank, then password will never expire.' At the bottom, there are buttons for 'Close', 'Reset', and 'Save', along with information and help icons.

Field	Value
Password
Account expires	No Expiry
Connection limit	-1

Figura 14.3: Configuración pgAdmin - 3

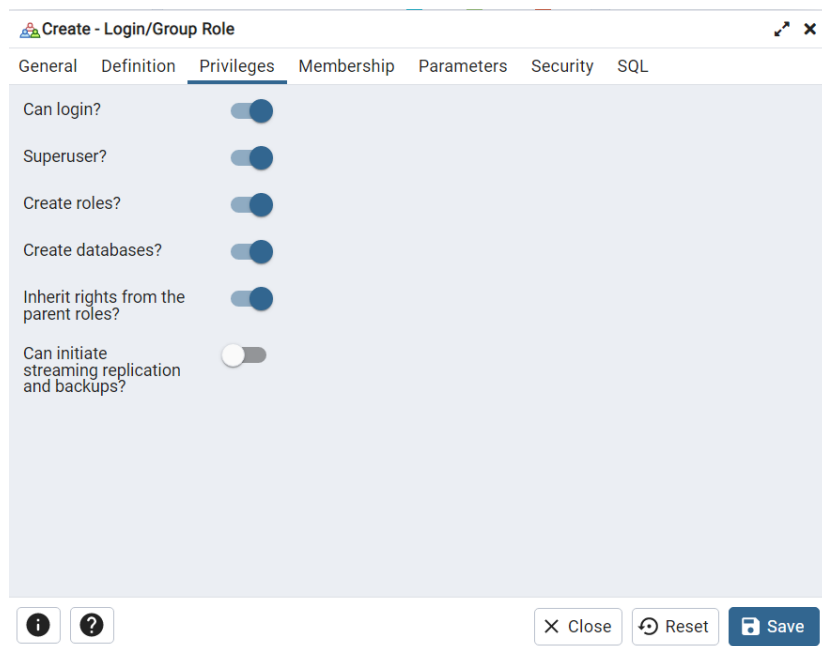


Figura 14.4: Configuración pgAdmin - 4

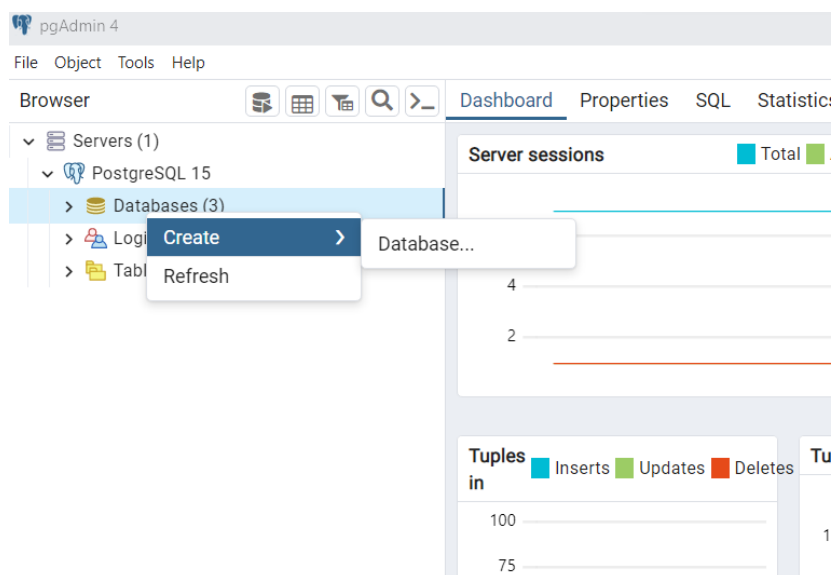


Figura 14.5: Configuración pgAdmin - 5

Le daremos un nombre a la base de datos, por ejemplo “tfg_db” y le asignaremos el usuario creado en el paso anterior, en mi caso “tfg_user” (ver Figura 14.6).

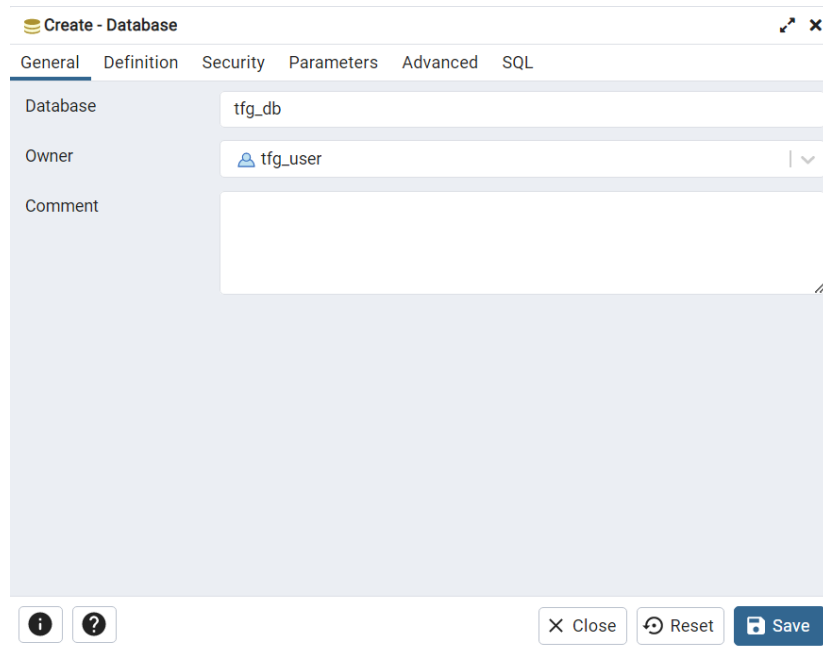


Figura 14.6: Configuración pgAdmin - 6

- c) Configuración en Django: Debemos crear un archivo de variables de entorno con la configuración de la base de datos. Para hacerlo nos dirigimos al directorio del proyecto llamado “project”. En este directorio encontraremos archivos como “celery.py” o “settings.py”. Crear un archivo llamado “.env” que contendrá la siguiente información.

Extracto de código 14.4: Archivo .env

```
DEBUG=1
SECRET_KEY=foo
DJANGO_ALLOWED_HOSTS=localhost 127.0.0.1
[ : : 1]
SQL_ENGINE=django.db.backends.postgresql
SQL_DATABASE=tfg_db
SQL_USER=tfg_user
SQL_PASSWORD=tfg_password
SQL_HOST=localhost
SQL_PORT=5432
```

Los campos podéis modificarlos según vuestra configuración de postgresql.

9. **Instalar Redis:** Para que funcione Celery es necesario instalar Redis. Se puede descargar desde el [sitio oficial](#). La versión usada por el desarrollador ha sido [Redis para windows 5.0.14](#).

Se debe ejecutar el comando `redis-server.exe` y para comprobar que funciona nos conectamos al servidor con el comando `redis-cli.exe` y escribimos “ping”.

El servidor responderá con “PONG” en el caso de que funcione bien (Figura 14.7).



Figura 14.7: Redis respuesta

10. **Despliegue:** Ya con todas las dependencias instaladas solo falta desplegar el proyecto. Primero abriremos una ventana de comandos en el directorio del proyecto y escribiremos:

Extracto de código 14.5: Comandos manage.py

```
#Realizar migraciones de la base de datos.
python manage.py makemigrations
python manage.py migrate

#Crear superusuario (opcional)
python manage.py createsuperuser

#Despliegue local
python manage.py runserver
```

En otra ventana diferente para poner en funcionamiento Celery escribiremos:

Extracto de código 14.6: Celery despliegue - 1

```
# project.celery es el archivo donde se configura
celery.
# --pool configura el grupo de procesos e hilos. En
windows usar el modo "threads" o "solo".
celery -A project.celery worker --pool=threads -l
info
```

Ignorando los “warnings”, el despliegue acabará cuando recibamos una información parecida a:

Extracto de código 14.7: Celery despliegue - 2

```
[2023-10-02 13:33:12,776: INFO/MainProcess] mingle:
searching for neighbors
[2023-10-02 13:33:19,882: INFO/MainProcess] mingle:
all alone
[2023-10-02 13:33:30,172: INFO/MainProcess]
celery@LAPTOP-011JQPAK ready.
```

Una vez seguidos estos pasos ya tendremos la aplicación corriendo en local.

14.2. Manual de usuario

En el manual de usuario se explicará y mostrará con imágenes la aplicación final desarrollada.

Análisis

La primera vista que obtendremos al entrar a la aplicación es directamente el formulario para realizar el análisis (ver Figura 14.8). En él se puede seleccionar el proyecto a analizar y el tipo de los datos (“RNAseq” o “miRNAs”). Tras seleccionar un proyecto se mostrará información de este recogido de la página de GDC Data Portal.

The screenshot shows the GenCancerAnalyzer application interface. The top navigation bar includes 'GenCancerAnalyzer', 'Analyze', 'Analyzed Projects', 'API', and 'Web Information'. The main content area is split into two panels. The left panel, titled 'Select the project', contains a 'Data type' dropdown menu currently set to 'RNAseq'. Below it is a 'Projects' section with a search bar containing 'tcga-c' and a list of project entries. The entry 'TCGA-CHOL: Cholangiocarcinoma' is highlighted. An 'Analyze' button is positioned to the right of the project list. The right panel, titled 'Selected Project Information', displays the following data: Name: Cholangiocarcinoma, Project ID: TCGA-CHOL, Disease Type: Adenomas and Adenocarcinomas, Primary Site: Liver and intrahepatic bile ducts, Other and unspecified parts of biliary tract, Gallbladder, and DB Gap Accession Number: null. At the bottom of this panel, the status 'Not analyzed' is displayed in red text.

Figura 14.8: App - Vista análisis

En la información del proyecto también se muestra si el proyecto seleccionado está analizado, en proceso o si todavía no. Al darle al botón de análisis, en el caso de que no esté aún actualizado, comenzará dicho análisis y seremos redireccionados a la vista de proyectos analizados.

Proyectos analizados

En esta vista (ver Figura 14.9) los usuarios pueden ver todos los proyectos que ya han sido o están siendo analizados, además de disponer de unos botones parecidos al formulario de análisis para filtrar los proyectos que se muestran.

En esta vista se irán actualizando los proyectos a medida que terminan los análisis.

Si pulsamos sobre el botón “Results” de alguno de los proyectos que ya han sido analizados seremos redireccionados a la vista de resultados de dicho caso de estudio.

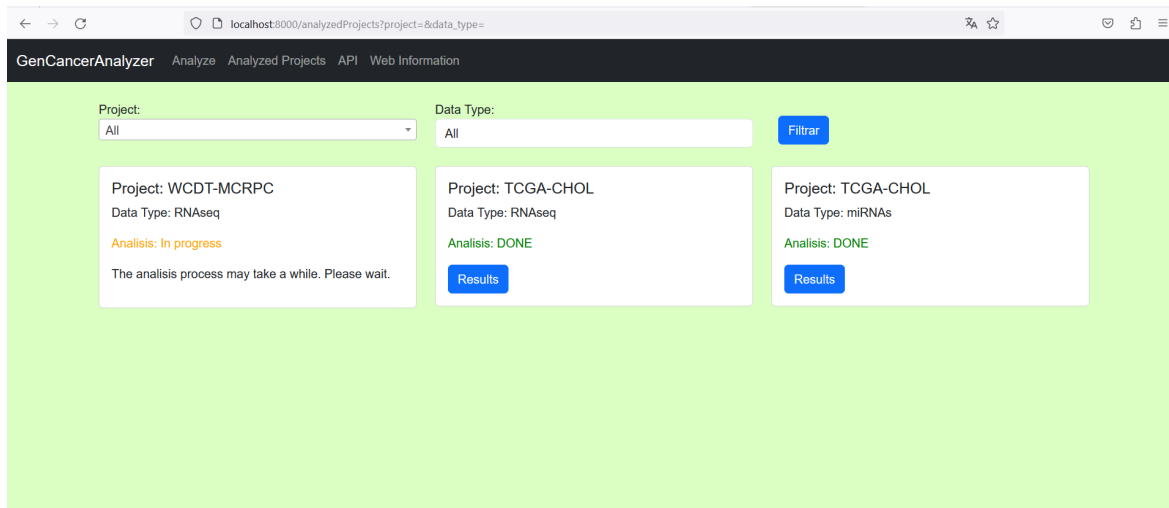


Figura 14.9: App - Vista proyectos analizados

API Información

A través de la barra de navegación podemos acceder a la vista de información de la API (ver Figura 14.10), una vista en la cual se resume las peticiones disponibles para que los usuarios puedan acceder a los datos de la base de datos.

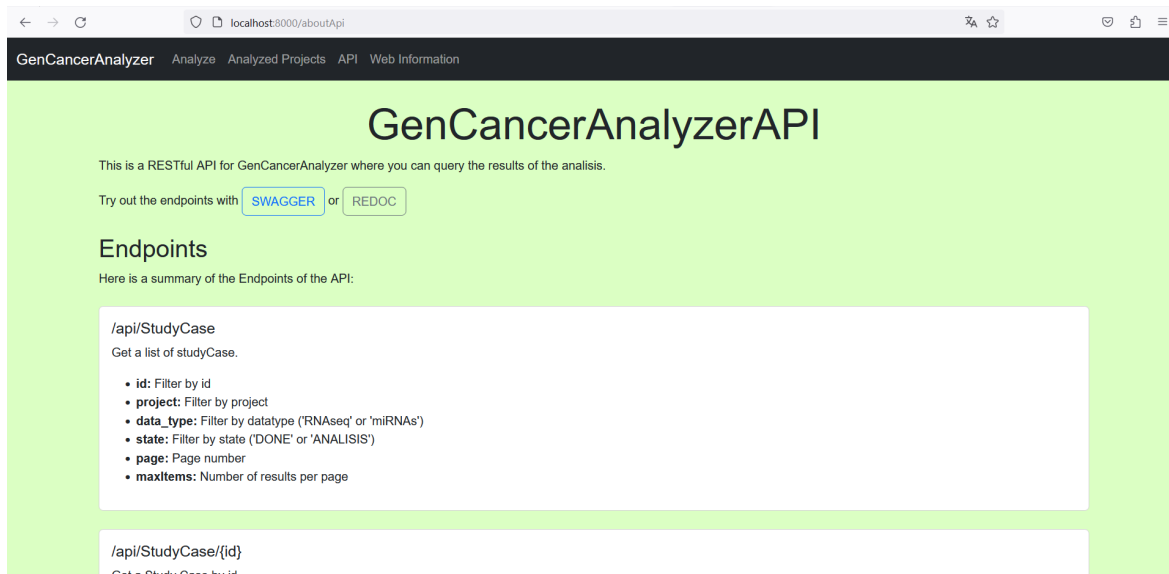


Figura 14.10: App - Vista información API

Los botones “SWAGGER” y “REDOC” son enlaces a la documentación de la API que ofrecen interfaces para probar las distintas consultas. En la vista de Swagger (ver Figura 14.11) ,por ejemplo, podemos probar las consultas y ver los parámetros que existen.

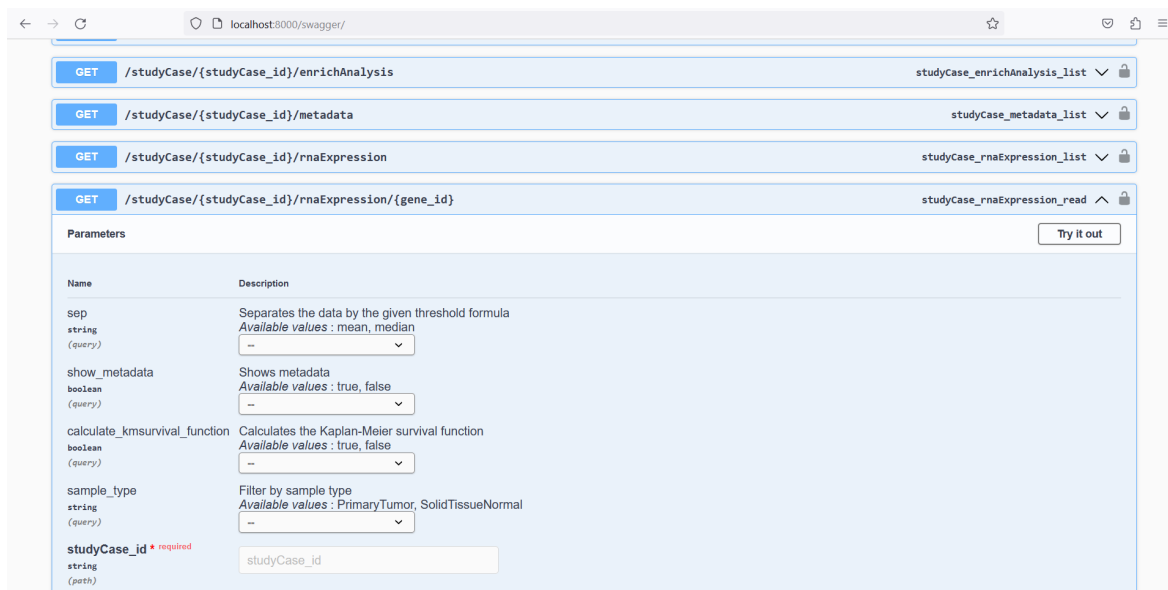


Figura 14.11: App - Vista Swagger

Información de la web

A través de la barra de navegación también podemos acceder a una vista con información (ver Figura 14.12) acerca de la aplicación, con datos de contacto y un enlace al repositorio.

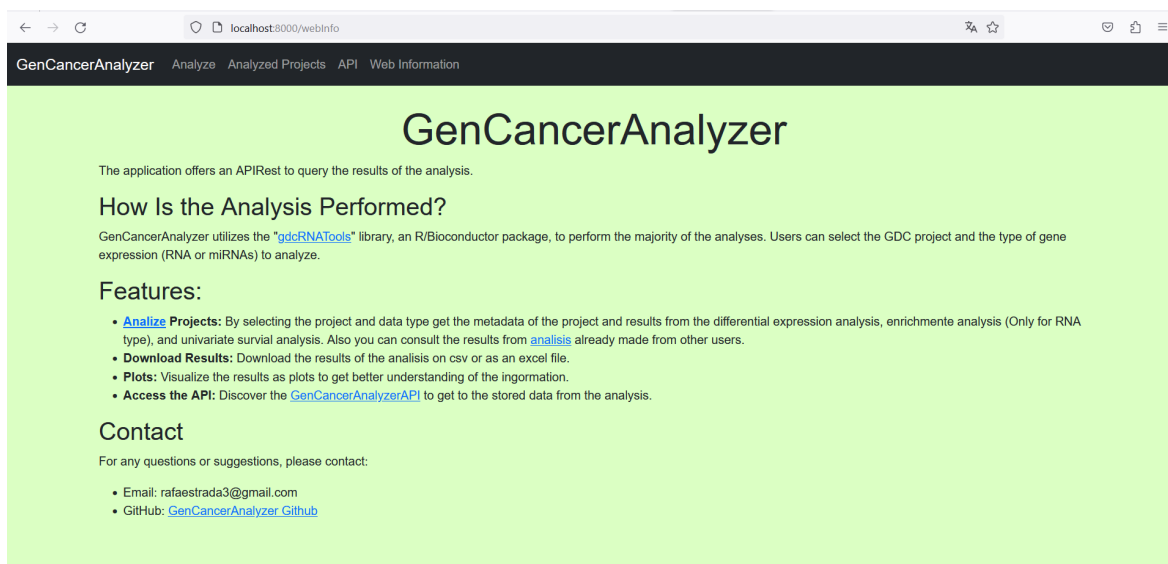


Figura 14.12: App - Vista información web

Resultados

La vista de resultados es la más importante de la aplicación y se compone de una vista para metadatos, para el análisis de expresión diferencial, para el análisis de enriquecimiento y para el análisis de supervivencia.

En la parte superior de las vistas de los resultados se muestra información del caso de estudio junto con botones para navegar entre los distintas vistas de resultados (ver Figura 14.13).

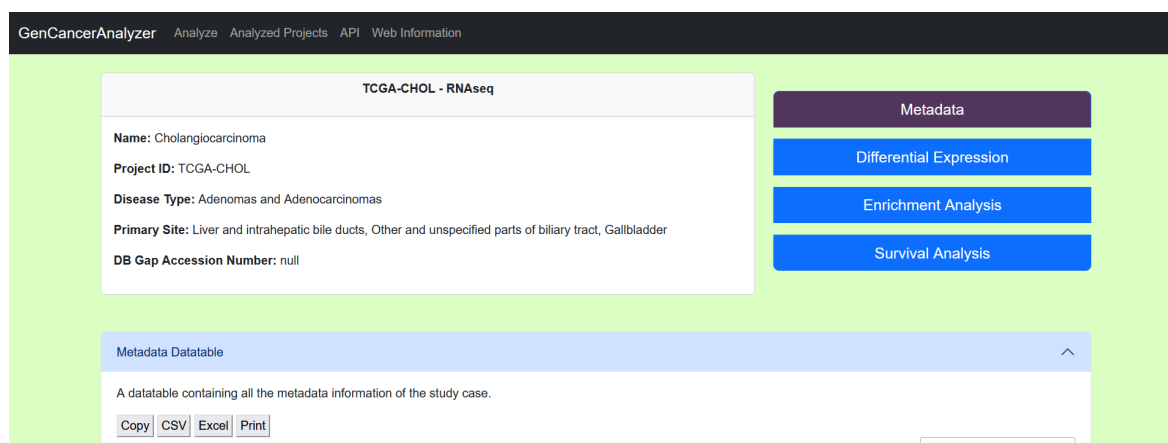


Figura 14.13: App - Vista resultados - Información

Datatables

En cada vista de resultados se mostrarán tablas (ver Figura 14.14 con la información almacenada. Algunas características que ofrecen son:

- Resultados paginados.
- Se puede buscar datos por los valores de la primera columna.
- se puede ordenar los valores ascendentemente y descendentemente pulsando en las columnas por las cuales ordenar.
- Hay botones que permiten descargar el contenido en distintos formatos, como CSV o Excel.

Resumen metadatos

En la vista de resultados de metadatos, obtenemos un resumen de las variables numéricas obteniendo las medias, medianas, mínimos, máximos y percentiles (ver Figura 14.15).

Metadata Datable

A datatable containing all the metadata information of the study case.

Copy CSV Excel Print

Search: AAV

	file_name	file_id	patient	sample	submitter_id	enti
TCGA-3X-AAV9-01A	913eff0d-52b9-4e73-9351-970393b2f523.ma_seq.augmented_star_gene_counts.tsv	43ac405a-93dc-4241-86be-3f034ba1eafb	TCGA-3X-AAV9	TCGA-3X-AAV9-01	TCGA-3X-AAV9-01A	TCC 72R
TCGA-3X-AAVA-01A	ad57a542-3089-47f7-8750-382adff65e97.ma_seq.augmented_star_gene_counts.tsv	37f36a91-44ec-433a-a7ce-06f982b54f52	TCGA-3X-AAVA	TCGA-3X-AAVA-01	TCGA-3X-AAVA-01A	TCC 11R
TCGA-3X-AAVB-01A	40538197-4671-4420-928e-7262cf452396.ma_seq.augmented_star_gene_counts.tsv	74f1a110-9ebb-4f5b-a6d0-4575602b4195	TCGA-3X-AAVB	TCGA-3X-AAVB-01	TCGA-3X-AAVB-01A	TCC 31R
TCGA-3X-AAVC-01A	2cfef5ef-b6f6-43fa-a5f6-8959c24ed654.ma_seq.augmented_star_gene_counts.tsv	1615db37-9b6e-4aca-b4e1-354beaa5fc3a	TCGA-3X-AAVC	TCGA-3X-AAVC-01	TCGA-3X-AAVC-01A	TCC 21R
TCGA-3X-AAVE-01A	edb6e6ac-5199-445b-a9d6-c08ec61c9dea.ma_seq.augmented_star_gene_counts.tsv	9432a7f6-59ad-44e1-b18b-5b7a366b3b88	TCGA-3X-AAVE	TCGA-3X-AAVE-01	TCGA-3X-AAVE-01A	TCC 11R

Showing 1 to 5 of 5 entries (filtered from 44 total entries)

Previous 1 Next

Figura 14.14: App - Vista resultados - Datatables

Metadata Datable

Metadata Summary

An statistical summary of metadata information.

For every categorical attribute describes the counts of each category ignoring the null fields.

For every numeric attribute describes the mean, min value, max value and quantiles ignoring the null fields.

Gender Counts <p>male: 22</p> <p>female: 22</p>	Sample Type Counts <p>PrimaryTumor: 35</p> <p>SolidTissueNormal: 9</p>	Vital Status Counts <p>Dead: 23</p> <p>Alive: 21</p>
Field: age_at_diagnosis <p>Mean: 23646.727272 1st Qu.: 21310.5 Median: 24933 Min: 10659 Max: 30039 3rd Qu.: 26618.5</p>	Field: days_to_death <p>Mean: 615.78260869 1st Qu.: 170 Median: 555 Min: 21 Max: 1939 3rd Qu.: 741</p>	Field: days_to_last_follow_up <p>Mean: 766.95833333 1st Qu.: 168 Median: 679.5 Min: 0 Max: 1976 3rd Qu.: 1203</p>

Figura 14.15: App - Vista resultados - Metadata resumen

Gráficos

Los gráficos están implementados con la herramienta plotly.js que proporciona distintas funcionalidades. Destacamos:

- Interacción: Los gráficos pueden ser explorados y manipulados por los usuarios. Puedes hacer zoom, pan, y obtener información detallada al pasar el ratón sobre puntos específicos.
- Exportación: Puedes exportar tus gráficos como PNG.

Expresión diferencial - Volcano Plot

En los resultados del análisis de expresión diferencial encontramos un gráfico de volcano (ver Figura 14.16). Este gráfico representa los valores de “PValue” y “Fold Change” diferenciando genes significativos y no significativos según umbrales.

Los valores de los umbrales se pueden configurar si pulsamos en el botón de “Config”, que mostrará un formulario flotante con los inputs.

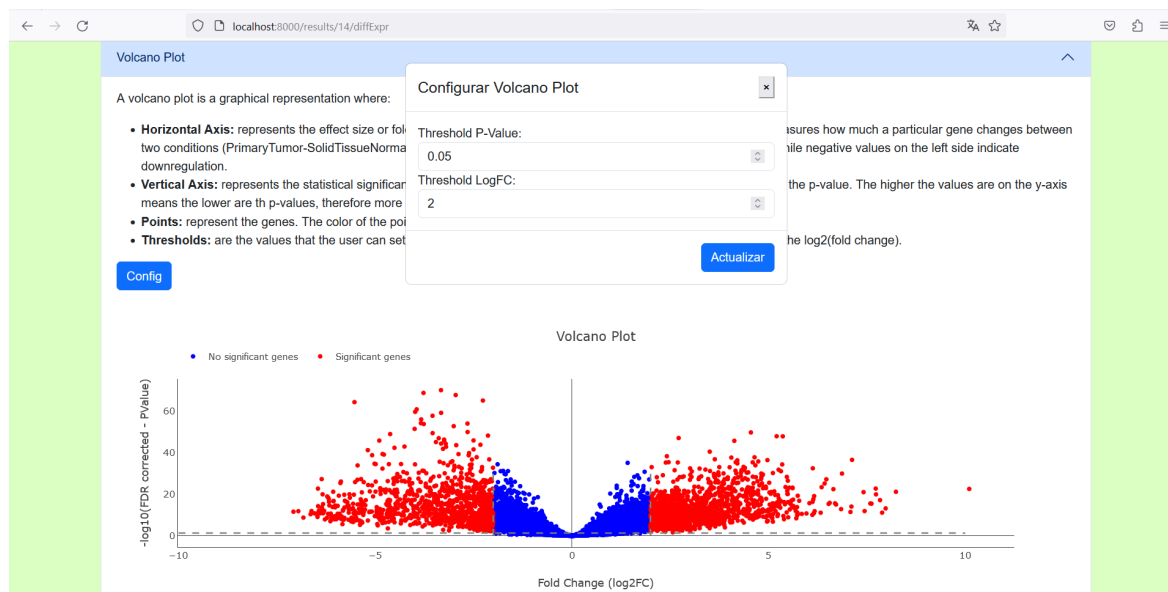


Figura 14.16: App - Vista resultados - Volcano plot

Expresión diferencial - Bar Plot

En los resultados del análisis de expresión diferencial encontramos un gráfico de barras (ver Figura 14.17) el cual divide los genes según sea sobreexpresados o subexpresados, y se hacen conteos de los genes agrupados por el grupo al que pertenezcan.

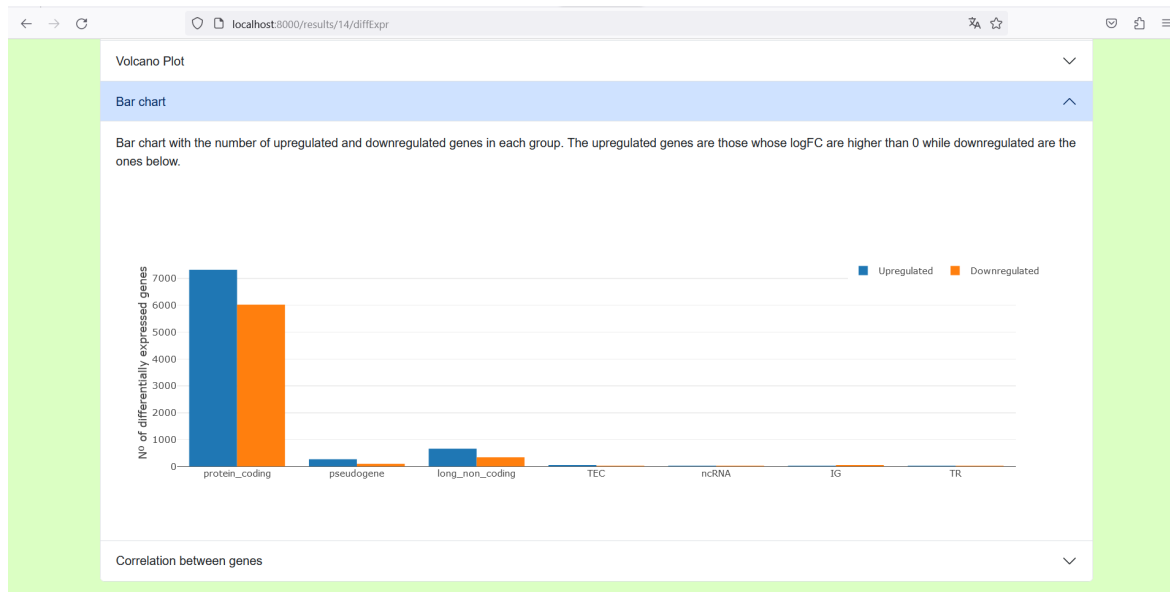


Figura 14.17: App - Vista resultados - Bar plot

Expresión diferencial - Correlation Plot

En los resultados del análisis de expresión diferencial encontramos un gráfico de correlación (ver Figura 14.18). Este gráfico relaciona dos genes, siendo los puntos los valores normalizados de cada muestra. La línea roja es la línea de tendencia de regresión lineal calculada para los valores.

Enriquecimiento - Bar plot

La gráfica que se muestra en la vista de los resultados del análisis de enriquecimiento se trata de un gráfico de barras (ver Figura 14.19) que es configurable. El gráfico se puede configurar según los siguientes valores:

- Plot type: Puedes transformar el gráfico en un gráfico de burbujas (ver Figura 14.20).
- Limit: El límite de resultados a mostrar en la gráfica.
- Category: La categoría a filtrar.
- Field to sort: Campo numérico que servirá de eje y.
- Order: Ordenar los valores de forma descendiente o ascendente.

Survival plot

En la vista de resultados de análisis de supervivencia se representa una gráfica de supervivencia Kaplan-Meier (ver Figura 14.21). Esta gráfica puede ser configurada según los siguientes parámetros:

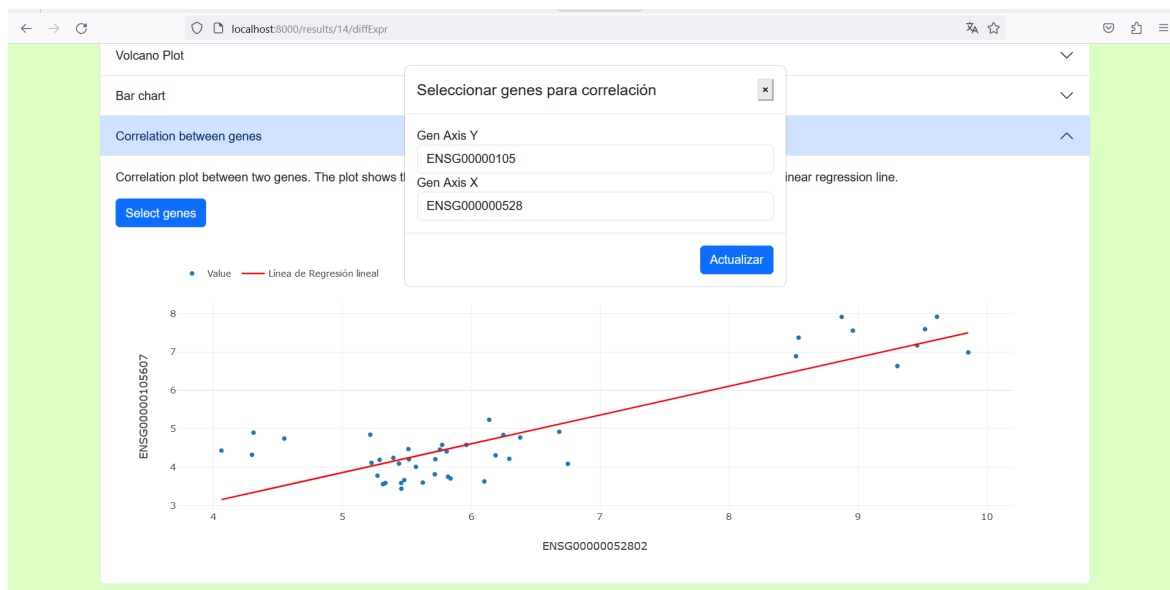


Figura 14.18: App - Vista resultados - Correlation plot

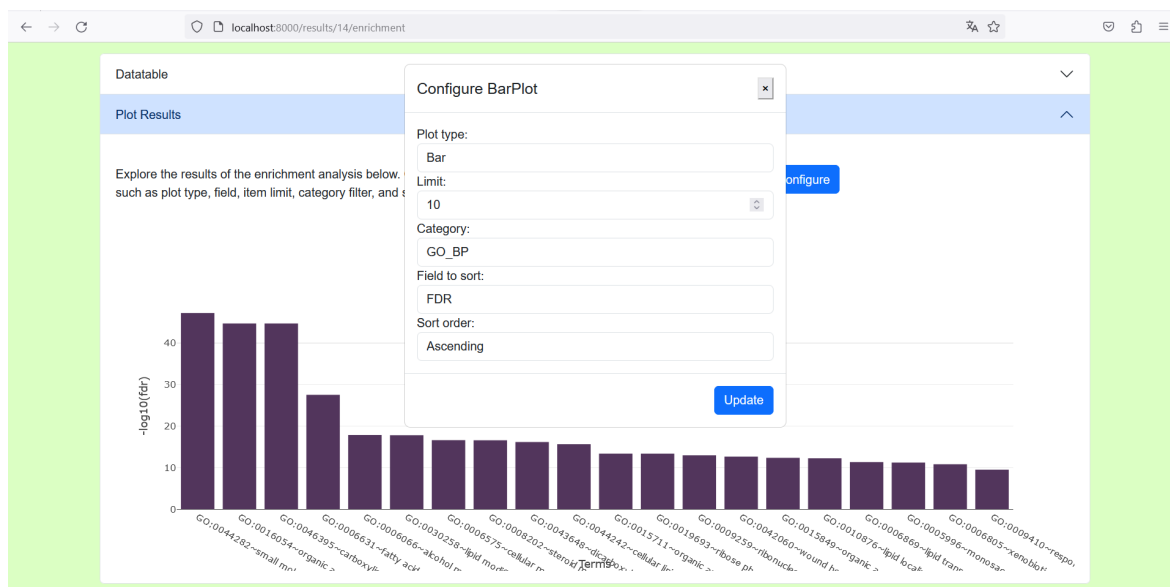


Figura 14.19: App - Vista resultados - Enrichment Bar plot

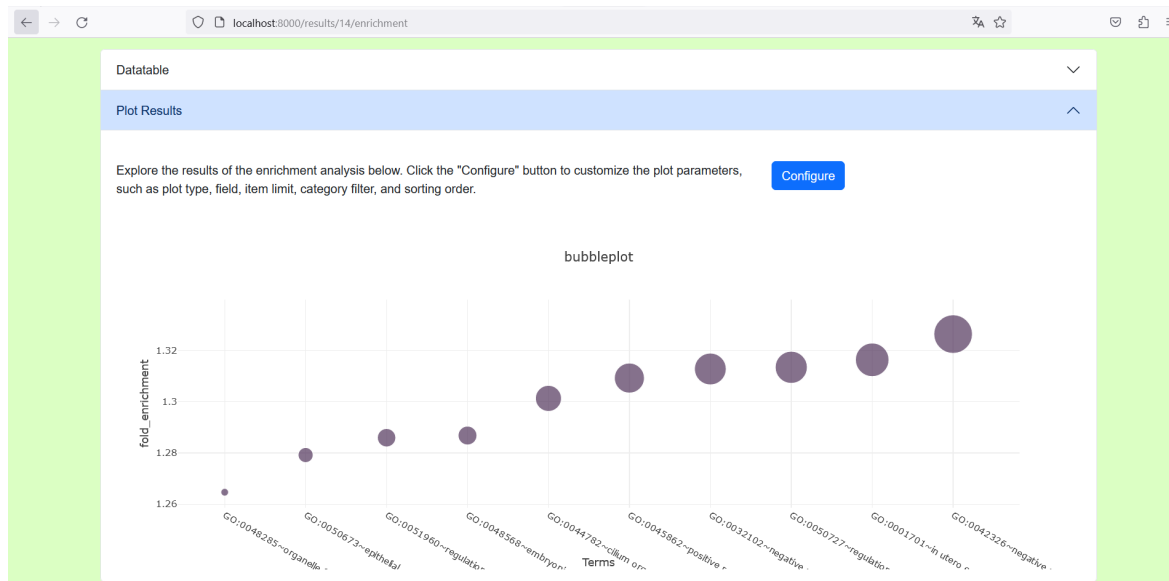


Figura 14.20: App - Vista resultados - Bubble plot

- gene.id: identificador del gen a analizar.
- separator: Fórmula por la cual separar las muestras. Puede ser la media de los valores o por la mediana. También se puede poner a None lo cual provoca que no se dividan.
- sample type: Tipo de los genes a filtrar. Puede ser “Primary Tumor” y “Solid tissue normal”.

14.3. Pruebas

En la aplicación se han diseñado un conjunto de pruebas. Dentro de la carpeta “apps>home>tests” encontramos dos archivos: el archivo “tests.py” son pruebas unitarias hechas para comprobar la funcionalidad de las consultas de la API y los modelos, mientras que el archivo “testsSelenium.py” son tests hechos con la herramienta Selenium que permite controlar el navegador y comprobar aspectos de la interfaz. Los tests de Selenium comprueban la navegabilidad entre la aplicación y si los resultados mostrados son los esperados.

Por otra parte, para validar los resultados obtenidos en la aplicación, se han comparado con un notebook[23] que realiza los mismos análisis para el proyecto TCGA-CHOL.

Para correr los tests nos servimos de los comandos.

Extracto de código 14.8: Comandos pruebas

```
#Pruebas unitarias
python manage.py test apps.home.tests.tests
#Pruebas selenium
```

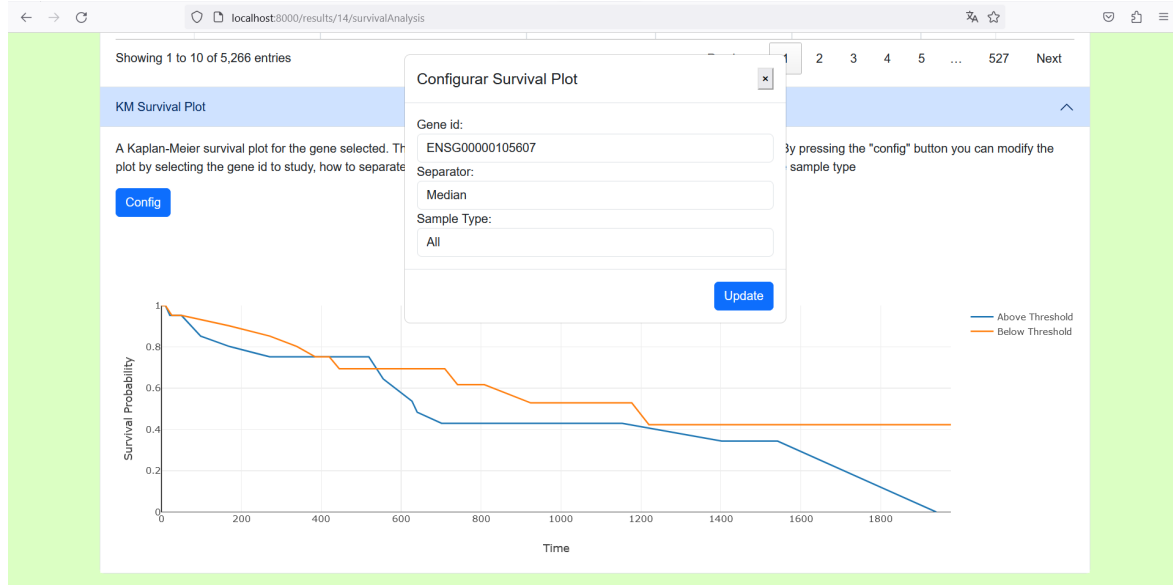


Figura 14.21: App - Vista resultados - Survival plot

```
python manage.py test apps.home.tests.testsSelenium
```

Las pruebas Selenium se realizan con Firefox por lo que es necesario que poseas este navegador. En el caso de que aún sigan sin funcionar, probablemente sea necesario instalar además el controlador para Firefox, llamado GeckoDriver (puedes descargarlo [aquí](#)) y añadirlo al Path.

Los tests de Selenium además se sirven de datos de prueba almacenados en la carpeta “DATA” para comprobar resultados mostrados en la aplicación.

15. Cumplimiento de objetivos

En este capítulo se explica como se cumplieron los objetivos propuestos previamente en el Capítulo 3.

Obj. 1 - Realizar una memoria del proyecto. La presente memoria ha sido validada por el tutor y será evaluada posteriormente.

Obj. 2 - Realizar una presentación final del proyecto. La presentación del proyecto será evaluada posteriormente en la defensa del proyecto.

Obj. 3 - Aplicación Web. Mediante el uso de Django como framework se ha conseguido desarrollar un software como aplicación web.

Obj. 4 - Obtención de datos desde GDC Portal. La descarga de los datos para su posterior análisis se realiza mediante un método proporcionado por el paquete de R/Bioconductor GDCRNATools [14], el cual accede a la API de GDC obteniendo los datos según el proyecto y el tipo de los datos.

Obj. 5 - Realizar análisis. Los análisis desarrollados en la aplicación son: análisis de expresión diferencial, análisis de enriquecimiento y análisis de supervivencia Kanplan-Meier. La librería GDCRNATools ofrecía métodos para desarrollar estos análisis. También se ha usado la librería Lifelines de Python para obtener las funciones de supervivencia.

Obj. 6 - Resultados de análisis. Los resultados de los análisis son mostrados en la aplicación a través de Datatables. Estas datatables pueden ser descargadas en distintos formatos como CSV o Excel. Además se ha desarrollado y documentado la API para acceder a los datos almacenados.

Obj. 7 - Gráficas resultados. En las vistas de resultados de la aplicación se muestran diferentes gráficas, la mayoría pudiendo ser configuradas por el usuario. Plotly.js añade valor a la aplicación permitiendo a los usuarios interactuar con las gráficas y descargarlas como PNG.

16. Coste final

En este capítulo se describe el coste final del proyecto y la desviación sufrida.

Primero veremos las desviaciones que se han sufrido a lo largo de las distintas fases del proyecto (ver Figura 16.1). Se observa que la fase de planificación no muestra variaciones, ya que la estimación se realizó específicamente para las tareas de las fases posteriores a la planificación. En el sprint 3, los costos reales son inferiores a los costos previstos, atribuibles principalmente a la adquisición de experiencia por parte del programador, lo que ha aumentado la productividad. Por otro lado, los costos del sprint 4 aumentan considerablemente debido a retrasos en tareas y la identificación de nuevos requisitos. Las demás fases también presentan costos superiores a los previstos, indicando en general una subestimación de las horas de trabajo.



Figura 16.1: Desviación de costes

Por otra parte, la Figura 16.2 muestra la evolución a lo largo de las distintas fases de los costes acumulados previstos y reales. Se observa como la variación no difiere excesivamente, siendo los costes acumulados reales mayores a los previstos. **El coste final sería de aproximadamente 10560 €** el cual es menor al presupuesto dado en la sección 8.4 de 10968,75 €, siendo por tanto un resultado bastante positivo.

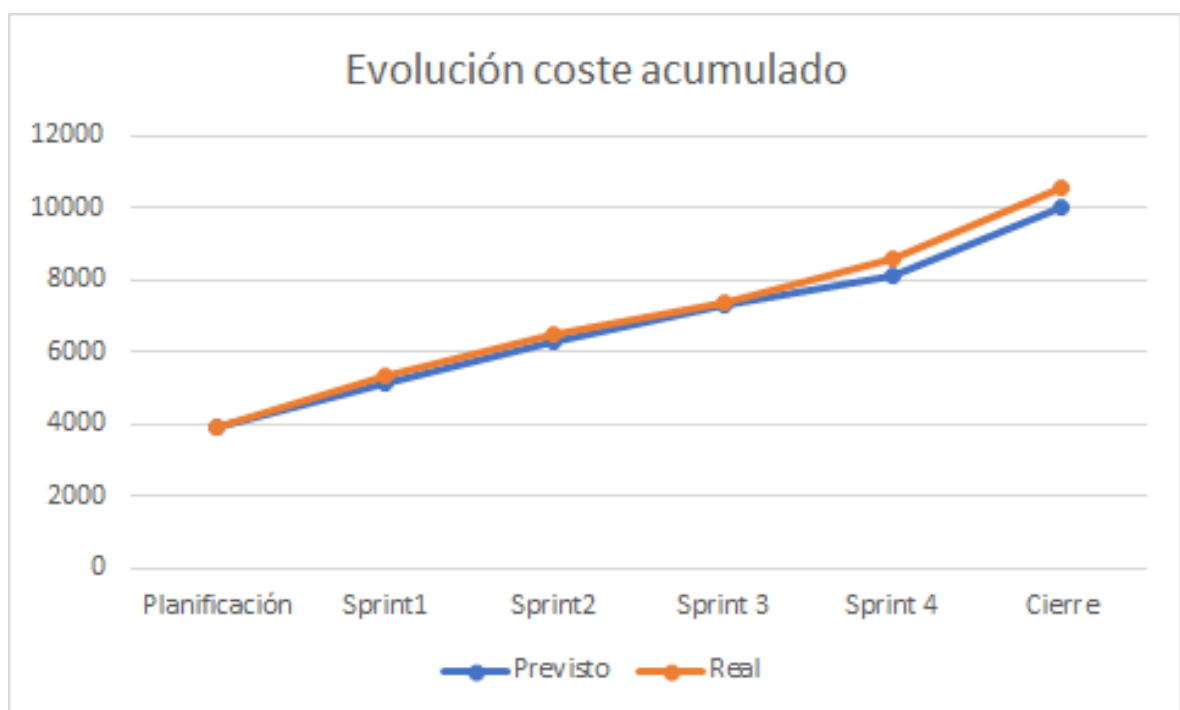


Figura 16.2: Evolución coste acumulado

17. Esfuerzo empleado

En este capítulo se detalla las horas totales de trabajo empleado en las distintas tareas del proyecto (ver Cuadro 17.1).

El esfuerzo total dedicado al trabajo es de **316,5 horas**.

Cuadro 17.1: Horas de trabajo reales

EDT	Título	Horas reales
1	Proyecto	316,5 horas
1.1	Planificación	103 horas
1.1.1	Recopilar requisitos	24 horas
1.1.2	Analizar requisitos	8 horas
1.1.3	Arquitectura del sistema	7 horas
1.1.4	Crear EDT y diccionario	15 horas
1.1.5	Identificar hitos	5 horas
1.1.6	Estimar duración de cada tarea	7 horas
1.1.7	Identificar recursos y adquisiciones	10 horas
1.1.8	Preparación del cronograma	8 horas
1.1.9	Calcular costes	8 horas
1.1.10	Identificar riesgos	6 horas
1.1.11	Metodología de desarrollo	5 horas
1.2	Ejecución y seguimiento	164,5 horas
1.2.1	Sprint 1	53 horas
1.2.1.1	Crear proyecto django	4 horas
1.2.1.2	Configurar base de datos postgresql	2 horas
1.2.1.3	Configurar R y descargar paquete GDCRNATools	3 horas
1.2.1.4	Configurar drf-yasg	2 horas
1.2.1.5	Configurar librería rpy2	4,5 horas
1.2.1.6	Plantilla base	2 horas
1.2.1.7	Barra de navegación	2 horas
1.2.1.8	Modelo StudyCase	2 horas
1.2.1.9	Formulario StudyCase	3 horas
1.2.1.10	Información de proyecto	2,5 horas
1.2.1.11	Descarga de datos por projectid y datatype	3,5 horas
1.2.1.12	Modelo Metadata	2 horas
1.2.1.13	Obtener metadatos	2,5 horas
1.2.1.14	Modelo RNAExpr	2 horas
1.2.1.15	Modelo DiffExpr	1,5 horas
1.2.1.16	API StudyCase	3 horas
1.2.1.17	API Metadata	2 horas
1.2.1.18	API RNAExpr	2 horas
1.2.1.19	API diffExpr	1,5 horas
Continúa en la siguiente página		

Cuadro 17.1 – Continuación desde la página anterior

EDT	Título	Horas reales
1.2.1.20	Seguimiento 1	6 horas
1.2.1.20.1	Informe de avance e incidencias	2 horas
1.2.1.20.2	Retrospectiva	2 horas
1.2.1.20.3	Informe de cambios	2 horas
1.2.2	Sprint2	41,5 horas
1.2.2.1	Analysis DE	3 horas
1.2.2.2	Analysis EA	3,5 horas
1.2.2.3	API pagination	2,5 horas
1.2.2.4	Documentar API Swagger	4 horas
1.2.2.5	Vista de resultados	3,5 horas
1.2.2.6	Datatable Metadata	3 horas
1.2.2.7	Datatable DE	1,5 horas
1.2.2.8	Volcano Plot DE	4,5 horas
1.2.2.9	Volcano Plot DE Config	2,5 horas
1.2.2.10	Bar Plot DE	3 horas
1.2.2.11	Correlation Genes Plot	4,5 horas
1.2.2.12	Seguimiento 2	6 horas
1.2.2.12.1	Informe de avance e incidencias	2 horas
1.2.2.12.2	Retrospectiva	2 horas
1.2.2.12.3	Informe de cambios	2 horas
1.2.3	Sprint3	32,5 horas
1.2.3.1	Descargar datatable csv	1 hora
1.2.3.2	Ocultar resultados EA para miRNAs	1 hora
1.2.3.3	Resumen estadístico metadata	3 horas
1.2.3.4	Resumen categorías metadata	2 horas
1.2.3.5	Datatable EA	2,5 horas
1.2.3.6	Bar Plot EA	4 horas
1.2.3.7	Bubble Plot EA	1,5 horas
1.2.3.8	Modelo SurvivalAnalysis	2 horas
1.2.3.9	Análisis Supervivencia univariado	3 horas
1.2.3.10	API SA	5 horas
1.2.3.11	Datatable SA	1,5 horas
1.2.3.12	Seguimiento 3	6 horas
1.2.3.12.1	Informe de avance e incidencias	2 horas
1.2.3.12.2	Retrospectiva	2 horas
1.2.3.12.3	Informe de cambios	2 horas
1.2.4	Sprint 4	37,5 horas
1.2.4.1	Plot Survival Analysis	4,5 horas
1.2.4.2	Configurar Celery	3,5 horas
1.2.4.3	Análisis tarea asíncrona	3 horas
1.2.4.4	Vista de proyectos analizados	3,5 horas
1.2.4.5	Detalle de proyectos: estado	2 horas
1.2.4.6	Actualizar estado en proceso análisis	1,5 horas
Continúa en la siguiente página		

Cuadro 17.1 – Continuación desde la página anterior

EDT	Título	Horas reales
1.2.4.7	Select2 implementar	2 horas
1.2.4.8	Vista web information	3 horas
1.2.4.9	Detallar gráficas resultado	3 horas
1.2.4.10	Pruebas selenium	5,5 horas
1.2.4.11	Seguimiento 4	6 horas
1.2.4.11.1	Informe de avance e incidencias	2 horas
1.2.4.11.2	Retrospectiva	2 horas
1.2.4.11.3	Informe de cambios	2 horas
1.3	Cierre	49 horas
1.3.1	Lecciones aprendidas	3 horas
1.3.2	Manual de usuario	7 horas
1.3.3	Finalización de la memoria	26 horas
1.3.4	Preparación de la presentación	13 horas

18. Conclusiones

Este proyecto culmina con el exitoso desarrollo de una aplicación diseñada para facilitar a investigadores, biólogos y profesionales de la salud el acceso y análisis de datos de expresión genética relacionados con el cáncer, eliminando la barrera de la necesidad de conocimientos especializados en tecnologías informáticas.

La aplicación ofrece capacidades avanzadas de análisis, incluyendo expresión diferencial, enriquecimiento y supervivencia, aplicables a secuencias de RNA y miRNA de proyectos de investigación almacenados en el Portal de Datos GDC. Los usuarios tienen la capacidad de descargar los resultados de estos análisis y, adicionalmente, la aplicación presenta los datos de manera visual a través de gráficas, muchas de las cuales son configurables. Esta funcionalidad permite a los usuarios obtener interpretaciones visuales significativas de los resultados, facilitando así la comprensión y aplicación de los hallazgos en sus respectivos campos de estudio.

Adicionalmente, la aplicación brinda a los usuarios documentación detallada sobre su API, facilitando a otros desarrolladores el acceso a la información almacenada mediante diversas consultas.

Para alcanzar estos resultados, el estudiante no solo ha aplicado los conocimientos adquiridos a lo largo de su carrera en Ingeniería Informática de Software, sino que también ha ganado experiencia práctica con diversas tecnologías empleadas en la aplicación, tales como “Plotly.js” y “Celery”, entre otras.

18.1. Lecciones aprendidas

En esta sección, se presentan las lecciones aprendidas durante el desarrollo del proyecto. Estas reflexiones son valiosas para mejorar prácticas, identificar áreas de mejora y aplicar conocimientos adquiridos en proyectos futuros.

Entendimiento del dominio: Es fundamental para definir bien los requisitos realizar un estudio previo del dominio entendiendo los datos con los que se va a trabajar y análisis.

Uso de librerías: Existen muchísimas librerías que pueden aportar métodos necesarios. En este caso el uso del paquete GDCRNATools de R/Bioconductor ha aportado la mayoría de las funciones de análisis lo cual ha reducido las horas para estas tareas. Además muchas de las librerías pueden aportar más valor a la aplicación como en este caso plotly.js que ofrece interactividad y exportación de los gráficos.

Problemas con windows: Hay mucho más soporte por parte de servicios y tecnología para entorno Unix. Con windows ha habido problemas en el desarrollo a la hora de integrar tecnologías como rpy2, Celery y Redis.

Uso de framework para front-end: Uno de los puntos negativos de la arquitectura del producto es que no usa unframework dedicado al front-end. Usando javascript

en la mayoría de vistas hubiera sido ideal usar una tecnología como React.

Tareas concretas A menudo durante el desarrollo se puede perder el foco de la tarea que se realiza. Es fundamental que las tareas tengan un objetivo claramente descrito para no dar lugar a ambigüedades.

Holguras en la planificación: Es ideal al estimar, dejar un margen de tiempo adicional a las tareas a desarrollar. Este tiempo adicional no solo es útil para gestionar los riesgos que surgen en el desarrollo y manejar las incertidumbres, sino que también ayuda a reducir el estrés del equipo.

18.2. Trabajo futuro

A continuación se proponen algunas funcionalidades adicionales para la aplicación en el caso de continuar su desarrollo en un futuro:

- **Análisis de otro data type:** Actualmente la aplicación solo analiza para los tipos RNAseq y miRNAs. La aplicación podría ofrecer soporte para otro tipo de datos.
- **Comparaciones:** Sería una idea interesante poder observar resultados de distinto proyectos y ver las diferencias que ofrecen entre sí.
- **Avisos de fin de análisis:** Los análisis tardan mucho tiempo en realizarse en el caso de proyectos con muchos datos. Se podría incluir una pestaña de notificación en la aplicación que avisara de los análisis acabados. Otra idea sería que los usuarios pudiesen dejar su correo para ser informados de cuando acaba un análisis.
- **Docker:** Instalar todas las dependencias y realizar el despliegue puede ser un proceso tedioso. Se podría crear la imagen docker del proyecto para acelerar este proceso.
- **Integrar React:** Integrar React en la aplicación puede mejorar la mantenibilidad del código a largo plazo y aceleraría el desarrollo.

Parte VII

APÉNDICE

19. Glosario

API API significa Interfaz de Programación de Aplicaciones. Es un conjunto de reglas y definiciones que permite que diferentes aplicaciones se comuniquen entre sí. En esencia, una API especifica cómo los componentes de software deben interactuar.

Análisis de enriquecimiento Consulta la definición en la Sección [2.3.4](#).

Análisis de expresión diferencial Consulta la definición en la Sección [2.3.3](#).

Análisis de supervivencia Consulta la definición en la Sección [2.3.5](#).

Fold change Consulta la definición en la Sección [2.3.3](#).

Metadatos Los metadatos son “datos que describen otros datos”, en otras palabras, son datos que describen características, propiedades y contextos relacionados con los datos principales. En el ámbito de la genómica y la bioinformática, los metadatos son esenciales para entender y contextualizar la información genómica.

miRNA (miARN) Consulta la definición en la Sección [2.1.1](#).

Project (proyecto) En el trabajo se refiere a proyectos de investigación que recaban datos sobre un tipo de cáncer.

P-Value Consulta la definición en la Sección [2.3.3](#).

RNA (ARN): Consulta la definición en la Sección [2.1.1](#).

Sincronicidad: En la programación, la sincronicidad hace referencia al estilo de programación en el que las operaciones se ejecutan en secuencia, una después de la otra, y se espera que cada operación se complete antes de pasar a la siguiente. Esto es contrario a la programación asíncrona, donde las operaciones pueden ejecutarse de manera concurrente y no bloqueante.

Sprint Un sprint es un concepto utilizado en la metodología ágil de desarrollo de software. Se refiere a un período de tiempo definido y fijo durante el cual un equipo de desarrollo trabaja en un conjunto específico de tareas o metas.

20. Bibliografía

- [1] Saad Ali. Setup your django project with celery, celery beat, and redis, Jan 2023. URL <https://saadali18.medium.com/setup-your-django-project-with-celery-celery-beat-and-redis-644dc8a2ac4b>.
- [2] Guillermo Ayala. *Análisis estadístico de datos ómicos*, 2023. URL <https://www.uv.es/ayala/docencia/tami/tami13.pdf>.
- [3] Ceolevel. ¿que es pert? ¿para que se utiliza y como se calcula?, Oct 2021. URL <https://www.ceolevel.com/certificacion-pmp-que-es-pert-para-que-se-utiliza-y-como-se-calcula>.
- [4] Tom Christie and contributors. *Django Rest Framework Documentation*. Django REST framework, Various locations, 2021. URL <https://www.django-rest-framework.org/>.
- [5] Gene Ontology Consortium. Go enrichment analysis, 2023. URL <https://geneontology.org/docs/go-enrichment-analysis/>.
- [6] Cameron Davidson-Pilon. lifelines: survival analysis in python. *Journal of Open Source Software*, 4(40):1317, 2019. doi: 10.21105/joss.01317. URL <https://doi.org/10.21105/joss.01317>.
- [7] Alexander Dobin, 2019. URL https://physiology.med.cornell.edu/faculty/skrabanek/lab/angsd/lecture_notes/STARmanual.pdf.
- [8] Maria Doyle. Galaxy training: Visualization of rna-seq results with volcano plot, May 2023. URL <https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/rna-seq-viz-with-volcanoplot/tutorial.html>.
- [9] Django Software Foundation. *Django Documentation*. Django Software Foundation, Lawrence, Kansas, 2021. URL <https://docs.djangoproject.com/>.
- [10] Laurent Gautier. rpy2. URL <https://pypi.org/project/rpy2/>.
- [11] GitHub. Osererror: Cannot load library ”\r.dll”: Error 0x7e - issue #958 - rpy2/rpy2. URL <https://github.com/rpy2/rpy2/issues/958>.
- [12] IBM. Análisis de supervivencia de kaplan-meier. URL <https://www.ibm.com/docs/es/spss-statistics/saas?topic=statistics-kaplan-meier-survival-analysis>.
- [13] Plotly Technologies Inc. *Plotly.js*. Plotly, Montreal, Quebec, Canada, 2021. URL <https://plotly.com/javascript/>.
- [14] National Cancer Institute. mirrna analysis pipeline, . URL https://docs.gdc.cancer.gov/Data/Bioinformatics_Pipelines/miRNA_Pipeline/.

- [15] National Cancer Institute. Estadísticas del cáncer, . URL <https://www.cancer.gov/espanol/cancer/naturaleza/estadisticas>.
- [16] National Cancer Institute. TCGA Metadata, 2023. URL <https://docs.cancercloud.org/docs/tcga-metadata>.
- [17] National Human Genome Research Institute. Rna (ribonucleic acid), 2023. URL <https://www.genome.gov/genetics-glossary/RNA-Ribonucleic-Acid>.
- [18] SpryMedia Ltd. *DataTables*. DataTables, Belfast, United Kingdom, 2021. URL <https://datatables.net/>.
- [19] Jacob Thornton Mark Otto. Bootstrap. URL <https://getbootstrap.com/>.
- [20] Teaching materials at the Harvard Chan Bioinformatics Core. Differential gene expression (DGE) analysis. URL https://hbctraining.github.io/Training-modules/planning_successful_rnaseq/lessons/sample_level_QC.html.
- [21] Bioconductor Project. *Bioconductor*. Bioconductor, Buffalo, NY, USA, 2021. URL <https://www.bioconductor.org/>.
- [22] Celery Project. *Celery Documentation*. Celery, Online, 2021. URL <https://docs.celeryproject.org/>.
- [23] Lu Qiong. Gene expression analysis of project tcga-chol, Apr 2022. URL https://brh.data-commons.org/dashboard/Public/notebooks/GDC_TCGA-CHOL_RNA_analysis_BRH_040722.html#Functional-enrichment-analysis.
- [24] Pere Rebasa. Conceptos básicos del análisis de supervivencia. *Cirugía Española*, 78(4):222–230, Oct 2005. doi: 10.1016/s0009-739x(05)70923-4.
- [25] Selectra. Tarifa luz hora. URL <https://tarifaluzhora.es/info/precio-kwh>.
- [26] TCGA. The Cancer Genome Atlas Program (TCGA). <https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga>, 2020.
- [27] Luis Tume, Carlos Cisneros, Josmell Sevillano, Romina Pacheco-Tapia, Daniel Matos, Román Acevedo-Espínola, Roberto Ubidia-Incio, and Wilder Rodríguez. Desregulación de microarn en el cáncer: Un enfoque terapéutico y diagnóstico. *Gaceta Mexicana de Oncología*, 15(5):298–304, Sep 2016. doi: 10.1016/j.gamo.2016.08.004.
- [28] Cristi Vîjdea. Yet another swagger generator for drf. URL <https://drf-yasg.readthedocs.io/en/stable/>.
- [29] John N. Weinstein et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013. ISSN 1061-4036. doi: 10.1038/ng.2764.
- [30] Matt Zabriskie and Contributors. *Axios*. Axios, San Francisco, CA, 2021. URL <https://axios-http.com/>.