



# Escuela de Ingenierías Industrial, Informática y Aeroespacial

## GRADO EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

APLICACIÓN WEB PARA CÁLCULOS DE ESTRUCTURAS  
DE ACERO

WEB APPLICATION FOR STEEL STRUCTURE  
CALCULATIONS

Autor: Javier Alonso Fernández  
Tutor: Héctor Alaiz Moretón  
Co-Tutor: Manuel Ignacio Guerra Romero

(Julio, 2022)

**UNIVERSIDAD DE LEÓN**  
**Escuela de Ingenierías Industrial, Informática y**  
**Aeroespacial**

**GRADO EN INGENIERÍA INFORMÁTICA**  
**Trabajo de Fin de Grado**

**ALUMNO:** Javier Alonso Fernández

**TUTOR:** Héctor Alaiz Moretón

**CO-TUTOR:** Manuel Ignacio Guerra Romero

**TÍTULO:** Aplicación web para cálculos de estructuras de acero

**TITLE:** Web application for steel structure calculations

**CONVOCATORIA:** Julio, 2022

**RESUMEN:**

A lo largo del proyecto se desarrollará una aplicación web utilizando Vue.js y Python. La aplicación constará de diversos apartados donde se realizarán cálculos diferentes sobre resistencias a esfuerzos en diferentes tipos y perfiles de acero. El proyecto constará de 4 apartados diferenciados, el primero será para realizar cálculos de resistencias para cada una de las secciones a cada uno de los esfuerzos. El objetivo de la segunda vista será hallar las resistencias al pandeo lateral. La tercera calculará la resistencia de la barra a compresión y verificará el esfuerzo. Y, por último, en la cuarta vista se calculará la interacción de los esfuerzos en barra, además de presentar un resumen de todas las verificaciones realizadas en las otras vistas. Para cada uno de estos apartados será necesario introducir datos de forma previa a la ejecución. Para ello, se deberán rellenar unos formularios para así obtener todos los datos del problema, estos formularios se irán actualizando de forma dinámica, para así poder obtener todos los datos correspondientes a cada apartado. Una vez se hayan introducido todos los datos, la aplicación ofrecerá un botón para iniciar los cálculos y deberá mostrarlos por pantalla ordenador en tablas. Además, cada apartado ofrecerá la opción de seguir un desarrollo por pasos de los cálculos, donde se mostrarán los cálculos realizados además de las fórmulas empleadas o demás imágenes que puedan ser útiles para el usuario. Para integrar Vue.js con Python se utilizará Flask, para así realizar los cálculos en Python y devolver el resultado a la vista correspondiente.

**ABSTRACT:**

Throughout the project a web application will be developed using Vue.js and Python. The application will consist in several sections where different calculations will be done. The project will consist of 4 different sections, the first one will be to perform resistance calculations for each of the sections to

each of the efforts. The objective of the second view will be to find the lateral buckling resistances. The third will calculate the compressive bar buckling and verify the efforts. And finally, the fourth view will calculate the bar efforts interaction and present a summary of all the verifications performed in the other views. For each of these sections it will be necessary to enter data before the execution. To do this, some forms must be filled in order to obtain all the data needed to solve the problem, these forms will be updated dynamically, in order to obtain all the data corresponding to each section. Once all the data have been entered, the application will offer a button to start the calculations and should display them on the computer screen in tables. In addition, each section will offer the option to follow a step-by-step development of the calculations, where the calculations performed will be shown in addition to the formulas used or other images that may be useful for the user. To integrate Vue.js with Python, Flask will be used, the calculations will be done in Python and return the result to the corresponding view.

**Palabras clave:** aplicación web, Python, Vue.js, vistas, Flask.

**Firma del alumno:**

**VºBº Tutor/es:**

# ÍNDICE

ÍNDICE .....	4
ÍNDICE DE FIGURAS .....	7
ÍNDICE DE TABLAS.....	9
<b>1 INTRODUCCIÓN.....</b>	<b>10</b>
1.1 JUSTIFICACIÓN DEL PROYECTO .....	10
1.2 OBJETIVOS DEL PROYECTO .....	10
1.3 ESTUDIO DE LA SITUACIÓN ACTUAL .....	11
1.4 EVALUACIÓN ALTERNATIVAS .....	11
1.5 METODOLOGÍA .....	11
1.5.1 FASES DEL PROYECTO .....	11
1.6 ESTRUCTURA DE LA MEMORIA.....	14
<b>2 ASPECTOS TEORICOS.....</b>	<b>15</b>
2.1 APLICACIÓN WEB.....	15
2.2 BACKEND .....	15
2.3 FRONTEND .....	15
2.4 API REST .....	15
2.5 MÉTODOS DE PETICIÓN HTTP .....	15
<b>3 GESTIÓN Y PLANIFICACIÓN DEL PROYECTO.....</b>	<b>17</b>
3.1 ALCANCE DEL PROYECTO .....	17
3.2 PLAN DE TRABAJO .....	17
3.3 GESTIÓN DE RECURSOS.....	18
3.4 PLANIFICACIÓN.....	18
<b>4 SITUACIÓN ACTUAL .....</b>	<b>19</b>
<b>5 ANÁLISIS .....</b>	<b>21</b>
5.1 DEFINICIÓN DEL SISTEMA .....	21
5.1.1 DETERMINACIÓN DEL ALCANCE DEL SISTEMA .....	21
5.2 REQUISITOS DEL SISTEMA.....	21

5.2.1 REQUISITOS FUNCIONALES.....	21
5.2.2 REQUISITOS NO FUNCIONALES .....	23
5.2.3 IDENTIFICACIÓN DE ACTORES DEL SISTEMA .....	24
5.2.4 CASOS DE USO .....	24
5.3 DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS .....	29
5.3.1 DIAGRAMA DE CLASES .....	29
5.3.2 DESCRIPCIÓN DE LAS CLASES.....	30
5.4 ANÁLISIS DE CASOS DE USO Y ESCENARIOS.....	30
5.4.1 CASO DE USO 1 .....	30
5.4.2 CASO DE USO 2 .....	31
5.4.3 CASO DE USO 3 .....	31
5.4.4 CASO DE USO 4 .....	31
5.4.5 CASO DE USO 5 .....	31
5.4.6 CASO DE USO 6 .....	32
5.4.7 CASO DE USO 7 .....	32
5.4.8 CASO DE USO 8 .....	32
5.4.9 CASO DE USO 9 .....	33
5.5 ANÁLISIS DE INTERFACES DE USUARIO .....	33
<b>6 DISEÑO DEL SISTEMA.....</b>	<b>37</b>
6.1 ARQUITECTURA DEL SISTEMA .....	37
6.2 DISEÑO DE CLASES .....	38
6.3 DIAGRAMAS DE INTERACCIÓN .....	39
6.4 DIAGRAMA DE ACTIVIDADES .....	41
6.5 DISEÑO DE LA INTERFAZ .....	42
6.5.1 PRIMERA FASE INTERFAZ.....	42
6.5.2 FASE FINAL INTERFAZ .....	43
<b>7 IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>45</b>
7.1 LENGUAJE HERRAMIENTAS Y LIBRERÍAS UTILIZADAS .....	45
7.2 CREACIÓN DEL SISTEMA .....	46
7.2.1 PROBLEMAS ENCONTRADOS .....	47
7.2.2 DESCRIPCIÓN DETALLADA DE LAS CLASES.....	48
7.2.3 ARCHIVOS PRECARGADOS DEL SISTEMA.....	50

<b>8</b>	<b>RESULTADOS.....</b>	<b>51</b>
8.1	RESULTADOS CÁLCULO DE RESISTENCIAS .....	51
8.2	RESULTADOS PANDEO DE BARRAS A COMPRESIÓN .....	54
8.3	RESULTADOS PANDEO LATERAL.....	56
8.4	RESULTADOS INTERACCIÓN DE ESFUERZOS EN BARRA .....	57
8.5	CONCLUSIONES RESULTADOS .....	60
<b>9</b>	<b>DESARROLLO DE LAS PRUEBAS .....</b>	<b>61</b>
9.1	PRUEBAS UNITARIAS .....	61
9.2	PRUEBAS DE INTEGRACIÓN .....	62
<b>10</b>	<b>MANUALES DEL SISTEMA .....</b>	<b>63</b>
10.1	MANUAL DE INSTALACIÓN .....	63
10.2	MANUAL DE EJECUCIÓN .....	64
10.3	MANUAL DE USUARIO .....	67
10.4	MANUAL DEL PROGRAMADOR.....	67
<b>11</b>	<b>CONCLUSIONES Y AMPLIACIONES .....</b>	<b>68</b>
11.1	CONCLUSIONES .....	68
11.2	AMPLIACIONES Y TRABAJOS FUTUROS .....	68
<b>12</b>	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>70</b>

# ÍNDICE DE FIGURAS

Ilustración 1 Diagrama de Gantt .....	18
Ilustración 2 Diagrama de clases .....	29
Ilustración 3 Análisis de interfaz: Barra de navegación.....	33
Ilustración 4 Análisis de la interfaz: Formulario .....	33
Ilustración 5 Análisis de la interfaz: Desarrollo paso a paso .....	34
Ilustración 6 Análisis de la interfaz: Fórmulas.....	35
Ilustración 7 Análisis de la interfaz: Tabla de resultados .....	35
Ilustración 8 Diagrama de navegación .....	36
Ilustración 9 Estructura de directorios.....	38
Ilustración 10 Diagrama de interacción: Cambio de vista .....	39
Ilustración 11 Diagrama de interacción: Ejecución normal .....	39
Ilustración 12 Diagrama de interacción: Ejecución por pasos, paso siguiente ....	40
Ilustración 13 Diagrama de interacción: Ejecución por pasos, paso anterior .....	40
Ilustración 14 Diagrama de interacción: Introducir datos no válidos .....	41
Ilustración 15 Diagrama de actividades.....	41
Ilustración 16 Boceto inicial de la interfaz.....	42
Ilustración 17 Interfaz fase previa: Error .....	43
Ilustración 18 Interfaz fase final .....	44
Ilustración 19 Logo Vue.js .....	45
Ilustración 20 Logo Flask .....	45
Ilustración 21 Logo Python .....	46
Ilustración 22 Código: Flask-CORS .....	47
Ilustración 23 Código: Ejemplo Flask-CORS @cross_origin.....	47
Ilustración 24 Código: Ciclo de vida de Vue.js .....	49
Ilustración 25 Ejemplo datos perfil Excel .....	51
Ilustración 26 Resultados cálculo de resistencias Excel.....	52
Ilustración 27 Resultado cálculo de resistencias aplicación.....	52
Ilustración 28 Resultados no válidos cálculo de resistencias Excel .....	53
Ilustración 29 Resultados no válidos cálculo de resistencias aplicación .....	53
Ilustración 30 Resultados cálculo pandeo de barras a compresión Excel .....	54

Ilustración 31 Resultados cálculo pandeo de barras a compresión aplicación ....	54
Ilustración 32 Resultados no válidos cálculo pandeo de barras a compresión Excel .....	55
Ilustración 33 Resultados no válidos cálculo pandeo de barras a compresión aplicación .....	55
Ilustración 34 Resultados cálculo pandeo lateral Excel .....	56
Ilustración 35 Resultados cálculo pandeo lateral aplicación .....	56
Ilustración 36 Resultados no válidos cálculo pandeo lateral Excel.....	57
Ilustración 37 Resultados no válidos cálculo pandeo lateral aplicación.....	57
Ilustración 38 Resultados interacción de esfuerzos en barra, datos mal introducidos .....	58
Ilustración 39 Resultados interacción de esfuerzos en barra Excel .....	58
Ilustración 40 Resultados interacción de esfuerzos en barra aplicación, coeficientes K .....	58
Ilustración 41 Resultados interacción de esfuerzos en barra aplicación .....	59
Ilustración 42 Resultados no válidos interacción de esfuerzos en barra Excel ....	59
Ilustración 43 Resultados no válidos interacción de esfuerzos en barra aplicación, coeficientes K .....	59
Ilustración 44 Resultados interacción de esfuerzos en barra aplicación .....	60
Ilustración 45 Ejemplo pruebas unitarias .....	61
Ilustración 46 Repositorio github .....	63
Ilustración 47 Manual de ejecución frontend .....	64
Ilustración 48 Manual de ejecución run and debug .....	65
Ilustración 49 Manual de ejecución Flask.....	66
Ilustración 50 Manual de ejecución servidor lanzado .....	66

# ÍNDICE DE TABLAS

Tabla 1 Comparación con otros productos .....	20
Tabla 2 Requisito funcional: Cálculo de resistencias.....	21
Tabla 3 Requisito funcional: Cálculo de pandeo lateral .....	21
Tabla 4 Requisito funcional: Cálculo de pandeo de barras a compresión .....	21
Tabla 5 Requisito funcional: Cálculo de interacción de esfuerzos en barra .....	22
Tabla 6 Requisito funcional: Introducción de datos .....	22
Tabla 7 Requisito funcional: Comprobación de datos.....	22
Tabla 8 Requisito funcional: Realización de formularios acordes a los csv .....	22
Tabla 9 Requisito funcional: Desarrollo por pasos del cálculo de resistencias ...	22
Tabla 10 Requisito funcional: Desarrollo por pasos del cálculo del pandeo lateral .....	23
Tabla 11 Requisito funcional: Desarrollo por pasos del cálculo del pandeo de barras a compresión .....	23
Tabla 12 Requisito funcional: Desarrollo por pasos del cálculo de la interacción de esfuerzos en barra.....	23
Tabla 13 Requisito no funcional: Facilidad de uso.....	23
Tabla 14 Requisito no funcional: Fiabilidad.....	24
Tabla 15 Caso de uso 01 .....	24
Tabla 16 Caso de uso 02 .....	25
Tabla 17 Caso de uso 03 .....	25
Tabla 18 Caso de uso 04 .....	26
Tabla 19 Caso de uso 05 .....	26
Tabla 20 Caso de uso 06 .....	27
Tabla 21 Caso de uso 07 .....	27
Tabla 22 Caso de uso 08 .....	28
Tabla 23 Caso de uso 09 .....	28

# 1 INTRODUCCIÓN

## 1.1 JUSTIFICACIÓN DEL PROYECTO

Este proyecto se inició para suplir una necesidad de mi Co-Tutor Manuel Ignacio Guerra Romero, profesor de la universidad de León, del departamento de ingeniería y ciencias agrarias y perteneciente al grupo de investigación ingeniería de materiales y eco-eficiencia. Actualmente, él tiene un Excel que le realiza los cálculos que hará esta aplicación web, pero necesitaba algo que se viese más amigable, presentase los resultados de manera más cómoda y fuese sencilla de usar. Además, se pensó en añadir una reproducción por pasos para que el usuario pudiese comprender mejor de dónde se obtiene cada resultado y cuál sería la forma de calcularlo.

## 1.2 OBJETIVOS DEL PROYECTO

El objetivo principal del proyecto es desarrollar una aplicación web que realice los cálculos necesarios para devolver por pantalla los resultados de las resistencias a los esfuerzos que el usuario introduce.

El objetivo principal de esta aplicación será la fiabilidad, ya que se necesita que los resultados sean correctos y no haya ningún fallo de cálculo.

Además de esto, cada usuario deberá de ser capaz de introducir los datos que desee de forma que tenga un uso personalizado de la aplicación. Por otro lado, la aplicación debe ser cómoda y sencilla de usar, para que, de esta forma, sea sencilla de comprender y no surjan problemas con su uso.

Por último, los resultados se deben mostrar por pantalla, de forma que el usuario no tenga problemas para realizar un estudio de los mismos y verificarlos.

### 1.3 ESTUDIO DE LA SITUACIÓN ACTUAL

Como se ha mencionado previamente, Manuel Ignacio tiene un Excel que le realiza estos cálculos. Pero, este Excel es complicado de utilizar y entender, razón por la cual se desarrollará esta aplicación web.

Este Excel está compuesto por diferentes hojas, donde realiza diferentes cálculos, nuestra aplicación solo recogerá los datos del perfil seleccionado para posteriormente realizar los cálculos en Python. De esta forma será más sencillo para el usuario utilizarla y obtener los resultados deseados que en el Excel actual.

### 1.4 EVALUACIÓN ALTERNATIVAS

La aplicación web se desarrolló en local. Aun así, hemos encontrado diferentes alternativas a este proyecto. Un ejemplo podría ser el desarrollo de una aplicación de escritorio en lugar de una aplicación web, de forma que el usuario simplemente necesite ejecutar esa aplicación para obtener los resultados. Otra alternativa podría ser la de migrar nuestra aplicación web de localhost a un servidor, de forma que tenga un más fácil acceso. Esto podrían ser ampliaciones de este mismo proyecto.

### 1.5 METODOLOGÍA

Para el desarrollo del proyecto se decidió dividir el mismo en diferentes fases. Debido a una situación personal, no se podía tener reuniones periódicas, por lo que se decidió simplemente dividir el proyecto en las diferentes fases a realizar.

#### 1.5.1 FASES DEL PROYECTO

En este apartado se comentarán las diferentes fases por las que ha pasado el proyecto. Esta sería la manera en la que se organizó y se podrá apreciar de mejor forma en el apartado 3. Gestión y planificación del proyecto.

### **Definición del proyecto**

Es la primera fase del proyecto, en ella se decide como se va a enfocar el proyecto y las tecnologías que se utilizarán. Además, se revisan las fórmulas que serán utilizadas y las condiciones que habrá que implementar posteriormente. Por último, se definen los objetivos principales del proyecto.

### **Familiarización con el entorno**

En esta fase, se realizaron cursos sobre las tecnologías a utilizar, ya que la mayoría de ellas, era la primera vez que se trabajaba en profundidad con ellas. Se realizaron cursos de Python, Vue.js y Flask para aprender sobre su funcionamiento y aplicación.

### **Desarrollo primera parte del backend**

Esta es la primera fase del desarrollo. Una vez conocemos más sobre Python y las librerías a utilizar comienza el desarrollo. En esta fase se realizaría la primera funcionalidad, que sería la del cálculo de resistencias a esfuerzos.

### **Desarrollo primera vista del frontend**

Tras la fase anterior, desarrollamos la primera vista, esta vista será una vista muy simple para verificar que todo funciona como queremos. Se deberán meter datos por pantalla y que el programa los reciba de manera satisfactoria. Además, se realizará la integración con el backend utilizando Flask.

### **Desarrollo completo del backend**

Una vez tenemos la primera parte de nuestra aplicación completamente funcional, utilizaremos Python para realizar los demás cálculos necesarios. En esta fase se realizará lo que sería el backend de las demás vistas. Además, se realizarán las correcciones necesarias a la funcionalidad ya existente.

### Desarrollo completo del frontend

Cuando ya tenemos toda la parte del backend realizado, realizaremos la vista correspondiente a cada funcionalidad, además de integrarlas utilizando Flask. Todas las vistas se realizan con Vue.js y tendrán un estilo similar.

### Desarrollo de desplegados en función de csv

En nuestros formularios habrá desplegados que tienen que ser ampliables para el usuario, es decir, que si necesita introducir un nuevo tipo de acero pueda hacerlo de forma sencilla. Para ello, estos desplegados se harán a partir de un archivo csv, en el que tendremos la información, y se cargarán directamente al ejecutar nuestra aplicación.

### Implementación de la ejecución por pasos

Ahora que ya tenemos todas las funcionalidades de nuestra aplicación web implementaremos una ejecución por pasos. En ella, el usuario podrá seguir los cálculos realizados paso a paso, además de tener la opción de volver al paso anterior si así lo necesita. En cada paso se mostrará la fórmula utilizada además de la explicación de dónde se obtiene cada dato.

### Verificación de los resultados

Cuando ya ha acabado la fase de desarrollo verificamos los resultados con la ayuda de Manuel Ignacio y su Excel, por si hubiese que añadir alguna pequeña corrección en las fórmulas.

### Estilo

Esta sería la última fase. En ella daríamos estilo a nuestra aplicación web, para que se vea más amigable y sea más sencilla de entender.

## 1.6 ESTRUCTURA DE LA MEMORIA

En este apartado se hará una breve explicación de los contenidos que contendrá cada uno de los apartados de esta memoria.

- **Introducción:** En la introducción se encuentra la justificación del proyecto, además de resumir las fases del mismo.
- **Aspectos Teóricos:** Este apartado contendrá definiciones consideradas importantes de cara a la comprensión del proyecto.
- **Gestión y Planificación del Proyecto:** Se definirá el alcance del proyecto, además de presentarse las fases del mismo.
- **Análisis:** Aquí se encuentra la definición del sistema, sus requisitos y los casos de uso. Además, se incluirá un análisis de la interfaz.
- **Diseño del Sistema:** En este apartado, se detalla la estructura de la aplicación, incluyendo los diagramas de interacción además de las distintas fases por las que pasó la interfaz.
- **Implementación del Sistema:** Este apartado incluirá una explicación más detallada de las librerías y herramientas utilizadas en para crear la aplicación. Además, se describe de forma detallada el funcionamiento principal de cada una de las clases.
- **Resultados:** Se realizará un análisis de los resultados, comparándolos con los del Excel inicial. Además, se mostrarán ejemplos de resultados obtenidos en ejecuciones de la aplicación.
- **Desarrollo de Pruebas:** Aquí se encuentran las pruebas realizadas a la aplicación para comprobar su funcionamiento.
- **Manuales del Sistema:** Incluye los manuales de instalación, ejecución, usuario y del programador.
- **Conclusiones y Ampliaciones:** Detalla las conclusiones del proyecto, además de las posibles mejoras planteadas a la aplicación.

## 2 ASPECTOS TEORICOS

### 2.1 APLICACIÓN WEB

Una aplicación web es un sitio web que contiene páginas con contenido sin determinar, parcialmente o en su totalidad. Dicho contenido se determinará según la interacción o solicitud del usuario. [1]

### 2.2 BACKEND

Backend es la capa de acceso a datos de un software o cualquier dispositivo que no es accesible directamente por los usuarios. Además, contiene la lógica que servirá para realizar las funcionalidades necesarias de la aplicación. [2]

### 2.3 FRONTEND

El frontend en cambio, es la parte a la que accede directamente el usuario, es decir, lo que se ve directamente en una aplicación web. Consiste de todas las tecnologías y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios. [2]

### 2.4 API REST

Una API, es un conjunto de reglas que definen la forma en la que se comunican y se conectan las aplicaciones o dispositivos. Una API Rest, es una API que cumple los principios del diseño del estilo de arquitectura REST. [3]

### 2.5 MÉTODOS DE PETICIÓN HTTP

Estos métodos son solicitudes que se realizan para que se lleve a cabo una acción. En el caso de esta aplicación, estos métodos se utilizarán cuando el frontend

necesita información que le proporcionará el backend. Consisten en una serie de mensajes que el cliente web realiza al servidor. [4]

# 3 GESTIÓN Y PLANIFICACIÓN DEL PROYECTO

## 3.1 ALCANCE DEL PROYECTO

El alcance final del proyecto es el desarrollo de una aplicación web capaz de realizar cálculos sobre resistencia para diferentes tipos de acero. Además, deberá adjuntarse una memoria que recoja el proceso de desarrollo de este proyecto para ser entregada para la asignatura de Trabajo de Fin de Grado.

Por lo tanto, era necesario desarrollar una aplicación web sencilla de utilizar y fiable en cuanto a los resultados ofrecidos. Además, aparte de ofrecer los resultados por pantalla, ofrecerá una resolución por pasos del problema que este resolviendo.

## 3.2 PLAN DE TRABAJO

Como ya se comentó previamente, debido a una situación personal no se pudo establecer plazos concretos o reuniones periódicas a lo largo del desarrollo, por lo que se decidió dividir el proyecto en fases y estimar las fechas aproximadamente.

El proyecto inició en el mes de marzo de 2022 con una primera reunión. En esta reunión se realizó una breve explicación sobre el funcionamiento del Excel original, que ya se ha mencionado previamente, y los objetivos que debería cumplir nuestra aplicación web.

Dada la situación personal, las fases del proyecto se fueron cumpliendo según la disponibilidad del alumno. Con la finalización de cada una de las fases, se realizaba un análisis de los resultados y una valoración de los problemas o mejoras que existían.

### 3.3 GESTIÓN DE RECURSOS

Ya que todo el proyecto fue realizado por la misma persona, para el desarrollo del mismo solo fue necesario el uso de un ordenador personal además de una conexión a internet tanto para el desarrollo como para la búsqueda de información necesaria.

### 3.4 PLANIFICACIÓN

A continuación, se muestra el diagrama de Gantt de las fechas estimadas para cada fase del proyecto.

#### Diagrama de Gantt

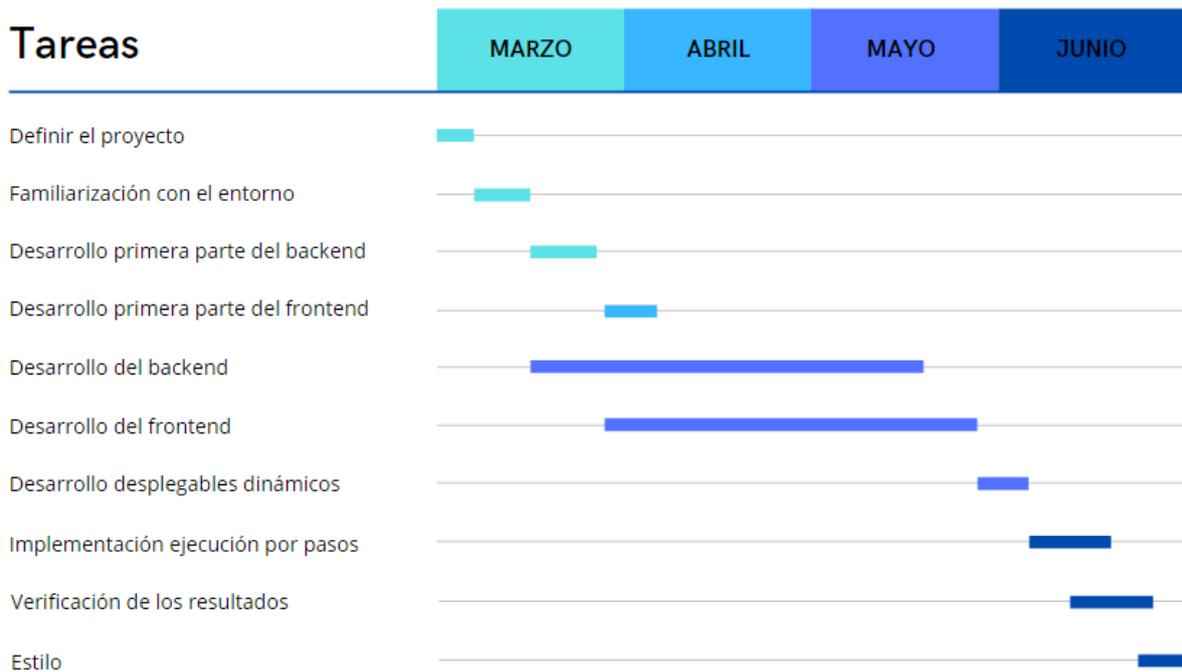


Ilustración 1 Diagrama de Gantt

## 4 SITUACIÓN ACTUAL

En este apartado, se comentarán aplicaciones similares existentes y los precios que tienen sus licencias en el mercado. Además de comentar las principales diferencias con esta aplicación.

La aplicación más similar la encontramos en el software de Dlubal, que se encarga de realizar software de análisis y dimensionamiento de estructuras. Este software analizaría las tensiones y los esfuerzos, al igual que esta aplicación, también mostraría los resultados en forma de tabla, además de documentar todos los resultados. [5]

Por otro lado, los precios de este producto estarían entre 550 y 920 euros por licencia. Por esto, sería una opción muy poco realista para utilizar en este caso. [5]

Luego, se han encontrado más programas y aplicaciones que realizan cálculos similares, pero más completos, ya que están destinados a edificación y obra civil. Estos programas serían Cypecad o Autodesk Robot, estas aplicaciones también serían de pago. En el caso de Cypecad, su precio iría desde los 1000 a los 9000 euros, mientras que Autodesk Robot rondaría los 350 euros al mes. [6]

Como se puede ver, esta aplicación ofrecería soluciones al problema inicial y ahorrando una cantidad grande de dinero. Además, muchas de estas aplicaciones realizan cálculos más generales, ya que están pensadas para diseños más grandes.

	DLUBAL	CYPECAD	AUTODESK ROBOT	ESTA APLICACIÓN
Tipo de licencia	Monopuesto, vitalicia	Monopuesto, Vitalicia	Mensual	Vitalicia
Producto que ofrece	Cálculos de resistencias en barras y superficies.	Diseño y cálculo de estructuras metálicas	Cálculo de uniones y estructuras de acero	Cálculo de resistencias en estructuras de acero

Precio de licencia	920 euros	Desde 1100 a 9998 euros	370\$ al mes, 2965\$ al año y 8005\$ 3 años.	Ahora mismo sería de código abierto.
--------------------	-----------	-------------------------	--	--------------------------------------

*Tabla 1 Comparación con otros productos*

Como se puede ver en la tabla anterior, los precios de las demás opciones son excesivamente caros. Esto significa que este producto tendría cabida para un sector más específico, ya que la función que cumple es más específica que las demás aplicaciones. Además, al proporcionar la explicación del desarrollo por pasos, sería una buena herramienta educativa o apoyo para los alumnos.

# 5 ANÁLISIS

## 5.1 DEFINICIÓN DEL SISTEMA

### 5.1.1 DETERMINACIÓN DEL ALCANCE DEL SISTEMA

El sistema deberá cumplir con los objetivos definidos anteriormente, para ello deberá mostrar en la pantalla todos los resultados de cada cálculo y que estos sean fiables. Además, en cuanto a la aplicación, debe ser sencilla de utilizar para el usuario y amigable.

## 5.2 REQUISITOS DEL SISTEMA

### 5.2.1 REQUISITOS FUNCIONALES

Requisito funcional: Cálculo de resistencias.	
Descripción	La aplicación realizará el cálculo de resistencias.
Prioridad	Alta
Estado	Desarrollado

*Tabla 2 Requisito funcional: Cálculo de resistencias*

Requisito funcional: Cálculo de pandeo lateral.	
Descripción	La aplicación realizará el cálculo del pandeo lateral.
Prioridad	Alta
Estado	Desarrollado

*Tabla 3 Requisito funcional: Cálculo de pandeo lateral*

Requisito funcional: Cálculo de pandeo de barras a compresión.	
Descripción	La aplicación realizará el cálculo del pandeo de barras a compresión.
Prioridad	Alta
Estado	Desarrollado

*Tabla 4 Requisito funcional: Cálculo de pandeo de barras a compresión*

Requisito funcional: Cálculo de interacción de esfuerzos en barra.	
Descripción	La aplicación realizará el cálculo de la interacción de esfuerzos en barra.
Prioridad	Alta
Estado	Desarrollado

*Tabla 5 Requisito funcional: Cálculo de interacción de esfuerzos en barra*

Requisito funcional: Introducción de datos.	
Descripción	El usuario podrá introducir los datos deseados en cada una de las vistas.
Prioridad	Alta
Estado	Desarrollado

*Tabla 6 Requisito funcional: Introducción de datos*

Requisito funcional: Comprobación de datos.	
Descripción	La aplicación comprobará si los datos introducidos son válidos.
Prioridad	Alta
Estado	Desarrollado

*Tabla 7 Requisito funcional: Comprobación de datos*

Requisito funcional: Realización de formularios acordes a los csv.	
Descripción	La aplicación realizará los desplegables de los formularios acorde al contenido de los csv.
Prioridad	Alta
Estado	Desarrollado

*Tabla 8 Requisito funcional: Realización de formularios acordes a los csv*

Requisito funcional: Desarrollo por pasos del cálculo de resistencias.	
Descripción	La aplicación ofrecerá un desarrollo por pasos del cálculo de resistencias.
Prioridad	Media
Estado	Desarrollado

*Tabla 9 Requisito funcional: Desarrollo por pasos del cálculo de resistencias*

Requisito funcional: Desarrollo por pasos del cálculo del pandeo lateral.	
Descripción	La aplicación ofrecerá un desarrollo por pasos del cálculo del pandeo lateral.
Prioridad	Media
Estado	Desarrollado

*Tabla 10 Requisito funcional: Desarrollo por pasos del cálculo del pandeo lateral*

Requisito funcional: Desarrollo por pasos del cálculo del pandeo de barras a compresión.	
Descripción	La aplicación ofrecerá un desarrollo por pasos del cálculo del pandeo de barras a compresión.
Prioridad	Media
Estado	Desarrollado

*Tabla 11 Requisito funcional: Desarrollo por pasos del cálculo del pandeo de barras a compresión*

Requisito funcional: Desarrollo por pasos del cálculo de la interacción de esfuerzos en barra.	
Descripción	La aplicación ofrecerá un desarrollo por pasos del cálculo de la interacción de esfuerzos en barra.
Prioridad	Media
Estado	Desarrollado

*Tabla 12 Requisito funcional: Desarrollo por pasos del cálculo de la interacción de esfuerzos en barra*

### 5.2.2 REQUISITOS NO FUNCIONALES

Requisito no funcional: Facilidad de uso.	
Descripción	La aplicación será fácil de utilizar para el usuario.
Prioridad	Alta
Estado	Desarrollado

*Tabla 13 Requisito no funcional: Facilidad de uso*

Requisito no funcional: Fiabilidad.	
Descripción	Los resultados ofrecidos por la aplicación serán fiables.
Prioridad	Alta
Estado	Desarrollado

Tabla 14 Requisito no funcional: Fiabilidad

### 5.2.3 IDENTIFICACIÓN DE ACTORES DEL SISTEMA

En este caso, solo existiría un actor del sistema para las ejecuciones, el cual sería el usuario de nuestra aplicación. Para nuestro usuario se definen los siguientes casos de uso.

### 5.2.4 CASOS DE USO

ID	CU_01
Nombre	Ejecución cálculo resistencias.
Descripción	El usuario ejecutará la parte de cálculo de resistencias.
Precondiciones	<ol style="list-style-type: none"> <li>1. El usuario debe introducir los datos necesarios.</li> <li>2. El usuario debe pulsar el botón iniciar</li> </ol>
Escenario	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. En la parte de cálculo de resistencias introduce los datos.</li> <li>3. Pulsa el botón "iniciar"</li> </ol>
Postcondiciones	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrará el botón de siguiente para acceder al desarrollo por pasos</li> </ol>

Tabla 15 Caso de uso 01

ID	CU_02
Nombre	Ejecución cálculo pandeo lateral.
Descripción	El usuario ejecutará la parte de pandeo lateral.
Precondiciones	<ol style="list-style-type: none"> <li>1. El usuario debe cambiar a la vista Pandeo Lateral.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> </ol>

<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. En la barra de navegación accede a Pandeo Lateral.</li> <li>3. El usuario introduce los datos</li> <li>4. El usuario pulsa el botón iniciar</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrará el botón de siguiente para acceder al desarrollo por pasos</li> </ol>

Tabla 16 Caso de uso 02

<b>ID</b>	<b>CU_03</b>
<b>Nombre</b>	Ejecución cálculo pandeo de barras a compresión.
<b>Descripción</b>	El usuario ejecutará la parte de pandeo de barras a compresión.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe cambiar a la vista Pandeo de barras a compresión.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. En la barra de navegación accede a Pandeo de barras a compresión.</li> <li>3. El usuario introduce los datos</li> <li>4. El usuario pulsa el botón iniciar</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrará el botón de siguiente para acceder al desarrollo por pasos</li> </ol>

Tabla 17 Caso de uso 03

<b>ID</b>	<b>CU_04</b>
<b>Nombre</b>	Ejecución cálculo interacción de esfuerzos en barra.
<b>Descripción</b>	El usuario ejecutará la parte de interacción de esfuerzos en barra.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe cambiar a la vista Interacción de esfuerzos en barra.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación</li> <li>2. En la barra de navegación accede a Interacción de esfuerzos en barra.</li> </ol>

	<ol style="list-style-type: none"> <li>3. El usuario introduce los datos</li> <li>4. El usuario pulsa el botón iniciar</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrará el botón de siguiente para acceder al desarrollo por pasos</li> </ol>

Tabla 18 Caso de uso 04

<b>ID</b>	<b>CU_05</b>
<b>Nombre</b>	Ejecución por pasos cálculo de resistencias
<b>Descripción</b>	El usuario ejecutará el desarrollo paso a paso del cálculo de resistencias
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar en la vista cálculo resistencias.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> <li>4. El usuario debe pulsar el botón siguiente.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. Se comenzará a partir de CU_01.</li> <li>2. Se pulsará el botón siguiente</li> <li>3. Se pulsará el botón siguiente o anterior según el usuario lo desee.</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrarán el botón de siguiente y anterior para moverte por los diferentes pasos del cálculo.</li> </ol>

Tabla 19 Caso de uso 05

<b>ID</b>	<b>CU_06</b>
<b>Nombre</b>	Ejecución por pasos cálculo de pandeo lateral.
<b>Descripción</b>	El usuario ejecutará el desarrollo paso a paso del cálculo del pandeo lateral.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar en la vista pandeo lateral.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> <li>4. El usuario debe pulsar el botón siguiente.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. Se comenzará a partir de CU_02.</li> <li>2. Se pulsará el botón siguiente</li> <li>3. Se pulsará el botón siguiente o anterior según el usuario lo desee.</li> </ol>

<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrarán el botón de siguiente y anterior para moverte por los diferentes pasos del cálculo.</li> </ol>
------------------------	---

Tabla 20 Caso de uso 06

<b>ID</b>	<b>CU_07</b>
<b>Nombre</b>	Ejecución por pasos cálculo de pandeo de barras a compresión.
<b>Descripción</b>	El usuario ejecutará el desarrollo paso a paso del cálculo de pandeo de barras a compresión
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar en la vista pandeo de barras a compresión.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> <li>4. El usuario debe pulsar el botón siguiente.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. Se comenzará a partir de CU_03.</li> <li>2. Se pulsará el botón siguiente</li> <li>3. Se pulsará el botón siguiente o anterior según el usuario lo desee.</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrarán el botón de siguiente y anterior para moverte por los diferentes pasos del cálculo.</li> </ol>

Tabla 21 Caso de uso 07

<b>ID</b>	<b>CU_08</b>
<b>Nombre</b>	Ejecución por pasos cálculo de interacción de esfuerzos en barra.
<b>Descripción</b>	El usuario ejecutará el desarrollo paso a paso de la interacción de esfuerzos en barra.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar en la vista interacción de esfuerzos en barra.</li> <li>2. El usuario debe introducir los datos necesarios.</li> <li>3. El usuario debe pulsar el botón iniciar.</li> <li>4. El usuario debe pulsar el botón siguiente.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. Se comenzará a partir de CU_04.</li> <li>2. Se pulsará el botón siguiente</li> <li>3. Se pulsará el botón siguiente o anterior según el usuario lo desee.</li> </ol>

<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Se mostrarán los resultados por pantalla.</li> <li>2. Se mostrarán el botón de siguiente y anterior para moverte por los diferentes pasos del cálculo.</li> </ol>
------------------------	---

Tabla 22 Caso de uso 08

<b>ID</b>	<b>CU_09</b>
<b>Nombre</b>	Dato introducido erróneo.
<b>Descripción</b>	El usuario meterá un dato erróneo.
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. El usuario debe estar en cualquier vista.</li> <li>2. El usuario debe introducir un dato no válido.</li> </ol>
<b>Escenario</b>	<ol style="list-style-type: none"> <li>1. Se comenzará en cualquier vista.</li> <li>2. El usuario introducirá un dato no válido.</li> <li>3. Se pulsará el botón introducir datos.</li> </ol>
<b>Postcondiciones</b>	<ol style="list-style-type: none"> <li>1. Saltará una alerta.</li> <li>2. Se mostrará un mensaje que indique que la entrada no es válida.</li> </ol>

Tabla 23 Caso de uso 09

## 5.3 DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS

### 5.3.1 DIAGRAMA DE CLASES

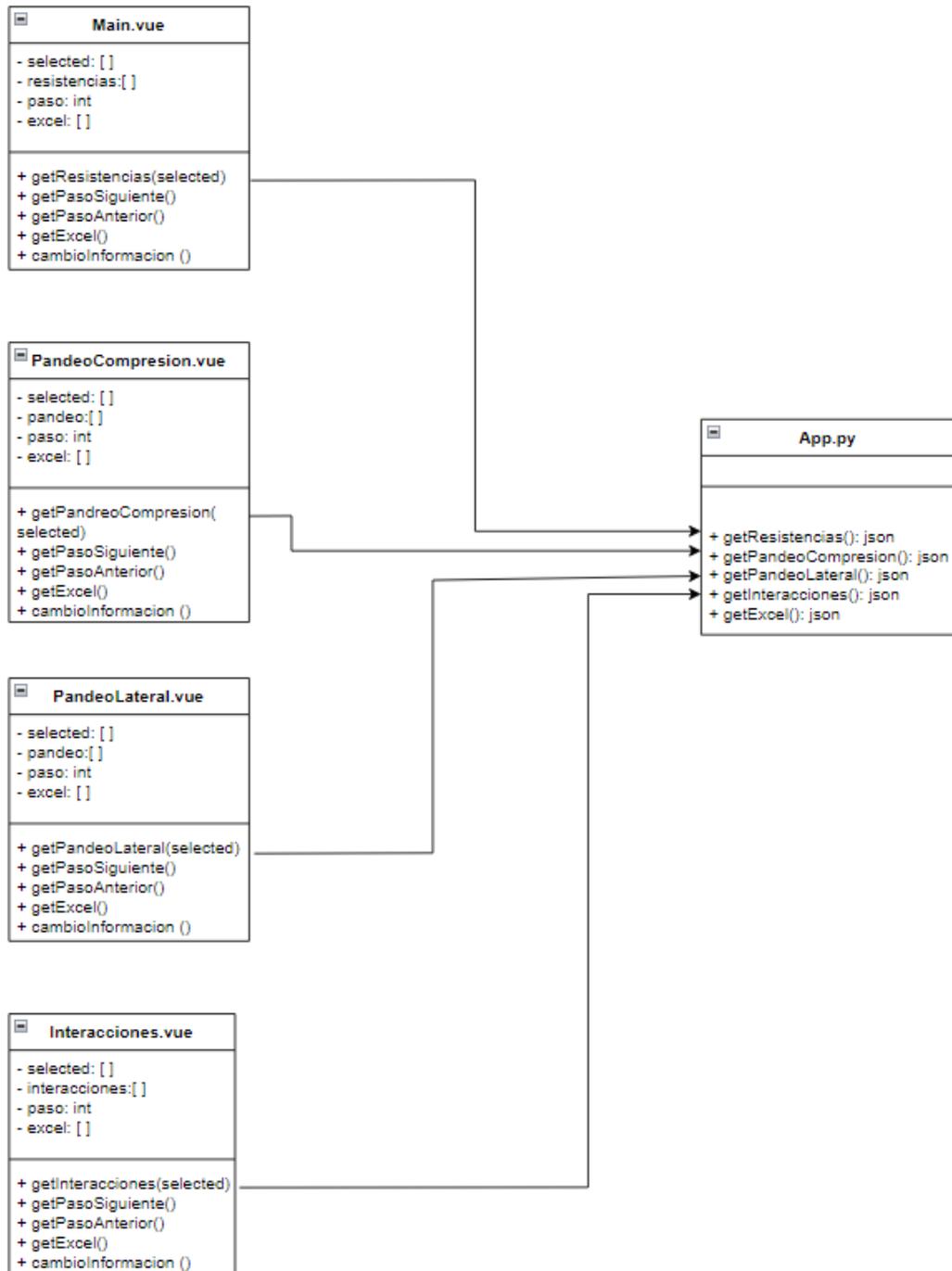


Ilustración 2 Diagrama de clases

En este diagrama, se pueden ver las clases utilizadas y sus principales métodos, que serán comentados más adelante.

### 5.3.2 DESCRIPCIÓN DE LAS CLASES

En la aplicación, podemos encontrar cuatro clases principales en el frontend, una por cada vista. Además, tenemos la única clase del backend, que será la que ejecute sus métodos al recibir la llamada de cada uno de los componentes respectivamente.

Las clases del frontend están formadas de la misma manera. Todas tienen sus métodos para cambiar la información, el paso siguiente y anterior y el de cargar el Excel, que sería una petición GET al backend que devolvería en formato json los datos recogidos de un csv para así realizar los despletables. Por último, tendrían cada uno su método para realizar los cálculos en el backend, este método, sería una petición POST al backend, con la que le mandarían los datos necesarios para el cálculo. Y el backend devolvería los resultados en formato json nuevamente.

En cuanto a nuestra clase del backend, sus métodos principales serían los de realizar los cálculos, que ejecutaría al recibir las peticiones POST. Además, tendría el método getExcel, que ejecutaría cuando recibe una petición GET. Todos devolverán el resultado en formato json.

## 5.4 ANÁLISIS DE CASOS DE USO Y ESCENARIOS

### 5.4.1 CASO DE USO 1

En el primer caso de uso (CU-01) se describe la ejecución del apartado de cálculo de resistencias. En esta ejecución, el usuario deberá estar en la vista de cálculo de resistencias, que es la vista predeterminada. Una vez ahí, el usuario deberá introducir de manera correcta todos los datos pedidos por el usuario. Una vez introducidos todos los datos debe pulsar el botón 'iniciar, el programa ofrecerá los resultados en forma de tabla por pantalla. Además, aparecerá el botón 'siguiente', para que así el usuario tenga acceso al desarrollo por pasos.

#### 5.4.2 CASO DE USO 2

En este caso de uso (CU-02), el usuario debe primero acceder a la vista del pandeo lateral utilizando la barra de navegación superior. Una vez ahí, el usuario deberá introducir los datos correspondientes a este cálculo. Cuando los datos han sido introducidos pulsando el botón 'iniciar', el programa ofrecerá los resultados en forma de tabla por pantalla. Además, aparecerá el botón 'siguiente' de forma que el usuario tenga acceso al desarrollo por pasos.

#### 5.4.3 CASO DE USO 3

Para la realización de este caso de uso (CU-03), el usuario debe acceder, utilizando la barra de navegación, a la vista del pandeo de barras a compresión. Cuando el usuario ya está en esa vista, tiene que introducir los datos necesarios previos al cálculo. Ahora que los datos ya han sido introducidos pulsando el botón 'iniciar', los resultados se mostrarán por pantalla en forma de tabla y aparecerá el botón 'siguiente' para que el usuario tenga acceso al desarrollo por pasos.

#### 5.4.4 CASO DE USO 4

En cuanto a este caso de uso (CU-04), el usuario tiene que acceder primero a la vista de interacción de esfuerzos en barra. Para ello, debe utilizar la barra de navegación. Una vez estemos en la vista respectiva, habrá que introducir los datos pedidos por la aplicación. Con todos los datos introducidos, el usuario podrá utilizar el botón 'iniciar' para que así se muestren todos los resultados por pantalla. Además, aparecerá el botón 'siguiente' con el que el usuario podrá acceder al desarrollo paso a paso.

#### 5.4.5 CASO DE USO 5

Este caso de uso (CU-05) parte de la realización del caso de uso 1 (CU-01). Por lo tanto, los resultados ya han sido mostrados por pantalla y aparecerá el botón 'siguiente'. Una vez se presione ese botón se entrará en el desarrollo por pasos, donde el usuario podrá ver una explicación de cada paso además de las fórmulas

empleadas. Para navegar entre los diferentes pasos de la ejecución, el usuario tendrá además otro botón 'anterior' para volver al paso previo.

#### 5.4.6 CASO DE USO 6

Para este caso de uso (CU-06) será necesaria la realización del caso de uso 2 (CU-02). Una vez hayamos pulsado el botón 'iniciar', los datos se ofrecerán por pantalla, además aparecerá el botón 'siguiente'. Utilizando el botón 'siguiente', el usuario comenzará el desarrollo paso a paso, donde se le mostrarán las fórmulas y resultados de cada paso intermedio. Además, tendrá otro botón 'anterior' con el que podrá volver al paso previo si así lo desea.

#### 5.4.7 CASO DE USO 7

En este caso de uso (CU-07), será necesaria la realización previa del caso de uso 3 (CU-03). Una vez se haya pulsado el botón iniciar y aparezcan los resultados por pantalla, el usuario también tendrá la opción de pulsar el botón 'siguiente'. Con este botón, el usuario accederá al desarrollo por pasos de este cálculo. Una vez estemos utilizando el desarrollo por pasos, el usuario tendrá la opción de presionar el botón 'anterior' en caso de querer revisar un paso previo, además de poder seguir el desarrollo con el botón 'siguiente'.

#### 5.4.8 CASO DE USO 8

Previo a este caso de uso (CU-08), se realizará previamente el caso de uso 4 (CU-04). Cuando ya hayamos introducido los datos, aparecerá el botón 'siguiente', presionándolo, podremos acceder al desarrollo paso a paso, donde también tendremos un botón de 'anterior' para movernos entre los diferentes pasos del cálculo.

### 5.4.9 CASO DE USO 9

En este caso de uso (CU-09) el usuario intentará introducir en cualquiera de las vistas un valor erróneo. Una vez trate de introducir el valor erróneo, saltará una alerta indicando que ha habido un problema a la hora de introducir los datos.

## 5.5 ANÁLISIS DE INTERFACES DE USUARIO

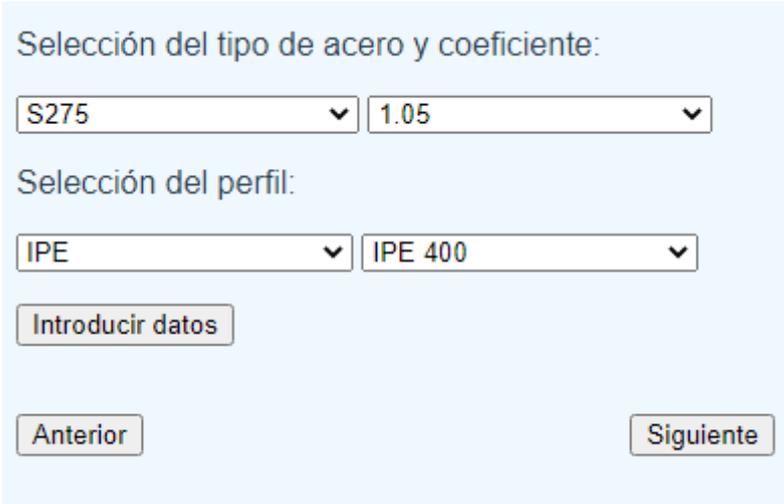
La interfaz de usuario constará de cuatro vistas diferentes. A las que accederemos utilizando nuestra barra de navegación.



*Ilustración 3 Análisis de interfaz: Barra de navegación*

En ella, podemos ver cuatro botones, que cada una de ellas nos redirigirá a cada una de las vistas. Las vistas tienen todas el mismo formato. La parte superior estará dividida en tres partes más pequeñas, formulario, desarrollo por pasos y el apartado de fórmulas.

En el apartado del formulario, además de la información a introducir tendremos los botones necesarios para el desarrollo por pasos.

Un formulario con un fondo azul claro. En la parte superior, el texto 'Selección del tipo de acero y coeficiente:' precede a dos campos de selección con flecha hacia abajo. El primer campo contiene 'S275' y el segundo '1.05'. Debajo, el texto 'Selección del perfil:' precede a dos campos de selección con flecha hacia abajo. El primer campo contiene 'IPE' y el segundo 'IPE 400'. Abajo de estos campos hay un botón rectangular con el texto 'Introducir datos'. En la parte inferior del formulario, hay dos botones rectangulares: 'Anterior' a la izquierda y 'Siguiete' a la derecha.

*Ilustración 4 Análisis de la interfaz: Formulario*

Como podemos ver en la imagen anterior, además del botón para introducir los datos tenemos los respectivos botones al desarrollo por pasos. Estos botones desaparecerán si no fuesen necesarios, por ejemplo, el botón 'anterior' desaparecería si estuviésemos en el primer paso.

#### Desarrollo paso a paso

Paso 1 Calculo de resistencia a esfuerzo axial:

Para la realización de este cálculo, solo necesitaremos utilizar el  $F_y$  obtenido de la clase de acero y su coeficiente. Además, la  $A$  es un valor que obtenemos de la tabla del perfil, que en este caso es: IPE 400

Una vez tenemos el resultado, tendríamos que compararlo con el  $N_{ed}$  introducido previamente, ya que  $N_{ed}$  debe ser menor o igual que dicho resultado.

Si esta condición no se cumple, el resultado estará resaltado en rojo en la tabla.

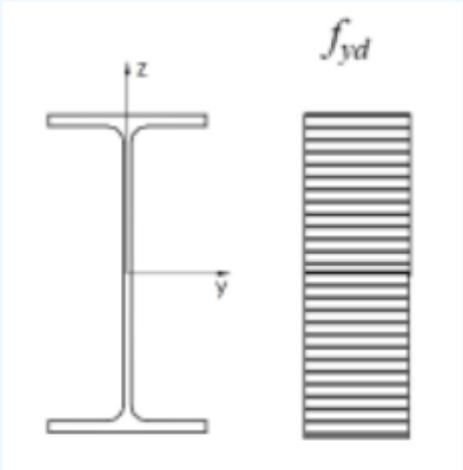
Resistencia a esfuerzo axial = 2213.1 kN.

*Ilustración 5 Análisis de la interfaz: Desarrollo paso a paso*

En la imagen anterior podemos ver lo que sería la parte del desarrollo por pasos. En esta parte nos muestra información sobre qué datos son necesarios para realizar las fórmulas, cual sería necesario utilizar y el resultado. Además de el paso en el que estemos en ese momento.

**Fórmulas:**

$$N_{pl,Rd} = A \frac{f_y}{\gamma_{M0}}$$



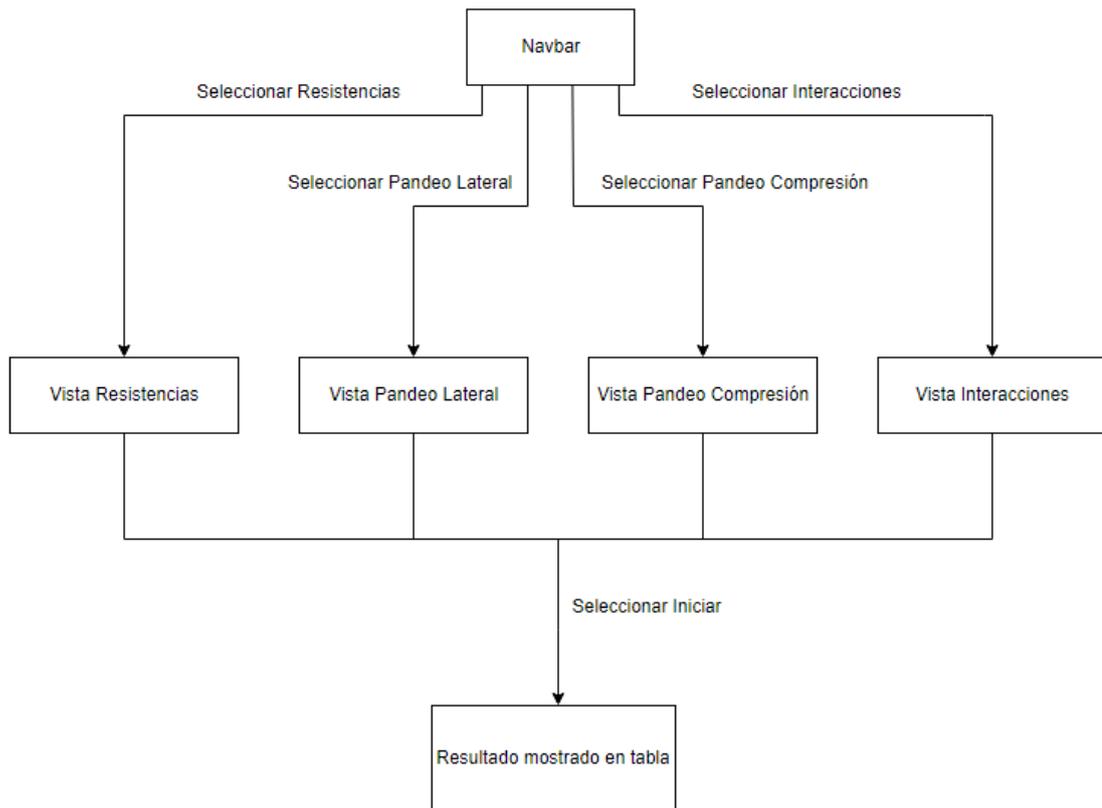
*Ilustración 6 Análisis de la interfaz: Fórmulas*

En este caso, podemos ver cómo sería el apartado para las fórmulas, donde se mostrarían, además de las fórmulas utilizadas, información que se considere interesante.

	N (kN)	My (kN m)	Mz (kN m)	Interaccion CTE	Interaccion EC-3	Vy (kN)	Vz (kN)
Ed	500	78	0	0.45	0.05	0	0
Rd	2213.1	342.3	60			632.2	645.5
Clase	3	1	1	2		Interaccion V:	0.45

*Ilustración 7 Análisis de la interfaz: Tabla de resultados*

Por último, tendríamos en la parte inferior la tabla de resultados. Donde se mostrarán los resultados de cada ejecución, para que así, el usuario, tenga fácil acceso a ellos.



*Ilustración 8 Diagrama de navegación*

En este diagrama se podría comprobar como la aplicación nos permite movernos entre las diferentes vistas de la misma.

# 6 DISEÑO DEL SISTEMA

## 6.1 ARQUITECTURA DEL SISTEMA

Para el desarrollo del proyecto se ha seguido una estructura de directorios que permitiese entender de forma más sencilla donde colocar cada archivo. La estructura sería la siguiente.

- Directorio principal: En este directorio podemos encontrar todas las carpetas. Además, es donde tenemos los archivos csv y el Excel de los que obtenemos toda la información. Por último, también tenemos nuestro archivo App.py que será nuestro backend.
- Frontend: En este directorio, como su propio nombre indica, estarán los archivos que necesita el frontend. En concreto, está el directorio src, donde tendremos los diferentes archivos necesarios para la ejecución. Además, tenemos los directorios de configuración para Vue.js.
- Src: En el directorio src podemos encontrar 3 diferentes carpetas, assets, components y Router. Además, también tendremos los archivos App.vue y el main.js.
- Assets: Este directorio contiene las imágenes a mostrar, se han dividido en carpetas según la vista a la que pertenecen. En el caso de que una imagen pertenezca a varias vistas, no estará en ninguna de las carpetas.
- Components: Carpeta que contiene nuestros componentes de Vue.js, son las diferentes vistas que se mostrarán en nuestra aplicación. Además, también está la barra de navegación.
- Router: En esta carpeta tenemos nuestro archivo index.js. Este archivo es el que contiene las rutas de nuestros componentes para que podamos utilizarlos desde Vue.js.

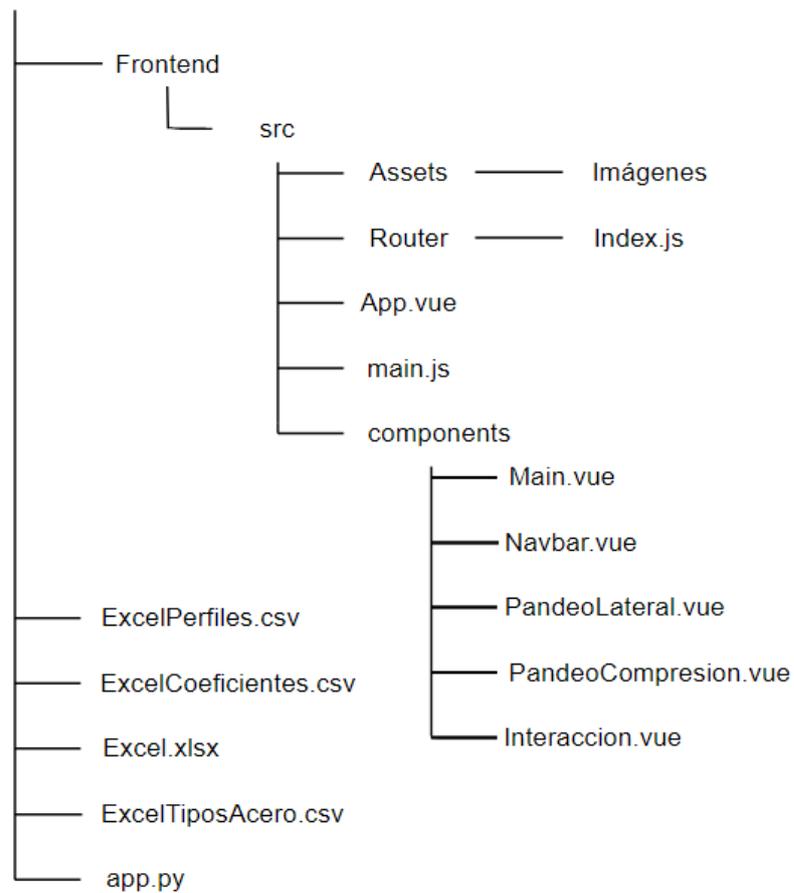


Ilustración 9 Estructura de directorios

## 6.2 DISEÑO DE CLASES

Como ya se ha mencionado previamente, en la carpeta components tendríamos los diferentes componentes de Vue.js. Cada componente tendrá su modelo, que será la vista a mostrar, su script, que contendrá las funciones que ejecutará al pulsar los botones y su style, que se encargará de darle estilo a la página y formato a las imágenes.

Además, tenemos nuestro archivo App.py, que se encargará de ejecutar las funciones del backend que necesite cada vista.

### 6.3 DIAGRAMAS DE INTERACCIÓN

A continuación, se muestran los diagramas de interacción del usuario con la aplicación web. Se incluirán los de cambio de vista, introducir resultados, desarrollo por pasos y el mensaje de error.

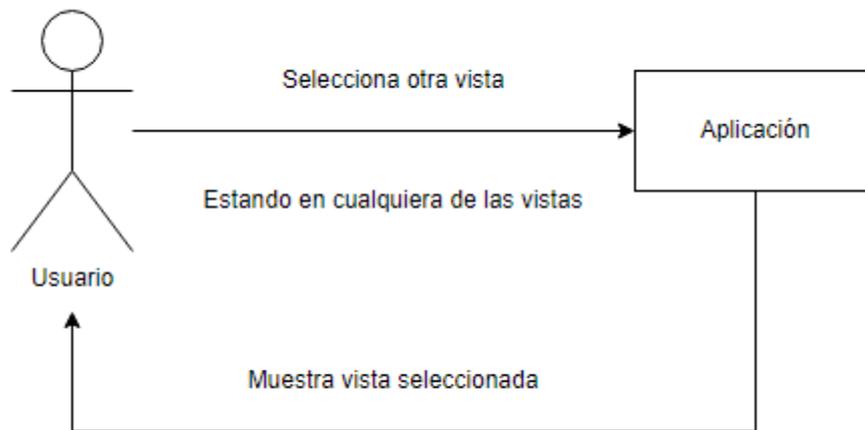


Ilustración 10 Diagrama de interacción: Cambio de vista

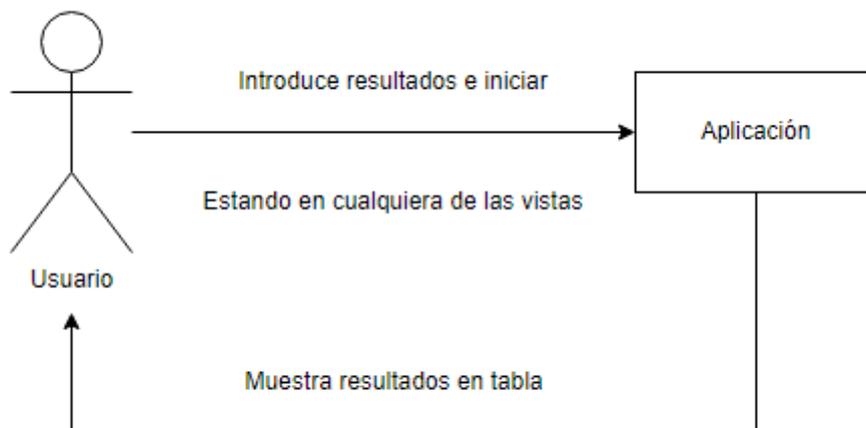


Ilustración 11 Diagrama de interacción: Ejecución normal

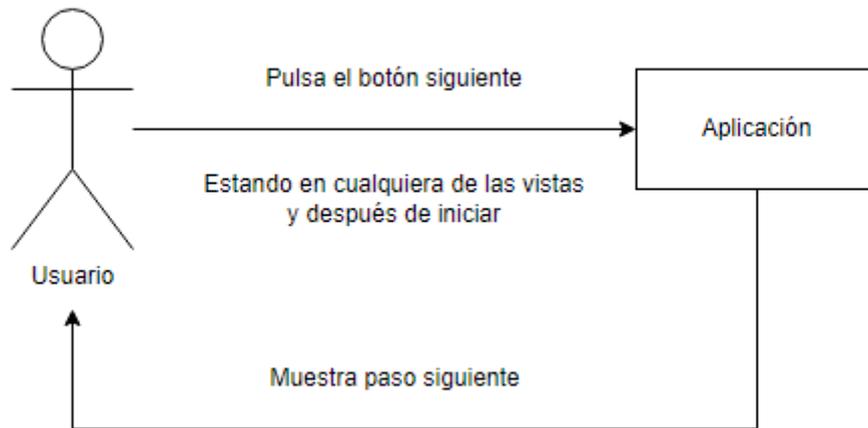


Ilustración 12 Diagrama de interacción: Ejecución por pasos, paso siguiente

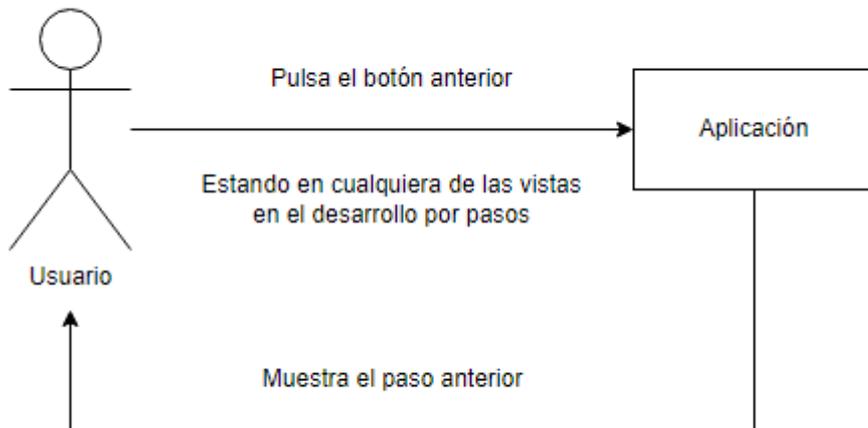


Ilustración 13 Diagrama de interacción: Ejecución por pasos, paso anterior

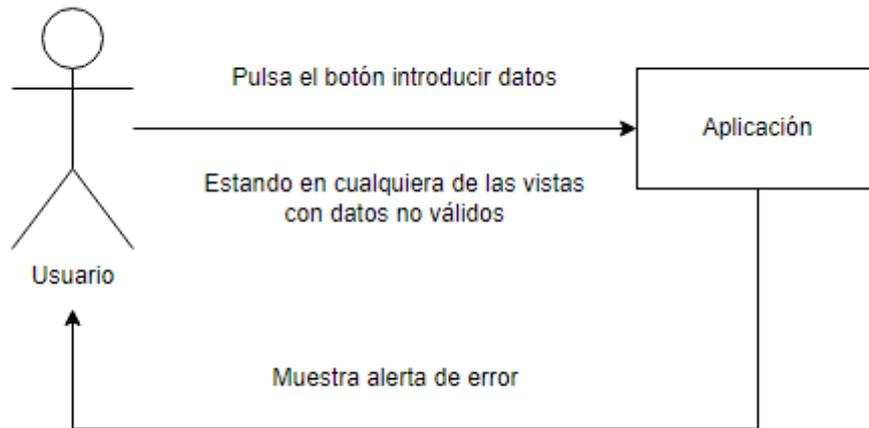


Ilustración 14 Diagrama de interacción: Introducir datos no válidos

## 6.4 DIAGRAMA DE ACTIVIDADES

En el siguiente diagrama se puede apreciar como la aplicación actúa según las acciones del usuario y, de ser necesario, como mostraría el mensaje de error.

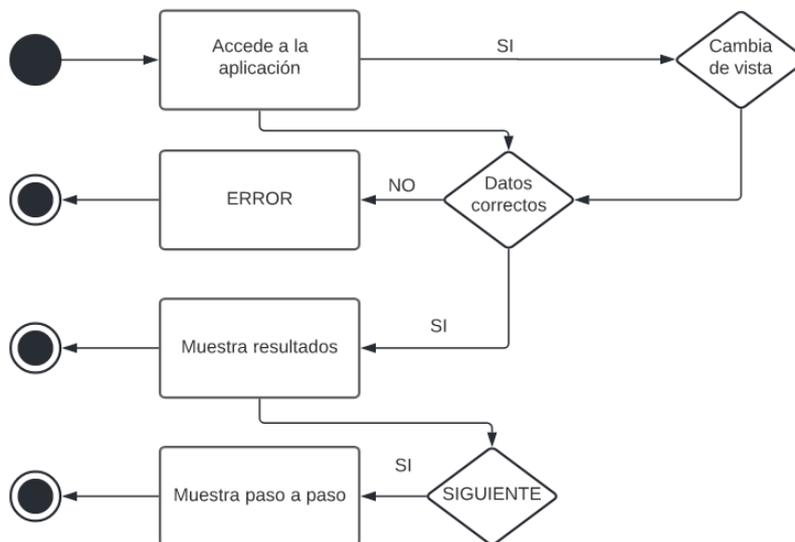
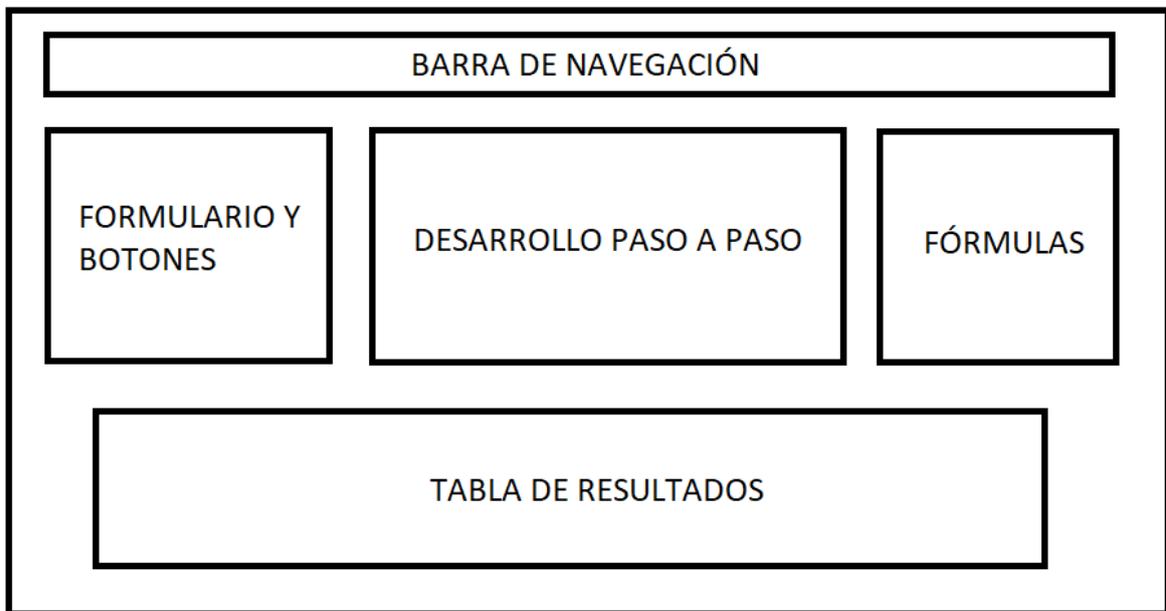


Ilustración 15 Diagrama de actividades

## 6.5 DISEÑO DE LA INTERFAZ

### 6.5.1 PRIMERA FASE INTERFAZ

Para el desarrollo del proyecto, se llevó a cabo el diseño de una interfaz, explicando donde debería ir cada una de las partes de la aplicación. El diseño debería ser similar a lo mostrado en la siguiente ilustración.



*Ilustración 16 Boceto inicial de la interfaz*

En la ilustración 16, se puede observar que en la parte superior tenemos la barra de navegación. Luego en cuanto a la página, la podríamos separar en cuatro partes diferenciadas, el formulario, el desarrollo paso a paso, las fórmulas y la tabla de resultados. En la parte del formulario, vemos el botón de 'introducir datos' que servirá para actualizar la información introducida. Además, nos presenta los botones de 'anterior' y 'siguiente' para movernos por el desarrollo paso a paso.

En la parte central, se podrá ver la información sobre el paso a paso, se muestra información respectiva a los cálculos utilizados y las fórmulas empleadas, que aparecen en la parte derecha. Por último, en la parte inferior se muestra una tabla con los resultados completos de la ejecución.

En la ilustración 17, se muestra un ejemplo de cuando alguno de los datos introducidos no es correcto. Por ejemplo, se introducen letras en lugar de números para alguno de los valores.

Calculo Resistencias Pandeo Lateral Pandeo de barras a compresión Interacción de esfuerzos en barra

**Calculo de resistencias en tipos de acero**  
Desarrollo paso a paso

Introduzca Ned y la clase: 500   
 Introduzca Myed y la clase: esdasd   
 Introduzca Mzed y la clase: 0   
 Seleccione la clase de flexocompresión: 2

El tipo de acero seleccionado es S275.  
 El coeficiente seleccionado es 1.05.  
 El perfil seleccionado es IPE 400.

A continuación, debemos introducir los esfuerzos en las secciones y la clase correspondiente.

La clase viene dada por la tabla según el perfil. Además se debe introducir la clase a flexocompresión correspondiente.

Seguindo la tabla, vemos que las casillas sombreadas significan que un perfil nunca podrá ser de esa clase y el asterisco indica que la sección no cambia para los valores de Ned superiores a esa clase.

**Formulas:**

Acero	Clase de sección			Flexocompresión		
	Compresión	Tensión	Flexión pura	Clase 1	Clase 2	Clase 3
S275	1	1	1	*	*	*
S355	1	1	1	*	*	*
S460	1	1	1	*	*	*
S550	1	1	1	*	*	*
S690	1	1	1	*	*	*
S960	1	1	1	*	*	*
S1100	1	1	1	*	*	*
S1300	1	1	1	*	*	*
S1500	1	1	1	*	*	*
S1700	1	1	1	*	*	*
S1900	1	1	1	*	*	*
S2100	1	1	1	*	*	*
S2350	1	1	1	*	*	*
S2600	1	1	1	*	*	*
S2900	1	1	1	*	*	*
S3200	1	1	1	*	*	*
S3550	1	1	1	*	*	*
S3900	1	1	1	*	*	*
S4300	1	1	1	*	*	*
S4700	1	1	1	*	*	*
S5100	1	1	1	*	*	*
S5600	1	1	1	*	*	*
S6100	1	1	1	*	*	*
S6700	1	1	1	*	*	*
S7300	1	1	1	*	*	*
S7900	1	1	1	*	*	*
S8600	1	1	1	*	*	*
S9300	1	1	1	*	*	*
S10000	1	1	1	*	*	*

Valores introducidos incorrectos.

	N (kN)	My (kN m)	Mz (kN m)	Interaccion CTE	Interaccion EC-3	Vy (kN)	Vz (kN)
Ed	500	esdasd	0			0	0
Rd							
Clase	3	1	1	2		Interaccion V.	

Ilustración 17 Interfaz fase previa: Error

### 6.5.2 FASE FINAL INTERFAZ

En la siguiente ilustración, tendríamos un ejemplo de una de las vistas en su versión final. Se añadieron bordes, para diferenciar de manera más sencilla cada sección y quedase todo más organizado.

Calculo Resistencias   Pandeo Lateral   Pandeo de barras a compresión   Interacción de esfuerzos en barra

**Introducir Datos:**

Selección del tipo de acero y coeficiente:

S275 | 1.05

Selección del perfil:

IPE | IPE 400

**Calculo de resistencias en tipos de acero**  
Desarrollo paso a paso

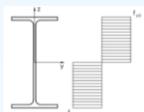
Paso 3 Calculo de resistencia a flexión en el eje z:

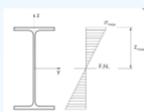
Como en el paso anterior, disponemos de dos formulas para calcular la resistencia. La superior para las clases 1 y 2, y la inferior para la clase 3. En esta ejecución, nuestra clase es 3, por lo que usamos la formula correspondiente. La diferencia entre ambas fórmulas es si usamos el W elastico o el W plastico, que ambos son datos obtenidos del perfil.

De nuevo, comparamos el resultado con nuestro Myed introducido previamente.

Resistencia a flexión en el eje z = 38.3kN m.

**Fórmulas:**

$$M_{z,el,Rd} = W_{z,el} \frac{f_y}{\gamma_{M0}}$$


$$M_{z,pl,Rd} = W_{z,pl} \frac{f_{yk}}{\gamma_{M0}}$$


IPE 400	N (kN)	My (kN m)	Mz (kN m)	Interaccion CTE	Interaccion EC-3	Vy (kN)	Vz (kN)
Ed	2300	78	0	1.3	1.3	15	15
Rd	2213.1	302.8	38.3			632.2	645.5
Clase	3	1	1	3		Interaccion V.	1.3

Ilustración 18 Interfaz fase final

# 7 IMPLEMENTACIÓN DEL SISTEMA

## 7.1 LENGUAJE HERRAMIENTAS Y LIBRERÍAS UTILIZADAS

A lo largo de este proyecto se desarrollará una aplicación web para realizar cálculos de estructuras de acero. Esta aplicación se desarrollará utilizando Vue.js y Python, y se utilizará Flask para integrarlos.

Vue.js es un marco de trabajo de JavaScript destinado a la construcción de interfaces de usuario. Trabaja sobre HTML, CSS y JavaScript, y proporciona un modelo de programación declarativo basado en componentes, que ayuda a desarrollar eficientemente dichas interfaces, ya sean simples o complejas. Se utilizará para realizar las diferentes vistas de la aplicación. [7]



*Ilustración 19 Logo Vue.js*

Flask es un microframework creado para facilitar el desarrollo de aplicaciones web en Python. Es utilizado para construir servicios web como APIs Rest o aplicaciones de contenido estático. En este proyecto se utilizará como API Rest. [8]



*Ilustración 20 Logo Flask*

Python es un lenguaje de programación de alto nivel, flexible y diseñado para ser fácil de interpretar. Se utiliza para desarrollar todo tipo de aplicaciones.[9] En este caso se utiliza para la realización de los cálculos necesarios.



*Ilustración 21 Logo Python*

En cuanto a las librerías utilizadas, comenzaremos con python.

- Para obtener los datos del Excel y los archivos csv se utilizó la librería **pandas** [10]. Además, con la librería **csv** [11] obtenemos los datos de los archivos csv para hacer los desplegables dinámicos.
- Para poder realizar las peticiones al backend desde nuestro frontend utilizábamos la librería **Flask** [12] mencionada previamente. Además, fue necesario el uso de la librería **Flask-CORS** [13] para permitir conexiones desde cualquier origen.
- En cuanto a los cálculos, se comenzó utilizando la librería **math** [14], pero tras realizar diferentes pruebas y Python ofrece soluciones para realizar todas las operaciones necesarias para los cálculos se decidió realizar todo siguiendo la información sobre las operaciones que nos ofrece Python. [15]

Por otro lado, en Vue.js importamos solamente la librería **VueSimpleAlert** [16]. Esta librería nos permite mostrar pop-ups de alertas cuando un usuario introduce un dato no válido, como podrían ser letras en lugar de números.

## 7.2 CREACIÓN DEL SISTEMA

Para la creación del sistema se utilizó Visual Studio Code [17], que nos permite instalar las librerías necesarias tanto para Python, como para Vue.js. Además, nos facilita mucho la ejecución tanto del frontend como del backend.

### 7.2.1 PROBLEMAS ENCONTRADOS

Los principales problemas que surgieron estaban relacionados con el uso de la librería Flask [12]. Inicialmente, esta librería generaba muchos problemas, ya que no nos permitía integrar el frontend con el backend. Por lo que hubo que utilizar la librería Flask-CORS [13], como se menciona previamente.

Para solucionar dichos problemas, utilizamos la función `@cross_origin`, perteneciente a la librería Flask-CORS [13], además de añadir al código las líneas mostradas en la ilustración 22.

```
app = Flask(__name__)
Cors = CORS(app)
CORS(app, resources={r'/*': {'origins': 'http://localhost'}}, CORS_SUPPORTS_CREDENTIALS = True)
app.config['CORS_HEADERS'] = 'Content-Type'
```

Ilustración 22 Código: Flask-CORS

Con estas líneas de código, hacíamos que nuestra aplicación usase Flask-CORS [13] y además permitíamos las peticiones desde todos los orígenes. En la ilustración 23, se muestra un ejemplo de cómo se utiliza la librería Flask-CORS [13] para realizar una petición GET.

```
@cross_origin
@app.route("/data", methods=['GET'])
def getExcel():
    excel = {}
    with open("ExcelPerfiles.csv", "r") as f:
        datos = f.readlines()
        excel['Tipos'] = list(filter(lambda x: len(x.strip())>0, datos[0].split(";")))
        i = 1
        for tipo in excel['Tipos']:
            excel[tipo] = list(filter(lambda x: len(x.strip())>0, datos[i].split(";")))
            i = i+1

    return jsonify(excel)
```

Ilustración 23 Código: Ejemplo Flask-CORS @cross\_origin

En la ilustración 23 podemos ver como se utiliza la función `@cross_origin` antes de indicar la ruta. Además, utilizamos `jsonify` perteneciente a la librería Flask [12] para devolver los datos al frontend en formato json.

### 7.2.2 DESCRIPCIÓN DETALLADA DE LAS CLASES

En este apartado comentaremos el contenido de las clases de una manera más detallada. Comentado los métodos más importantes y las funcionalidades de los mismos.

#### App.py

En el caso de App.py, nuestro backend que se encarga de recibir las peticiones y devolver el resultado al frontend, tendrá siete métodos principales.

- Los métodos getExcel: Son los encargados de enviar la información de los archivos csv a nuestro frontend. Un ejemplo, sería la Ilustración anterior, donde se muestra uno de estos métodos getExcel.
- Método getResistencias: Este método ejecuta la parte del cálculo de resistencias. Para ello, primero debe recibir unos valores desde el frontend al realizar la petición POST. Una vez recibidos dichos valores, realizará los cálculos necesarios y devolverá los resultados en formato json, como se ha mencionado en el apartado anterior.
- Método getPandeoLateral: Tiene el mismo funcionamiento que el método anterior, solo que los cálculos a realizar son distintos, ya que en este caso realizamos los cálculos correspondientes al pandeo lateral. Los datos a recibir también son distintos, y serán los del formulario que se muestre en nuestro frontend.
- Método getPandeoCompresion: Al igual que los dos anteriores, este método realizará el cálculo del pandeo de barras por compresión. Su funcionamiento es el mismo, solo varían los cálculos realizados.
- Método getInteracciones: Por último, tenemos el método que realizará los cálculos relacionados con la parte de las interacciones. Tendrá el mismo funcionamiento que los anteriores.

#### Componentes.vue

En cuanto a los métodos de los componentes se hablarán en conjunto, ya que todas las vistas siguen una misma estructura y funcionan de la misma manera.

Javier Alonso Fernández

- Métodos `cambiarInformación`: Estos métodos tienen como objetivo actualizar la información que sale por pantalla durante la fase de rellenar los formularios con los datos del problema. Actualizarán los formularios y la información mostrada que será necesaria para rellenar cada uno de ellos.
- Métodos `getCalculo`: Cada uno de nuestros componentes tendrá el propio, pero todos funcionan de la misma manera. Realizan una petición POST, enviando todos los datos necesarios, a nuestro backend para realizar los cálculos necesarios. Son los métodos que ejecutarán los mencionados previamente en el apartado de `App.py`.
- Método `getExcel`: Este será el método que obtendrá la información de los csv para los despleables. Realizará una petición GET para que el backend le devuelva la información en formato json. Y con ella, realizará los menús despleables dinámicamente. Para que se ejecute de forma previa a la visualización de la página, es necesario utilizar el ciclo de vida de `Vue.js` como se muestra en la ilustración 24.

```
beforeMount () {  
  this.getExcel()  
}
```

*Ilustración 24 Código: Ciclo de vida de `Vue.js`*

- Método `getPasoSiguiente`: El encargado de la reproducción por pasos será este método. Utilizando una variable `paso`, se encargará de ir actualizando la información mostrada según el paso en el que nos encontremos. De esta forma, el usuario podrá moverse entre los diferentes pasos del desarrollo.
- Método `getAnterior`: Por último, este método se encargará de volver al paso previo. Para ello, modificará la variable `paso` mencionada previamente. Y realizará una llamada al método `getPasoSiguiente`, de forma que, la información mostrada por pantalla sea la del paso anterior.

### 7.2.3 ARCHIVOS PRECARGADOS DEL SISTEMA

Los archivos precargados de este sistema serían los archivos csv y el Excel de donde se obtienen los datos para los cálculos.

Los archivos csv se cargarían al cambiar a cada una de las vistas, ya que el frontend realizaría una petición GET al backend para obtener los respectivos datos para hacer los formularios. Esto está implementado para cada una de las cuatro vistas.

Por otro lado, los datos del archivo Excel se cargarían cada vez que hacemos una petición POST, ya que son datos necesarios para realizar cada uno de los cálculos de nuestra aplicación. Esto se haría con la librería pandas. [10]

## 8 RESULTADOS

En este apartado, se analizarán los datos de cada uno de los cálculos comparándolos con los resultados obtenidos en el Excel inicial, del que parte la aplicación.

Para hacer las operaciones, la aplicación obtiene del Excel solamente los datos de cada perfil. En la ilustración 25, hay un ejemplo de parte de los datos que son necesarios de un perfil. Para ello, con la librería pandas [10], se extraen los datos necesarios para las operaciones.

		G	h	b	t <sub>w</sub>	t <sub>f</sub>	r	A	h <sub>i</sub>	d	Ø	P <sub>min</sub>	P <sub>max</sub>	A <sub>L</sub>	A <sub>G</sub>
		kg/m	mm	mm	mm	mm	mm	mm <sup>2</sup> x10 <sup>2</sup>	mm	mm		mm	mm	m <sup>2</sup> /m	m <sup>2</sup> /ft
IPE	IPE 80	6	80	46	3,8	5,2	5	7,64	69,6	59,6	-	-	-	0,328	54,6
IPE	IPE 100	8,1	100	55	4,1	5,7	7	10,3	88,6	74,6	-	-	-	0,4	49,3
IPE	IPE 120	10,4	120	64	4,4	6,3	7	13,2	107,4	93,4	-	-	-	0,475	45,8
IPE	IPE 140	12,9	140	73	4,7	6,9	7	16,4	126,2	112,2	-	-	-	0,551	42,7
IPE	IPE 160	15,8	160	82	5	7,4	9	20,1	145,2	127,2	-	-	-	0,623	39,5
IPE	IPE 180	18,8	180	91	5,3	8	9	23,9	164	146	M10	48	48	0,698	37,1
IPE	IPE 200	22,4	200	100	5,6	8,5	12	28,5	183	159	M10	54	58	0,768	34,4
IPE	IPE 220	26,2	220	110	5,9	9,2	12	33,4	201,6	177,6	M12	60	62	0,848	32,4
IPE	IPE 240	30,7	240	120	6,2	9,8	15	39,1	220,4	190,4	M12	66	68	0,922	30
IPE	IPE 270	36,1	270	135	6,6	10,2	15	45,9	249,6	219,6	M16	72	72	1,041	28,9
IPE	IPE 300	42,2	300	150	7,1	10,7	15	53,8	278,6	248,6	M16	72	86	1,16	27,5
IPE	IPE 330	49,1	330	160	7,5	11,5	18	62,6	307	271	M16	78	96	1,254	25,5
IPE	IPE 360	57,1	360	170	8	12,7	18	72,7	334,6	298,6	M22	88	88	1,353	23,7
IPE	IPE 400	66,3	400	180	8,6	13,5	21	84,5	373	331	M22	96	98	1,467	22,1
IPE	IPE 450	77,6	450	190	9,4	14,6	21	98,8	420,8	378,8	M24	100	102	1,605	20,7
IPE	IPE 500	90,7	500	200	10,2	16	21	116	468	426	M24	102	112	1,744	19,2
IPE	IPE 550	106	550	210	11,1	17,2	24	134	515,6	467,6	M24	110	122	1,877	17,8
IPE	IPE 600	122	600	220	12	19	24	156	562	514	M27	116	118	2,015	16,5

Ilustración 25 Ejemplo datos perfil Excel

### 8.1 RESULTADOS CÁLCULO DE RESISTENCIAS

En este caso, se analizarán los resultados del cálculo de resistencias, comparándolos con el mismo apartado del Excel y comentando los detalles más importantes.

RESISTENCIA DE LAS SECCIONES (en kN y m)								
Selección del Perfil		N	$M_y$	$M_z$	Interac.	según EC-3	$V_z$	$V_y$
DATOS (Esfuerzos)	$E_d =$	500,0 kN	78,0 kNm	0,0 kNm	0,45	0,05	11,3 kN	2,5 kN
IPE	IPE 400	$R_d =$	2213,1	342,3	60,0	$\frac{N_{Ed}}{N_{Rd}} + \frac{M_{Ed}}{M_{Rd}} + \frac{M_{Ed}}{M_{Rd}} \leq 1$	646	632
	Clase:	(calculada)	3	1	1	2	$V_{Ed} \leq 1/2 V_{Rd}$	¡Si!
	Interacción:		0,2	0,2	0,0	0,2	0,0	0,0040176
¿Hay que hacer la Interacción Flexión-cortante			No	$\rho$	Auxiliar	$M_{v,Rd,y}$		Inte- Cort
				Cort. $V_z$	0,931	218	342,3	kN m
				Cort. $V_y$	0,98			0,45

Ilustración 26 Resultados cálculo de resistencias Excel

En la ilustración 26, se pueden ver los resultados que se obtienen. Los datos introducidos serían los escritos en negro y los resultados dados estarían en rojo. Para introducir los datos, la aplicación exige rellenar los formularios antes de iniciar los cálculos, por lo que no habría ningún problema previo.

A continuación, se realizará una prueba con los mismos datos que hemos introducido en el ejemplo del Excel.

	N (kN)	My (kN m)	Mz (kN m)	Interaccion CTE	Interaccion EC-3	Vy (kN)	Vz (kN)
Ed	500	78	0	0.45	0.05	2.5	11.3
Rd	2213.1	342.3	60			632.2	645.5
Clase	3	1	1	2		Interaccion V:	0.45

Ilustración 27 Resultado cálculo de resistencias aplicación

En la ilustración 27, se pueden ver los resultados obtenidos por la aplicación en una ejecución con los mismos datos. Los resultados serían los mismos, por lo que procedemos a analizar casos especiales.

Estos casos especiales serían casos en los que algún resultado sería no válido, por lo tanto, deben resaltarse en rojo. Por ejemplo, los esfuerzos no pueden ser mayores que las resistencias calculadas, ni las interacciones mayores que 1.

A continuación, se realiza otro ejemplo con datos que sabemos previamente que nos darán un resultado no válido para ver cómo se comporta la aplicación.

RESISTENCIA DE LAS SECCIONES (en kN y m)									
Selección del Perfil			N	M <sub>y</sub>	M <sub>z</sub>	Interac.	según EC-3	V <sub>z</sub>	V <sub>y</sub>
DATOS (Esfuerzos)		E <sub>d</sub> =	2300,0 kN	78,0 kNm	0,0 kNm	1,30	1.30	11,3 kN	2,5 kN
IPE	IPE 400	R <sub>d</sub> =	2213,1	302,8	38,3	$\frac{N_{Ed}}{N_{Rd}} + \frac{M_{Ed}}{M_{Rd}} + \frac{M_{Ed}}{M_{Rd}} < 1$		646	632
Clase:		(calculada)	3	1	1	3		V <sub>Ed</sub> ≤ 1/2 V <sub>Rd</sub>	¡Sí!
Interacción:			1,0	0,3	0,0	0,3		0,0	0,0040176
¿Hay que hacer la Interacción Flexión-cortante				No	ρ	Auxiliar	M <sub>v,Rd,y</sub>		Inte- Cort
				Cort. V <sub>z</sub>	0,931	179	302,8	kN m	1,30
				Cort. V <sub>y</sub>	0,98				

Ilustración 28 Resultados no válidos cálculo de resistencias Excel

Para este ejemplo solo hemos cambiado el valor de el esfuerzo en N para que sea superior a la resistencia resultado. De esta manera vemos los valores que incumplirían las normas resaltados en rojo.

	N (kN)	My (kN m)	Mz (kN m)	Interaccion CTE	Interaccion EC-3	Vy (kN)	Vz (kN)
Ed	2300	78	0	1.3	1.3	2.5	11.3
Rd	2213.1	302.8	38.3			632.2	645.5
Clase	3	1	1	3		Interaccion V:	1.3

Ilustración 29 Resultados no válidos cálculo de resistencias aplicación

Por otro lado, en la ilustración 29, se puede ver la misma ejecución, pero esta vez en la aplicación. De nuevo, los resultados son correctos y, como se puede observar en dicha ilustración, los valores que no cumplen las normas establecidas se resaltan en color rojo. Además, estas normas vienen explicadas en el desarrollo por pasos, ya sea en las propias fórmulas o con una pequeña explicación en forma de texto.

## 8.2 RESULTADOS PANDEO DE BARRAS A COMPRESIÓN

De nuevo, la diferencia entre como realizan los cálculos la aplicación y el Excel es de dónde obtienen los datos previos. En el caso de la aplicación, como se ha mencionado previamente, solo coge datos del perfil seleccionado. Mientras que el Excel utiliza cálculos ya realizados, por lo que sería más complicado encontrar un error en caso de que lo hubiese.

A continuación, se va a realizar una comparación de los resultados obtenidos con los mismos datos.

RESISTENCIA DE LAS BARRAS:		PANDEO DE BARRAS A COMPRESIÓN							
	$\beta$	$L_k$	$\lambda_{mec}$	$\lambda_{reducida}$	curva	$\chi$	$N_{CR}$	$N_{b,Rd}$	interac.
Eje Y	<b>0,7</b>	3500 mm	21,1	0,24	a	0,99	39160 kN	<b>2192 kN</b>	<b>0,23</b>
Eje Z	<b>1</b>	5000 mm	126,6	1,46	b	0,36	1093 kN	<b>792 kN</b>	<b>0,63</b>
L (m)=	<b>5,00 m</b>								

Ilustración 30 Resultados cálculo pandeo de barras a compresión Excel

Para esta ejecución, serán necesarios unos nuevos datos que vienen resaltados en la ilustración 30 en negrita. Estos datos, se pedirán por pantalla de manera previa a la ejecución.

Para compararlos, se realiza una ejecución de nuestra aplicación introduciendo los mismos datos por pantalla. Cómo se puede ver en la ilustración 31, los resultados volverían a coincidir con los mostrados por el Excel.

	$L_k$ (mm)	$\lambda_{mecanica}$	$\lambda_{reducida}$	curva	$\chi$	$N_{b,Rd}$ (kN)	interac.
Eje Y	3500	21.1	0.24	a	0.99	2191.8	0.23
Eje Z	5000	126.6	1.46	b	0.36	792.3	0.63

Ilustración 31 Resultados cálculo pandeo de barras a compresión aplicación

De nuevo, las interacciones no podrían ser superiores a 1. Por lo tanto, si incumplen esta condición, aparecerán resaltadas en rojo indicando que su valor no es correcto. Esto también vendría explicado en el desarrollo por pasos.

RESISTENCIA DE LAS BARRAS:		PANDEO DE BARRAS A COMPRESIÓN							
	$\beta$	$L_k$	$\lambda_{mec}$	$\lambda_{reducida}$	curva	$\chi$	$N_{CR}$	$N_{b,Rd}$	interac.
Eje Y	0,7	3500 mm	21,1	0,24	a	0,99	39160 kN	2192 kN	1,05
Eje Z	1	5000 mm	126,6	1,46	b	0,36	1093 kN	792 kN	2,90
L (m)=	5,00 m								

Ilustración 32 Resultados no válidos cálculo pandeo de barras a compresión Excel

En este caso, se han cambiado los datos introducidos para que se produzca un resultado no válido en las interacciones. Como podemos ver en la ilustración 32, los valores de las interacciones serían superiores a 1, por lo tanto, aparecerían resaltados en rojo.

A continuación, se realiza el mismo ejemplo en la aplicación. Como se puede observar en la ilustración 33, los resultados vuelven a ser correcto y los valores no válidos vuelven a resaltarse en rojo como deberían.

	$L_k$ (mm)	$\lambda_{mecanica}$	$\lambda_{reducida}$	curva	$\chi$	$N_{b,Rd}$ (kN)	interac.
Eje Y	3500	21.1	0.24	a	0.99	2191.8	1.05
Eje Z	5000	126.6	1.46	b	0.36	792.3	2.9

Ilustración 33 Resultados no válidos cálculo pandeo de barras a compresión aplicación

Con esto, se comprueba que los datos de este apartado también serían correctos y su funcionamiento es el esperado. Por lo tanto, analizaremos a continuación el apartado de pandeo lateral y los resultados obtenidos en su ejecución.

### 8.3 RESULTADOS PANDEO LATERAL

Como en los casos anteriores, para analizar los resultados se ha utilizado un ejemplo con los mismos datos tanto en el Excel, como en la aplicación, para así, comprobar el correcto funcionamiento de la misma.

En cuanto al Excel, los resultados serían los mostrados a continuación. En este caso, se decidió cambiar ligeramente el orden en el que se mostraban los datos, para que así, el usuario tuviese más fácil comparar ciertos valores.

RESISTENCIA DE LAS BARRAS:			PANDEO LATERAL				momentos en kNm		
DATOS	$L$ (m)	$\beta_{LT}$	$L_C$ (m)	$C_1$	$h/b$	$\alpha_{LT}$	curva	$i_{f,z}$ (cm)	$L_C$ límite
IPe 400	5,00 m	1	5,0 m	1,55	2,2	0,34	b	4,5	2,080 m
EC3						0,34	b		
RESULT.	$M_{LTw}$	$M_{LTv}$	$k_2$	$M_{CR}$	$\lambda_{LT}$	$\Phi$	$\chi_{LT}$	$M_{b,Rd}$	interac.
DB SE-A	232 kNm	328 kNm	0,7	281 kNm	1,13	1,30	0,52	176,9 kNm	0,44
EC-3			1	463 kNm	0,88	1,00	0,67	230,4 kNm	0,34

Ilustración 34 Resultados cálculo pandeo lateral Excel

Además, por decisión del Co-Tutor Manuel Ignacio Guerra Romero, los resultados mostrados en la aplicación serán los que siguen la norma EC-3, ya que es la norma principal que sigue toda la aplicación.

Datos:	$L$ (m)	$\beta_{it}$	$L_c$ (m)	$L_c$ límite (m)	$C_i$	$h/b$	$\alpha_{LT}$	Curva
IPe 400	5	1	5	2.08	1.55	2.22	0.34	b
Resultados:	$M_{ltw}$ (kNm)	$M_{lrv}$ (kNm)	$M_{cr}$ (kNm)	$\lambda_{lt}$	$\Phi$	$\chi_{lt}$	$M_{brd}$ (kNm)	interac.
	231.8	327.5	462.5	0.88	1	0.67	230.39	0.34

Ilustración 35 Resultados cálculo pandeo lateral aplicación

De nuevo, los resultados vuelven a ser correctos. Como se ha comentado previamente, se realizaron ciertos cambios en la disposición de estos resultados. Lo que sería  $L_c$  límite, está justo al lado de  $L_c$ , esto se decidió hacer de esta manera para que el usuario tenga más facilidad a la hora de comparar dichos valores.

RESISTENCIA DE LAS BARRAS:			PANDEO LATERAL				momentos en kNm		
DATOS	L (m)	$\beta_{LT}$	$L_C$ (m)	$C_1$	h/b	$\alpha_{LT}$	curva	$i_{f,z}$ (cm)	$L_C$ límite
IPE 400	5,00 m	1	5,0 m	1,55	2,2	0,34	b	4,5	2,080 m
EC3						0,34	b		
RESULT.	$M_{LTw}$	$M_{LTv}$	$k_2$	$M_{CR}$	$\lambda_{LT}$	$\Phi$	$\chi_{LT}$	$M_{b,Rd}$	interac.
DB SE-A	232 kNm	328 kNm	0,7	281 kNm	1,13	1,30	0,52	176,9 kNm	2,83
EC-3			1	463 kNm	0,88	1,00	0,67	230,4 kNm	2,17

Ilustración 36 Resultados no válidos cálculo pandeo lateral Excel

En la ilustración 36, se muestran resultados no válidos para las interacciones. Al igual que en los casos anteriores, estos resultados aparecen resaltados en rojo. Esto es algo que debemos aplicar también en la aplicación, para que el usuario comprenda de manera más sencilla los resultados y su significado.

Datos:	L (m)	$\beta_{it}$	Lc (m)	Lc limite (m)	$C_1$	h/b	$\alpha_{LT}$	Curva
IPE 400	5	1	5	2.08	1.55	2.22	0.34	b
Resultados:	$M_{ltw}$ (kNm)	$M_{ltv}$ (kNm)	$M_{cr}$ (kNm)	$\lambda_{lt}$	$\Phi$	$\chi_{lt}$	$M_{brd}$ (kNm)	interac.
	231.8	327.5	462.5	0.88	1	0.67	230.39	2.17

Ilustración 37 Resultados no válidos cálculo pandeo lateral aplicación

De nuevo, los resultados mostrados serían correctos y se puede observar que la interacción, al tener un valor erróneo, se vuelve a resaltar en color rojo.

## 8.4 RESULTADOS INTERACCIÓN DE ESFUERZOS EN BARRA

En esta ejecución, además de mostrar los resultados de la interacción de esfuerzos en barra, se decidió mostrar los resultados de todas las interacciones previas, de forma que el usuario tenga una manera más sencilla de acceder a todos los resultados previos.

En esta ejecución, ciertos valores introducidos por pantalla no pueden ser superiores a 1 ni inferiores a 0.4, por lo que, en caso de introducir valores

incorrectos, se muestra información por pantalla para que el usuario entienda cual fue el problema.

Recuerde que estos valores no pueden ser superiores a 1, ni inferiores a 0.4.



Ilustración 38 Resultados interacción de esfuerzos en barra, datos mal introducidos

A continuación, se va a analizar un ejemplo introduciendo los mismos datos en el Excel que en la aplicación.

RESISTENCIA DE LAS BARRAS:			INTERACCIÓN EN LA BARRA				EAE		
Pandeo en el plano débil (z)			Interac. EC3	Inter. SE-A			$C_{m,y}$	$C_{m,z}$	$C_{m,LT}$
				Ec. (6,53)			<b>0,74</b>	<b>0,74</b>	<b>0,74</b>
$\frac{N_{Ed}}{N_{b,z,Rd}} + k_{zy} \frac{M_{y,Ed}}{M_{b,Rd}} + k_{zz} \frac{M_{z,Ed}}{M_{z,Rd}} \leq 1$	<b>0,926</b>	<b>1,015</b>							
Pandeo en el plano fuerte (y)			Interac. EAE	Inter. SE-A	EAE	0,747	0,836	0,871	1,394
				Ec. (6,51)			$k_y$	$k_z$	$k_{yLT}$
$\frac{N_{Ed}}{N_{b,y,Rd}} + k_{yy} \frac{M_{y,Ed}}{M_{b,Rd}} + k_{yz} \frac{M_{z,Ed}}{M_{z,Rd}} \leq 1$	<b>0,48</b>	0,56			SE-A	1,01	1,88	0,87	

Ilustración 39 Resultados interacción de esfuerzos en barra Excel

Para esta ejecución, también se decidió realizar los cálculos que siguen la norma EC3, los valores introducidos por pantalla serían los mismos que en las demás ejecuciones, añadiéndole los nuevos valores resaltados en negrita en la parte superior derecha de la ilustración 39.

Datos	Kyy	Kyz	Kzy	Kzz
	0.75	0.84	0.87	1.39

Ilustración 40 Resultados interacción de esfuerzos en barra aplicación, coeficientes K

Para realizar este cálculo, primero se deben hallar los coeficientes K, mostrados en la ilustración 40. Como se puede ver, los resultados serían los mismos que los del Excel, por lo tanto, se comprobarán los resultados de las interacciones.

interac. en barras z	interac. en barras y
0.93	0.48

Ilustración 41 Resultados interacción de esfuerzos en barra aplicación

Comparando los resultados, se ve que son correctos siguiendo la norma EC3, como se ha explicado previamente. Nuevamente, estos valores no pueden ser superiores a 1, por lo tanto, se analizará a continuación un ejemplo que muestre resultados no válidos.

RESISTENCIA DE LAS BARRAS:			INTERACCIÓN EN LA BARRA				EAE		
Pandeo en el plano débil (z)			Interac. EC3	Inter. SE-A			$C_{m,y}$	$C_{m,z}$	$C_{m,LT}$
				Ec. (6.53)			0,74	0,74	0,74
$\frac{N_{Ed}}{N_{b,z,Rd}} + k_{zy} \frac{M_{y,Ed}}{M_{b,Rd}} + k_{zz} \frac{M_{z,Ed}}{M_{z,Rd}} \leq 1$			3,159	3,228					
Pandeo en el plano fuerte (y)			Interac. EAE	Inter. SE-A	EAE	$k_{yy}$	$k_{yz}$	$k_{zy}$	$k_{zz}$
				Ec. (6.51)		0,854	2,029	0,704	2,029
$\frac{N_{Ed}}{N_{b,y,Rd}} + k_{yy} \frac{M_{y,Ed}}{M_{b,Rd}} + k_{yz} \frac{M_{z,Ed}}{M_{z,Rd}} \leq 1$			1,36	1,44			$k_y$	$k_z$	$k_{yLT}$
					SE-A	1,15	2,74	0,70	

Ilustración 42 Resultados no válidos interacción de esfuerzos en barra Excel

En este caso, se han cambiado los datos introducidos para que los resultados vuelvan a ser no válidos. Como se puede observar en la ilustración 42, las interacciones según EC3 toman valores superiores a 1, por lo tanto, vuelven a aparecer resaltadas en rojo.

Datos	Kyy	Kyz	Kzy	Kzz
	0.85	2.03	0.7	2.03

Ilustración 43 Resultados no válidos interacción de esfuerzos en barra aplicación, coeficientes K

Al realizar la misma ejecución, se observa que los valores de los coeficientes K vuelven a ser los correctos según el Excel. Por lo tanto, se analizarán los resultados de las interacciones.

interac. en barras z	interac. en barras y
3.16	1.36

*Ilustración 44 Resultados interacción de esfuerzos en barra aplicación*

Por último, al analizar estos resultados, vuelven a ser correctos y, además, se vuelven a resaltar en rojo al ser superiores a 1. Por lo tanto, se ha comprobado que el funcionamiento es el que debería y todo estaría correcto.

## 8.5 CONCLUSIONES RESULTADOS

Para concluir con el análisis de los resultados, se han realizado comprobaciones de cada una de las funcionalidades, utilizando los mismos datos tanto en el Excel como en la aplicación. Incluso cambiando los datos, se puede observar que los resultados siguen coincidiendo, por tanto, la aplicación cumple con todas las funcionalidades de manera satisfactoria.

# 9 DESARROLLO DE LAS PRUEBAS

## 9.1 PRUEBAS UNITARIAS

Para el desarrollo de estas pruebas, se utilizó la herramienta Pytest [18]. De esta manera se comprobó el correcto funcionamiento del backend, utilizando los resultados obtenidos en el Excel con diferentes datos introducidos. Se realizaron pruebas del correcto funcionamiento de cada uno de los diferentes cálculos de nuestra aplicación.

Para ello, se daban los datos necesarios para acceder a cada una de las condiciones de nuestros cálculos, estos datos se obtenían del Excel inicial. Además, se comprobaba que los resultados fuesen los esperados y todo funcionase de la forma correcta.

Por ejemplo, para probar el código mostrado en la siguiente ilustración, se introducirían diferentes tipos de acero y perfiles de forma que se comprobasen todos los posibles resultados.

```
def calcularCurvaPandeoY(claseAcero, hdb):  
    if claseAcero == 'S450' or claseAcero == 'S460':  
        if hdb >1.2:  
            return "a0"  
        else:  
            return "a"  
    else:  
        if hdb >1.2:  
            return "a"  
        else:  
            return "b"
```

*Ilustración 45 Ejemplo pruebas unitarias*

Para ello, primero se probaría con diferentes tipos de acero, para comprobar si entra de forma correcta en el primer condicional y, a continuación, se realizarían

pruebas con distintos perfiles que tengan un 'h' y 'b' diferentes y que, de esta manera, hagan cumplir el resto de condiciones.

Una vez se comprueba que siempre se obtiene el resultado deseado, comprobaríamos el resultado del resto de los métodos. Para verificar los resultados se utiliza el Excel inicial, dónde se verá si los resultados son los correctos y cada método funciona como debería.

## 9.2 PRUEBAS DE INTEGRACIÓN

Una vez se comprobó el correcto funcionamiento del backend. Se realizaron pruebas de integración. En estas pruebas, se comprobó que los elementos que componen el software funcionaban correctamente probándolos juntos. Y, además, la comunicación entre los componentes era la esperada.

# 10 MANUALES DEL SISTEMA

## 10.1 MANUAL DE INSTALACIÓN

Antes de instalar ejecutar o trabajar en el proyecto, hay que instalar las siguientes características en el equipo.

Primero, se deberá descargar e instalar node.js y npm de la página oficial de node.js [19]. Una vez instalado, ahora descargaremos e instalaremos tanto pip [20] como Python [21] desde las páginas oficiales. Una vez realizado este paso, ya sería posible comenzar con la clonación del repositorio y la instalación de las librerías necesarias.

Para clonar el repositorio de este proyecto [22], deberíamos utilizar en enlace del mismo, que se indica en la siguiente ilustración.

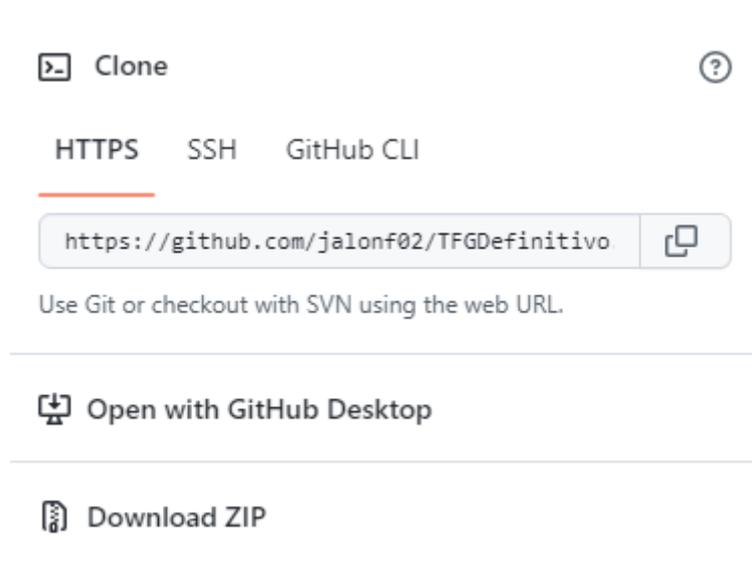


Ilustración 46 Repositorio github

Una vez clonado, procederemos a la instalación de librerías. Para ello, se aconseja utilizar Visual Studio Code [17], ya que fue el entorno de desarrollo utilizado a lo largo de todo el proyecto.

Primero, accederemos al directorio del frontend y utilizaremos el comando `npm install vue`, para instalar vue. A continuación, necesitaremos introducir el comando `npm install vue-simple-alert [16]` para instalar una de las librerías utilizadas. Una vez instalado, procederemos a instalar las librerías del backend.

Para instalar estas librerías, debemos volver al directorio del proyecto. Allí, introduciremos los siguientes comandos:

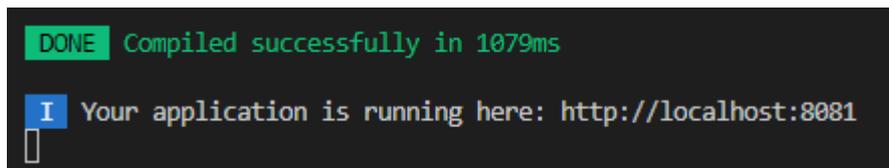
- `pip install pandas`, para instalar la librería pandas [10], que será la encargada de extraer los datos necesarios del Excel.
- `pip install flask`, con esto, instalaremos la librería Flask [12], que nos servirá para recibir las peticiones de nuestro frontend.
- `pip install flask_cors`, para instalar la librería Flask-CORS [13], que nos permitirá permitir que se acceda a las rutas establecidas en nuestro backend.

Una vez se han instalado todas las librerías. El proyecto estaría listo para la ejecución.

## 10.2 MANUAL DE EJECUCIÓN

Para la ejecución, sería necesario realizar las instalaciones previas mostradas en el apartado anterior. Una vez se ha instalado todo correctamente, la aplicación podrá ser ejecutada. Para ello se utilizaría Visual Studio Code [17].

Primera, en el directorio del frontend, se debería poner el comando `npm run dev`. Con esto, se lanzaría el frontend y se indicaría en la terminal donde ha sido lanzada nuestra aplicación. Por defecto será en `http://localhost:8080`.

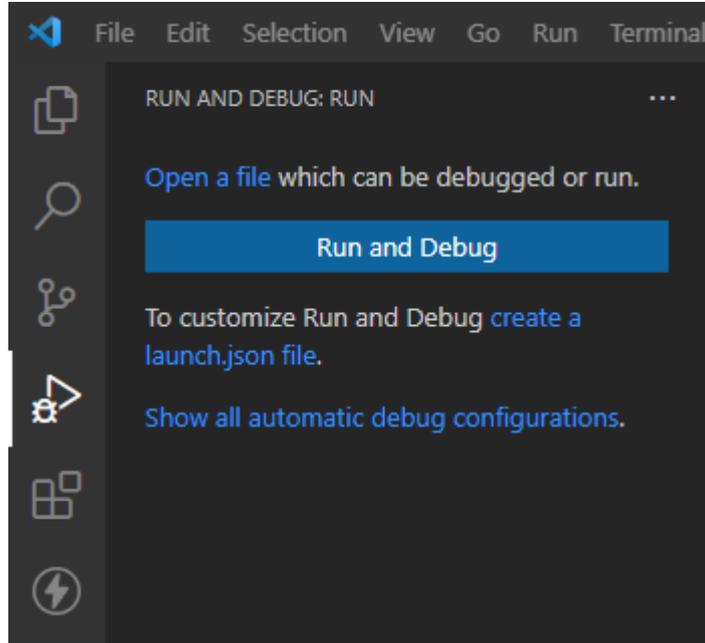


```
DONE Compiled successfully in 1079ms
I Your application is running here: http://localhost:8081
█
```

*Ilustración 47 Manual de ejecución frontend*

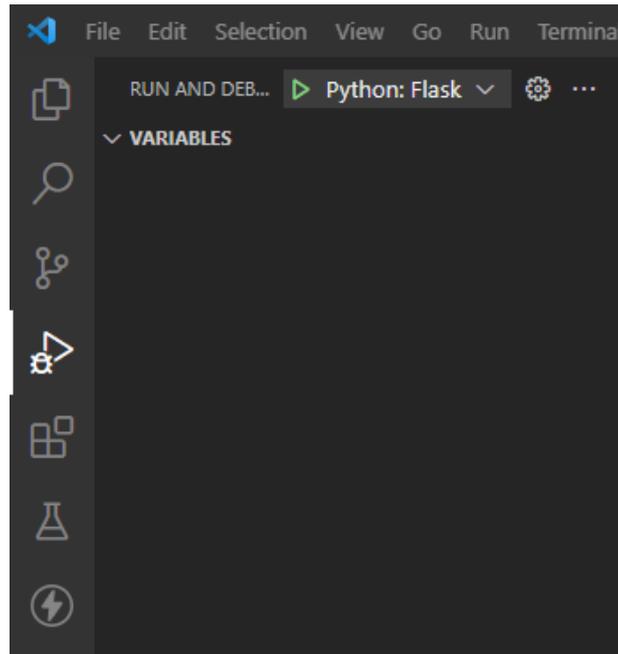
El mensaje debería ser cómo el mostrado en la anterior ilustración.

A continuación, tenemos que ejecutar lo que sería nuestro servidor y backend. Para ello, tenemos dos opciones, la más recomendable sería utilizar la pestaña de run and debug como se muestra en la siguiente ilustración.



*Ilustración 48 Manual de ejecución run and debug*

En esa pestaña, haremos click en run and debug y se debe seleccionar python en el desplegable que aparezca. A continuación, aparecerá un nuevo desplegable donde se debe seleccionar la opción flask y seleccionar el archivo a ejecutar, que en este caso sería nuestro app.py.



*Ilustración 49 Manual de ejecución Flask*

Una vez hemos seleccionado el archivo que se ejecutará, lo que veremos en la pestaña run and debug sería lo mostrado en la ilustración anterior. Con esto, solo debemos darle al botón verde de iniciar para ejecutar nuestro servidor y backend. Por terminal se debería mostrar lo siguiente.

```
PS C:\Users\Javier\Desktop\Prueba2\TFGDefinitivo> & 'python'  
* Serving Flask app 'app.py' (lazy loading)  
* Environment: development  
* Debug mode: on  
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)  
* Restarting with stat
```

*Ilustración 50 Manual de ejecución servidor lanzado*

Con esto, ya tendríamos ambas partes de nuestra aplicación funcionando, por lo que ya podríamos acceder a ella sin ningún problema y utilizarla.

Otra manera de ejecutar la parte del backend y del servidor sería utilizando el comando `export FLASK_APP=app.py` y después utilizar `flask run` o bien `python -m flask run`. Ambas maneras son válidas, pero la primera de ellas es más visual y sencilla de seguir, además, después de realizar la primera configuración, siempre tendremos el botón de iniciar flask en la pestaña de run and debug directamente.

### 10.3 MANUAL DE USUARIO

Una vez ejecutado el sistema, el uso para el usuario es muy sencillo. El usuario simplemente deberá introducir la información pedida por pantalla siguiendo las indicaciones. Una vez introduzca dicha información, el usuario podrá darle al botón iniciar para obtener los resultados de la ejecución. Además, tendrá la opción de utilizar los botones siguiente y anterior, para así moverse por los diferentes pasos del desarrollo paso a paso.

### 10.4 MANUAL DEL PROGRAMADOR

De cara a que otro desarrollador pudiese retomar este trabajo, se adjunta este manual.

El entorno de desarrollo utilizado para el proyecto fue Visual Studio Code [17], utilizando la versión de Python 3.9.6 [21] y la 21.1.3 de pip [20].

En cuanto a Vue.js [7], la versión utilizada fue la 3.2.31. Las librerías utilizadas y problemas surgidos se pueden revisar en el apartado [7.1 Lenguajes herramientas y librerías utilizadas](#).

Además, se presenta un manual de instalación en el apartado [10.1 Manual de instalación](#).

Una vez instalado, ya se podrá trabajar sobre el código existente. Para revisar la ejecución, se debe consultar el [10.2 Manual de ejecución](#).

# 11 CONCLUSIONES Y AMPLIACIONES

## 11.1 CONCLUSIONES

En cuanto a la parte de las funcionalidades, se ha conseguido que la aplicación lleve a cabo todas las que se consideraron necesarias. Además, se añadió una ejecución paso a paso para añadir más información para el usuario y que pueda interactuar con ella de una manera más sencilla.

Lo más importante de la aplicación sería la fiabilidad, debido a las pruebas realizadas, se ha comprobado que los resultados son correctos y la aplicación funciona como se espera. Gracias al Excel inicial, hemos podido realizar pruebas con una cantidad grande de datos, para así asegurarnos de su correcto funcionamiento.

Por otro lado, la aplicación es muy sencilla de comprender y utilizar al tener las partes bien diferenciadas. Esto era otro objetivo inicial que se ha conseguido cumplir.

## 11.2 AMPLIACIONES Y TRABAJOS FUTUROS

En cuanto a las posibles mejoras o ampliaciones de la aplicación, durante el desarrollo surgieron diferentes ideas, pero por la falta de tiempo no se pudieron aplicar.

La principal mejora sería rehacer los formularios de forma que, en cada una de las vistas, el formulario se mantenga y el usuario no tenga que volver a introducir los datos para llevar a cabo los cálculos. De esta manera, el usuario ahorraría tiempo y podría realizar los cálculos de manera más rápida.

Además, en la aplicación actual se pide por pantalla la clase del perfil. Otra posible mejora sería la de calcular dicha clase. De esta forma, reduciríamos la

información que debe introducir el usuario, por lo que haríamos el uso de la aplicación aún más sencillo.

Otra posible mejora sería añadir más tipos de perfiles, por ejemplo, los perfiles tubulares. Los cálculos para este tipo de perfiles serían prácticamente los mismos, solo habría que realizar pequeños retoques en el backend, donde se realizan los cálculos.

Por último, también se podría añadir una nueva función, con la que el usuario pudiese elegir el archivo dónde se guardarán los datos de la ejecución si así lo desea. Con esto, el usuario podría tener un fichero en el que guardaría los datos de las ejecuciones que considerase necesarios.

# 12 REFERENCIAS BIBLIOGRÁFICAS

- [1] Adobe, “Aspectos básicos de las aplicaciones web,” 2021. <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html> (accessed Jun. 18, 2022).
- [2] Nicole and Maldeadora, “Qué es Frontend y Backend: diferencias y características,” 2018. <https://platzi.com/blog/que-es-frontend-y-backend/> (accessed Jun. 19, 2022).
- [3] IBM, “API REST.” <https://www.ibm.com/es-es/cloud/learn/rest-apis> (accessed Jun. 20, 2022).
- [4] Felipe, “Métodos HTTP de petición que debes conocer.” <https://www.hostingplus.com.es/blog/metodos-http-de-peticion-que-debes-conocer/> (accessed Jun. 19, 2022).
- [5] Dlubal, “Análisis de tensiones de superficies y barras.” <https://www.dlubal.com/es/productos/complementos-para-rfem-6-y-rstab-9/calculo/analisis-tension-deformacion> (accessed Jun. 27, 2022).
- [6] Structuralia, “Software utilizado para el diseño y cálculo de estructuras.” <https://blog.structuralia.com/5-software-utilizados-para-el-diseno-y-calculo-de-estructuras-en-edificacion-y-obra-civil> (accessed Jun. 27, 2022).
- [7] Vue.js, “Vue.js.” <https://vuejs.org/guide/introduction.html> (accessed Jun. 15, 2022).
- [8] J. D. Muñoz, “Qué es Flask.” <https://openwebinars.net/blog/que-es-flask/> (accessed Jun. 20, 2022).
- [9] Datademia, “¿Qué es Python?” <https://datademia.es/blog/que-es-python> (accessed Jun. 17, 2022).
- [10] Pandas, “Pandas.” <https://pandas.pydata.org/> (accessed Jun. 10, 2022).
- [11] Python.org, “csv - CSV File Reading and Writing.” <https://docs.python.org/3/library/csv.html> (accessed Jun. 10, 2022).

- [12] Flask, “Flask.” <https://flask.palletsprojects.com/en/2.1.x/> (accessed Jun. 13, 2022).
- [13] C. Dolphin, “Flask-CORS.” <https://flask-cors.readthedocs.io/en/latest/> (accessed Jun. 14, 2022).
- [14] Python, “math - Mathematical functions.” <https://docs.python.org/3/library/math.html> (accessed Jun. 04, 2022).
- [15] B. S. Marco, “Python operaciones matemáticas.” <https://www.mclibre.org/consultar/python/lecciones/python-operaciones-matematicas.html> (accessed Jun. 04, 2022).
- [16] Vuejsexamples, “Simple Alert for vue.js.” <https://vuejsexamples.com/simple-alert-for-vue-js/> (accessed Jun. 16, 2022).
- [17] V. S. Code, “Visual Studio Code.” <https://code.visualstudio.com/> (accessed Jun. 01, 2022).
- [18] pytest, “pytest: helps you write better programs,” 2015. <https://docs.pytest.org/en/7.1.x/> (accessed Jun. 21, 2022).
- [19] Node.js, “Descargas node.js.” <https://nodejs.org/es/download/> (accessed Jun. 23, 2022).
- [20] Pypi.org, “pip.” <https://pypi.org/project/pip/> (accessed May 02, 2022).
- [21] Python.org, “Python.” <https://www.python.org/downloads/> (accessed Jun. 23, 2022).
- [22] J. A. Fernández, “jalonf02/TFGDefinitivo,” 2022. <https://github.com/jalonf02/TFGDefinitivo>.