



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Modelos de Mezcla
Documentación Técnica**



Presentado por Humberto Marijuán
Santamaría
en Universidad de Burgos — 22 de septiembre
de 2022
Tutores: Luis R. Izquierdo y José Manuel
Galán

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	7
Apéndice B Especificación de Requisitos	13
B.1. Introducción	13
B.2. Objetivos generales	13
B.3. Catalogo de requisitos	13
B.4. Especificación de requisitos	15
Apéndice C Especificación de diseño	21
C.1. Introducción	21
C.2. Diseño de datos	21
C.3. Diseño procedimental	22
C.4. Diseño arquitectónico	25
Apéndice D Documentación técnica de programación	27
D.1. Introducción	27
D.2. Estructura de directorios	27
D.3. Manual del programador	32

D.4. Compilación, instalación y ejecución del proyecto	33
D.5. Pruebas del sistema	36
Apéndice E Documentación de usuario	39
E.1. Introducción	39
E.2. Requisitos de usuarios	39
E.3. Instalación	39
E.4. Manual del usuario	39
Bibliografía	47

Índice de figuras

A.1. Sprint: 17 Mayo - 31 Mayo, 2022	4
A.2. Sprint: 31 Mayo - 14 Junio, 2022	5
A.3. Sprint: 14 Mayo - 28 Junio, 2022	6
A.4. Sprint: 06 Septiembre - 20 Septiembre, 2022	7
A.5. Régimen General de la Seguridad Social - Impuestos por contin- gencias comunes	9
A.6. Régimen General de la Seguridad Social - Impuestos por otros	9
C.1. Flujo-01	23
C.2. Flujo-02	24
C.3. Flujo traducciones	25
D.1. Estructura directorios, repositorio GitHub	27
D.2. Estructura directorio web-app/src	29
D.3. Estructura directorio web-app/src/app	29
D.4. Estructura directorio web-app/src/assets	30
D.5. Visual Studio Code extensiones	33
D.6. Comando ng new nombre-proyecto	34
D.7. Configuración angular - angular.json	35
D.8. Configuración angular - tsconfig.json	35
D.9. Configuración angular - tsconfig.app.json	35
D.10. Resumen análisis código Sonarcloud	36
D.11. Sonarcloud issues brand main	37
D.12. Sonarcloud líneas de código	38
E.1. Top Navigation Bar	40
E.2. Ventana inicial web	41
E.3. Inputs datos matriz de fuentes	41

E.4. Arrays solución máximos y mínimos	42
E.5. Export datos entrada problema	43
E.6. Export solución completa	44
E.7. Export resolución paso intermedio	45
E.8. Botón seleccionar archivo	45
E.9. Importar plantilla	45
E.10.Importar visual web	46

Índice de tablas

A.1. Tabla costes totales	10
A.2. Licencias de herramientas utilizadas	11
B.1. CU-01 Importar datos entrada	15
B.2. CU-02 Exportar datos entrada	16
B.3. CU-03 Exportar pasos intermedio	17
B.4. CU-04 Exportar solución completa	18
B.5. CU-05 Traducciones al Inglés	18
B.6. CU-06 Resolver sistemas de ecuaciones	19
B.7. CU-07 Apartado guía de uso aplicación	19
B.8. CU-08 Selección marcadores y fuentes	20
B.9. CU-09 Subida a un servidor	20

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apéndice se ha recogido la planificación temporal, estudio de viabilidad y viabilidad legal.

A.2. Planificación temporal

Para realizar la planificación temporal se ha utilizado la metodología de gestión ágil de proyectos (Scrum). La planificación del proyecto se realizó desde GitHub junto con la extensión Zenhub. En estas herramientas de gestión de versiones se realizaron varios sprints de trabajo con una duración aproximada de 2 semanas. En estos periodos de tiempo se han definido tareas a completar.

Estas tareas o *issues* están asignadas a una persona, tienen una etiqueta de cara a su organización y una estimación de la dificultad. En cuanto a las etiquetas o labels, en este proyecto se han trabajado con las predefinidas por GitHub siendo estas:

- **Enhancement:** Añade una nueva característica o petición. Se puede ver con mayor detalle en [Enhancement](#).
- **Bug:** Corrección de errores o funcionamientos incorrectos de la aplicación. Se puede ver con mayor detalle en [Bug](#).

- **Documentation:** Tareas de documentación, en las que se espera añadir o mejorar documentos. Se puede ver con mayor detalle en [Documentation](#).
- **Duplicate:** Referentes a issues ya existentes.
- **Good first issue:** Tareas para facilitar el uso a nuevos usuarios. Se puede ver con mayor detalle en [Good first issue](#).
- **Help wanted:** Tareas que requieren ayuda externa.
- **Invalid:** Tareas incorrectas.
- **Question:** Se solicita más información.
- **Wontfix:** Tareas que no han sido resueltas o que no están funcionando en ese momento.

Sprints del proyecto

Antes de comenzar con el desarrollo de funcionalidades para la aplicación se realizaron algunas tareas previas como: decisión de lenguajes con los que trabajar, herramientas a utilizar y un periodo de formación.

Antes de empezar con cualquier desarrollo se realizó un curso de L^AT_EX (Más información en la issue [Curso de para aprender a usar Latex](#)) con el objetivo de poder documentar a la vez que se añadían funcionalidades. A continuación, se realizaron algunas tareas de análisis de la herramienta GitHub para comprender en profundidad sus posibilidades. Se leyeron algunos artículos sobre las [Milestones y Sprints](#), junto con una aproximación a los [GitHub Reports](#).

También se preparó el entorno de desarrollo, descargando algunas herramientas como: Instalación Visual Studio Code y configuración extensiones para trabajar con Angular, Postman, extensión de ZenHub, Debian y Men-deley inicialmente.

Sprints análisis

Previo al desarrollo de funcionalidades se lleva a cabo una etapa de análisis. Durante estas semanas se realizó un análisis en profundidad del artículo *Using n-alkanes to estimate diet composition of herbivores* [5]. Durante este periodo se aprendió sobre cómo resolver sistemas de ecuaciones

con restricciones. Se resolvieron ejercicios en Matlab para entender aun más en profundidad los pasos para resolver estos sistemas.

Una vez se definió y entendió el problema planteado se realizó una búsqueda de librerías y lenguajes para desarrollar posteriormente la aplicación.

Sprints iniciales

Durante los primeros meses se realizó la instalación de la librería GLPK [1] y se realizaron algunos ejemplos para llamar desde una proyecto front a una API que realizara funcionalidades de la librería, en concreto resolviera problemas de programación lineal, y recibiéramos esos resultados desde el proyecto front.

Sprint: 17 Mayo - 31 Mayo, 2022

Una vez se conectaron los dos proyectos se empezaron a desarrollar de forma simultánea tanto la API como el front, utilizando la herramienta Postman para hacer pruebas de peticiones.

Este sprint se centró en realizar un flujo en el cual se introducían los datos desde el front y por medio de una petición al API escrita en laravel se recibía la respuesta al problema. A partir de la cual se hizo una primera visual de los resultados.

Las tareas completadas son representadas en el gráfico de la figura A.1.

- **#4-Resolver problemas Non-negative least square, min y max desde api laravel.**
- **#5-Realizar ciclo de recepción y mostrar los resultados desde la web.**
- **#6-Subida documentación y códigos.**



Figura A.1: Sprint: 17 Mayo - 31 Mayo, 2022

Sprint: 31 Mayo - 14 Junio, 2022

Este sprint se centró principalmente en preparar los documentos para importar y exportar, preparando también todo el sistema de carga de archivos de la web. Además de configurar los ficheros de traducciones en la web.

- #7-Funcionamiento paquete GLPK.
- #8-Alert proyección no coincidente con punto.
- #9-Exports solución y datos problemas.
- #10-Descargar .mod de cada elemento de la solución.
- #11-Import datos problema.
- #12-Sistema de Traducciones.
- #14-Export csv traducidos.

Podemos ver el gráfico correspondiente a las tareas anteriores en la figura [A.2](#).

Sprint: 14 Junio - 28 Junio, 2022

Este Sprint está marcado principalmente por la subida a un servidor de nuestro proyecto, aunque también se hicieron pequeños ajustes de maqueta-ción y procedimientos de cara a agilizar el uso de la aplicación

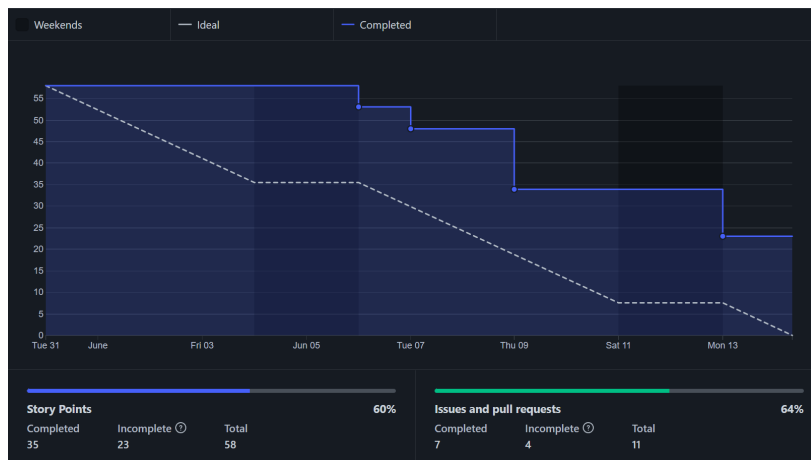


Figura A.2: Sprint: 31 Mayo - 14 Junio, 2022

- #13-Slider web.
- #15-Ajustes maquetación solución.
- #16-Subida a servidor.
- #17-Función reset arrays.
- #19-Añadir línea min max en exports solución.

Podemos ver el gráfico "*Burndown report*" correspondiente al sprint en la figura A.3.

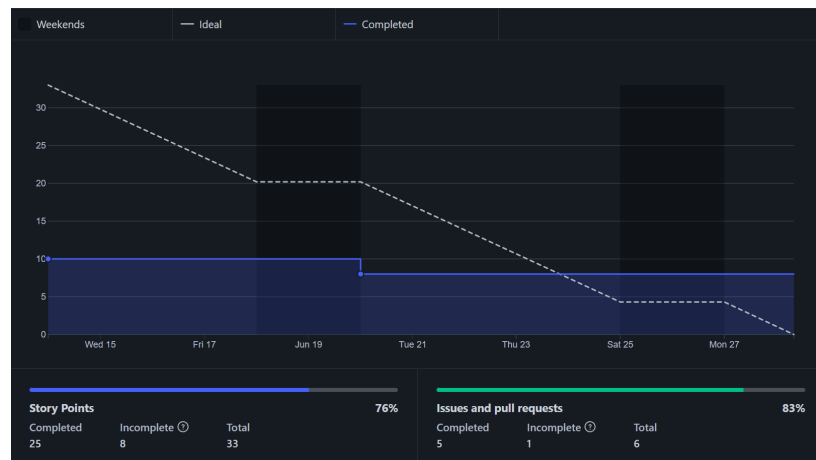


Figura A.3: Sprint: 14 Mayo - 28 Junio, 2022

Sprint: 6 Septiembre - 20 Septiembre, 2022

En este último sprint se han añadido cambios para adaptar por completo las funciones a la nueva librería, se han resuelto algunos bugs, y se ha realizado la primera versión de los documentos memoria y anexos.

- #18-Primera versión memoria
- #23-Búsqueda funciones resuelvan las funciones usadas en el API
- #24-Cambio función api "deleteAllfiles().al proyecto front
- #25-Cambio función API - "getTextFiles().al proyecto front
- #26-Cambio función API - "downloadFile().al proyecto front
- #28-Cambio título web
- #29-Error recargar la página al cambiar de idioma
- #30-Primera versión documento anexos
- #32-Redondeo proyecciones a tres decimales
- #33-Cargar archivo y darle a crear se queda nombre archivo
- #34-Corrección refresh web netlify

Podemos ver el gráfico correspondiente a las tareas anteriores en la figura A.4.

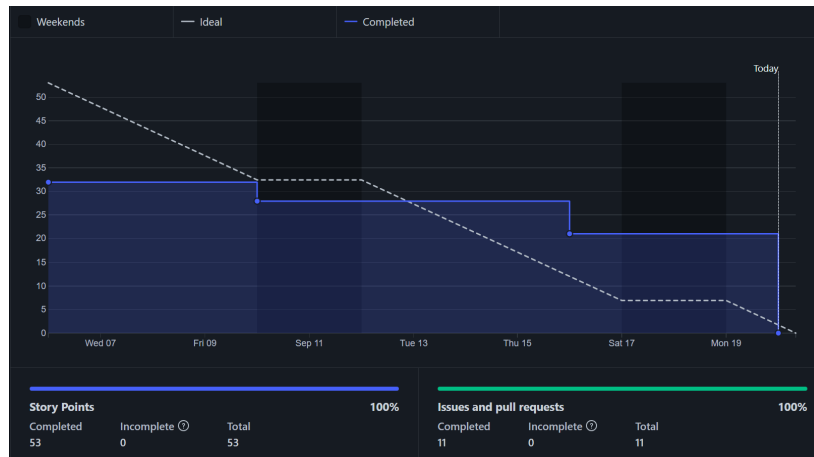


Figura A.4: Sprint: 06 Septiembre - 20 Septiembre, 2022

A.3. Estudio de viabilidad

En esta sección vamos a realizar un estudio de la viabilidad del proyecto tanto en el ámbito económico dando una aproximación de cuál sería la cantidad a pagar por un empresario, como de la viabilidad legal.

Viabilidad económica

De cara a la viabilidad económica vamos a centrarnos únicamente en el gasto que ha significado este proyecto, y no en el posible beneficio que podría dar. Para ello dividiremos los costes en cuatro:

- Costes análisis
- Costes desarrollo
- Costes software
- Costes Hardware

Costes análisis

En cuanto a los costes de análisis en este proyecto solo ha trabajado un único analista web, que según Indeed [2] el sueldo medio de un analista programador en España es de 28 350 € al año. Lo que estimaría una tarifa media de **19,68 €** por hora ($28\,350 / (30 \text{ horas} \times 48 \text{ semanas})$). Se han dedicado 10 semanas (estas semanas han sido en su mayoría antes de empezar el desarrollo, pero se han requerido algunos periodos temporales de análisis durante el desarrollo, ya cuantificados en esas 10 semanas) al análisis del proyecto con jornadas de 6 horas diarias, es decir, unas 30 horas semanales.

$$30 \text{ horas/semana} \times 10 \text{ semanas} \times 19,68\text{€/hora} = \mathbf{5\,904\, \text{€}} \text{ total}$$

A este salario habría que añadirle los impuestos por seguridad social que actualmente aplicarían:

- 23.6 % **contingencias comunes**.
- 5.5 % **tipo general de desempleo para contrato indefinido**.
- 0.20 % **FOGASA** (Fondo de Garantía Salarial).
- 0.60 % para **formación profesional**.

Estos datos (figuras A.5 y A.6) provienen de la página de la Seguridad Social [4], en el apartado de *Régimen General de la Seguridad Social*.

El empleador debería añadir estos impuestos al total de pago tanto en la parte de desarrollo como de análisis.

$$5\,904 * (1 + 0,236 + 0,055 + 0,002 + 0,006) = \mathbf{7\,669,296\, \text{€}}$$

Costes desarrollo

En cuanto a los costes de desarrollo en este proyecto solo ha trabajado un único desarrollador web, que según Indeed [2] el sueldo medio de desarrollador web en España es de 25 822 € al año. Lo que estimaría una tarifa media de **17,93 €** por hora ($25\,822 / (30 \text{ horas} \times 48 \text{ semanas})$). El proyecto se ha desarrollado en 5 periodos de 2 semanas con unas 6 horas al día, es decir, unas 30 horas semanales.

$$30 \text{ horas/semana} \times (5 \text{ periodos} \times 2 \text{ semanas de duración}) \times 17,93\text{€/hora} = \mathbf{5\,379\, \text{€}} \text{ (total trabajador sin impuestos)}$$

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) Exoneración en la cotización por contingencias comunes (salvo IT), desempleo, fondo garantía salarial y formación profesional, prevista en el art. 152 del RD Legislativo 8/2015:

- aplicable durante el año 2022 a trabajadores cuenta ajena o a socios trabajadores o de trabajo de cooperativas, que continúen trabajando tras haber alcanzado la edad de 65 años si acreditan cotizaciones de 37 años y 6 meses o más; o tras haber alcanzado la edad de 66 años y 2 meses, si acreditan cotizaciones inferiores a 37 años y seis meses.
- Tipo de cotización aplicable durante el año 2022 por IT por contingencias comunes: 1,55 %, del que el 1,30% será a cargo de la empresa, y el 0,25 % a cargo del trabajador.

(2) Los contratos de duración determinada por tiempo inferior a 30 días tendrán una cotización adicional a cargo del empresario que se abonará a su finalización, y que durante el año 2022 tendrá un importe de 27,53 €. Esta cotización adicional no se aplicará a los contratos celebrados con trabajadores incluidos en el Sistema Especial para trabajadores Cuenta Ajena Agrarios, en el Sistema Especial de Empleados de Hogar o en el Régimen Especial para la Minería del Carbón, ni a los contratos por sustitución.

Figura A.5: Régimen General de la Seguridad Social - Impuestos por contingencias comunes

DESEMPLEO	EMPRESA	TRABAJADORES	TOTAL
Tipo General: Contratación indefinida, incluidos los contratos indefinidos a tiempo parcial y fijos discontinuos, contratación de duración determinada en las modalidades de contratos formativos en prácticas y para la formación y el aprendizaje, de relevo, interinidad y contratos, cualquiera que sea la modalidad, realizados con trabajadores discapacitados	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
FGGASA	0,20		0,20

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

Figura A.6: Régimen General de la Seguridad Social - Impuestos por otros

Al igual que en el apartado anterior aplicamos los impuestos por seguridad social:

$$5\,379 \cdot (1 + 0,236 + 0,055 + 0,002 + 0,006) = \mathbf{6\,987,321\,€}$$

Costes software

Las herramientas utilizadas son totalmente gratuitas. Por lo que no hay coste en este apartado.

Costes hardware

El proyecto se ha desarrollado en un portátil Huawei KLVL-WXX9 con AMD Ryzen 7 4800h y 16GB de RAM con un precio de 700€ en enero de 2022. Partiendo que Hacienda marca que la amortización de ordenadores es de un 26 % multiplicado por 0,75 (11 meses de uso de noviembre 2021 a septiembre 2022, menos 2 meses de no trabajo da 9 meses de trabajo entre 12 que tiene el año da 0,75 siendo los meses de uso del portátil en un año) el portátil amortizado significaría una cantidad de $700 \times 26 \% = 182$ al año, multiplicado por 0,75 = **136,5** sería el gasto en la duración de este proyecto.

Los gastos totales por tanto de este proyecto vienen representados en tabla de costes de la figura A.1.

Tipo de Costes	Total
<i>Análisis</i>	7 669, 296 €
<i>Desarrollo</i>	6 987,321 €
<i>Software</i>	0 €
<i>Hardware</i>	136,5 €
Total	14 793,117 €

Tabla A.1: Tabla costes totales

Viabilidad legal

Las herramientas utilizadas en el desarrollo del proyectos disponen de licencias gratuitas o son open source. En la tabla A.2 se muestra las licencias de cada herramientas utilizada.

Herramientas	Licencia
GitHub	GNU
Visual Studio Code	MIT
Debian	DFSG
Angular	MIT
Angular Translate	MIT
Angular Material	MIT
Bootstrap	MIT
Node.js	MIT
npm	MIT
Heroku	EULA
Netlify	MIT
GLPK	GPL
TeXStudio	GPL v2

Tabla A.2: Licencias de herramientas utilizadas

Uno de los requisitos del proyecto es que debe cederse bajo licencia GNU GPL 3.0. La GNU General Public License [3] es una licencia libre y sin copyleft para software. Al contrario de la mayoría de licencias que se centran en restringir la libertad para compartir la obra, la licencia GNU esta destinada a garantizar su libertad al compartir y cambiar las versiones del programa. Siendo un proyecto de software libre.

La licencia GNU no impide la venta de las copias, cuando se habla de libertad se refiere a libertad de distribución de copias de software, pero no impide el cobrar por esas copias en caso de que el autor lo requiera.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice se describe la especificación de requisitos. Se detallarán aquellos requisitos funcionales y no funcionales que definen el desarrollo del proyecto.

B.2. Objetivos generales

Los objetivos generales son:

- Crear una aplicación web para estimar diferencias y composición de la dieta de vertebrados herbívoros.
- Realizar análisis matemáticos a partir de datos proporcionados por el usuario. En particular, la aplicación deberá resolver sistemas de ecuaciones lineales con restricciones.
- Poner a disposición de la comunidad científica de forma gratuita.

B.3. Catalogo de requisitos

Requisitos funcionales del proyecto:

- **Importar datos entrada:** La aplicación debe proporcionar una opción de carga de datos mediante fichero.

- **Exportar datos entrada:** La aplicación debe proporcionar una opción de exportar los datos entrada usados a partir de los cuales se ha llegado al resultado actual.
- **Exportar pasos intermedios:** La aplicación debe permitir exportación de los pasos intermedios o en este caso el sistema de ecuaciones, restricciones y función objetivo que se han resuelto, para cada valor de la solución.
- **Exportar solución completa:** La aplicación debe ofrecer una opción de exportar la solución completa, incluyendo datos de entrada, proyecciones de los puntos y los arrays de solución.
- **Traducciones al Inglés:** permitir traducir todos los textos al Inglés, incluyendo los documentos exportados por la aplicación
- **Apartado guía de uso aplicación:** añadir un apartado de guía de uso en la aplicación. Se ha optado por hacer un vídeo explicativo del ciclo habitual de uso.
- **Resolver sistemas de ecuaciones:** Usar la librería GLPK [1] que permite calcular funciones objetivo sobre un sistema de ecuaciones con restricciones
- **Selección marcadores y fuentes:** Permitir al usuario seleccionar los rangos de matriz de datos de entrada y permitir cambiar los tamaños de esta.
- **Subida a un servidor:** la aplicación debe estar subida a un servidor, donde pueda ser accedida por los usuarios online preferiblemente sin requerir una descarga previa.

Requisitos no funcionales del proyecto:

- **Usabilidad:** la aplicación debe ser fácil de usar para el usuario, y fácil de entender para usuarios familiarizados con el problema
- **Compatibilidad:** la web debe ser accesible para la mayor parte de navegadores.
- **Internacionalización:** La aplicación debe incorporar traducciones a todos los textos de la aplicación. Inicialmente al español e inglés

- **Licencia :** a ser posible debe tener la licencia GNU General Public License v3.0 garantizando a los usuarios finales la libertad de estudiar, modificar, usar y compartir el software.

B.4. Especificación de requisitos

CU-01	Importar datos entrada
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	Permite al usuario cargar plantillas de datos
Precondición	Disponer de una plantilla en formato .csv acorde con la esperada por la aplicación
Secuencias	1.Accede página principal 2.Botón seleccionar archivo 3.Seleccionar archivo entre los directorios de tu equipo
Postcondición	Se espera una carga correcta de los datos de la plantilla
Excepciones	En caso de tener otro formato o ser datos incorrectos no carga el archivo
Importancia	Alta

Tabla B.1: CU-01 Importar datos entrada

CU-02	Exportar datos entrada
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	Permite al usuario descargar plantilla de datos
Precondición	Haber introducido previamente datos de entrada y ejecutado la resolución del problema obteniendo una solución
Secuencias	1. Usuario presiona el botón Exportar problema. 2. Se descarga un archivo con extensión .csv en forma de descarga de navegador
Postcondición	Se espera tener un archivo plantilla con los datos de entrada del problema resuelto actual
Excepciones	En caso de no tener solución no se puede exportar
Importancia	Media

Tabla B.2: CU-02 Exportar datos entrada

CU-03	Exportar pasos intermedios
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	Exportar documento que contiene el sistema de ecuaciones con restricciones y la función objetivo
Precondición	Haber introducido previamente datos de entrada y ejecutado la resolución del problema obteniendo una solución
Secuencias	1. Usuario presiona botón sobre valores solución 2. Se descarga un archivo por descarga de navegador con extensión .csv.
Postcondición	Se espera un documento con el sistema de ecuaciones con restricciones y su función objetivo
Excepciones	En caso de no llegar a la solución previamente no se puede exportar
Importancia	Bajo

Tabla B.3: CU-03 Exportar pasos intermedio

CU-04	Exportar solución completa
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	Permite al usuario descargar documento con datos de entrada y solución
Precondición	Haber introducido previamente datos de entrada y ejecutado la resolución del problema obteniendo una solución
Secuencias	1. Usuario presiona el botón Exportar solución. 2. Se descarga un archivo con extensión .csv en forma de descarga de navegador
Postcondición	Se espera disponer un archivo que contenga datos de entrada y la solución de los mismo
Excepciones	En caso de no tener solución no se puede exportar
Importancia	Media

Tabla B.4: CU-04 Exportar solución completa

CU-05	Traducciones al Inglés
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	Permitir al usuario cambiar el idioma de la aplicación en este caso al menos a dos idiomas español e inglés
Precondición	Encontrarse en la aplicación
Secuencias	1. click en el select del idioma actual 2. Se cambia al deseado 3. Se recarga la web
Postcondición	Se espera que los textos de la aplicación cambien al idioma seleccionado
Excepciones	Ninguna
Importancia	Alta

Tabla B.5: CU-05 Traducciones al Inglés

CU-06	Resolver sistemas de ecuaciones
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	Resolver ecuaciones con restricciones, para ello es necesario utilizar una librería que resuelva este tipo de problemas, y además recibir estos datos y almacenarlos
Precondición	Encontrarse en la aplicación
Secuencias	1. Usuario marca el tamaño de la matriz creando una matriz de ceros 2. El usuario rellena los valores de la matriz configurando las ecuaciones a resolver 2. El usuario pulsa botón Resolver
Postcondición	Se espera que la aplicación muestre los arrays solución máximos y mínimos por cada mezcla
Excepciones	No tiene solución
Importancia	Alta

Tabla B.6: CU-06 Resolver sistemas de ecuaciones

CU-07	Apartado guía de uso aplicación
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	La aplicación debe contener un apartado guía de uso que explique el funcionamiento de sus funcionalidades
Precondición	Encontrarse en la aplicación
Secuencias	1. Usuario se dirige al apartado Video 2. El usuario encontrará un vídeo explicativo de la aplicación
Postcondición	Se espera que el usuario pueda ver un ejemplo de uso y algunos de los flujos principales de la aplicación
Excepciones	Ninguna
Importancia	Media

Tabla B.7: CU-07 Apartado guía de uso aplicación

CU-08	Selección marcadores y fuentes
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	La web debe permitir cambiar el número y constantes de las ecuaciones según quiera
Precondición	Encontrarse en la aplicación
Secuencias	1. Usuario introduce el campo "marcadores" 2. Usuario introduce el campo fuentes 3. El usuario pulsa el botón Crear
Postcondición	Se espera que la aplicación cree una matriz de ceros a partir de los tamaños fijados por el usuario
Excepciones	Ninguna
Importancia	Media

Tabla B.8: CU-08 Selección marcadores y fuentes

CU-09	Subida a un servidor
Versión	1.0
Autor	Humberto Marijuán Santamaría
Descripción	La web debe estar alojada en un servidor y permitir de esta forma el acceso a sus funcionalidades a cualquier usuario con conexión a wifi
Precondición	Código del proyecto desarrollado
Secuencias	1. Usuario introduce la url de la aplicación en un navegador 2. Usuario accederá a la ventana inicial de la web
Postcondición	Se espera que el usuario consiga abrir y utilizar la aplicación en un navegador
Excepciones	Bugs y posibles fallos no encontrados
Importancia	Alta

Tabla B.9: CU-09 Subida a un servidor

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice se presentará aquello referente al diseño de la aplicación. En una primera parte analizaremos los datos con los que trabaja la aplicación y también qué datos exporta. A continuación veremos en profundidad los principales procedimientos que contiene la aplicación y en una última parte, el diseño arquitectónico centrándonos en el código del proyecto.

C.2. Diseño de datos

Para resolver el problema planteado hemos utilizado variables dentro del código que nos permiten guardar todos los datos tanto para resolver como para mostrar sistemas de ecuaciones con restricciones.

- **Formulario tamaño matriz:** el primer requerimiento fue contar con un elemento encargado de definir el tamaño de filas por columnas de nuestra matriz de datos de entrada. Este formulario va a requerir dos únicos valores: marcadores y fuentes.
- **Formulario matriz:** matriz con un tamaño definido por los índices del primer formulario, este *FormGroup* que contiene tantos *FormControls* como *marcadores x fuentes*, tiene un tamaño fijo, pero el usuario puede crear otro nuevo mediante el primer formulario. En el código se encuentra como **Matrix**

- **Formulario mezclas:** formulario de marcadores de las fuentes. En el caso de este formulario su tamaño de marcadores es fijo marcado por el primer formulario y su segundo índice parte de uno y el usuario puede añadir tantas mezclas como quiera. En el código se nombra como **Results**.

C.3. Diseño procedimental

En esta sección recorreremos los principales flujos de la aplicación y veremos la secuencia de operaciones que realizan los componentes entre sí.

Flujo-01

En este primer flujo se parte del estado inicial de la aplicación. La ruta por defecto de la aplicación es el componente *"MatrixComponent"* que sería el encargado de resolver el sistema de ecuaciones. Por esa razón el texto escrito en html `<router-outlet>` nos redirige al componente *"MatrixComponent"*. Una vez situados en el punto de partida el usuario rellena el primer formulario con los tamaños de la matriz fuentes, lanzando la función *crearMatrix()*. Como respuesta nos devuelve una matriz de arrays de fuentes y de mezclas a rellenar por el usuario (el usuario puede añadir con un botón tantas mezclas como quiera). Cuando se complete la introducción de datos el usuario pulsa a resolver que lanzará de nuevo una función en *"matrix.component.ts"*, esta función hace tantas iteraciones como mezclas se han configurado y retorna dos arrays: uno de máximo y otro de mínimos por mezcla.

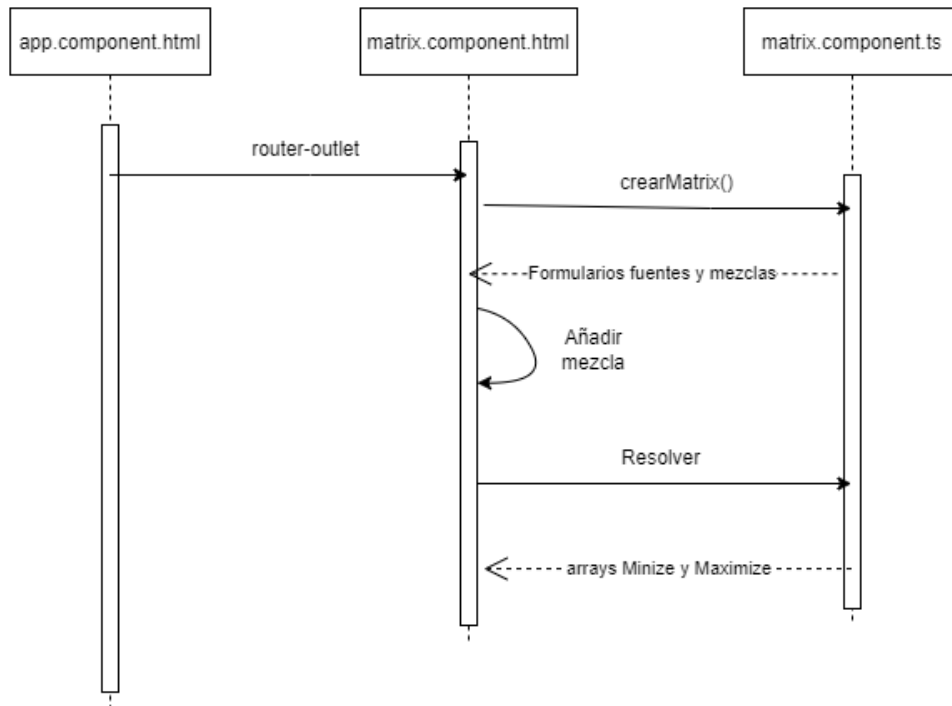


Figura C.1: Flujo-01

Flujo-02

En este segundo flujo partimos del estado donde termino el flujo de la figura C.1, el cual es justo cuando el usuario visualiza la solución. Justo debajo de la solución se podrían encontrar dos botones “Exportar problema” y “Exportar solución”. El primero activa la función `exportProblem()` que genera un documento con los datos de entrada y llama a un componente de angular encargado de generar el archivo a descargar en el navegador. Este proceso es repetido de la misma forma al pulsar “Exportar solución”, aunque añadiendo los datos de los arrays solución. Por último el usuario puede hacer click sobre cualquiera de los números de la solución y esto provocara una llamada a la función `download()` que genera un documento con la resolución de los pasos intermedios y este es descargado en el propio navegador al igual que los documentos anteriores.

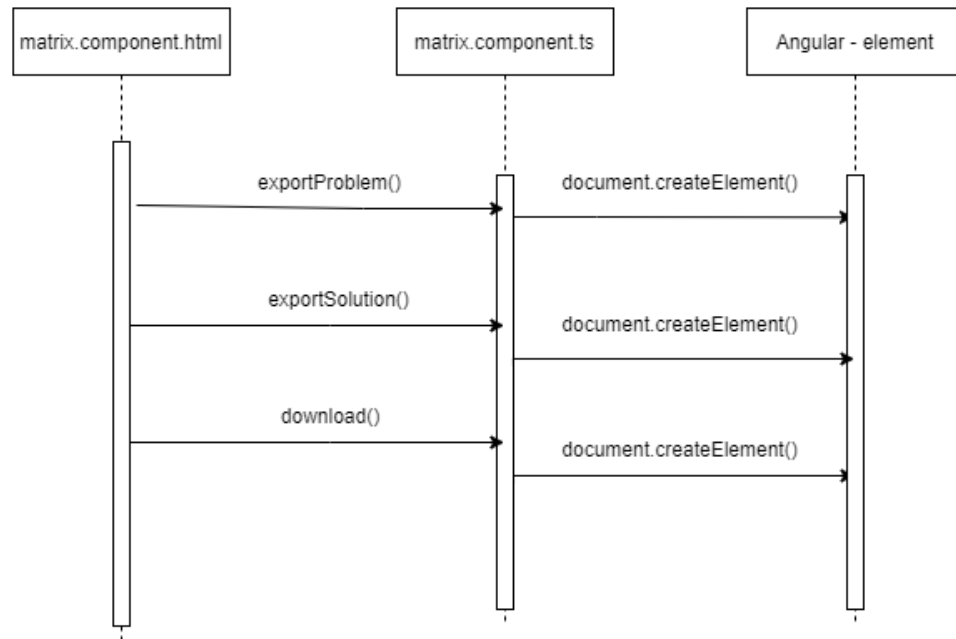


Figura C.2: Flujo-02

Flujo cambio idioma - traducciones

En este flujo hemos realizado el procedimiento llevado por la aplicación para realizar el cambio de idioma de las traducciones. Partimos de cualquier estado ya que el menú superior siempre está presente en la aplicación. Por tanto en el momento que cambia el select al otro idioma disponible se lanza un evento de cambio *changeLang(event.target())* este evento cambia una variable ubicada en el local storage de la aplicación y toma como valor el seleccionado por el usuario. Esto hace que al realizar las traducciones tome el fichero correspondiente al idioma marcado *assets/i18n/es.json* o *assets/i18n/en.json*.

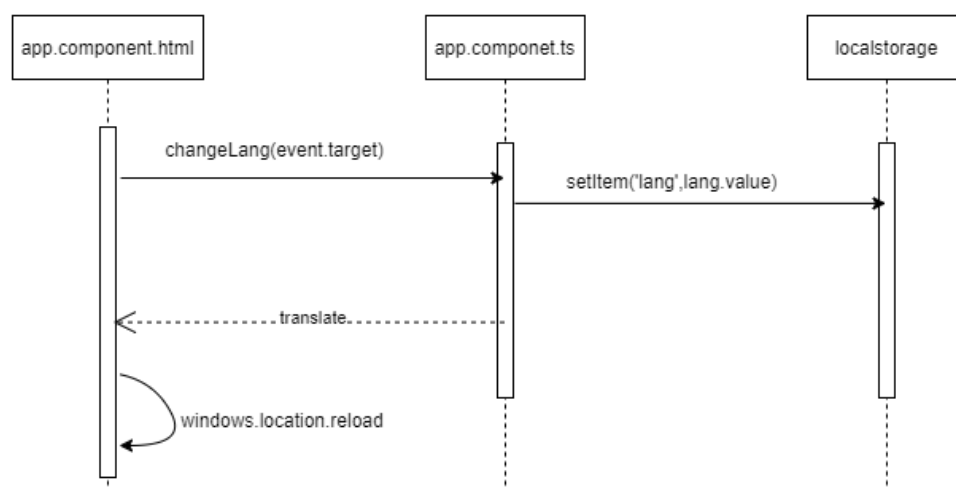


Figura C.3: Flujo traducciones

C.4. Diseño arquitectónico

Para realizar la arquitectura funcional de la aplicación se ha utilizado Angular, que una de sus características es que permite la creación de páginas tipo Single Page Application (SPA). Se ha desarrollado el código siguiendo el patrón de diseño MVVM que divide la estructura del código en dos claras partes modelo y vista. En angular nos encontramos con dos tipos de archivos `.ts` encargándose de la lógica y las operaciones, y `.html` junto a `.css` encargados de la vista.

El proyecto de angular está dividido en componentes que están formados por:

- **Archivo `.ts`:** Vendría a ser el “Modelo” y contendría los datos a los que se accede desde la vista, además de métodos que modifican esos datos. La propiedad *“two way binding”* hace un enlace bidireccional permitiendo a los componentes de la aplicación compartir datos, permitiendo escuchar eventos y actualizar valores simultáneamente.
- **Archivo `.html`:** Junto a el archivo `.css` que aporta los estilos formarían la “Vista” de la aplicación, presentando al usuario los datos en una interfaz fácil de entender.

Servicios

Un Servicio o *Service* es una clase, comúnmente decorada con el decorador Inyector de Angular, que indica que el Servicio puede inyectar otras dependencias de la aplicación, ya sean otros servicios como `Http` o hacer consultas AJAX.

Módulos

Los `NgModules` son contenedores para un bloque cohesivo de código dedicado al dominio de la aplicación, un flujo de trabajo o importaciones de librerías. Puede contener componentes, proveedores de servicios y otros archivos. Estos módulos se suelen dividir en las mismas partes: *declarations*, *imports*, *exports* y *providers*.

Apéndice *D*

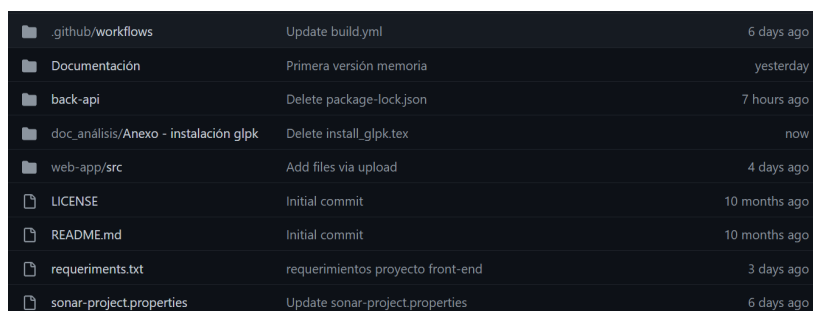
Documentación técnica de programación

D.1. Introducción

En este apéndice se detallan aspectos relacionados a la documentación técnica. Se describirá la estructura de directorios, un apartado dedicado al montaje de desarrollo con el cual se ha trabajado, y un último apartado, dedicado a pruebas del sistema.

D.2. Estructura de directorios

La estructura de directorios (ver figura [D.1](#)) del repositorio GitHub contienen la aplicación web, la API y la documentación del trabajo.



📁 .github/workflows	Update build.yml	6 days ago
📁 Documentación	Primera versión memoria	yesterday
📁 back-api	Delete package-lock.json	7 hours ago
📁 doc_análisis/Anexo - instalación glpk	Delete install_glpk.tex	now
📁 web-app/src	Add files via upload	4 days ago
📄 LICENSE	Initial commit	10 months ago
📄 README.md	Initial commit	10 months ago
📄 requirements.txt	requerimientos proyecto front-end	3 days ago
📄 sonar-project.properties	Update sonar-project.properties	6 days ago

Figura D.1: Estructura directorios, repositorio GitHub

.github.workflows

Contiene archivos de configuración para el análisis de los códigos desde la herramienta Sonarcloud.

Documentación

Este directorio contiene los documentos de la memoria y los anexos. Así como los archivos .tex y las imágenes que forman los documentos.

back-api

Carpeta contiene el código de la API, que es un proyecto Laravel. Es importante destacar que este documento **no se ha usado en la parte de la subida al servidor**. Debido a problemas con la instalación de la librería GLPK [1]. Pero hemos querido añadirlo, ya que ha sido un proyecto que ha servido como aprendizaje sobre como se comunican un proyecto front con un back.

doc__análisis

Carpeta contiene documentos de análisis en L^AT_EX. Algunas partes de estos documentos se han añadido a la memoria. Pero en su mayoría su objetivo ha sido documentar algunas lecturas interesantes del proceso de aprendizaje de herramientas. Y realizar algún documento de práctica con la nomenclatura de L^AT_EX.

web-app

En este directorio hemos subido el contenido de la carpeta **src** del proyecto Angular. En la figura D.2 podemos ver los directorios default que tiene cualquier proyecto de angular.

- **Directorio app:** index.html es la página que contiene los componentes de las aplicaciones Angular. Estos componentes se encuentran dentro del directorio app (ver figura D.3).

app	16/09/2022 18:09	Carpeta de archivos	
assets	16/09/2022 18:09	Carpeta de archivos	
environments	16/09/2022 18:09	Carpeta de archivos	
constantes.ts	16/09/2022 18:09	Archivo TS	1 KB
favicon.ico	16/09/2022 18:09	Icono	1 KB
glpk.all.wasm	16/09/2022 18:09	Archivo WASM	919 KB
index.html	16/09/2022 18:09	Chrome HTML Docu...	1 KB
main.ts	16/09/2022 18:09	Archivo TS	1 KB
polyfills.ts	16/09/2022 18:09	Archivo TS	3 KB
styles.css	16/09/2022 18:09	Documento de hoja ...	1 KB
test.ts	16/09/2022 18:09	Archivo TS	1 KB

Figura D.2: Estructura directorio web-app/src

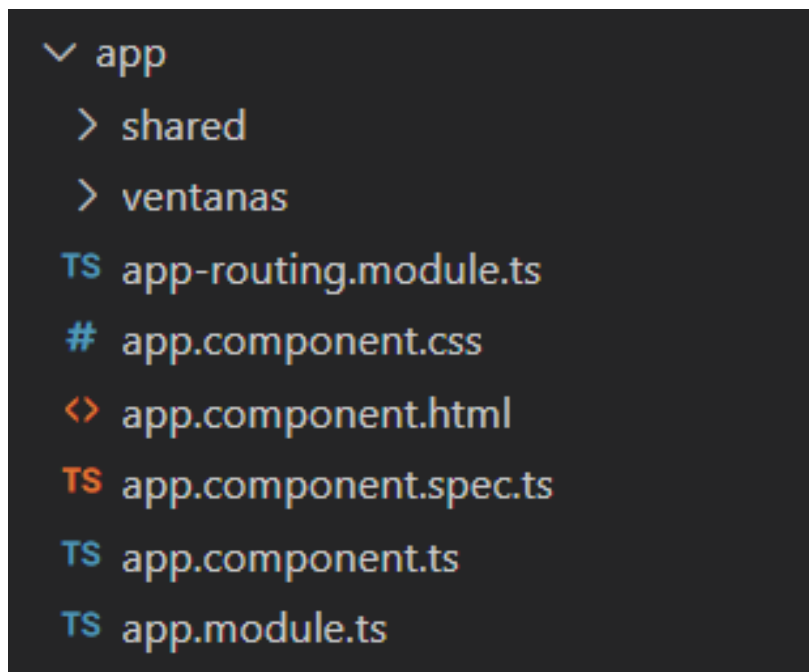


Figura D.3: Estructura directorio web-app/src/app

Directorios incluidos:

- **app-routing.module.ts:** contiene las rutas de la aplicación, están asignadas a componentes o módulos
- **app.component.css:** hoja de estilos del archivo *app.component.html*
- **app.component.html:** archivo con la vista de la aplicación, archivo al que se apunta desde *index.html*.
- **app.component.spec.ts:** archivo de pruebas del componente *AppComponent*, que es el componente inicial de la aplicación.

- **app.component.ts:** archivo incluye la lógica del componente de arranque.
 - **app.module.ts:** contiene la configuración de los imports y exports de módulos y componentes.
 - **carpeta shared:** se ha configurado como un módulo que contiene componentes comunes en toda la aplicación.
 - **carpeta ventanas:** contiene los componentes de las rutas principales de la aplicación.
- **Directorio assets:** es el encargado de almacenar elementos estáticos como imágenes, pdfs, archivos mp3, etc (ver figura D.4).

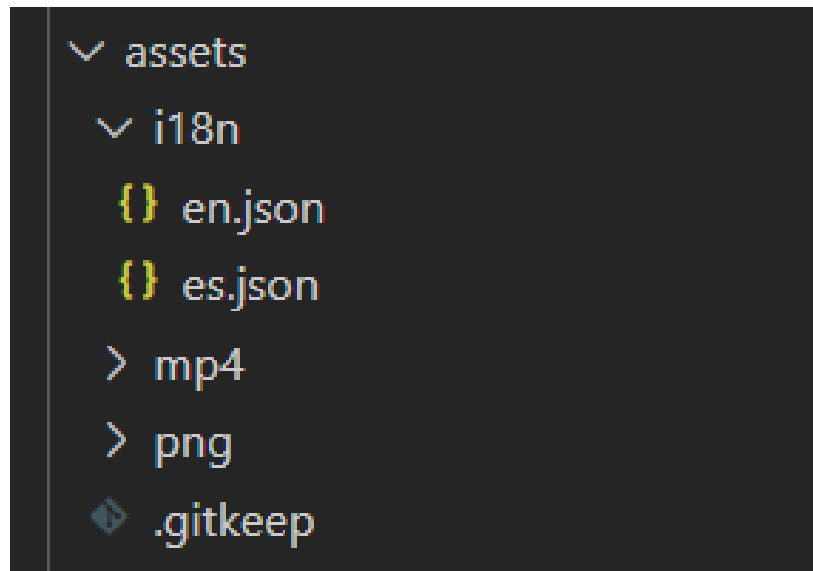


Figura D.4: Estructura directorio web-app/src/assets

Directorios incluidos:

- **Carpeta i18n:** contiene los archivos *.json* con las traducciones tanto al inglés como al español.
 - **Carpeta mp4:** contiene archivos con formato *.mp4*, en este caso únicamente el vídeo guía de uso de la aplicación.
 - **Carpeta png:** carpeta que contiene todas las imágenes e iconos usados en la aplicación
-
- **Directorio environments:** nos permitir definir configuraciones para desplegar en un entorno local o producción.
 - **constantes.ts:** archivo destinado a definir constantes globales para toda la aplicación.
 - **favicon.ico:** es 16x16 icon que sirve de icono de la web, permitiendo localizar al usuario la aplicación cuando se trabaja con múltiples ventanas
 - **glpk.all.wasm:** archivo librería instalada GLPK.
 - **index.html:** estructura inicial de documento HTML5; en este archivo se realiza la carga de scripts, estilos y dependencias necesarias. Además este archivo hace una llamada al fichero *favicon.ico*
 - **main.ts:** Este fichero es el encargado de definir qué módulo es el de arranque. Apuntando normalmente al AppComponent.
 - **polyfills.ts:** Contiene algunos archivos para que nuestra aplicación sea compatible con algunos navegadores antiguos. El código se transpila a ES6 que no es compatible con algunas versiones de Internet Explorer o Firefox (ES5 si que es compatible)
 - **styles.css:** hoja de estilos asociada al archivo *index.html*. Es llamado desde el archivo *angular.json* configurándolo como la hoja principal de estilos de la aplicación.
 - **test.ts:** Define la configuración que va a utilizar Karma. En este archivo se define el entorno de prueba.

Ficheros adicionales

- **LICENSE:** archivo contiene información referente a la licencia marcada en el repositorio.
- **README.md:** Archivo presentación del repositorio GitHub y del proyecto.
- **requirements.txt:** contiene parte del archivo **package.json** del proyecto en Angular. Está formado por las dependencias de la aplicación, y se usa en el proceso de montaje del proyecto.
- **sonar-project.properties:** archivo de configuración de SonarCloud en el repositorio

D.3. Manual del programador

En esta sección se va a realizar la configuración de un entorno de desarrollo con las herramientas principales con las que se ha trabajado.

Instalación Visual Studio Code

Visual Studio Code se ha usado como framework de desarrollo. Se puede descargar mediante en enlace <https://code.visualstudio.com/download>, donde podremos descargar la versión correspondiente a nuestro sistema operativo.

Algunas de las extensiones que hemos utilizado en el desarrollo son (figura D.5):

Instalación Postman

Para ejecutar las pruebas funcionales de la API, hemos utilizado la herramienta Postman que se puede descargar en el siguiente link de descarga: <https://www.postman.com/downloads/>

Instalación Nodejs y npm

Para el proyecto angular es necesario tener una versión de Nodejs instalada. Podemos obtener este link de descarga en su página oficial: <https://nodejs.org/es/download/>, seleccionando la versión dependiendo nuestro sistema. Después de la instalación es recomendable reiniciar el equipo.

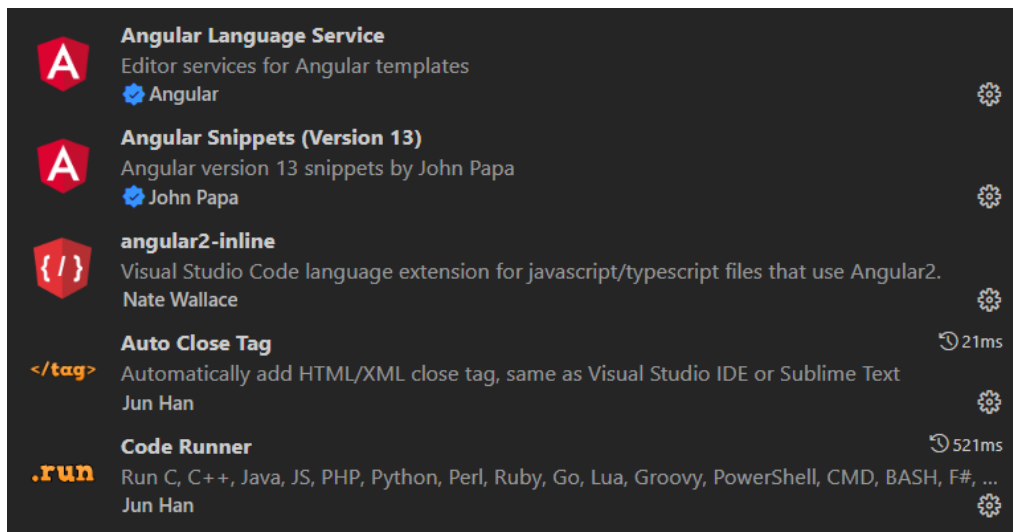


Figura D.5: Visual Studio Code extensiones

Instalación Debian

Hemos usado Debian para simular un subsistema Linux en un sistema principal Windows. Link de descarga en: <https://apps.microsoft.com/store/detail/debian/9MSVKQC78PK6?hl=es-es&gl=es>

D.4. Compilación, instalación y ejecución del proyecto

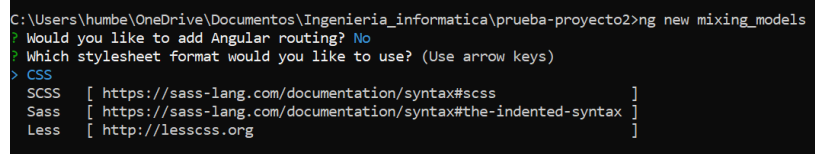
Para utilizar el proyecto es necesario dirigirse al Github en https://github.com/humbertoms99/Mixing_models, y descargar el zip del repositorio.

Instalación Angular y librerías Node

Previamente debemos tener instalado Nodejs, ya que vamos a necesitar ejecutar algunos comandos npm.

Nos posicionamos en la ruta donde queremos montar el proyecto y ejecutamos el comando `npm install -g @angular/cli` que instalará las dependencias angular/cli. Creamos un proyecto angular `ng new mixing_models` (ver figura D.6), y entramos en la carpeta creada con `cd mixing_models`. El siguiente

paso es instalar las librerías (Recomendable hacerlo con la opción más abajo):



```
C:\Users\humbe\OneDrive\Documentos\Ingenieria_informatica\prueba-proyecto2>ng new mixing_models
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SCSS [ https://sass-lang.com/documentation/syntax#scss ]
  Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less [ http://lesscss.org ]
```

Figura D.6: Comando `ng new nombre-proyecto`

Instalación librería Fast Combinatorial Non-negative Least Squares:

$$npm\ i\ ml - fcnnls$$

Instalación GLPK interface for TypeScript:

$$npm\ install\ glpk - ts$$

Angular Material:

$$ng\ add\ @angular/material$$

Angular ngx-translate:

$$npm\ install\ @ngx - translate/core\ @ngx - translate/http - loader$$

GLPK compiled to wasm:

$$npm\ i\ glpk - wasm$$

Y los comandos:

$$npm\ install\ @types/node - --save - dev$$

$$npm\ install\ @angular/localize - --save$$

En vez de realizar estas instalaciones podemos abrir el archivo **package.json** y copiar el contenido del archivo *requeriment.txt* del GitHub. Y copiar de esta forma las dependencias del proyecto, y para instalarlas ejecutamos el comando:

$$npm\ install - --legacy - peer - deps$$

En el archivo **angular.json**:

```
"assets": [  
  "src/favicon.ico",  
  "src/assets",  
  "src/glpk.all.wasm"  
],
```

Figura D.7: Configuración angular - angular.json

En el archivo **tsconfig.json**:

```
"target": "es2017",
```

Figura D.8: Configuración angular - tsconfig.json

En el archivo **tsconfig.app.json**

```
"types": [ "node" ],  
"typeRoots": [ "../node_modules/@types" ]
```

Figura D.9: Configuración angular - tsconfig.app.json

Por último, **borramos la carpeta src** y pegamos la carpeta src que se encuentra dentro la carpeta web-app del zip descargado del repositorio del GitHub.

Ejecución del proyecto

Una vez configurado el proyecto, abrimos la terminal y nos posicionamos en path del carpeta del proyecto con el comando:

Y para ejecutar la aplicación:

`ng serve -o`

Con este comando no abrirá automáticamente una ventana en nuestro navegador, con la dirección `http://localhost:4200/`

D.5. Pruebas del sistema

Las pruebas de sistema validan el sistema completo y totalmente integrado. El objetivo de las pruebas de sistema es evaluar las especificaciones del sistema en su totalidad.

Para realizar estas pruebas hemos utilizado la herramienta SonarCloud que es una plataforma para evaluar la calidad del código fuentes. Es una herramienta de software libre y está vinculada con GitHub para obtener métricas de calidad de código.

Para poder realizar los análisis se han añadido los archivos `sonar-project.properties` y `.github/workflows/build.yml` y una key en el apartado de configuración del repositorio de GitHub. Esto hace que Sonarcloud analice el contenido actual de la carpeta del proyecto Github.

Podemos ver el resumen del análisis del código en la figura [D.10](#).



Figura D.10: Resumen análisis código Sonarcloud

- **Bugs:** errores de código que pueden romper el código y deben ser corregidos cuando antes.
- **Vulnerabilities:** partes del código que pueden ser explotadas por un hacker.
- **CodeSmells:** códigos que son confusos, y por tanto difíciles de mantener.
- **Security Hotspots:** código sensible a la seguridad que requiere una revisión manual para evaluar si existe o no una vulnerabilidad.

- **Duplications:** porcentaje de códigos duplicados en las nuevas líneas
- **Líneas de códigos y lenguajes utilizados.**

Además si accedemos a una rama concreta podemos ver análisis más específicos, como por ejemplo el listado de issues (figura D.11). Consideramos muy útil este apartado porque aparte de ofrecernos el listado completo de fallos pendientes por arreglar, nos da una aproximación de tiempo de cada issue y un tiempo total aproximado para todas, establece importancia y permite añadir etiquetas. Además permite asignar estas tareas al estar sincronizado con GitHub.

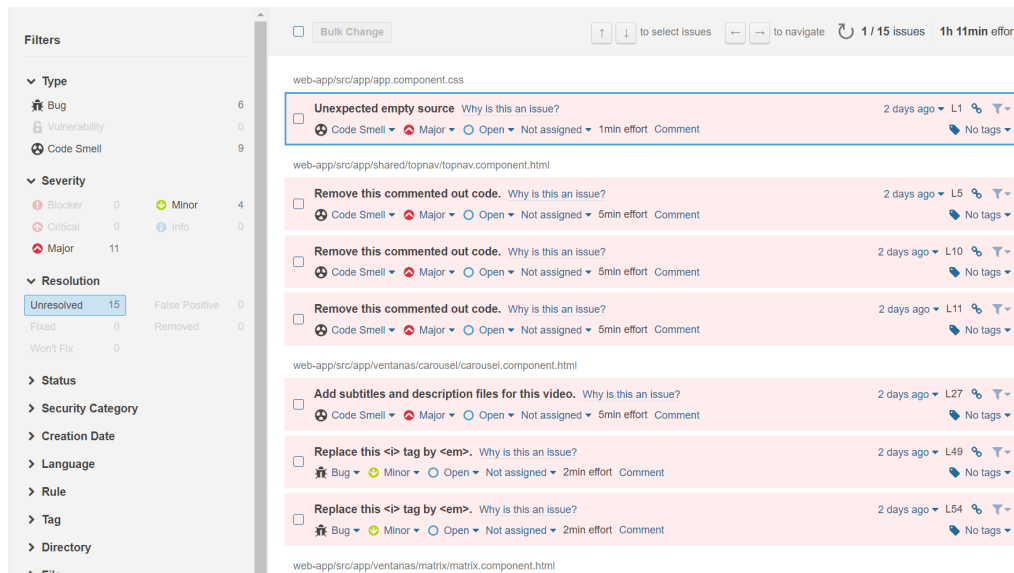


Figura D.11: Sonarcloud issues brand main

Por último, podemos entrar al apartado de **Code** (ver figura D.12). En este apartado veremos un listado de todos los archivos, sus líneas de código y los Bugs, vulnerabilities, Code Smells, Security Hotspots, Coverage y Duplications. Apartado que puede servir para ver una visión general del código.

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
 ventanas	447	6	0	5	0	—	0.0%
 carousel	193	2	0	1	0	—	0.0%
 interceptors	—	0	0	0	0	—	0.0%
 matrix	241	4	0	1	0	—	0.0%
 nav	8	0	0	0	0	—	0.0%
 Services	—	0	0	0	0	—	0.0%
 video	5	0	0	3	0	—	0.0%
 ventanas-routing.module.ts	—	0	0	0	0	—	0.0%
 ventanas.module.ts	—	0	0	0	0	—	0.0%

Figura D.12: Sonarcloud líneas de código

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice se detallan los requisitos para acceder al proyecto por parte de un usuario. Además se han descrito las funcionalidades de la aplicación en un manual de usuario.

E.2. Requisitos de usuarios

Al ser una aplicación web únicamente es necesario tener un navegador web y acceder mediante el enlace <https://mixingmodels.netlify.app/>. Por otra parte si se quiere poder ver el contenido .csv, se requiere alguna herramienta de lectura de hojas de cálculo.

E.3. Instalación

No requiere ninguna instalación.

E.4. Manual del usuario

Esta sección es dedicada a un apartado de guía de usuario, orientando a las personas que utilicen la web, ofreciendo asistencias a los usuarios y recopilando todas las funcionalidades que ofrece.

Menú superior

En la parte de superior de la pantalla encontramos un menú superior **E.1** que organiza la web en algunos apartados:



Figura E.1: Top Navigation Bar

1. **Problema:** vista principal contiene la resolución de problemas de sistemas ecuaciones con restricciones, recepción y exportación de datos.
2. **Video:** vista contiene un vídeo guía, que permite ver las principales funcionalidades de la aplicación al usuario aportando un ejemplo básico de resolución.
3. **Icono GitHub:** enlace directo al repositorio GitHub, que contiene el código de la aplicación y documentación del proyecto.
4. **About us:** desplegable que muestra los desarrolladores del proyecto.
5. **Select de traducciones:** seleccionable que nos permite cambiar entre los dos idiomas de la aplicación Inglés y Español, únicamente dando click en al idioma que queramos cambiar.

Ventana resolución problema

En esta ventana vemos la barra de navegación superior, y el contenido de la aplicación. El funcionamiento de la web es que según se introducen datos y se va creando la estructura del problema a resolver, se van desbloqueando apartados, terminando el último con la solución del problema y los botones de exportar.

En la Figura **E.2** aparece el primer formulario de la aplicación, destinado a marcar el tamaño de la matriz de fuentes de datos. Es importante recalcar que el botón “*Crear*” no se activa mientras los valores de marcadores y fuentes sean inferiores o iguales a cero. Una vez rellenemos el formulario con valores válidos se habilitara el botón. Por otra parte disponemos de un botón adicional (Figura **E.9**) que permite al usuario cargar una plantilla de datos. Esta parte se comentará en profundidad más adelante analizando la plantilla que requiere la aplicación.

Problema Video About Español

Modelos de mezcla

Los modelos de mezcla son una herramienta estadística que nos permite estimar contribuciones de fuentes a una mezcla.
Esta página permite resolver sistemas de ecuaciones lineales con restricciones, y calcular la función objetivo de estas ecuaciones

Introducción datos:

Número de marcadores
0

Número de fuentes
0

Crear

Seleccionar archivo Ninguno a...hivo selec.

Figura E.2: Ventana inicial web

Una vez se habilita y pulsamos “*Crear*”, muestra un nuevo formulario (Figura E.3). Siendo una matriz del tamaño fijado en el primer formulario, las filas son el número de fuentes y las columnas el número de marcadores. Se permite la introducción con números decimales, y para marcar la coma decimal se realiza con ".". Por otra parte, se ha añadido la opción de nombrar a las fuentes según queramos ofreciendo más personalización al problema.

	marcador 1	marcador 2
Fuente- 1	01	2
Fuente- 2	4	3
Fuente- 3	8	4

Mezcla-1	1	4	+
-----------------	---	---	---

Resolver

Figura E.3: Inputs datos matriz de fuentes

En la parte inferior se encuentra la introducción de datos de las mezclas, teniendo estas el mismo número de marcadores que las fuentes. Inicialmente se encuentran datos de entrada para una única mezcla, pero podemos añadir con el botón situado a su derecha en la figura E.3, al pulsar, crea una nueva mezcla con el mismo tamaño de marcadores que el resto.

Una vez hemos configurado la matriz de datos de entrada tanto de fuentes como de mezclas damos a “*Resolver*”. En este momento se ejecuta una función que primero calcula las proyecciones de las mezclas respecto al plano de las fuentes y posteriormente se usa la librería GLPK para resolver el sistema. El ejercicio resuelve el sistema de ecuaciones con la restricción de que todas las x cantidades deben ser superiores a 0, esto es porque no se puede tener negativos de una cantidad.

Por último se muestra la solución del sistema de ecuaciones mostrando las cantidades mínimas y máximas que pueden tener las fuentes respecto a las mezclas (Figura E.4).

Las siguientes mezclas no pueden haber sido generadas por las fuentes proporcionadas
Se ha tomado la mezcla compatible más cercana a la proporcionada por el usuario
(1 4) -> (1.800 3.600)

	Mezcla 1		Mezcla 2	
	[Min	, Max]	[Min	, Max]
Fuente- 1	[1.80	, 1.80]	[0.00	, 1.33]
Fuente- 2	[0.00	, 0.00]	[0.00	, 2.00]
Fuente- 3	[0.00	, 0.00]	[0.00	, 0.83]

Exportar problema
Exportar solución

Figura E.4: Arrays solución máximos y mínimos

En caso de que los “*puntos mezclas*” se encuentre fuera del plano de fuentes, en una primera parte nos mostrará las proyecciones, las cuales son los valores utilizados en la resolución del sistema.

Justo debajo de las proyecciones encontramos los arrays de solución de mínimos y máximos para cada mezcla sobre las fuentes. En la figura E.4 vemos entonces que la *Fuente-1* tendría una proporción mínima de 1,80 y una máxima de 1,80 sobre la *Mezcla 1*, y una proporción mínima de 0,00 y una máxima de 1,33 sobre la *Mezcla 2*, comentar que estos valores solución se han redondeado a dos decimales y las proyecciones a tres.

Una vez hemos resuelto el ejercicio, la aplicación ofrece diferentes opciones de exportar los datos de la solución, a continuación veremos las tres posibilidades:

Exportar problema: en la figura E.4 si presionamos “*Exportar problema*” nos descarga un archivo con nombre “*InputData.csv*” que contiene los datos de la figura E.5. Manteniendo siempre la misma estructura: las dos primeras filas marcan el tamaño de la matriz de datos **marcadores** y **fuentes**, una fila en blanco y justo debajo una fila por cada fuente que se haya especificado. Por último, dejando una línea de separación encontramos una fila para cada mezcla.

El objetivo de este documento es servir como plantilla que permita cargar este mismo problema en futuros usos.

	A	B	C	D
1	Marcadores:	2		
2	Fuentes:	3		
3				
4	Fuente- 1:	1	2	
5	Fuente- 2:	4	3	
6	Fuente- 3:	8	4	
7				
8	Mezcla- 1:	1	4	
9	Mezcla- 2:	9	6	
10				

Figura E.5: Export datos entrada problema

Exportar solución: en la figura E.4 si presionamos “*Exportar solución*” nos descarga un archivo de nombre “*Solution.csv*”, que contiene la estructura de datos de la figura E.6. El objetivo de este documento es tanto mostrar los datos de entrada como la solución del sistema, en las primeras filas encontramos el mismo diseño que con “*Exportar problema*”. A continuación, se deja una línea en blanco y se muestra las proyecciones de las mezclas en caso de haberlas. Se deja otra fila en blanco y se muestra una estructura de tabla muy parecida a la visual de la web.

	A	B	C	D	E	F	G
1	Marcadores:	2					
2	Fuentes:	3					
3							
4	Fuente- 1:	1	2				
5	Fuente- 2:	4	3				
6	Fuente- 3:	8	4				
7							
8	Mezcla- 1:	1	4				
9	Mezcla- 2:	9	6				
10							
11	Mezcla -> proyeccion: (1 4) -> (1.7999999999999998 3.5999999999999996)						
12							
13		Mezcla- 0			Mezcla- 1		
14		[Min	,Max]		[Min	,Max]	
15	Fuente- 1:	1.8	1.8		0	1	
16	Fuente- 2:	0	0		0	1.5	
17	Fuente- 3:	0	0		0.375	1	
18							

Figura E.6: Export solución completa

Exportar pasos intermedios: para descargar este documento podemos pulsar encima del número de la solución del cual queramos ver sus pasos intermedios, en el caso de la figura E.7 se ha pulsado sobre el valor 1,80 de la *Fuente-1* y *Min* de la *Mezcla-1*.

La estructura del documento está compuesta por unas primeras filas con restricciones del sistema, la función objetivo y las ecuaciones del sistema de ecuaciones.

A partir del documento de la figura E.5 que funciona como plantilla de entrada podemos pulsar el botón "*Seleccionar archivo*" de la figura E.8. Esta acción abre el nuestro sistema de directorios (Figura E.9) y podemos seleccionar la plantilla de datos exportada anteriormente.

	A	B	C	D
1	var x1 >=0	# Variable definition		
2	var x2 >=0	# Variable definition		
3	var x3 >=0	# Variable definition		
4	minimize z: x1	# Objective function		
5	subject to con1: 1*x1 + 4*x2 + 8*x3 = 1.8	# Constraint 1		
6	subject to con2: 2*x1 + 3*x2 + 4*x3 = 3.6	# Constraint 2		
7	end			
8				

Figura E.7: Export resolución paso intermedio

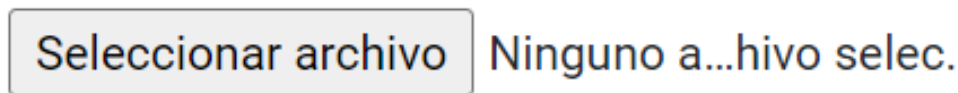


Figura E.8: Botón seleccionar archivo

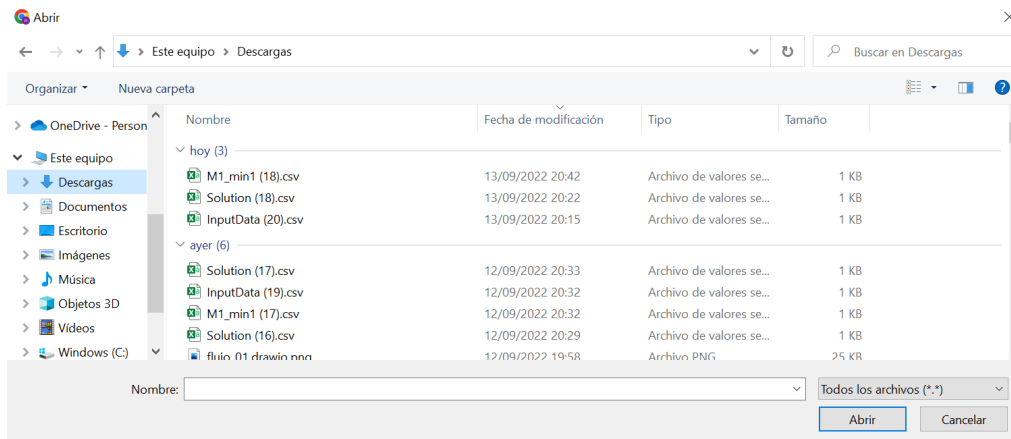


Figura E.9: Importar plantilla

El resultado de seleccionar una plantilla de datos es el de la figura E.10, con la matriz de fuentes y mezclas con los valores y tamaños de la plantilla. Esto deja la aplicación justo antes de dar a “Resolver”.

Problema

Video

About

Español

Modelos de mezcla

Los modelos de mezcla son una herramienta estadística que nos permite estimar contribuciones de fuentes a una mezcla.

Esta página permite resolver sistemas de ecuaciones lineales con restricciones, y calcular la función objetivo de estas ecuaciones

Introducción datos:

Número de marcadores

2

Número de fuentes

3

Crear

Seleccionar archivo

InputData (21).csv

	marcador 1	marcador 2
Fuente- 1	1	2
Fuente- 2	4	3
Fuente- 3	8	4
Mezcla-1	1	4
Mezcla-2	8	6

Resolver

Figura E.10: Importar visual web

Bibliografía

- [1] Inc. Free Software Foundation. Glpk (gnu linear programming kit). <https://www.gnu.org/software/glpk/>, 2012. [Intenet; descargado 10-febrero-2022].
- [2] Indeed. Web oficial indeed. <https://es.indeed.com/>. [Online].
- [3] GNU General Public License. Gnu general public license. <https://www.gnu.org/licenses/gpl-3.0.html>, 2007. [Online].
- [4] Seguridad Social y Migraciones Ministerio de Inclusión. Web seguridad social. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/3653>. [Online].
- [5] M. L. Campagnolo P. Barcia, M. N. Bugalho and J. O. Cerdeira. Using n-alkanes to estimate diet composition of herbivores: a novel mathematical approach. *Animal(2007)*, 1:141–149, 2007.