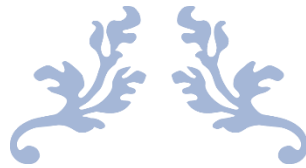




UNIVERSIDAD DE BURGOS



TFG DEL GRADO DE INGENIERÍA INFORMÁTICA - GII 22.40 – ANEXOS



ADCO

Presentado por Enrique Diez Fernández en
Universidad de Burgos el 9 de junio de 2023

Luis Rodrigo Izquierdo Millán

José Manuel Galán Ordax



Tabla de contenido

Índice de ilustraciones.....	5
Índice de tablas	6
Anexo I – Planificación de proyecto software.....	7
1.1. Introducción	7
1.2. Planificación temporal.....	7
1.2.1. Sprint 1 (24/02/2023 – 10/03/2023).....	7
1.2.2. Sprint 2 (10/03/2023 – 24/03/2023).....	8
1.2.3. Sprint 3 (24/03/2023 – 10/04/2023).....	9
1.2.4. Sprint 4 (14/04/2023 – 28/04/2023).....	9
1.2.5. Sprint 5 (27/04/2023 – 12/05/2023).....	10
1.2.6. Sprint 6 (12/05/2023 – 29/05/2023).....	11
1.2.7. Sprint 7 (29/05/2023 – 08/06/2023).....	12
1.3. Resumen.....	13
1.4. Estudio de viabilidad económica.....	15
1.5. Estudio de viabilidad legal.....	19
Anexo II – Especificación de requisitos	21
2.1. Introducción	21
2.2. Objetivos generales.....	21
2.3. Catálogo de requisitos.....	21
2.3.1. Requisitos funcionales.....	21
2.3.2. Requisitos no funcionales	23
2.4. Especificación de requisitos	24
2.5. Actores	24
2.6. Casos de uso.....	25
2.7. Diagrama de casos de uso	45
Anexo III – Especificación de diseño	47
3.1. Introducción	47
3.2. Model-View-ViewModel (MVVM).....	47
3.3. Paquetes.....	49
3.4. Diseño de interfaces.....	50
Anexo IV – Documentación técnica de programación.....	55
4.1. Introducción	55
4.2. Estructura del directorio	55

4.3.	Manual del programador	56
4.3.1.	Instalación y ejecución del proyecto en local:	56
4.3.2.	Internacionalización de la aplicación:	57
Anexo V – Documentación de usuario		59
5.1.	Introducción	59
5.2.	Requisitos de usuario	59
5.3.	Instalación	59
5.4.	Manual del usuario.....	60
5.4.1.	Introducción	60
5.4.2.	Barra de navegación.....	60
5.4.3.	Introducción de datos	61
5.4.4.	ACP	63
5.4.5.	Distancias	63
5.4.6.	Clustering	64
5.4.7.	Outliers.....	67
6.	Bibliografía	69



Índice de ilustraciones

Ilustración 1 - Tabla Sprint 1.....	8
Ilustración 2 - Tabla Sprint 2.....	8
Ilustración 3 - Tabla Sprint 3.....	9
Ilustración 4 - Tabla Sprint 4.....	10
Ilustración 5 - Tabla Sprint 5.....	11
Ilustración 6 - Tabla Sprint 6.....	12
Ilustración 7 - Tabla Sprint 7.....	13
Ilustración 8 - Número de tareas por sprint.....	14
Ilustración 9 - Agrupamiento de tareas por grupo.....	14
Ilustración 10 - Grafico de tareas por sprint	14
Ilustración 11 - Impuestos de salario	16
Ilustración 12 - Model-View-ViewModel	48
Ilustración 13 - Estructura del directorio src. Fuente: elaboración propia	49
Ilustración 14 - Pantalla inicial del prototipado.	51
Ilustración 15 - Introducción manual de datos	51
Ilustración 16 - Introducción de datos por fichero	52
Ilustración 17 - Ventana ACP.....	52
Ilustración 18 - Ventana distancias y clustering.....	53
Ilustración 19 - Ventana Outliers	53
Ilustración 20 - Menú de la aplicación	54
Ilustración 21 - Barra de navegación.....	60
Ilustración 22 - Datos de ejemplo	61
Ilustración 23 - Introducir datos paso 1	61
Ilustración 24 - Introducir datos paso 2	61
Ilustración 25 - Introducir datos paso 3	62
Ilustración 26 - Tabla final de introducir resultados	62
Ilustración 27 - Introducción del número de componentes principales.....	63
Ilustración 28 - Selector de distancias.....	64
Ilustración 29 - Selector de distancias.....	64
Ilustración 30 - Selector algoritmo clustering	65
Ilustración 31 - Ejemplo DBSCAN	65
Ilustración 32 - Ejemplo OPTICS	66
Ilustración 33 - Ejemplo k-means.....	66

Índice de tablas

Tabla 1- Licencias de las herramientas y librerías empleadas	19
Tabla 2 - Caso de uso 1.....	25
Tabla 3 - Caso de uso 2.....	25
Tabla 4 - Caso de uso 3.....	26
Tabla 5 - Caso de uso 4.....	26
Tabla 6 - Caso de uso 5.....	27
Tabla 7 - Caso de uso 6.....	27
Tabla 8 - Caso de uso 7.....	28
Tabla 9 - Caso de uso 8.....	28
Tabla 10 - Caso de uso 9.....	29
Tabla 11 - Caso de uso 10.....	29
Tabla 12 - Caso de uso 11.....	30
Tabla 13 - Caso de uso 12.....	30
Tabla 14 - Caso de uso 13.....	31
Tabla 15 - Caso de uso 14.....	31
Tabla 16 - Caso de uso 15.....	32
Tabla 17 - Caso de uso 16.....	32
Tabla 18 - Caso de uso 17.....	33
Tabla 19 - Caso de uso 18.....	33
Tabla 20 - Caso de uso 19.....	34
Tabla 21 - Caso de uso 20.....	34
Tabla 22 - Caso de uso 21.....	35
Tabla 23 - Caso de uso 22.....	35
Tabla 24 - Caso de uso 23.....	36
Tabla 25 - Caso de uso 24.....	36
Tabla 26 - Caso de uso 25.....	37
Tabla 27 - Caso de uso 26.....	37
Tabla 28 - Caso de uso 27.....	38
Tabla 29 - Caso de uso 28.....	38
Tabla 30 - Caso de uso 29.....	39
Tabla 31 - Caso de uso 30.....	39
Tabla 32 - Caso de uso 31.....	40
Tabla 33 - Caso de uso 32.....	40
Tabla 34 - Caso de uso 33.....	41
Tabla 35 - Caso de uso 34.....	41
Tabla 36 - Caso de uso 35.....	42
Tabla 37 - Caso de uso 36.....	42
Tabla 38 - Caso de uso 37.....	43
Tabla 39 - Caso de uso 38.....	43
Tabla 40 - Caso de uso 39.....	44
Tabla 41 - Diagrama de casos de uso	45



Anexo I – Planificación de proyecto software

1.1. Introducción

En este primer anexo, se busca explicar el ciclo de desarrollo del proyecto mediante una planificación temporal. Además, se presenta la viabilidad económica del proyecto, incluyendo una aproximación de los costos asociados a su desarrollo. Por último, se realiza un estudio de viabilidad legal, considerando la legislación que pueda afectar al proyecto.

1.2. Planificación temporal

En este apartado, se describe el desarrollo del proyecto aplicando una metodología ágil, específicamente SCRUM. Se realizaron reuniones cada dos semanas para revisar los avances, evaluar el estado actual del proyecto y discutir mejoras o futuras funcionalidades.

El proceso comenzó con una reunión inicial en la que se presentó la idea del proyecto y se resumieron todas las funcionalidades que debían implementarse. En esta reunión, se acordaron las técnicas y herramientas que se utilizarían en el desarrollo.

Después de la reunión inicial, se me concedió una semana para familiarizarme con el entorno, las técnicas y herramientas, así como el tema a desarrollar.

Durante los siguientes meses, se llevaron a cabo reuniones cada dos semanas, programadas los viernes, para asegurar que cada sprint tuviera una duración adecuada que permitiera implementar mejoras significativas en la aplicación.

En cuanto al seguimiento de los sprints, inicialmente se planteó utilizar la extensión de GitHub llamada ZenHub, que permite visualizar las tareas planificadas en forma de sprints. Sin embargo, después de un mes utilizando esta extensión, se decidió cambiar a Trello, una plataforma similar pero gratuita, que resultó más cómoda y fácil de familiarizarse con ella.

1.2.1. Sprint 1 (24/02/2023 – 10/03/2023)

Las primeras dos semanas del proyecto, se enfocaron en la configuración inicial del proyecto en GitHub, la elección del entorno de desarrollo y la plataforma de implementación de la aplicación, así como la selección de las herramientas que ayudarían en el desarrollo.

Los primeros días, se centraron en crear un prototipo de la página web, que sirviera como referencia y guía para desarrollar los primeros apartados de la página. Se comenzó por generar la sección de información y la funcionalidad de introducción de datos manuales, ya que forman la base fundamental de la aplicación.

Además, se inició la documentación de las herramientas utilizadas, con el objetivo de evitar retrasos en esta parte que podrían haberse convertido en una carga de trabajo significativa en etapas posteriores del proyecto.

Planificación temporal:

	Documentación	Programación	Prototipado	pruebas	PDF	Lectura	Excel
Creación introducción de datos manual							
Portada de la aplicación e introducción							
Creación de la base de la aplicación							
Elección de herramientas para repositorio							
Elección de editor de texto							
Elección de herramienta para bibliografía							
Elección de software para la documentación automática							
Elección de herramienta de prototipado							
Creación del prototipo de la web							
Lectura y entendimiento de ACP							
Documentación de Mendeley							
Documentación de ZenHub							
Documentación de Angular							
Documentación de Netlify							
ocumentación de VSCode							

Ilustración 1 - Tabla Sprint 1

1.2.2. Sprint 2 (10/03/2023 – 24/03/2023)

Durante el segundo sprint, el enfoque estuvo en desarrollar la sección de análisis de componentes principales en la aplicación y su respectiva documentación.

En los primeros días, me centré investigar y comprender en qué consistía el análisis de componentes principales y cómo se podría incorporar en la aplicación. Durante esta fase de investigación, descubrió que ya existían librerías creadas por otros desarrolladores que cumplían con las funciones requeridas, en la página de librerías npm. Esto permitió aprovechar el trabajo previo realizado y evitar tener que desarrollar los algoritmos desde cero.

En lugar de ser un sprint en el que se debía implementar algoritmos, se trató más bien de comprender cómo funcionaban estas librerías y cómo se podían integrar en el código del proyecto. Esto implicó estudiar la documentación de las librerías encontradas y adaptarlas a las necesidades específicas de la aplicación.

Planificación temporal:

	Documentación	Programación	Prototipado	pruebas	PDF	Lectura	Excel
Funcionalidad de insertar datos por Excel							
Documentar ACP							
Buscar librerías ACP							
Incorporar librerías ACP							
Documentar librerías ACP							
Prototipado de ACP							
Creación del apartado de ACP							
Búscar librerías de gráficos							
Incorporar gráficos en ACP							
Documentar librerías de graficos							

Ilustración 2 - Tabla Sprint 2



1.2.3. Sprint 3 (24/03/2023 – 10/04/2023)

Durante este sprint se realizaron cortes en los periodos de trabajo debido a las vacaciones de Semana Santa. A pesar de ello, se lograron avances significativos en el proyecto.

En primer lugar, se llevó a cabo el cambio de la herramienta de seguimiento de tareas de ZenHub a Trello. Esta transición se realizó exitosamente, lo que permitió mantener un mejor control y organización de las tareas del proyecto.

Además, se finalizó el apartado de Análisis de Componentes Principales (ACP) en términos estéticos; es decir, se trabajó en la apariencia visual de la sección. También se implementaron algunas funcionalidades adicionales que se acordaron durante el sprint, mejorando así la experiencia del usuario en esa parte de la aplicación.

A pesar de las interrupciones ocasionadas por las vacaciones, se logró avanzar de manera efectiva en el proyecto durante este sprint.

Planificación temporal:

	Documentación	Programación	Prototipado	Pruebas	PDF	Lectura	Excel
Cambio a Trello							
Rediseño ACP							
Rediseño gráficas ACP							
Estandarizar datos							
Aplicar notaciones científicas							
Documentación ACP							

Ilustración 3 - Tabla Sprint 3

1.2.4. Sprint 4 (14/04/2023 – 28/04/2023)

Durante el cuarto sprint se llevó a cabo la incorporación del apartado de cálculo de distancias en la aplicación. Este es uno de los aspectos más importantes, ya que los algoritmos de clustering y detección de outliers utilizarán estos datos para realizar sus propios cálculos.

En primer lugar, se dedicaron varios días a buscar librerías en npm que pudieran simplificar el proceso de cálculo de distancias. Sin embargo, la mayoría de las librerías encontradas no resultaban satisfactorias, ya sea porque incorporaban funcionalidades complicadas de implementar o porque los cálculos realizados no eran precisos.

Ante esta situación, se optó por incorporar las distancias euclídeas, euclídea normalizada y de Mahalanobis en el proyecto mediante el uso de pseudocódigo y entendiendo cómo se realizan estos cálculos. Esto implicó estudiar y comprender los algoritmos necesarios para cada tipo de distancia, y luego implementarlos de manera adecuada en el código del proyecto.

A pesar de las dificultades encontradas con las librerías existentes, se logró avanzar en el sprint al realizar la incorporación de los cálculos de distancias de manera manual. Esto permitió garantizar la precisión y adecuación de los resultados obtenidos.

Planificación temporal:

	Documentación	Programación	Prototipado	pruebas	PDF	Lectura	Excel
Documentar distancias (Mahalanobis, Euclídea y Euclídea normalizada)							
Documentar Clustering							
Reajustar diseño ACP							
Buscar librerías y algoritmos de distancias							
Creación tabla para mostrar distancias en función del tipo.							
Modificar los pipes de notación científica							
Cambiar nombres de los botones al usarse							
Comprobaciones de los cálculos de distancias							
Finalizar documentación ACP							

Ilustración 4 - Tabla Sprint 4

1.2.5. Sprint 5 (27/04/2023 – 12/05/2023)

Durante este sprint se dedicó tiempo al análisis de clustering de los datos iniciales, y es probablemente el sprint en el que se ha agregado más contenido a la aplicación.

En primer lugar, se realizó el análisis de los diferentes tipos de clustering que se iban a implementar, como DBSCAN, OPTICS y K-means. Se dedicó tiempo a comprender su funcionamiento, algoritmos y características. Esto implicó estudiar en profundidad cada método y su aplicación en el contexto del proyecto.

Además, se implementaron estos algoritmos de clustering utilizando pseudocódigo. Se trabajó en la traducción de los algoritmos a código funcional dentro de la aplicación, asegurándose de que se aplicaran correctamente a los datos y se obtuvieran los resultados esperados.

Por otra parte, se introdujo la automatización para generar archivos en formato Excel, lo cual permitiría al usuario exportar los resultados obtenidos en los diferentes apartados de la aplicación. Esto brinda una funcionalidad adicional y facilita la visualización y el manejo de los datos generados.

En resumen, este sprint fue dedicado al análisis de clustering, su implementación y la introducción de la automatización para generar archivos en formato Excel, ampliando así las capacidades y utilidad de la aplicación.



Planificación temporal:

	Documentación	Programación	Prototipado	pruebas	PDF	Lectura	Excel
Modificar cálculo de distancias a nivel matricial							
Buscar librerías y algoritmos de clustering							
Crear gráficas para clustering							
Implementar DBSCAN - Clustering							
Test DBSCAN - Clustering							
Implementar OPTICS - Clustering							
Test OPTICS - Clustering							
Aumentar cantidad de colores para clustering							
Implementar k-means - Clustering							
Test k-means - Clustering							
Guardado de los datos iniciales en Excel							
Guardado de distancias en Excel							
Añadir cabeceras a los Excel							
Guardar clustering en PDF							
Comprobar funcionamiento de Excel							
Comprobar funcionamiento de PDF							

Ilustración 5 - Tabla Sprint 5

1.2.6. Sprint 6 (12/05/2023 – 29/05/2023)

Este sprint, se ha enfocado en la documentación de los algoritmos de clustering empleados anteriormente, asegurándose de que estén debidamente documentados y explicados para facilitar su comprensión y futuras modificaciones.

Además, se ha comenzado la implementación de los algoritmos de detección de outliers. Se ha investigado y estudiado los algoritmos pertinentes, comprendiendo su funcionamiento y adaptándolos a la aplicación. Se ha trabajado en la integración de estos algoritmos en el código del proyecto.

Paralelamente, se han continuado realizando pruebas de carga sobre el programa, buscando optimizar su rendimiento y asegurándose de que pueda manejar grandes volúmenes de datos de manera eficiente. También se han realizado modificaciones mínimas en la estética de la aplicación, con el objetivo de crear un producto visualmente atractivo y estético.

Finalmente, para cerrar el sprint, se ha trabajado en la parte teórica de la memoria del proyecto y en la elaboración de los diferentes anexos. Esto implica documentar y explicar de manera clara y concisa los aspectos teóricos y técnicos del proyecto, así como los resultados obtenidos y las conclusiones relevantes.

En resumen, este sprint ha estado centrado en la documentación de algoritmos, la implementación de algoritmos de detección de outliers, pruebas de carga, mejoras estéticas y la elaboración de la memoria y anexos correspondientes.

Planificación temporal:

	Documentación	Programación	Prototipado	pruebas	PDF	Lectura	Excel
Documentación DBSCAN - Clustering							
Documentación OPTICS - Clustering							
Documentación k-means - Clustering							
Separar Distancias de Clustering							
Minimizar tabla de Distancias							
Implementar apartado Outliers							
Implementar Mahalanobis - Outliers							
Test Mahalanobis							
Implementar DBSCAN - Outliers							
Test DBSCAN							
Implementar OPTICS - Outliers							
Test OPTICS							
Implementar LOF - Outliers							
Test LOF							
Implementar k-nearest							
Test k-nearest							
Modificar botones de navegación							
Sacar resultados de outliers por PDF							
Corregir PDF para ficheros muy grandes							
Corregir ventana de introducción							
Añadir ejemplos de datos							
Documentación Outliers							
Documentación Mahalanobis - Outliers							
Documentación DBSCAN - Outliers							
Documentación OPTICS - Outliers							
Documentación LOF - Outliers							
Documentación k-nearest - Outliers							

Ilustración 6 - Tabla Sprint 6

1.2.7. Sprint 7 (29/05/2023 – 08/06/2023)

Durante este, me he enfocado en actividades clave como la documentación, generación de anexos, finalización de la memoria y pruebas de mi aplicación web. Aproveché este período para consolidar y concluir los aspectos importantes del proyecto, asegurando una entrega de calidad y lista para su implementación.

En los primeros días del sprint, me dediqué a la documentación detallada de los avances y resultados obtenidos hasta el momento. Elaboré informes, análisis y registros exhaustivos para documentar cada etapa del proyecto. Además, generé anexos que complementan y respaldan la información principal, brindando una visión completa y transparente del trabajo realizado.

Simultáneamente, me enfoqué en la finalización de la memoria del proyecto. Revisé y pulí el contenido existente, asegurándome de que estuviera completo y bien estructurado. También añadí los detalles finales, como conclusiones, resultados, lecciones aprendidas y posibles líneas de trabajo futuras. Me esforcé en presentar una memoria sólida que refleje el trabajo realizado y pueda ser utilizada como referencia en el futuro.

Además de la documentación, realicé pruebas exhaustivas en la aplicación web. Verifiqué que todas las funcionalidades estén implementadas correctamente y se comporten de acuerdo a los requisitos establecidos.



Planificación temporal:

	Documentación	Programación	Prototipado	pruebas	PDF	Lectura	Excel
Memoria - Conclusiones y líneas de trabajo futuras							
Memoria - Trabajos relacionados							
Memoria - Aspectos relevantes del desarrollo del proyecto							
Memoria - Introducción							
Memoria - Objetivos del proyecto							
Anexo I - Planificación del Proyecto Software							
Anexo II - Especificación de Requisitos							
Anexo III - Especificación de Diseño							
Anexo IV - Documentación técnica de programación							
Anexo V - Documentación de usuario							

Ilustración 7 - Tabla Sprint 7

1.3. Resumen

Durante el transcurso de los diferentes sprints, se ha observado un aumento gradual en la carga de trabajo, lo cual puede estar relacionado con el tiempo invertido en el proyecto. Esto se debe, en gran medida, a la familiarización con el entorno de trabajo y las tecnologías utilizadas. A medida que se adquiere más experiencia, se tiende a ser más eficiente en la realización de tareas específicas.

Además, se ha notado que en los últimos sprints se han desglosado más las tareas, con el objetivo de evitar la complejidad excesiva en las funcionalidades implementadas. Esta descomposición permite abordar de manera más efectiva cada tarea y asegurar un avance constante en el desarrollo del proyecto.

Los gráficos proporcionados por Trello reflejan claramente esta evolución en la carga de trabajo. En la imagen mencionada (ilustración 8), se puede observar cómo el sprint 6 se ha dividido en dos partes, lo que indica una mejor planificación y distribución de las tareas para mantener un progreso constante.

En resumen, a lo largo de los sprints se ha observado un aumento progresivo en la carga de trabajo, debido a la familiarización con las tecnologías y la experiencia ganada. Además, se ha aplicado una estrategia de desglose de tareas para evitar complejidades innecesarias y mantener un ritmo de trabajo constante.

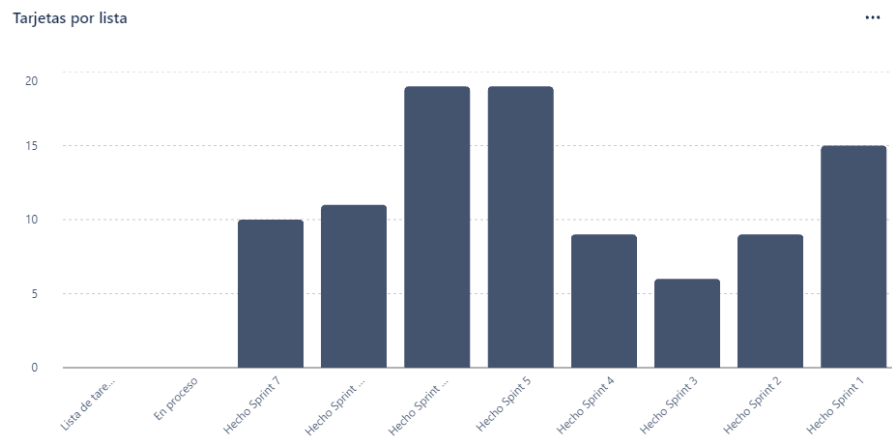


Ilustración 8 - Número de tareas por sprint

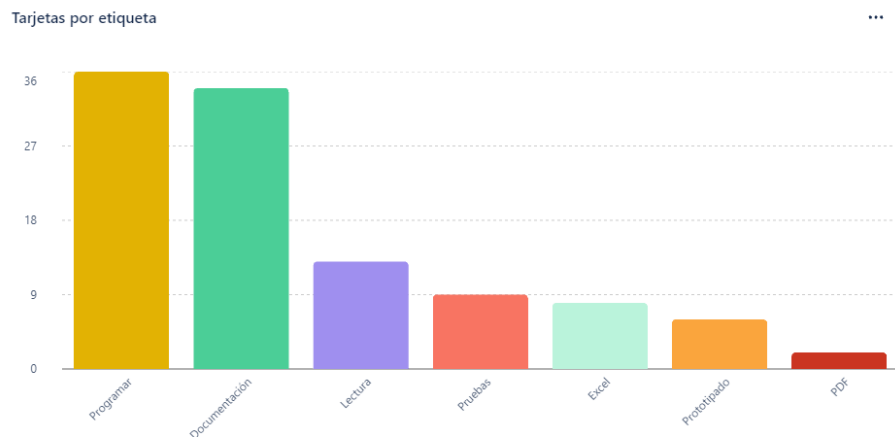


Ilustración 9 - Agrupamiento de tareas por grupo

Tarjetas por lista

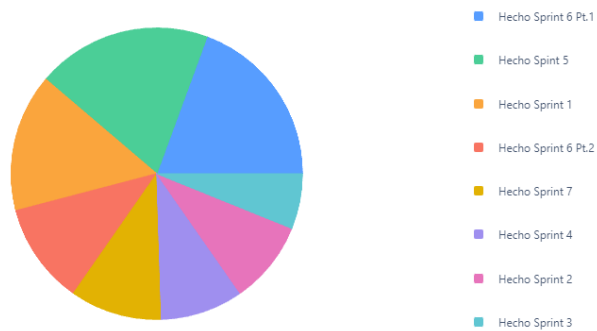


Ilustración 10 - Grafico de tareas por sprint



1.4. Estudio de viabilidad económica

En el estudio de costes y beneficios de esta aplicación, se deben tener en cuenta diferentes elementos para calcular los costes asociados a su desarrollo y funcionamiento. Los costes considerados son los siguientes:

1. Empleados: Este incluye los gastos asociados a los recursos humanos necesarios para desarrollar, mantener y administrar la aplicación. Esto puede involucrar desarrolladores, diseñadores y personal de soporte técnico, entre otros. Los costes de los empleados incluyen salarios, beneficios, impuestos y otros gastos relacionados con el personal.
2. Hardware: Se deben considerar los costes relacionados con la infraestructura tecnológica necesaria para ejecutar la aplicación. Esto puede incluir servidores, equipos de red, almacenamiento, dispositivos móviles u otros dispositivos necesarios para el desarrollo y despliegue de la aplicación.
3. Software: Los costes de software incluyen las licencias de software y herramientas necesarias para el desarrollo, pruebas y operación de la aplicación. Esto puede incluir licencias de sistemas operativos, bases de datos, herramientas de desarrollo, software de seguridad, entre otros.

Es importante tener en cuenta que estos costes pueden variar según las necesidades y requisitos específicos de la empresa que desarrolle la aplicación. Además, es posible que haya otros costes indirectos, como costes de formación del personal, costes de marketing y publicidad, gastos legales, etc., que también deben ser considerados en el estudio de costes.

Es fundamental realizar un análisis detallado de los costes asociados a cada elemento mencionado, teniendo en cuenta tanto los costes iniciales como los costes recurrentes a lo largo del tiempo, para obtener una estimación precisa de los gastos totales relacionados con el desarrollo y operación de la aplicación.

Empleados:

El proyecto ha sido desarrollado únicamente por una persona, con una estimación de 300 horas de trabajo durante 14 semanas, a un promedio de 21,42 horas por semana.

Considerando un precio medio de 20€ brutos por hora para un recién graduado en ingeniería informática, el cálculo del coste bruto laboral mensual sería el siguiente:

$$21,42 \frac{\text{horas}}{\text{semana}} * 20 \frac{\text{€}}{\text{hora}} * 4 \frac{\text{semanas}}{\text{mes}} = 1714\text{€ al mes}$$

Por lo tanto, el coste de los empleados en este caso sería de aproximadamente 1714€ al mes.

Cabe mencionar que este cálculo es una estimación basada en las suposiciones proporcionadas, y los costes reales pueden variar según el contexto y las condiciones específicas de la empresa o proyecto. También es importante tener en cuenta que este cálculo

solo considera el coste de los empleados y no incluye otros gastos asociados, como el hardware, software u otros costes indirectos mencionados anteriormente

Esto se corresponde con el salario bruto del empleado, luego a esto se le añaden los impuestos que tiene que pagar la empresa. La siguiente información ha sido obtenida de [aquí](#). Estos impuestos se corresponden con:

- 26,3% de contingencias.
- 5,5% de desempleo.
- 0.20% de FOGASA.
- 0.60% de formación profesional.

TIPOS DE COTIZACIÓN (%)			
CONTINGENCIAS	EMPRESA	TRABAJADORES	TOTAL
Comunes	23,60	4,70	28,30
Horas Extraordinarias Fuerza Mayor	12,00	2,00	14,00
Resto Horas Extraordinarias	23,60	4,70	28,30

(1) Tipo de contingencias comunes (II): Trabajadores con 65 años y 0 a 7 meses de edad y 36 años y 9 meses o más de cotización ó 65 años y 8 meses ó más de edad y 35 años y 6 meses o más de cotización: 1,50 por 100 (1,25 por 100 - empresa- y 0,25 por 100 -trabajador-).

(2) En los contratos temporales de duración igual o inferior a cinco días, la cuota empresarial por contingencias comunes se incrementa en un 40 por ciento. No se aplica al Sistema Especial para Trabajadores por Cuenta Ajena Agrario, incluido en el Régimen General.

DESEMPLEO	EMPRESA	TRABAJADORES	TOTAL
Tipo General	5,50	1,55	7,05
Contrato duración determinada Tiempo Completo	6,70	1,60	8,30
Contrato duración determinada Tiempo Parcial	6,70	1,60	8,30

	EMPRESA	TRABAJADORES	TOTAL
FOGASA	0,20		0,20

	EMPRESA	TRABAJADORES	TOTAL
FORMACIÓN PROFESIONAL	0,60	0,10	0,70

Ilustración 11 - Impuestos de salario

Por lo que, si tenemos en cuenta los siguientes impuestos, nos sale que la empresa debería de pagar por un empleado:

$$1714 \frac{\text{€}}{\text{mes}} (1 + (0,236 + 0,055 + 0,002 + 0,006)) = 2226\text{€/mes}$$

Por otra parte, también se cuenta con dos profesores tutelando al alumno, con amplios conocimientos en el tema, lo que supondremos que cuesta unos 40€/hora. Los tutores trabajan unas 2 horas cada dos semanas (es decir, 1 hora a la semana). Por tanto:

$$1 \frac{\text{horas}}{\text{semana}} * 40 \frac{\text{€}}{\text{hora}} * 4 \frac{\text{semanas}}{\text{mes}} = 160\text{€ al mes por profesor}$$



Por lo tanto, si aplicamos los impuestos al grupo de los dos profesores:

$$320 \frac{\text{€}}{\text{mes}} (1 + (0,236 + 0,055 + 0,002 + 0,006)) = 416\text{€/mes}$$

De esta forma, finalmente obtenemos que la empresa deberá de abonar un coste mensual de 2642€ mensuales, lo que supone un gasto total del proyecto de aproximadamente 9247€.

Hardware:

En términos de hardware, el único gasto asociado ha sido el ordenador portátil utilizado para llevar a cabo el proyecto. En su momento, el costo de adquisición de este equipo fue de 900€. Sin embargo, teniendo en cuenta su uso continuo y el tiempo transcurrido desde su compra, se puede considerar que dicho costo ya ha sido amortizado. Por lo tanto, en este apartado, no se ha incurrido en ningún gasto adicional, ya que el equipo ya estaba totalmente amortizado en términos contables.

La utilización de un ordenador portátil existente ha permitido optimizar los recursos disponibles y minimizar los gastos en hardware. Aprovechar al máximo el equipo previamente adquirido es una práctica eficiente desde el punto de vista económico, ya que evita la necesidad de invertir en nuevas adquisiciones.

En conclusión, no ha sido necesario incurrir en gastos adicionales en hardware para llevar a cabo este proyecto, ya que se ha utilizado un ordenador portátil que ya estaba en posesión y obsoleto en términos contables. Esto ha contribuido a mantener los costos bajo control y maximizar el valor obtenido de los recursos existentes.

Software:

En cuanto al software utilizado en el proyecto, la mayoría de las herramientas y recursos han sido de uso gratuito. Sin embargo, es importante mencionar que se ha requerido una clave de activación para el sistema operativo Windows, con un costo de 145€, la cual ya poseía para uso personal. Además, se ha utilizado una licencia de Office para la documentación y pruebas con ficheros, la cual también estaba previamente disponible a través de la cuenta de la UBU y no ha sido considerada como un gasto adicional en este proyecto.

Al aprovechar recursos previamente adquiridos y utilizar software gratuito en su mayoría, se ha logrado minimizar los costos asociados al software en el desarrollo de esta aplicación. Esta estrategia ha permitido aprovechar eficientemente los recursos disponibles y maximizar el valor obtenido de las herramientas utilizadas.

Es importante destacar que la utilización de software con licencias adecuadas es fundamental para garantizar la legalidad y el cumplimiento de los derechos de autor. Al contar con las licencias necesarias para el sistema operativo y el software de Office, se ha asegurado el uso adecuado de estos recursos en el proyecto.

En resumen, se ha buscado minimizar los gastos relacionados con el software mediante el uso de herramientas gratuitas y el aprovechamiento de licencias previamente adquiridas. Esto ha permitido mantener los costos bajo control y cumplir con los requisitos legales correspondientes.

Total:

El gasto total, suponiendo que no ha habido gastos de hardware ni de software en términos contables, va a ser únicamente coste de empleados. No obstante, también se va a suponer un coste de un 15% por costes indirectos tales como electricidad, internet, uso de infraestructura, etc.

De esta forma podemos obtener un gasto total del proyecto como:

$$9247\text{€} * 1,15\% = 10634,05\text{€}$$

Beneficios:

Aunque económicamente no se esperan beneficios debido a que la aplicación se distribuirá de forma gratuita a la comunidad científica bajo una licencia MIT, personalmente este proyecto ha sido una valiosa oportunidad para adentrarme en el mundo profesional y adquirir experiencia en diversos lenguajes, entornos y herramientas utilizados durante todo el proceso de desarrollo.

Al compartir la aplicación de forma gratuita con la comunidad científica, se abre la puerta a otros tipos de beneficios. A través de esta iniciativa, espero obtener reconocimiento y visibilidad en el ámbito científico, así como la posibilidad de establecer colaboraciones y recibir valiosa retroalimentación. Estas interacciones pueden enriquecer mi experiencia y ampliar mi red profesional, lo cual es fundamental para mi crecimiento a largo plazo.

Aunque el enfoque no es obtener beneficios económicos directos, el valor de esta experiencia no se puede subestimar. Esta oportunidad me ha permitido fortalecer mis habilidades y conocimientos, y me ha brindado un sólido punto de partida para futuros proyectos. Además, el espíritu de contribución a la comunidad científica es una motivación adicional, ya que puedo aportar mi granito de arena en la promoción del avance científico y tecnológico.

En resumen, aunque los beneficios económicos sean limitados, el valor personal y profesional obtenido a través de este proyecto es significativo. La experiencia adquirida, las conexiones establecidas y el impacto potencial en la comunidad científica son recompensas valiosas que trascienden las ganancias financieras.



1.5. Estudio de viabilidad legal

Para este apartado se va a proceder al estudio de las licencias que poseen las librerías y herramientas utilizadas para el desarrollo del proyecto, así como la propia licencia que se establecerá para el proyecto en sí.

Para este proyecto se han utilizado las siguientes herramientas y/o librerías:

Herramienta y/o librería	Versión	Licencia
Angular	13.3.12	MIT
Html2canvas	1.4.1	MIT
jspdf	2.0.0	MIT
Bootstrap	5.2.3	MIT
Chart.js	3.9.1	MIT
Jquery	2.5.1	MIT
math	0.0.3	Nativo de JS
Math.js	1.1.46	MIT
matrix	1.0.3	MIT
ml-pca	4.1.1	MIT
Ng2-charts	3.1.2	MIT
npm	9.6.0	MIT
rxjs	7.8.0	Apache 2.0.
sheetsjs	2.0.0	MIT
xlsx	0.18.5	MIT

Tabla 1- Licencias de las herramientas y librerías empleadas

Es evidente que todas las herramientas empleadas en la ejecución del proyecto se encuentran disponibles para su uso sin restricciones. Por consiguiente, no existen limitaciones que justifiquen la aplicación de una licencia más restrictiva en nuestro proyecto.

Es por esta razón que se ha tomado la decisión de otorgar una licencia MIT al proyecto. Esto implica que la aplicación puede ser aprovechada en todos sus aspectos, permitiendo su uso, modificación, copia, publicación, venta y distribución de manera libre.

La licencia MIT brinda un marco de flexibilidad y libertad a los usuarios, alentando la colaboración, la adaptación y la expansión del proyecto. De esta manera, se fomenta la participación de la comunidad científica en la mejora continua de la aplicación y se promueve su utilidad y accesibilidad para aquellos que deseen beneficiarse de ella.

La elección de la licencia MIT refleja nuestro compromiso de ofrecer un producto de calidad a la comunidad científica, al tiempo que brindamos la oportunidad de utilizarlo y compartirlo de manera amplia y abierta.



Anexo II – Especificación de requisitos

2.1. Introducción

En esta sección, se detallarán los objetivos y requisitos que la aplicación debe cumplir, los cuales fueron establecidos desde el inicio del proyecto. Además de estos, a lo largo del desarrollo también se han incorporado algunos nuevos.

Este anexo incluye además las siguientes secciones:

- **Objetivos generales:** se establecerán los objetivos que se pretenden lograr en este proyecto.
- **Catálogo de requisitos:** se mencionarán todos los requisitos planteados en el proyecto.
- **Especificación de requisitos:** se incluirán los casos de uso y sus definiciones correspondientes.

2.2. Objetivos generales

En este proyecto, se han establecido los siguientes objetivos generales:

1. Desarrollar una aplicación capaz de realizar análisis de componentes principales, cálculo de distancias, clustering y detección de outliers sobre un conjunto de datos proporcionados por el usuario.
2. Facilitar al usuario la introducción de datos en la aplicación.
3. Brindar al usuario la posibilidad de visualizar los resultados obtenidos de manera intuitiva y sencilla.
4. Permitir al usuario guardar los resultados obtenidos en forma de archivos.

2.3. Catálogo de requisitos

En este apartado se van a exponer los requisitos funcionales y no funcionales del proyecto.

2.3.1. Requisitos funcionales

RF1. – Información de la aplicación.

RF2. – Acceso al repositorio de GitHub.

RF3. - Introducción de datos.

RF3.1. – Introducción de datos manualmente.

RF3.2. – Introducción de datos por fichero.

RF3.3. – Borra datos introducidos.

RF3.4. – Restaurar datos borrados.

RF3.5. – Exportar datos introducidos.

RF4. – Análisis de componentes principales (ACP)

RF4.1. – Establecer número de componentes principales

RF4.2. – Cálculo de Componentes Principales.

RF4.3. – Tablas con resultados del análisis.

RF4.4. – Gráfico con resultados.

RF4.5. – Exportar resultados de ACP a Excel.

RF5. – Cálculo de distancias

RF5.1. – Cálculo de distancia euclídea.

RF5.2. – Cálculo de distancia euclídea normalizada.

RF5.3. – Cálculo de distancia de Mahalanobis.

RF5.4. – Mostrar tabla resultado de distancias.

RF5.5. – Exportar resultados de distancias a Excel.

RF6. – Cálculo de clustering

RF6.1. – Elección del tipo de distancia a aplicar.

RF6.2. – Elección del algoritmo de clustering a usar.

RF6.3. – Cálculo de DBSCAN.

RF6.4. – Cálculo de OPTICS.

RF6.5. – Cálculo de K-means.

RF6.6. – Gráfico con resultados de agrupación.

RF6.7. – Tabla con resultados de agrupación.

RF6.8. – Exportar resultados de distancias a Excel.

RF7. – Cálculo de outliers

RF7.1. – Elección del tipo de algoritmo a aplicar.

RF7.2. – Cálculo de Mahalanobis.

RF7.3. – Cálculo de DBSCAN.

RF7.4. – Cálculo de OPTICS.

RF7.5. – Cálculo de KNN.

RF7.6. – Cálculo de LOF.



RF7.7. – Tabla con resultados de outliers.

RF7.8. – Exportar resultados de outliers a Excel.

RF8. – Cambiar el idioma de la aplicación

2.3.2. Requisitos no funcionales

RNF1: Usabilidad

La aplicación web debe ser intuitiva y fácil de usar, con una interfaz de usuario amigable y clara.

RNF2: Rendimiento

La aplicación web debe ser eficiente y capaz de manejar grandes volúmenes de datos sin retrasos significativos.

RNF3: Escalabilidad

La aplicación web debe poder adaptarse y manejar un aumento en la carga de trabajo sin que su rendimiento se vea perjudicado significativamente.

RNF4: Seguridad

La aplicación web debe garantizar la seguridad y protección de los datos ingresados por los usuarios.

RNF5: Portabilidad

La aplicación web debe ser compatible y funcionar correctamente en diferentes navegadores y dispositivos.

RNF6: Mantenibilidad

La aplicación web debe ser fácil de mantener, actualizar y mejorar en el futuro.

RNF7: Disponibilidad

La aplicación web debe estar disponible y accesible para los usuarios en todo momento, con un tiempo de inactividad mínimo.

2.4. Especificación de requisitos

En este apartado se va a descomponer cada caso de uso citado anteriormente dentro de la aplicación. Para ello se va a establecer una frecuencia de uso de cada uno de los casos que determinará de esta forma su importancia dentro de la aplicación.

- **Baja:** Un caso de uso se considera de baja frecuencia cuando se utiliza en menos del 30% de las veces que se usa la aplicación.
- **Media:** Un caso de uso se considera de frecuencia media cuando se utiliza entre el 30% y el 60% de las veces que se usa la aplicación.
- **Alta:** Un caso de uso se considera de alta frecuencia cuando se utiliza entre el 60% y el 100% de las veces que se usa la aplicación.

2.5. Actores

En la aplicación, se considera al usuario como el único actor principal. Esto se debe a que la aplicación está diseñada para ser utilizada directamente por él, ya que es él quien interactuará con ella para realizar diversas tareas y obtener resultados específicos. El usuario desempeña un papel fundamental al proporcionar los datos de entrada, tomar decisiones sobre los algoritmos y parámetros a utilizar, y analizar los resultados generados por la aplicación.

Si bien la propia aplicación puede tener cierta lógica interna y procesos automatizados, considero que su rol es secundario en comparación con la interacción y control del usuario. La aplicación actúa como una herramienta tecnológica que facilita las tareas de análisis de datos, cálculos y generación de resultados, pero no tiene autonomía ni capacidad de tomar decisiones por sí misma.



2.6. Casos de uso

Caso de uso 1	Información de la aplicación.	
Requisitos asociados	RF1	
Descripción	Proporcionar información detallada sobre la aplicación, sus características y funcionalidades.	
Precondición	N/A	
Acciones	Paso	Acción
	1	El usuario accede a la página.
	2	El usuario visualiza la información.
Postcondición	Ninguna	
Excepciones	Esta pantalla no tiene ninguna excepción.	
Importancia	Media	
Frecuencia	Alta	

Tabla 2 - Caso de uso 1

Caso de uso 2	Acceso al repositorio de GitHub.	
Requisitos asociados	RF2	
Descripción	Permitir el acceso directo al repositorio de GitHub asociado a la aplicación.	
Precondición	N/A	
Acciones	Paso	Acción
	1	El usuario pulsa el botón de GitHub.
	2	El usuario es redireccionado al enlace.
Postcondición	Ninguna	
Excepciones	Esta pantalla no tiene ninguna excepción.	
Importancia	Media	
Frecuencia	Baja	

Tabla 3 - Caso de uso 2

Caso de uso 3	Introducción de datos.	
Requisitos asociados	RF3	
Descripción	Permitir al usuario ingresar datos en la aplicación.	
Precondición	Haber pulsado el botón de empezar.	
Acciones	Paso	Acción
	1	Seleccionar el método de introducción.
Postcondición	Ninguna	
Excepciones	Esta pantalla no tiene ninguna excepción.	
Importancia	Alta	
Frecuencia	Alta	

Tabla 4 - Caso de uso 3

Caso de uso 4	Introducción manual de datos.	
Requisitos asociados	RF3, RF3.1	
Descripción	Permitir al usuario ingresar datos directamente en la interfaz de la aplicación, mediante formularios o campos designados.	
Precondición	Estar en la pestaña de introducir datos.	
Acciones	Paso	Acción
	1	Introducir el número de variables.
	2	Introducir el nombre de cada variable.
	3	Introducir los valores de las variables.
	4	Revisar que los datos sean correctos.
Postcondición	Debe de existir un mayor número de datos que de variables.	
Excepciones	1. Menor número de datos que de variables.	
Importancia	Alta	
Frecuencia	Alta	

Tabla 5 - Caso de uso 4



Caso de uso 5	Introducción de datos por fichero.	
Requisitos asociados	R.F3, RF3.2	
Descripción	Permitir al usuario cargar un archivo con los datos para su procesamiento.	
Precondición	Estar en la pestaña de introducir datos y que el fichero tenga el formato correcto.	
Acciones	Paso	Acción
	1	Pulsar el botón para cargar el archivo.
	2	Escoger el archivo a subir
	3	Revisar que los datos sean correctos.
Postcondición	Debe de existir un mayor número de datos que de variables.	
Excepciones	1. Menor número de datos que de variables.	
Importancia	Alta	
Frecuencia	Alta	

Tabla 6 - Caso de uso 5

Caso de uso 6	Borrar datos introducidos.	
Requisitos asociados	RF3, RF3.1, RF3.2, RF3.3	
Descripción	Proporcionar una opción para eliminar los datos ingresados previamente.	
Precondición	Existencia de datos	
Acciones	Paso	Acción
	1	Pulsar el botón para eliminar una fila de datos.
	2	Revisar que el borrado se ha aplicado correctamente.
Postcondición	Debe de existir un mayor número de datos que de variables.	
Excepciones	1. Menor número de datos que de variables.	
Importancia	Baja	
Frecuencia	Baja	

Tabla 7 - Caso de uso 6

Caso de uso 7	Restaurar datos borrados.	
Requisitos asociados	RF3, RF3.1, RF3.2, RF3.3, RF3.4.	
Descripción	Permitir al usuario recuperar los datos eliminados previamente, en caso de que sea necesario.	
Precondición	Haber realizado algún borrado.	
Acciones	Paso	Acción
	1	Pulsar el botón para restaurar los datos eliminados.
	2	Revisar que la restauración se ha aplicado correctamente.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Baja	
Frecuencia	Baja	

Tabla 8 - Caso de uso 7

Caso de uso 8	Exportar datos introducidos.	
Requisitos asociados	RF3, RF3.1, RF3.2, RF3.5.	
Descripción	Ofrecer la posibilidad de guardar los datos ingresados en un archivo externo para su posterior uso o análisis.	
Precondición	Haber introducido algún dato.	
Acciones	Paso	Acción
	1	Pulsar el botón para guardar los datos.
	2	Aceptar la descarga.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 9 - Caso de uso 8



Caso de uso 9	Análisis de componentes principales.	
Requisitos asociados	RF4	
Descripción	Realizar un análisis de componentes principales sobre los datos ingresados.	
Precondición	Haber pulsado el botón de ACP.	
Acciones	Paso	Acción
	1	Leer la descripción.
	2	Calcular componentes principales.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Alta	

Tabla 10 - Caso de uso 9

Caso de uso 10	Establecer número de componentes principales.	
Requisitos asociados	RF4, RF4.1.	
Descripción	Permitir al usuario definir la cantidad de componentes principales a utilizar en el análisis.	
Precondición	Haber pulsado el botón de ACP	
Acciones	Paso	Acción
	1	Introducir el número de componentes principales en el input.
	2	Pulsar el botón introducir.
Postcondición	N/A	
Excepciones	1. El número introducido es mayor al número de variables.	
Importancia	Alta	
Frecuencia	Alta	

Tabla 11 - Caso de uso 10

Caso de uso 11	Cálculo de componentes principales.	
Requisitos asociados	RF4, RF4.1, RF4.2.	
Descripción	Realizar el cálculo de los componentes principales basados en los datos proporcionados.	
Precondición	Haber pulsado el botón de ACP y haber introducido el número de CP.	
Acciones	Paso	Acción
	1	Esperar a que el programa realice los cálculos de ACP.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Alta	

Tabla 12 - Caso de uso 11

Caso de uso 12	Tablas con resultados del análisis.	
Requisitos asociados	RF4, RF4.1, RF4.2, RF4.3	
Descripción	Mostrar los resultados del análisis de componentes principales en forma de tablas, incluyendo la contribución de cada componente y la varianza explicada.	
Precondición	Haber pulsado el botón de ACP y haber introducido el número de CP.	
Acciones	Paso	Acción
	1	Visualizar las tablas con los resultados.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Alta	

Tabla 13 - Caso de uso 12



Caso de uso 13	Gráfico con resultados.	
Requisitos asociados	RF4, RF4.1, RF4.2, RF4.4	
Descripción	Representar visualmente los resultados del análisis de componentes principales mediante un gráfico, como un gráfico de dispersión o un gráfico de varianza explicada.	
Precondición	El número de CP introducido debe ser 1 o 2.	
Acciones	Paso	Acción
	1	Visualizar la gráfica con los resultados.
Postcondición	N/A	
Excepciones	1. Numero de CP mayor que 2.	
Importancia	Alta	
Frecuencia	Media	

Tabla 14 - Caso de uso 13

Caso de uso 14	Exportar resultados de ACP a Excel.	
Requisitos asociados	RF4, RF4.1, RF4.2, RF4.3, RF4.5.	
Descripción	Permitir al usuario exportar los resultados del análisis de componentes principales a un archivo de Excel para su posterior análisis o presentación.	
Precondición	Haber introducido un valor de CP valido.	
Acciones	Paso	Acción
	1	Pulsar el botón para guardar los datos.
	2	Aceptar la descarga.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 15 - Caso de uso 14

Caso de uso 15	Cálculo de distancias.	
Requisitos asociados	RF5	
Descripción	Calcular diferentes tipos de distancias entre los puntos de datos.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón Distancias.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Media	

Tabla 16 - Caso de uso 15

Caso de uso 16	Cálculo de distancia euclídea.	
Requisitos asociados	RF5, RF5.1.	
Descripción	Realizar el cálculo de la distancia euclidiana entre los puntos de datos.	
Precondición	Haber seleccionado la distancia euclídea.	
Acciones	Paso	Acción
	1	Seleccionar distancia euclídea.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 17 - Caso de uso 16



Caso de uso 17	Cálculo de distancia euclidiana normalizada.	
Requisitos asociados	RF5, RF5.2.	
Descripción	Calcular la distancia euclidiana normalizada entre los puntos de datos.	
Precondición	Haber seleccionado la distancia euclídea normalizada.	
Acciones	Paso	Acción
	1	Seleccionar distancia euclídea normalizada.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 18 - Caso de uso 17

Caso de uso 18	Cálculo de distancia de Mahalanobis.	
Requisitos asociados	RF5, RF5.3.	
Descripción	Realizar el cálculo de la distancia de Mahalanobis, que tiene en cuenta la correlación entre las variables.	
Precondición	Haber seleccionado la distancia de mahalanobis.	
Acciones	Paso	Acción
	1	Seleccionar distancia de mahalanobis.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 19 - Caso de uso 18

Caso de uso 19	Mostrar los resultados de las distancias en una tabla.	
Requisitos asociados	RF5, RF5.4.	
Descripción	Presentar los resultados de los cálculos de distancias en una tabla para visualizar las relaciones y las medidas de distancia entre los puntos de datos.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Visualizar los cálculos realizados.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 20 - Caso de uso 19

Caso de uso 20	Exportar los resultados de las distancias a Excel.	
Requisitos asociados	RF5, RF5.5.	
Descripción	Guardar los resultados de los cálculos de distancias en un archivo de Excel para su posterior análisis y uso.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón para guardar los datos.
	2	Aceptar la descarga.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 21 - Caso de uso 20



Caso de uso 21	Cálculo de clustering.	
Requisitos asociados	RF6	
Descripción	Realizar cálculos de agrupamiento (clustering) sobre los datos.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón Clustering.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 22 - Caso de uso 21

Caso de uso 22	Permitir al usuario seleccionar el tipo de distancia a utilizar en el algoritmo de agrupamiento.	
Requisitos asociados	RF6, RF6.1.	
Descripción	Brindar al usuario la opción de elegir la medida de distancia (euclídea, euclídea normalizada o mahalanobis) para utilizar en el algoritmo de agrupamiento.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Seleccionar el tipo de distancia a usar.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Media	

Tabla 23 - Caso de uso 22

Caso de uso 23	Permitir al usuario elegir el algoritmo de agrupamiento a aplicar.	
Requisitos asociados	RF6, RF6.2.	
Descripción	Permitir al usuario seleccionar el algoritmo de agrupamiento (DBSCAN, OPTICS o K-means) para utilizar en el análisis de agrupamiento de los datos.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Seleccionar el tipo de algoritmo a usar.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Media	

Tabla 24 - Caso de uso 23

Caso de uso 24	Calcular el agrupamiento utilizando el algoritmo DBSCAN.	
Requisitos asociados	RF6, RF6.1, RF6.2, RF 6.3.	
Descripción	Aplicar el algoritmo de agrupamiento DBSCAN para identificar grupos densos de puntos de datos en el espacio.	
Precondición	Haber escogido un tipo de distancia y el algoritmo DBSCAN.	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 25 - Caso de uso 24



Caso de uso 25	Calcular el agrupamiento utilizando el algoritmo OPTICS.	
Requisitos asociados	RF6, RF6.1, RF6.2, RF 6.4.	
Descripción	Aplicar el algoritmo de agrupamiento OPTICS para identificar estructuras de agrupamiento y detectar puntos de datos atípicos.	
Precondición	Haber escogido un tipo de distancia y el algoritmo OPTICS.	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 26 - Caso de uso 25

Caso de uso 26	Calcular el agrupamiento utilizando el algoritmo K-means.	
Requisitos asociados	RF6, RF6.1, RF6.2, RF 6.5.	
Descripción	Aplicar el algoritmo de agrupamiento K-means para dividir los puntos de datos en k grupos, donde k es un número predefinido por el usuario.	
Precondición	Haber escogido un tipo de distancia y el algoritmo K-means.	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 27 - Caso de uso 26

Caso de uso 27	Generar un gráfico con resultados de agrupación.	
Requisitos asociados	RF6, RF6.1, RF6.2, RF 6.3, RF 6.4, RF 6.5, RF 6.6.	
Descripción	Crear un gráfico visual que represente los resultados del análisis de agrupamiento, mostrando la distribución de los puntos de datos y resaltando los diferentes grupos identificados.	
Precondición	Haber seleccionado un algoritmo de clustering y que la dimensión de los datos iniciales sea 2.	
Acciones	Paso	Acción
	1	Visualizar la gráfica con los resultados.
Postcondición	N/A	
Excepciones	1. Dimensión diferente a 2.	
Importancia	Baja	
Frecuencia	Baja	

Tabla 28 - Caso de uso 27

Caso de uso 28	Tabla con resultados de agrupación.	
Requisitos asociados	RF6, RF6.1, RF6.2, RF 6.3, RF 6.4, RF 6.5, RF 6.7.	
Descripción	Presentar los resultados del análisis de agrupamiento en una tabla, mostrando la asignación de cada punto de datos a su respectivo grupo y proporcionando información adicional sobre los atributos o características de los grupos.	
Precondición	Haber seleccionado un algoritmo de clustering.	
Acciones	Paso	Acción
	1	Visualizar la tabla con los resultados.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Alta	

Tabla 29 - Caso de uso 28



Caso de uso 29	Exportar los resultados de outliers a PDF.	
Requisitos asociados	RF6, RF6.1, RF6.2, RF 6.3, RF 6.4, RF 6.5, RF 6.7, RF6.8.	
Descripción	Guardar los resultados de los cálculos de clustering en un archivo de PDF para su posterior análisis y uso.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón para guardar los datos.
	2	Aceptar la descarga.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 30 - Caso de uso 29

Caso de uso 30	Cálculo de outliers.	
Requisitos asociados	RF7	
Descripción	Realizar cálculos para identificar outliers en los datos.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón Outliers.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Alta	

Tabla 31 - Caso de uso 30

Caso de uso 31	Permitir al usuario elegir el algoritmo a utilizar para la detección de outliers.	
Requisitos asociados	RF7, RF7.1.	
Descripción	Brindar al usuario la opción de seleccionar el algoritmo de detección de outliers (por ejemplo, Mahalanobis, DBSCAN, KNN) a aplicar en el análisis de los datos.	
Precondición	Estar en el apartado de Outliers	
Acciones	Paso	Acción
	1	Escoger el tipo de algoritmo a aplicar.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Media	

Tabla 32 - Caso de uso 31

Caso de uso 32	Calcular los outliers utilizando el algoritmo de Mahalanobis.	
Requisitos asociados	RF7, RF7.1, RF7.2.	
Descripción	Utilizar el algoritmo de Mahalanobis para identificar puntos de datos atípicos basados en la distancia multivariante entre ellos y el resto del conjunto de datos.	
Precondición	haber seleccionado el algoritmo de Mahalanobis.	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 33 - Caso de uso 32



Caso de uso 33	Calcular los outliers utilizando el algoritmo DBSCAN.	
Requisitos asociados	RF7, RF7.1, RF7.3.	
Descripción	Aplicar el algoritmo DBSCAN para detectar outliers basados en la densidad de los puntos de datos en el espacio.	
Precondición	Haber seleccionado el algoritmo de DBSCAN	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 34 - Caso de uso 33

Caso de uso 34	Calcular los outliers utilizando el algoritmo OPTICS.	
Requisitos asociados	RF7, RF7.1, RF7.4.	
Descripción	Emplear el algoritmo OPTICS para identificar outliers considerando la estructura de agrupamiento y la densidad de los puntos de datos.	
Precondición	Haber seleccionado el algoritmo de OPTICS	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 35 - Caso de uso 34

Caso de uso 35	Calcular los outliers utilizando el algoritmo KNN.	
Requisitos asociados	RF7, RF7.1, RF7.5.	
Descripción	Utilizar el algoritmo KNN (K-Nearest Neighbors) para identificar outliers basados en la proximidad a sus vecinos más cercanos.	
Precondición	Haber seleccionado el algoritmo de KNN.	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 36 - Caso de uso 35

Caso de uso 36	Calcular los outliers utilizando el algoritmo LOF.	
Requisitos asociados	RF7, RF7.1, RF7.6.	
Descripción	Aplicar el algoritmo LOF (Local Outlier Factor) para identificar outliers considerando las características locales de densidad de los puntos de datos.	
Precondición	Haber seleccionado el algoritmo de LOF.	
Acciones	Paso	Acción
	1	Introducir los parámetros solicitados.
Postcondición	N/A	
Excepciones	Parámetros no válidos.	
Importancia	Alta	
Frecuencia	Media	

Tabla 37 - Caso de uso 36



Caso de uso 37	Mostrar los resultados de los outliers en una tabla.	
Requisitos asociados	RF7, RF7.1, RF7.2, RF7.3, RF7.4, RF7.5, RF7.6, RF7.7.	
Descripción	Presentar los resultados de la detección de outliers en una tabla, mostrando los puntos de datos identificados como outliers junto con sus atributos relevantes.	
Precondición	Haber seleccionado algún algoritmo.	
Acciones	Paso	Acción
	1	Esperar a que el programa muestre la tabla.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Alta	

Tabla 38 - Caso de uso 37

Caso de uso 38	Exportar los resultados de outliers a PDF.	
Requisitos asociados	RF7, RF7.1, RF7.2, RF7.3, RF7.4, RF7.5, RF7.6, RF7.7, RF7.8.	
Descripción	Guardar los resultados de los cálculos de outliers en un archivo de PDF para su posterior análisis y uso.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón para guardar los datos.
	2	Aceptar la descarga.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Media	
Frecuencia	Media	

Tabla 39 - Caso de uso 38

Caso de uso 39	Cambiar el idioma de la aplicación.	
Requisitos asociados	RF8.	
Descripción	Intercambiar el idioma de la aplicación entre español e inglés.	
Precondición	N/A	
Acciones	Paso	Acción
	1	Pulsar el botón para cambiar el idioma.
Postcondición	N/A	
Excepciones	N/A	
Importancia	Alta	
Frecuencia	Media	

Tabla 40 - Caso de uso 39



2.7. Diagrama de casos de uso

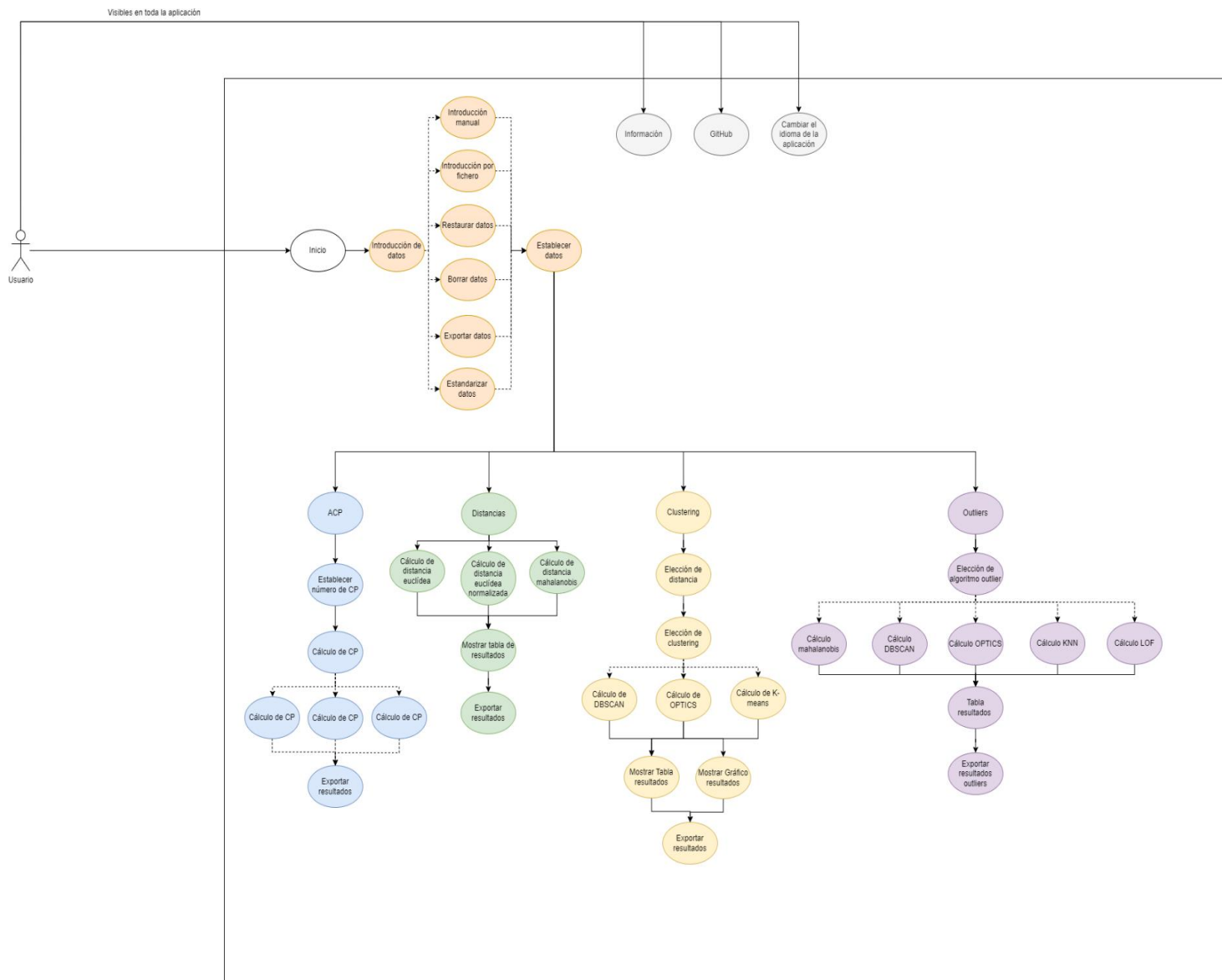


Tabla 41 - Diagrama de casos de uso



Anexo III – Especificación de diseño

3.1. Introducción

En este anexo, se abordarán los diversos patrones de diseño empleados para asegurar un sólido diseño de software en la aplicación, contribuyendo así a su mantenibilidad a largo plazo. Además, se describirán los diseños iniciales de la aplicación y las modificaciones clave que se han realizado, junto con la estructura de directorios adoptada.

Se ha puesto especial énfasis en la utilización de patrones de diseño reconocidos y probados, con el objetivo de lograr un código limpio, modular y fácilmente comprensible. Estos patrones ofrecen soluciones efectivas a problemas comunes en el desarrollo de software, permitiendo un diseño robusto y escalable.

Además de los patrones de diseño, se presentarán los diseños iniciales de la aplicación, que sirvieron como punto de partida para su desarrollo. A lo largo del proceso, se han realizado modificaciones significativas en estos diseños, con el fin de mejorar la experiencia del usuario, optimizar el rendimiento y garantizar una mayor flexibilidad en cuanto a futuras actualizaciones.

Asimismo, se detallará la estructura de los directorios adoptada en el proyecto, lo cual resulta fundamental para mantener un orden adecuado en el código fuente y facilitar la navegación y localización de los diferentes componentes de la aplicación.

En resumen, este apartado proporcionará una visión detallada de los patrones de diseño aplicados, los diseños iniciales y sus modificaciones, así como la organización de los directorios, todo ello en aras de lograr un diseño de software robusto, flexible y fácilmente mantenible.

3.2. Model-View-ViewModel (MVVM)

Para este proyecto, se ha optado por la implementación del patrón de diseño Model-View-ViewModel (MVVM), el cual incorpora Angular como su marco de trabajo nativo. Este enfoque es una variante evolucionada del patrón Modelo-Vista-Presentador (MVP).

En el patrón MVVM, el ViewModel asume el rol principal en la gestión de la lógica de negocio de la aplicación, mientras que la Vista se encarga de la presentación de la interfaz de usuario. Por su parte, el Model representa los datos y el estado de la aplicación, al igual que en los patrones MVC o MVP.

La principal ventaja del patrón MVVM radica en su capacidad para separar claramente las responsabilidades de cada componente. El ViewModel se encarga de preparar los datos y ofrecer métodos y propiedades que la View puede utilizar para interactuar con la lógica de negocio subyacente. Esto permite una mayor flexibilidad y facilita las pruebas unitarias, ya que el ViewModel puede ser probado independientemente de la Vista.

Al incorporar Angular como marco de trabajo para la implementación del patrón MVVM, se aprovechan las características propias de Angular, como la gestión de eventos, la inyección de

dependencias y la manipulación eficiente del DOM, para agilizar el desarrollo y mejorar la eficiencia del código.

En resumen, la adopción del patrón MVVM junto con Angular como marco de trabajo proporciona una estructura clara y eficiente para el desarrollo de la aplicación, promoviendo la separación de preocupaciones y facilitando el mantenimiento y la evolución del software.

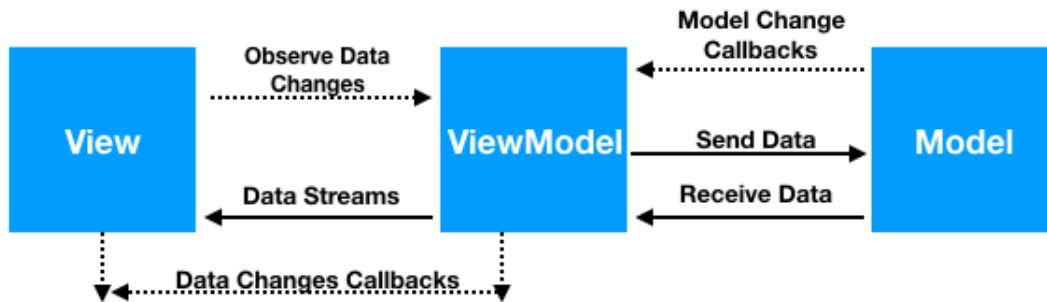


Ilustración 12 - Model-View-ViewModel

De esta forma, dentro de la aplicación podemos definir estos tres elementos como:

- **Model:** Representa los datos y el estado de la aplicación. Incluye la estructura de datos, la lógica de acceso a los datos y la lógica de negocio para procesar y manipular los datos. Se encarga de manejar la persistencia de los datos a través de servicios y proporciona una interfaz para que el controlador pueda interactuar con ellos.
- **View:** Es la interfaz de usuario, la parte con la que el usuario interactúa y la que muestra toda la información. Se ha realizado en Angular y se trata de varios componentes que realizan comunicaciones con el controlador para poder comunicarse entre distintas ventanas, mostrar gráficos, etc.
- **ViewModel:** Es el conjunto de reglas y procesos implementados en la aplicación para el correcto funcionamiento de ésta. Ésta incluye escalabilidad y flexibilidad, para que pueda ser modificada acorde a los futuros cambios del negocio y a las necesidades de los usuarios.



3.3. Paquetes

Los paquetes de la aplicación cuelgan de la carpeta src y cada subcarpeta que lo compone indica cada uno de los módulos de la aplicación.

Src, es la carpeta contenedora de todos los paquetes y componentes del proyecto.

Dentro de src, podemos encontrar las siguientes carpetas:

- **app:** es la carpeta contenedora de todos los componentes individuales, que van a contener la gran mayoría del contenido de la aplicación. Por otra parte, cuenta con los servicios que se utilizan y la configuración de arranque del proyecto.
- **assets:** esta carpeta está diseñada para contener las imágenes e iconos que se emplean dentro de la aplicación; en nuestro caso, únicamente contiene el logo que tiene el proyecto.
- **components:** esta carpeta contiene los componentes generales de la aplicación, estos son aquellos que, debido a su funcionalidad, se emplean en más de un punto de la aplicación por lo que conviene externalizarlos y generalizarlos para su uso.
- **pipes:** esta carpeta contiene los formateadores para las diferentes salidas por pantalla.

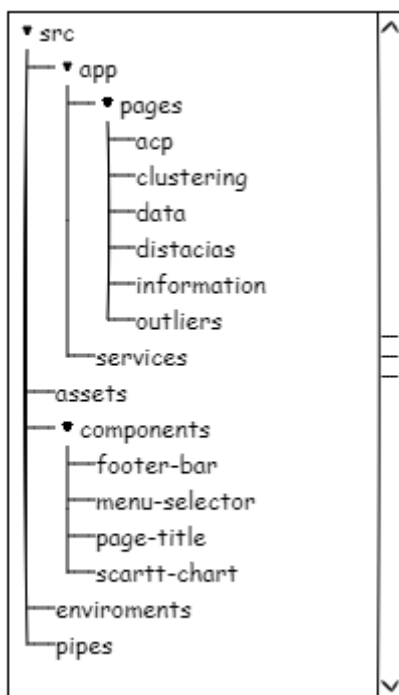


Ilustración 113 - Estructura del directorio src. Fuente: elaboración propia

Directorio app:

Como hemos dicho anteriormente, la carpeta app contiene la mayoría de los ficheros del proyecto que implican al modelo MVVM. Dentro de esta carpeta, podemos encontrar los siguientes directorios:

- **Pages:** representa el directorio que contiene todos los componentes individuales de la aplicación.
- **Services:** representa el directorio que contiene todos los servicios empleados en el proyecto y que pueden ser empleados por los diferentes componentes.
- **Ficheros de configuración de arranque:** estos ficheros contienen la configuración y estructura inicial del proyecto. Son los primeros en cargarse y compilarse.

Directorio pages:

Dentro del directorio pages, podemos encontrar los siguientes directorios:

- **Acp:** contiene los archivos empleados para el cálculo de Análisis de Componentes Principales.
- **Clustering:** dentro de este, podemos encontrar un componente general de clustering que contiene los algoritmos de DBSCAN, OPTICS y k-means.
- **Data:** este directorio contiene la entrada de datos, tanto de forma manual como por fichero.
- **Distancias:** podemos encontrar un componente que generaliza los diferentes cálculos de las distancias euclídea, euclídea normalizada y mahalanobis.
- **Information:** este directorio contiene la información principal del proyecto, donde se define el objetivo y uso del proyecto.
- **Outliers:** este directorio generaliza los diferentes algoritmos de detección de outliers, así como DBSCAN, OPTICS, mahalanobis, LOF y KNN.

3.4. Diseño de interfaces

Para el diseño de interfaces, se utilizó la herramienta Pencil, una aplicación gratuita y de código abierto que permite crear prototipos de interfaz de usuario de manera rápida y sencilla para aplicaciones web, móviles y de escritorio. Pencil ofrece una amplia variedad de elementos predefinidos, como botones, íconos, barras de herramientas y cuadros de texto, que los diseñadores pueden utilizar en sus diseños.

Una de las principales ventajas de Pencil es su interfaz intuitiva y fácil de usar. Además, ofrece la posibilidad de exportar los diseños en varios formatos, como HTML, PDF, PNG y SVG. También cuenta con funciones de simulación de interacción, lo que permite mostrar cómo funcionaría la interfaz en la práctica.



Durante el proceso de diseño, las interfaces iniciales han experimentado numerosas modificaciones con el objetivo de mejorar la experiencia del usuario. Se ha puesto énfasis en crear una interfaz amigable e intuitiva para que cualquier persona pueda utilizar la aplicación sin dificultad. A continuación, se presentan las interfaces iniciales que se han desarrollado:



Ilustración 14 - Pantalla inicial del prototipo.

Ilustración 15 - Introducción manual de datos

Website Page Title

[http://www.example.com/](#)

Información

Datos

ACP

Clustering

Outliers

GitHub

Numero de variables

Subir archivo

Tabla de datos

ID	Área	Numero de Baños	Numero de Habitaciones	Escuelas Cercanas	Crimen en el Área
1	150	1	1	2	20%
2	200	2	1	1	25%
3	270	2	2	1	15%
4	300	3	2	3	10%
5	400	3	3	2	5%

Ilustración 16 - Introducción de datos por fichero

Website Page Title

[http://www.example.com/](#)

Información

Datos

ACP

Clustering

Outliers

GitHub

Descripción

Texto sobre el ACP y su funcionamiento.

Introducir el nuevo número de componentes principales

Datos resultado

Graficas

Ilustración 17 - Ventana ACP



Website Page Title

http://www.example.com/

Información Datos ACP Clustering Outliers GitHub

Descripción
Texto sobre distancias y clustering y su funcionamiento.

Introduce el tipo de distancia

Distancia

Introduce el algoritmo de clustering

Clustering

Distancias

Clustering

Gráfica

Outliers

Ilustración 18 - Ventana distancias y clustering

Website Page Title

http://www.example.com/

Información Datos ACP Clustering Outliers GitHub

Descripción
Texto sobre outliers y su funcionamiento.

Introduce el algoritmo de outlier

Outliers

Resultados

Ilustración 19 - Ventana Outliers

Principales modificaciones de diseño:

Tras recibir comentarios y sugerencias por parte de los usuarios que probaron la aplicación, se implementaron mejoras significativas en el diseño y la experiencia del usuario.

Uno de los cambios más destacados fue la modificación del menú de ventanas. En lugar de utilizar etiquetas como secciones para las diferentes funcionalidades, se decidió incorporar un stepper, que proporciona una experiencia más dinámica e intuitiva para los usuarios. A continuación, se muestra el nuevo diseño del menú:



Ilustración 20 - Menú de la aplicación

De esta manera, se logró una mejor organización de los apartados, diferenciando la sección de introducción y GitHub de la parte funcional del proyecto, donde el usuario puede acceder a las diferentes ventanas. Se aseguró que el primer paso fuera la introducción de datos con los que se trabajará.

Otro cambio importante fue la separación del cálculo de distancias de la ventana de clustering. Esto permitió tener una funcionalidad independiente para calcular distancias entre puntos, sin necesidad de realizar clustering de datos.

Por último, se realizaron modificaciones en los botones inferiores de navegación. Se reemplazaron los nombres de los apartados por flechas, lo cual facilita la navegación intuitiva. Además, se sustituyeron textos en botones como "exportar a Excel" por iconos de las aplicaciones correspondientes. Esto permite al usuario interpretar rápidamente la funcionalidad de estos botones y saber qué acciones realizan.



Anexo IV – Documentación técnica de programación

4.1. Introducción

En este anexo, se proporcionará una explicación detallada de la organización del proyecto, el manual del programador y los requisitos necesarios para ejecutar correctamente el proyecto.

En primer lugar, se describirá la estructura y organización del proyecto. Se proporcionarán detalles sobre la arquitectura general, los módulos y componentes principales, así como la interconexión entre ellos. Esto permitirá a los lectores tener una visión clara de cómo está diseñado el proyecto y cómo se relacionan las diferentes partes entre sí.

Posteriormente, se presentará el manual del programador. Este manual contendrá instrucciones detalladas sobre cómo configurar el entorno de desarrollo, compilar el código fuente y realizar pruebas. Se incluirán también pautas de codificación y convenciones para asegurar un desarrollo consistente y de alta calidad. Además, se proporcionarán recomendaciones para el mantenimiento y la gestión del proyecto a lo largo del tiempo.

Finalmente, se enumerarán los requisitos necesarios para ejecutar el proyecto de manera adecuada. Estos requisitos podrían incluir el sistema operativo compatible, las dependencias de software, las versiones mínimas de bibliotecas y cualquier otro componente esencial para el correcto funcionamiento del proyecto. Al proporcionar estos requisitos de manera clara y concisa, los usuarios podrán preparar su entorno correctamente y evitar posibles problemas durante la ejecución.

4.2. Estructura del directorio

Este proyecto está subido al repositorio de [GitHub](https://github.com), donde se puede comprobar la estructura y descargar el proyecto.

La estructura de carpetas es la siguiente:

src: Esta carpeta es el punto de partida principal de tu proyecto. Aquí se encuentran todos los archivos fuente de tu aplicación Angular, como los archivos HTML, CSS, TypeScript y otros recursos.

src/app: Esta carpeta contiene los componentes, módulos y otros archivos relacionados con la lógica y la presentación de tu aplicación. Aquí es donde normalmente crearás tus propios componentes y servicios.

src/app/pages: En esta carpeta puedes organizar tus componentes específicos de páginas. Por lo general, cada página o vista de tu aplicación tendrá su propio componente dentro de esta carpeta. Esto ayuda a mantener una estructura clara y modular para tu aplicación.

src/app/services: Aquí es donde puedes almacenar tus servicios, que son clases reutilizables que proporcionan funcionalidades específicas a través de la inyección de dependencias. Los servicios se utilizan para realizar tareas como obtener datos de un servidor, autenticación, comunicación entre componentes, etc.

src/assets: Esta carpeta se utiliza para almacenar archivos estáticos como imágenes, archivos de configuración, estilos CSS globales, fuentes personalizadas, etc. Estos archivos pueden ser accedidos por tu aplicación durante la ejecución.

src/components: En esta carpeta puedes organizar componentes reutilizables que pueden ser utilizados en diferentes partes de tu aplicación. Estos componentes pueden ser independientes y no estar directamente relacionados con las páginas específicas.

src/environments: Esta carpeta contiene archivos de configuración para diferentes entornos, como desarrollo, producción, pruebas, etc. Puedes definir variables y configuraciones específicas para cada entorno, lo que te permite cambiar fácilmente la configuración según sea necesario al compilar y desplegar tu aplicación.

src/pipes: Aquí puedes crear y almacenar tus propios pipes personalizados. Los pipes en Angular son transformaciones que puedes aplicar a los datos en tus plantillas HTML, como el filtrado, ordenamiento, formato de fechas, etc.

4.3. Manual del programador

En este apartado, se proporcionará una explicación detallada sobre dos aspectos clave:

1. **Instalación del proyecto en local:** Se proporcionará una guía detallada para clonar el repositorio del proyecto, configurar el entorno de desarrollo, instalar las dependencias necesarias y ejecutar el servidor de desarrollo localmente para poder visualizar y editar la aplicación.
2. **Internacionalización de la aplicación:** Se explicará cómo configurar la aplicación para admitir múltiples idiomas y culturas, incluyendo el marcado de las cadenas de texto a traducir, la generación de archivos de traducción y la implementación de la funcionalidad para cambiar dinámicamente el idioma en la aplicación.

4.3.1. Instalación y ejecución del proyecto en local:

Para ejecutar la aplicación en un entorno local, es necesario seguir los siguientes pasos:

Descarga del proyecto: Descarga el proyecto desde el repositorio principal mencionado anteriormente. Asegúrate de tener acceso al código fuente del proyecto antes de continuar.

Instalación de las herramientas: Antes de proceder, asegúrate de tener instaladas las herramientas necesarias. En este caso, necesitaremos el programa Visual Studio Code, para poder abrir el repositorio y tener el código expuesto.



Una vez instalado VSC, debemos instalar node.js, una herramienta de gestión de paquetes que podemos encontrar en el siguiente enlace: <https://nodejs.org/es>. Lo mejor es instalar la versión más compatible, la 18.16.0 LTS. Se descargará un instalador que instalará node.js en nuestro sistema.

Una vez hecho todo esto, necesitaremos instalar todas las dependencias y librerías de nuestro proyecto. Para hacerlo, navega hasta la carpeta del proyecto en tu terminal y ejecuta el siguiente comando:

```
$ npm install
```

Esto descargará e instalará todas las dependencias requeridas para el proyecto.

Una vez completados estos pasos, el proyecto estará listo para ser ejecutado en tu entorno local. Puedes utilizar el siguiente comando para iniciar la aplicación:

```
$ npm run start
```

Este comando compilará el proyecto y lo ejecutará localmente. Luego, podrás acceder a la aplicación a través de tu navegador web en la dirección <http://localhost:4200>.

Recuerda que, si realizas cambios en el código fuente, la aplicación se actualizará automáticamente en el navegador.

4.3.2. Internacionalización de la aplicación:

La internacionalización es un proceso esencial para hacer que la aplicación sea accesible para audiencias de diferentes idiomas y culturas.

Para lograrlo, se utilizan técnicas de internacionalización y localización. A continuación, se presentan los pasos del proceso:

1. Configuración del soporte de internacionalización: Angular proporciona una herramienta integrada llamada Angular i18n que facilita la internacionalización de la aplicación. Primero, se deben agregar las configuraciones necesarias en el archivo angular.json para habilitar el soporte de internacionalización.
2. Marcado de texto para traducción: En la aplicación, debes marcar todas las cadenas de texto que deben ser traducidas. Esto se hace mediante directivas especiales de Angular, como {{ 'texto' | translate }}, donde 'texto' es la cadena de texto que se traducirá.
3. Generación de archivos de traducción: Se deben generar archivos de traducción para cada idioma que se vaya a soportar. Estos archivos contienen las traducciones correspondientes a las cadenas de texto marcadas en el paso anterior. Puedes generar los archivos utilizando el comando ng xi18n y luego crear un archivo de traducción para cada idioma. Estos archivos tendrán estructura de diccionario, donde se deben de mantener las claves como referencias iguales en todos los idiomas y los valores son las diferentes traducciones de cada idioma. Ver los casos de los diccionarios es.json y en.json en la carpeta src/assets/i18n.

4. Cambio dinámico de idioma: Implementa la lógica en tu aplicación para permitir a los usuarios cambiar el idioma dinámicamente. En este caso, al incorporar solo dos idiomas español e inglés, se ha escogido la opción de introducir un botón que en función del idioma escogido cambie de una bandera a otra.



Anexo V – Documentación de usuario

5.1. Introducción

En esta sección, abordaremos aspectos relacionados con el uso de la aplicación por parte del usuario final. Se proporcionarán detalles sobre los requisitos necesarios para utilizar la aplicación, así como instrucciones de instalación y una guía del usuario que describirá todas las pantallas de la aplicación para facilitar la comprensión de su funcionamiento.

Se incluirá información detallada sobre lo que el usuario necesita tener para ejecutar la aplicación, así como instrucciones paso a paso para la instalación y el inicio. La guía del usuario guiará al usuario a través de cada pantalla de la aplicación, brindándole una comprensión clara de cómo funciona y qué acciones puede realizar en cada una de ellas.

5.2. Requisitos de usuario

La aplicación ofrece dos opciones de acceso para los usuarios: remota y local. Para acceder de forma remota, el usuario solo necesita ingresar la dirección de la página de Netlify en su navegador. El enlace es el siguiente: <https://main--tfg-edf.netlify.app/>. Por otro lado, si el usuario desea acceder de forma local, puede hacerlo a través de <http://localhost:4200/>. Esta opción permite al usuario acceder y modificar el código que compone los diferentes apartados de la aplicación.

Dado que se trata de una aplicación web, el usuario solo necesita tener un navegador web para acceder a la página. Los requisitos mínimos son tener una de las siguientes versiones de navegador instaladas:

- Safari 9 o posterior.
- Opera 28 o posterior.
- Google Chrome 41 o posterior.
- Mozilla Firefox 40 o posterior.
- Microsoft Edge 12 o posterior.

5.3. Instalación

No es necesario que el usuario realice ninguna instalación, ya que la aplicación está desplegada en la web y solo se requiere un navegador web para acceder a ella.

Se ha considerado la posibilidad de generar un ejecutable de la aplicación para su uso local, pero como se explica en el manual del programador, actualmente no es necesario realizar ninguna instalación adicional para utilizar la aplicación como usuario regular.

Por lo tanto, el acceso a la aplicación no requiere instalación alguna y el usuario puede comenzar a utilizarla simplemente abriendo el enlace en su navegador web preferido.

5.4. Manual del usuario

En esta sección se proporcionará una descripción detallada del flujo de uso de la aplicación y sus diferentes pestañas y secciones. Esto permitirá al usuario orientarse y comprender el funcionamiento de la aplicación en caso de sentirse perdido o tener dificultades para utilizarla. El objetivo principal de este documento es facilitar el aprendizaje y comprensión de cada una de las pantallas de la aplicación por parte del usuario.

El primer paso para utilizar la aplicación como usuario es acceder a la página web a través del siguiente enlace: <https://main--tfg-edf.netlify.app/>.

5.4.1. Introducción

En los siguientes apartados se proporcionará una descripción detallada del uso normal de cada una de las ventanas de la aplicación, junto con ejemplos prácticos que ayudarán al usuario a comprender mejor su funcionamiento.

A continuación, se detallarán los pasos a seguir y las acciones a realizar en cada ventana, lo que permitirá al usuario seguir el ejemplo y obtener una mejor comprensión del proceso.

5.4.2. Barra de navegación

La barra de navegación se compone de tres apartados principales.

- **Introducción:** este apartado incluye los objetivos del proyecto y un pequeño resumen de uso con unas instrucciones básicas.
- **Empezar:** este apartado contiene toda la lógica de la aplicación y todos los apartados que contienen funcionalidades dentro del programa.
- **GitHub:** este último apartado es un enlace directo al repositorio de GitHub, donde se encuentra alojado el proyecto.
- **Idioma:** este botón sobre el menú principal realiza la funcionalidad de internalización de la aplicación. Gracias a esto, podemos escoger entre utilizar el programa en español o inglés.



Ilustración 21 - Barra de navegación



5.4.3. Introducción de datos

Este apartado reviste una gran importancia, ya que constituye la base del programa. Sin la introducción de datos, ningún otro apartado puede funcionar correctamente.

Para comenzar, existen dos formas de introducir datos: manualmente o mediante la carga de un archivo.

La introducción manual de datos requiere que el usuario indique inicialmente el número de variables que se van a declarar para cada objeto con el que se trabajará. Se recomienda utilizar el método de carga de archivo si el usuario planea introducir más de 10 variables diferentes, ya que puede resultar tedioso manejar una gran cantidad de entradas en la ventana.

Una vez que el usuario ha indicado el número de variables, se mostrarán una serie de campos de entrada correspondientes a cada variable. En estos campos, el usuario deberá ingresar un nombre para cada una de las variables. Posteriormente, se generarán campos adicionales donde se podrán introducir los valores de las variables para cada columna correspondiente.

Como ejemplo, supóngase que se quiere generar una tabla de datos con la siguiente estructura:

X	Y	Z
2	3	4
5	4	3
7	1	4
6	5	4

Ilustración 22 - Datos de ejemplo

Para ello, deberemos de seguir los siguientes pasos:

1. Introducir el número de variables:

Número de variables:

Introduce un valor *

3

Enviar

Ilustración 23 - Introducir datos paso 1

2. Introducir los nombres de las variables:

Introduce el nombre de las diferentes variables:

Variable 0 *

X

Variable 1 *

Y

Variable 2 *

Z

Enviar

Ilustración 24 - Introducir datos paso 2

3. Introducir los valores de las variables:

A form titled "Introduce el valor de las variables:" with three input fields labeled "Variable 0 *", "Variable 1 *", and "Variable 2 *". The values 2, 3, and 4 are entered into the respective fields. A blue "Introducir" button is located to the right of the fields.

Introduce el valor de las variables:

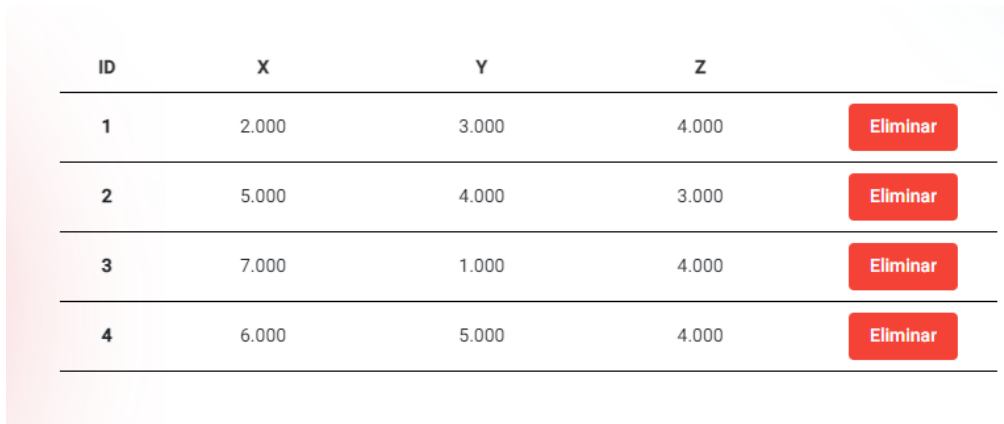
Variable 0 * Variable 1 * Variable 2 *

2 3 4

Introducir

Ilustración 25 - Introducir datos paso 3

De esta forma, el usuario obtendrá la siguiente tabla dentro de la aplicación:

A table with 5 columns: ID, X, Y, Z, and a button column. It contains 4 rows of data. Each row has a red "Eliminar" button in the last column.

ID	X	Y	Z	
1	2.000	3.000	4.000	Eliminar
2	5.000	4.000	3.000	Eliminar
3	7.000	1.000	4.000	Eliminar
4	6.000	5.000	4.000	Eliminar

Ilustración 26 - Tabla final de introducir resultados

Mediante este proceso, se generará una tabla con los datos ingresados manualmente.

En el caso de utilizar la carga de datos mediante un archivo, es importante recordar que el formato óptimo es .xlsx y que las cabeceras deben estar ubicadas en la primera fila del archivo.

Para cargar los datos desde un archivo, el usuario debe seleccionar el botón "Seleccionar archivo" y elegir el archivo que desea cargar. A continuación, la tabla se cargará con los datos y las cabeceras correspondientes.

Una vez que los datos han sido cargados, se encontrarán dos botones. El primer botón permite estandarizar los datos, lo cual es recomendable cuando las unidades de las diferentes variables presentan una gran disparidad entre sí. El segundo botón, identificado con el logotipo de Excel, permite exportar los datos de la tabla junto con las cabeceras, lo que permite al usuario guardarlos en su equipo y realizar una carga más rápida de los datos en el futuro.



5.4.4. ACP

En la ventana ACP, se nos ofrece la opción de realizar un análisis de componentes principales sobre nuestro conjunto de datos. Un uso común del ACP es reducir la dimensionalidad de los datos.

Una vez que los datos han sido cargados, simplemente debemos seleccionar el número deseado de componentes principales en el campo correspondiente. Para hacerlo, introduciremos el valor deseado en el campo de entrada.

Ilustración 27 - Introducción del número de componentes principales

En este caso, se recomienda redimensionar a 2 o 1 dimensión siempre que la calidad de los datos no se vea demasiado afectada. Esto nos permitirá visualizar los datos de manera gráfica en las primeras componentes principales.

Una vez que hemos introducido este parámetro, podemos acceder a los siguientes apartados:

- **Gráfica:** En el caso de haber seleccionado 1 o 2 como número de componentes, podremos observar los datos proyectados en la primera o primera y segunda componente principal.
- **Proyecciones sobre los componentes principales:** En esta sección, se genera una tabla con los valores de cada dato y su correspondiente proyección sobre los componentes principales.
- **Propiedades:** Este apartado muestra algunas propiedades generales de las componentes principales, como su varianza, autovalores y desviación estándar.
- **Autovectores:** En esta parte se genera una tabla que muestra los autovectores de las componentes principales.

Además, encontraremos un botón con el logotipo de Excel, que nos permite exportar los resultados de las tablas a un archivo en formato .xlsx. Esto proporciona al usuario la posibilidad de guardar los resultados obtenidos.

5.4.5. Distancias

En el apartado de distancias, el usuario tiene la opción de calcular diferentes tipos de distancias como la euclídea, euclídea normalizada y Mahalanobis sobre los conjuntos de datos introducidos en la aplicación.

Este apartado es muy sencillo de utilizar. El usuario tiene que seleccionar el tipo de distancia que desea calcular y luego la aplicación mostrará una tabla con las distancias ya calculadas para el conjunto de datos proporcionado.

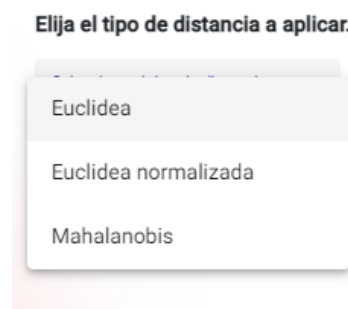


Ilustración 28 - Selector de distancias

Al igual que en otros apartados, incluye también un botón con el logotipo de Excel, que permite generar un archivo en formato .xlsx que contiene las tablas con los resultados de los tres tipos de distancias ya calculadas. De esta manera, el usuario podrá guardar y utilizar estos datos de forma conveniente.

5.4.6. Clustering

Dentro de este apartado se encuentra una amplia gama de métodos de clustering disponibles, como DBSCAN, OPTICS o K-means. Además, el usuario tendrá la posibilidad de seleccionar el tipo de distancia que desea utilizar en estos algoritmos.

El primer paso consiste en elegir el tipo de distancia que se desea aplicar. Para hacerlo, simplemente se despliega el primer selector y elige una de las siguientes opciones:

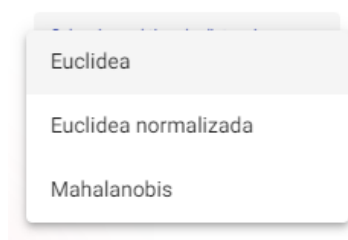


Ilustración 29 - Selector de distancias

Lo siguiente es escoger el tipo de algoritmo de clustering a aplicar entre las siguientes opciones:

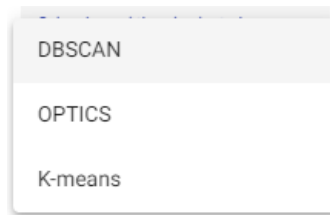


Ilustración 30 - Selector algoritmo clustering

En el caso de seleccionar DBSCAN como método de clustering, se deben proporcionar dos parámetros necesarios para generar los grupos correspondientes.

El primer parámetro es la distancia máxima, que define el alcance máximo de la vecindad de un punto. El segundo parámetro indica el número mínimo de puntos que se debe alcanzar para formar un grupo.

Una vez que se introducen estos parámetros, el usuario puede visualizar los grupos resultantes en forma de una lista coloreada. En el caso de que los datos sean bidimensionales, también se proporciona una representación gráfica de los grupos, donde cada grupo se distingue por un color correspondiente al de la lista.

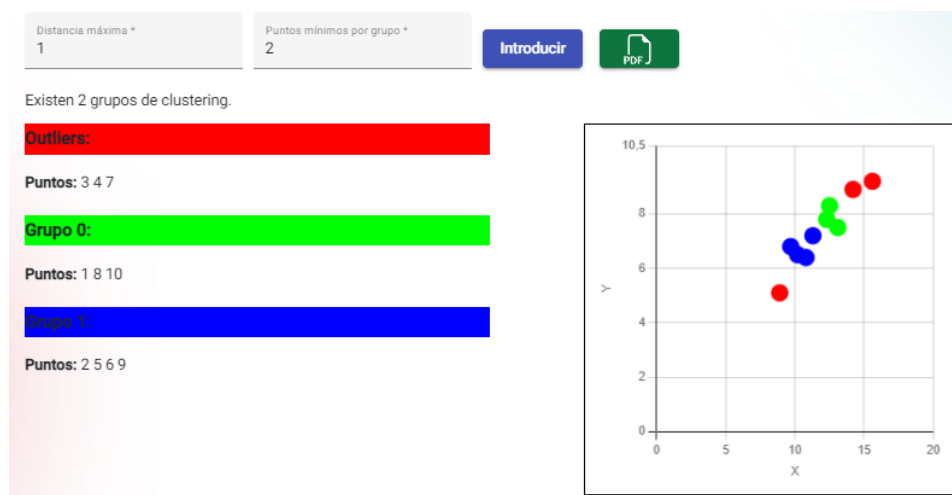


Ilustración 31 - Ejemplo DBSCAN

Si se selecciona OPTICS como método de clustering, se deben proporcionar dos parámetros necesarios para generar los grupos correspondientes.

El primer parámetro es la distancia máxima, que define el alcance máximo de un punto en relación con su vecindad. El segundo parámetro indica el número mínimo de puntos que un se debe alcanzar para formar un grupo.

Una vez introducidos estos parámetros, el usuario puede visualizar los grupos resultantes en forma de una lista coloreada. En el caso de que los datos sean bidimensionales, también se proporciona una representación gráfica de los agrupamientos, donde cada grupo se distingue por un color correspondiente al de la lista.

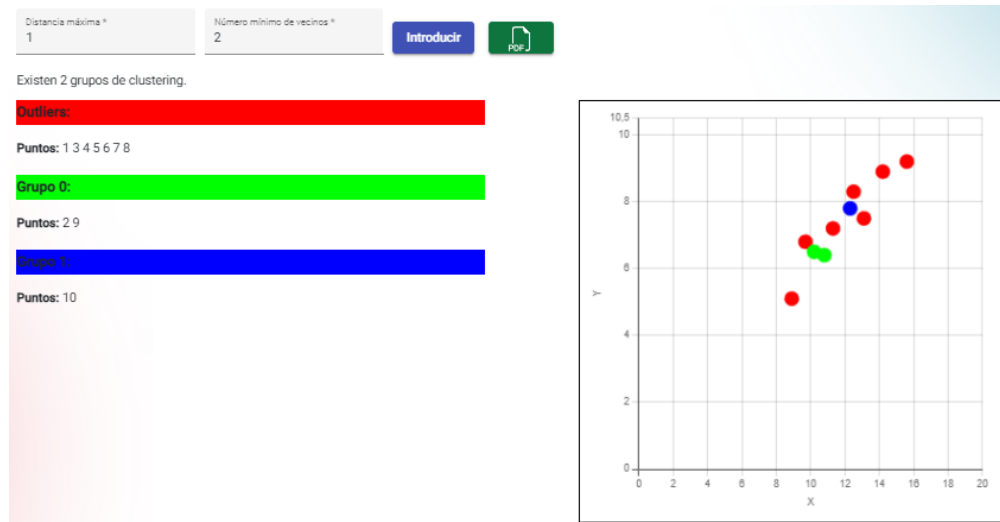


Ilustración 32 - Ejemplo OPTICS

El último caso es el de K-means, el cual solo requiere de un parámetro de entrada; el número de grupos en los que se van a organizar los diferentes puntos.

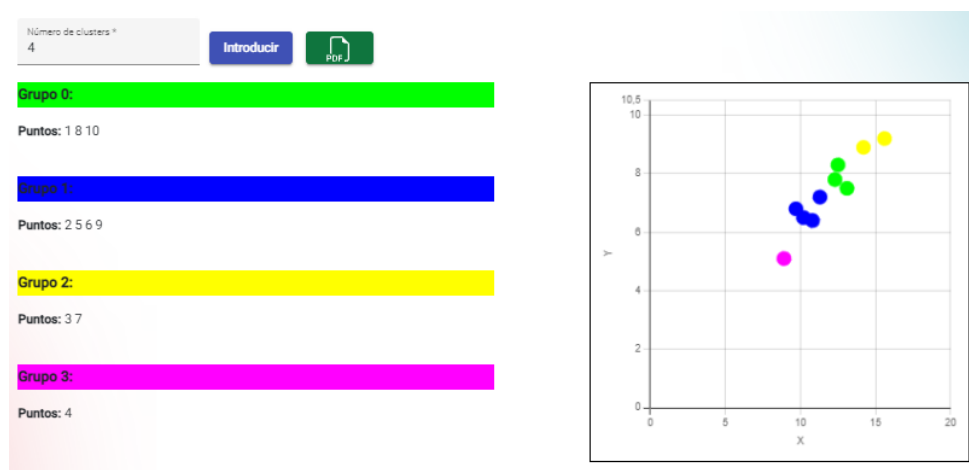


Ilustración 33 - Ejemplo k-means



Como se puede observar, para cada uno de los algoritmos de clustering, existe un botón con un icono de PDF, el cual va a generar una copia de lo que estamos viendo por pantalla y lo va a introducir en un PDF, para que podamos guardar de una forma más visual los resultados obtenidos.

5.4.7. Outliers

Este apartado del programa tiene como objetivo detectar y obtener anomalías dentro del conjunto de datos proporcionado por el usuario. Permite identificar y eliminar datos que pueden ser considerados errores o incertidumbres en las mediciones.

Se ofrecen cinco tipos de detección de outliers: Mahalanobis, DBSCAN, OPTICS, K-Nearest Neighbors y LOF.

Mahalanobis:

Para utilizar Mahalanobis, solo se requiere un parámetro de entrada: la distancia máxima a aplicar en el algoritmo. El resultado es una tabla que contiene los identificadores de los puntos considerados outliers, junto con la distancia al centro y sus coordenadas.

DBSCAN y OPTICS:

Estos dos algoritmos funcionan como se ha explicado en el apartado de clustering.

K-Nearest Neighbors:

KNN requiere que el usuario especifique el número de puntos más alejados. El algoritmo ordena los puntos de menor a mayor distancia, con respecto al origen, lo que nos permite identificar los puntos más distantes y considerarlos como outliers. Se muestra una tabla con los identificadores de los puntos y sus coordenadas.

LOF:

El algoritmo LOF requiere el valor de distancia máxima a aplicar. Devuelve los puntos cuya desviación promedio, una vez aplicada la distancia, es menor que la media de la población circundante. El resultado es una tabla que contiene los identificadores de los puntos, el valor LOF y sus coordenadas en el espacio.

En cada apartado de cada algoritmo, hay un botón con un icono de PDF que permite exportar las tablas de resultados en formato PDF.



6. Bibliografía

- Academia Serrano. (2022). *Análisis de componentes principales (PCA)* - YouTube.
https://www.youtube.com/watch?v=7My_PBhxeP4
- Adobe XD. (2023). *Adobe XD Learn & Support*. <https://helpx.adobe.com/support/xd.html>
- Armstrong Jim. (2018). *TSDDataStats/DataStats.ts at master · theAlgorithmist/TSDDataStats · GitHub*. <https://github.com/theAlgorithmist/TSDDataStats/blob/master/src/DataStats.ts>
- Angular. (2023). *Angular - What is Angular?* <https://angular.io/guide/what-is-angular>
- Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering Points To Identify the Clustering Structure. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 49–60. <https://doi.org/10.1145/304182.304187>
- Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1), 243–256.
<https://doi.org/10.1016/j.patcog.2012.07.021>
- Avila Leyanis, Mendoza Niusvel, & Alonso Andres. (2019). *Detección de anomalías basada en aprendizaje profundo: Revisión*.
https://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992019000300107
- Badge Codacy. (2023). *@swimlane/ngx-charts - npm*.
<https://www.npmjs.com/package/@swimlane/ngx-charts>
- Bello Omar. (2021). *Análisis de Componentes Principales: Maximización de varianza* - YouTube.
<https://www.youtube.com/watch?v=-blx46lDhFY>
- Benites Luis. (2022a). ▷ *Distancia de Mahalanobis: definición simple, ejemplos en 2023* → STATOLOGOS®. <https://statologos.com/distancia-mahalanobis/>
- Benites Luis. (2022b). ▷ *Distancia de Mahalanobis: definición simple, ejemplos en 2023* → STATOLOGOS®. <https://statologos.com/distancia-mahalanobis/>
- Binani Harsh. (2020). *R and Javascript : Execution, Libraries, Integration* | HackerNoon.
<https://hackernoon.com/r-and-javascript-execution-libraries-integration-40a30726f295>
- Bootstrap. (2023). *Get started with Bootstrap · Bootstrap v5.3*.
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Bootstrap. (2023). *Angular powered Bootstrap*. <https://ng-bootstrap.github.io/#/home>
- Bostock Mike. (2021). *Scatterplot / D3 | Observable*.
<https://observablehq.com/@d3/scatterplot>
- Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (1999). Principles of Data Mining and Knowledge Discovery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 1704). Springer Verlag. https://doi.org/10.1007/978-3-540-48247-5_28

- Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-based Local Outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 29(2), 93–104. <https://doi.org/10.1145/335191.335388>
- Capterra. (2020). *Netlify - Opiniones, precios y características - Capterra España 2023*. <https://www.capterra.es/software/154989/netlify>
- Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1), 200–210. <https://doi.org/10.1016/j.eswa.2012.07.021>
- ChartJS. (2023). *Chart.js | Open source HTML5 Charts for your website*. <https://www.chartjs.org/>
- Clare Timothy, & Potix Corporation. (2012). ZK MVVM. Potix. http://books.zkoss.org/wiki/Small_Talks/2012/February/New_Features_of_ZK_6#ZK_MV_VM
- Clusterfck. (2010). *clusterfck - JavaScript hierarchical clustering*. <https://harthur.github.io/clusterfck/>
- Coomans, D., & Massart, D. L. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 136(C), 15–27. [https://doi.org/10.1016/S0003-2670\(01\)95359-0](https://doi.org/10.1016/S0003-2670(01)95359-0)
- Cotton Ben. (2016). Making documentation easy with Read the Docs. *Opensource.Com*. <https://opensource.com/business/16/8/introduction-read-docs>
- Coyler Chris. (2013). *A Complete Guide to Flexbox | CSS-Tricks - CSS-Tricks*. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- Cypress. (2023). *JavaScript Web Testing and Component Testing Framework | cypress.io*. <https://www.cypress.io/>
- D3Gallery. (2023). *D3 Gallery / D3 | Observable*. <https://observablehq.com/@d3/gallery>
- Damian A. (2015). *GitBook, escribe documentación para tus proyectos desde Ubuntu | UbuLog*. <https://ubunlog.com/gitbook-editor-documentacion-ubuntu/>
- De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1), 1–18. [https://doi.org/10.1016/S0169-7439\(99\)00047-7](https://doi.org/10.1016/S0169-7439(99)00047-7)
- Deza, M. M., & Deza, E. (2013). Encyclopedia of Distances. *Encyclopedia of Distances*. <https://doi.org/10.1007/978-3-642-30958-8>
- Dixon, & Moe. (2023). *Códigos de Colores HTML*. <https://htmlcolorcodes.com/es/>
- Edix. (2022). *¿Qué es Adobe XD y para qué sirve? ¡Te lo contamos todo!* <https://www.edix.com/es/instituto/adobe-xd/>
- Ekstein Nikki. (2019). Using Trello to Plan Your Next Vacation (Really). *Www.Bloomberg.Com*. <https://www.bloomberg.com/tosv2.html?vid=&uuid=a1895840-ec37-11e9-bd54->



8dc30bda8b5f&url=L25ld3MvYXJ0aWNsZXNmMjAxOS0wMy0xNC9ob3ctdG8tdXNlLXRyZWxsby10by1wbGFuLXRyYXZlbC1ib29rLWZsaWdodHMtbWFWLXZhY2F0aW9ucw==

Esri. (2019). *Cómo funciona el clustering basado en densidad—ArcGIS Pro | Documentación*.
<https://pro.arcgis.com/es/pro-app/latest/tool-reference/spatial-statistics/how-density-based-clustering-works.htm>

Estivill-Castro, V., & Estivill-Castro, V. (2002). Why so many clustering algorithms — A Position Paper. *ACM SIGKDD Explorations Newsletter*, 4(1), 65–75.
<https://doi.org/10.1145/568574.568575>

Figma. (2023). *Free Design Tool for Websites, Graphic Design and More | Figma*.
<https://www.figma.com/design/>

FontAwesome. (2023). *Find Icons with the Perfect Look & Feel | Font Awesome*.
<https://fontawesome.com/icons>

Fowler, M. (2004). *The Presentation Model Design Pattern*. Martin Fowler.com.
<http://martinfowler.com/eaDev/PresentationModel.html>

Gabriel João. (2017). *Clustering with Javascript — Part 3: Clustering Algorithms in Practice | by João Gabriel Lima | Medium*. <https://medium.com/@joaogabriellima/clustering-with-javascript-part-3-clustering-algorithms-in-practice-75631b241917>

García-Pérez, C. A. (2016). Me gusta citar. *Geotechnical, Geological and Earthquake Engineering*, 16, 129–145. <https://doi.org/10.1007/978>

Geekscoach. (2020). *K-Means | Clustering. K-means o en español seria K medias es... | by Geekscoach | Medium*. <https://geekscoach.medium.com/k-means-clustering-cebbcb4e38ec>

Geun Kim, M. (2000). Multivariate outliers and decompositions of Mahalanobis distance. *Communications in Statistics – Theory and Methods*, 29(7), 1511–1526.
<https://doi.org/10.1080/03610920008832559>

GitBook. (2023a). *GitBook - Where technical teams document*. <https://www.gitbook.com/>

GitBook. (2023b). *Introduction to GitBook - GitBook Documentation*. <https://docs.gitbook.com/>

GitBook. (2023c). *¿Qué es Gitbook? · GitBook*. <https://ull-esit-dsi-1617.github.io/estudiar-las-rutas-en-expressjs-alberto-diego/Alberto/gitbook/queesgitbook.html>

González Guillermo. (2020). *▷ Análisis de componentes principales (PCA): mejor explicado | Machine Learning Studio® 2023*. <https://mlstudio.jaol.net/principal-components-analysis-pca-better-explained/>

Gossman, J. (2005). *Tales from the Smart Client: Introduction to Model/View/ViewModel pattern for building WPF apps*. <https://docs.microsoft.com/en-us/archive/blogs/johngossman/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps>

- Gossman, J. (2006). *Tales from the Smart Client: Advantages and disadvantages of M-V-VM*. <https://docs.microsoft.com/en-gb/archive/blogs/johngossman/advantages-and-disadvantages-of-m-v-vm>
- Harmouch Mahmoud. (2021). *17 Clustering Algorithms Used In Data Science and Mining | by Mahmoud Harmouch | Towards Data Science*. <https://towardsdatascience.com/17-clustering-algorithms-used-in-data-science-mining-49dbfa5bf69a>
- Holscher, E. (2022). Read the Docs 2021 Stats. *Read the Docs Blog*. <https://blog.readthedocs.com/read-the-docs-2021-stats.html>
- Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. 561.
- Hsu, D., Kakade, S. M., & Zhang, T. (2012). A spectral algorithm for learning Hidden Markov Models. *Journal of Computer and System Sciences*, 78(5), 1460–1480. <https://doi.org/10.1016/j.jcss.2011.12.025>
- Jia, Z., & Song, L. (2020). Weighted k-Prototypes Clustering Algorithm Based on the Hybrid Dissimilarity Coefficient. *Mathematical Problems in Engineering*, 2020. <https://doi.org/10.1155/2020/5143797>
- Jos de Jong. (2023). *josdejong/mathjs: An extensive math library for JavaScript and Node.js*. <https://github.com/josdejong/mathjs>
- Kaggle. (2023). *Kaggle: Your Home for Data Science*. <https://www.kaggle.com/>
- KeepCoding Team. (2023). *¿Qué es el patrón de arquitectura MVVM?* <https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/>
- Li, J., Song, S., Zhang, Y., & Zhou, Z. (2016). Robust K-median and K-means clustering algorithms for incomplete data. *Mathematical Problems in Engineering*, 2016. <https://doi.org/10.1155/2016/4321928>
- Liu Shuyi, & Wu Wei. (2017). *GENERALIZED MAHALANOBIS DEPTH IN POINT PROCESS AND ITS APPLICATION IN NEURAL CODING on JSTOR*. <https://www.jstor.org/stable/26362214>
- López José Francisco, & Coll Francisco. (2020a). *Covarianza - Qué es, definición y concepto | 2023 | Economipedia*. https://economipedia.com/definiciones/covarianza.html?nab=1&utm_referrer=https%3A%2F%2Fsearch.brave.com%2F
- López José Francisco, & Coll Francisco. (2020b). *Varianza - Qué es, definición y significado | 2023 | Economipedia*. https://economipedia.com/definiciones/varianza.html?nab=1&utm_referrer=https%3A%2F%2Fsearch.brave.com%2F
- Madhukumar, S., & Santhiyakumari, N. (2015). Evaluation of k-Means and fuzzy C-means segmentation on MR images of brain. *Egyptian Journal of Radiology and Nuclear Medicine*, 46(2), 475–479. <https://doi.org/10.1016/J.EJRM.2015.02.008>
- Maisam Muhammad. (2023). *Calcular la distancia de Mahalanobis en Python | Delft Stack*. <https://www.delftstack.com/es/howto/python/python-mahalanobis-distance/>
- Massey, S. (2011). *Presentation Patterns in ZK*. <http://www.slideshare.net/simbo1905/design-patterns-in-zk-java-mvvm-as-modelviewbinder>



- MathJS. (2020). *math.js / an extensive math library for JavaScript and Node.js*.
<https://mathjs.org/>
- Matlab. (2023). *MATLAB - El lenguaje del cálculo técnico*.
<https://es.mathworks.com/products/matlab.html>
- MDBootstrap. (2023a). *Bootstrap 5 & Angular 12 - Free Material Design UI KIT*.
<https://mdbootstrap.com/docs/angular/>
- MDBootstrap. (2023b). *Bootstrap 5 & Angular 12 - Free Material Design UI KIT*.
<https://mdbootstrap.com/docs/angular/>
- Microsoft. (2005). Introduction to Model/View/ViewModel pattern for building WPF apps. *Microsoft Developer Network*.
<https://blogs.msdn.microsoft.com/johngossman/2005/10/08/introduction-to-modelviewviewmodel-pattern-for-building-wpf-apps/>
- Microsoft. (2012). The MVVM Pattern. *Msdn.Microsoft.Com*. <https://msdn.microsoft.com/en-us/library/hh848246.aspx>
- Microsoft. (2022). How to implement MVVM (Model–View–ViewModel) in TDD (test-driven development). *Microsoft Developer Network*. <https://code.msdn.microsoft.com/How-to-implement-MVVM-71a65441>
- Modzilla. (2023). *Making content editable | MDN*.
https://developer.mozilla.org/es/docs/conflicting/Web/HTML/Global_attributes/contenteditable
- Moreno, I. (2023). *Introducción al Clustering DBSCAN - StatDeveloper*.
<https://www.statdeveloper.com/clustering-dbscan/>
- Murtagh, F., & Kurtz, M. J. (2012). *A History of Cluster Analysis Using the Classification Society's Bibliography Over Four Decades*. <http://arxiv.org/abs/1209.0125>
- Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5), 1003–1016.
<https://doi.org/10.1109/TKDE.2002.1033770>
- NGX-Charts. (2022a). *Introduction - ngx-charts*. <https://swimlane.gitbook.io/ngx-charts>
- NGX-Charts. (2022b). *NgxCharts*. <https://swimlane.github.io/ngx-charts/#/ngx-charts/bar-vertical>
- NIST. (2010). 6.5.4.1. *Mean Vector and Covariance Matrix*.
<https://www.itl.nist.gov/div898/handbook/pmc/section5/pmc541.htm>
- Orange Data Mining. (2023). *Orange Data Mining - Data Mining*.
<https://orangedatamining.com/>
- Osmami Addy. (2012). Understanding MVVM: A Guide for JavaScript Developers. *AddysOnmani.Com*. <http://addyosmani.com/blog/understanding-mvvm-a-guide-for-javascript-developers/>

- Palacio Marta. (2022). *Scrum Master Libro*.
https://www.scrummanager.com/files/scrum_master.pdf
- Parzibyte. (2020). *HTML a PDF con JavaScript - Parzibyte's blog*.
<https://parzibyte.me/blog/2020/09/05/html-pdf-javascript/>
- Paula, & Cansu. (2023). *Biplot for PCA Explained (Example & Tutorial) - How to Interpret*.
<https://statisticsglobe.com/biplot-pca-explained>
- Pavlutin Dimitri. (2022). *Covariance and Contravariance in TypeScript*.
<https://dmitripavlutin.com/typescript-covariance-contravariance/>
- Piechocki Miłosz. (2020). *Strict function types in TypeScript: covariance, contravariance and bivariance - codewithstyle.info*. <https://codewithstyle.info/Strict-function-types-in-TypeScript-covariance-contravariance-and-bivariance/>
- Ponce, R. V., Luis, J., & Alcaraz, G. (2013). Evaluation of Technology using TOPSIS in Presence of Multi-collinearity in Attributes: Why use the Mahalanobis distance? *Rev. Fac. Ing. Univ. Antioquia N*, 31–42.
- Popper. (2023). *Popper - Tooltip & Popover Positioning Engine*. <https://popper.js.org/>
- prabhjotkushparmar. (2023). *Covariance Matrix - Definition, Formula, Examples, Properties and FAQs*. <https://www.geeksforgeeks.org/covariance-matrix/>
- Prabhnkaran Selva. (2019). *Mahalanobis Distance - Understanding the math with examples (python) - Machine Learning Plus*.
<https://www.machinelearningplus.com/statistics/mahalanobis-distance/>
- R Documentation. (2022). *depth.Mahalanobis: Calculate Mahalanobis Depth in ddalpha: Depth-Based Classification and Calculation of Data Depth*.
<https://rdr.io/cran/ddalpha/man/depth.Mahalanobis.html>
- RapidMiner. (2023). *RapidMiner | Amplify the Impact of Your People, Expertise & Data*.
<https://rapidminer.com/>
- Read The Docs. (2023). *Inicio | Read the Docs*. <https://readthedocs.org/>
- Read the Docs. (2023). *Read the Docs: documentation simplified — Read the Docs user documentation 9.7.0 documentation*. <https://docs.readthedocs.io/en/stable/>
- RedHat. (2022). *¿Qué es la metodología ágil?* <https://www.redhat.com/es/devops/what-is-agile-methodology>
- Reynolds, D. (2009). Gaussian Mixture Models. *Encyclopedia of Biometrics*, 659–663.
https://doi.org/10.1007/978-0-387-73003-5_196
- Rodó Paula, & Sevilla Andrés. (2019). *Vectores y valores propios - Qué es, definición y concepto | 2023 | Economipedia*. https://economipedia.com/definiciones/vectores-y-valores-propios.html?nab=1&utm_referrer=https%3A%2F%2Fsearch.brave.com%2F
- Rodriguez Txema. (2014). *GitBook, crea documentación técnica y libros usando Markdown y Git/Github de forma flexible*. <https://www.genbeta.com/desarrollo/gitbook-crea-documentacion-tecnica-y-libros-usando-markdown-y-git-github-de-forma-flexible>



- Salton Kevin. (2017). *How DBSCAN works and why should we use it?* / by Kelvin Salton do Prado / Towards Data Science. <https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80>
- Sanz Francisco. (2018). *Algoritmo K-Means - Clustering y cómo funciona*. <https://www.themachinelearners.com/k-means/>
- Sass. (2023). *Sass: Documentation*. <https://sass-lang.com/documentation/>
- Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3). <https://doi.org/10.1145/3068335>
- Schubert, E., Zimek, A., & Kriegel, H. P. (2012). Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1), 190–237. <https://doi.org/10.1007/s10618-012-0300-z>
- SciPy Documentation. (2022). *scipy.spatial.distance.mahalanobis — SciPy v1.10.1 Manual*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.mahalanobis.html>
- Sejda. (2023). *Convertir HTML a PDF en línea*. <https://www.sejda.com/es/html-to-pdf>
- Shifflett, K. (2023). *Learning WPF M-V-VM*. <http://karlshifflett.wordpress.com/2008/11/08/learning-wpf-m-v-vm/>
- Shvets Alexander. (2020). *Patrones de diseño / Design patterns*. <https://refactoring.guru/es/design-patterns>
- Sibson, R., & Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1), 30–34. <https://doi.org/10.1093/comjnl/16.1.30>
- Smith, J. (2009). WPF Apps with the Model–View–ViewModel Design Pattern. *MSDN Magazine*. <http://msdn.microsoft.com/en-us/magazine/dd419663.aspx>
- Statistics Kingdom. (2022). *Cluster analysis*. <https://www.statskingdom.com/cluster-analysis.html>
- Statlogos. (2021). ▷ *Cómo calcular la distancia euclidiana en Excel en 2023* → STATOLOGOS®. <https://statlogos.com/distancia-euclidiana-excel/>
- Stonis Michael, Jain Tarun, & Pine David. (2022). *Model-View-ViewModel | Microsoft Learn*. <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>
- Tailwind CSS. (2023). *Installation - Tailwind CSS*. <https://tailwindcss.com/docs/installation>
- Trello. (2016a). Breve historia de Trello. *Trello.Com*. <https://trello.com/about>
- Trello. (2016b). What is Trello? *Trello.Com*. <http://help.trello.com/article/708-what-is-trello>
- twelch. (2021). *@turf/clusters-dbscan - npm*. <https://www.npmjs.com/package/@turf/clusters-dbscan>

- UniOviedo. (2016). *kmeans*.
https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html
- Universidad de Burgos. (2023). *Aul@Abierta: formación 2022-2023 | Universidad de Burgos*.
<https://www.ubu.es/aprendizaje-e-investigacion/aulabierta-formacion-guias/aulabierta-formacion-2022-2023>
- Valdes Michael. (2018). *javascript - Como convertir un HTML a PDF con jsPDF, sin perder los estilos CSS - Stack Overflow en español*.
<https://es.stackoverflow.com/questions/156647/como-convertir-un-html-a-pdf-con-jspdf-sin-perder-los-estilos-css>
- Veltman. (2020). *mahalanobis - npm*. <https://www.npmjs.com/package/mahalanobis>
- Villadiego, C. /, & Burgos. (2023). *UNIVERSIDAD DE BURGOS ESCUELA POLITÉCNICA SUPERIOR Pag. 1 de 8 REGLAMENTO SOBRE TRABAJO FIN DE GRADO Y TRABAJO FIN DE MASTER*.
- WanaTop. (2021). *3 mejores herramientas de prototipado | Experts Academy*.
<https://expertsacademy.es/blog/mejores-herramientas-prototipado/>
- Weisstein, E. W., & Weisstein, E. W. (2020). Covariance Matrix. *MathWorld*.
<http://mathworld.wolfram.com/CovarianceMatrix.html>
- Wicklin Rick. (2012). *What is Mahalanobis distance? - The DO Loop*.
<https://blogs.sas.com/content/iml/2012/02/15/what-is-mahalanobis-distance.html>
- Wikipedia. (2023a). *Aprendizaje no supervisado - Wikipedia, la enciclopedia libre*.
https://es.wikipedia.org/wiki/Aprendizaje_no_supervisado
- Wikipedia. (2023b). *Distancia de Mahalanobis - Wikipedia, la enciclopedia libre*.
https://es.wikipedia.org/wiki/Distancia_de_Mahalanobis
- Wikipedia. (2023c). *Distancia de Mahalanobis - Wikipedia, la enciclopedia libre*.
https://es.wikipedia.org/wiki/Distancia_de_Mahalanobis
- Wikipedia. (2023d). *GitHub - Wikipedia, la enciclopedia libre*.
<https://es.wikipedia.org/wiki/GitHub>
- Wikipedia. (2023e). *k-means clustering - Wikipedia*. https://en.wikipedia.org/wiki/K-means_clustering
- Wikipedia. (2023f). *Matriz de covarianza - Wikipedia, la enciclopedia libre*.
https://es.wikipedia.org/wiki/Matriz_de_covarianza
- Wikipedia. (2023g). *Model–view–viewmodel - Wikipedia*.
<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>
- Wikipedia. (2023h). *Rayleigh quotient - Wikipedia*.
https://en.wikipedia.org/wiki/Rayleigh_quotient
- Wikipedia. (2023i). *Sample mean and covariance - Wikipedia*.
https://en.wikipedia.org/wiki/Sample_mean_and_covariance
- Wildermuth, S. (2010). *Windows Presentation Foundation Data Binding: Part 1*. Microsoft.
<http://msdn.microsoft.com/en-us/library/aa480224.aspx>



Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2), 165–193. <https://doi.org/10.1007/S40745-015-0040-1>

York University. (2020). *Zotero vs Mendeley Comparison - York University Libraries*. <https://www.library.yorku.ca/web/research-learn/citing-your-work-academic-integrity/citations/zotero-vs-mendeley-comparison/>

Yufeng. (2022). *Understanding OPTICS and Implementation with Python | by Yufeng | Towards Data Science*. <https://towardsdatascience.com/understanding-optics-and-implementation-with-python-143572abdfb6>

Zenhub Team. (2015). *ZenHub 2.0 : Project management, evolved*. <https://blog.zenhub.com/zenhub-2-0-the-evolution-of-zenhub/>